**Switch H/W Generic**

OCP Command Test Plan

Version:1

Date: 2016.02

Alpha Corporation

No.8, Li-shing Road VII,

Science-based Iudustrial Park,

Hsinchu, Taiwan, R.O.C.

Phone: 886-3-563-6666

Fax: 886-3-578-1861

**Table of Contents**

[1.1 Endurance Test 3](#_Toc442274277)

[1.2 Basic TX/RX Test 4](#_Toc442274278)

[Port Function Test - 10/100M/1000M/10G/40G Auto-negotiation 4](#_Toc442274279)

[Basic TX/RX Wire Speed Test 4](#_Toc442274280)

[Varying Cable length, Data Pattern, Packet Rate, Packet Size Test, Packet Type 5](#_Toc442274281)

[1.3 Switch function and performance Test 8](#_Toc442274282)

[Buffer Size 8](#_Toc442274283)

[Filtering Rate and Address Rate: 9](#_Toc442274284)

[Forwarding the Reserved Multicast Packets 9](#_Toc442274285)

[Forwarding the BPDU packets 9](#_Toc442274286)

[Forwarding the Reserved multicast packets 9](#_Toc442274287)

[MAC addresses supported/Illegal MAC Test 11](#_Toc442274288)

[Aging Time 12](#_Toc442274289)

[Overwrite the Forwarding Table 12](#_Toc442274290)

[Update the Forwarding Table 12](#_Toc442274291)

[Learning Algorism Test 12](#_Toc442274292)

[Filter Illegal Frames Test 12](#_Toc442274293)

[1.4 Flow Control Tests 13](#_Toc442274294)

[Pause Frame in Flow Control Test 13](#_Toc442274295)

[Flow Control/Backpressure Tests 14](#_Toc442274296)

[Priority Queues Test 14](#_Toc442274297)

[Buffer Clean Test 15](#_Toc442274298)

[10M/100M/1000M/10G/40G Hybrid Test 15](#_Toc442274299)

[Broadcast Mesh Test 15](#_Toc442274300)

[1.5 Power ON/OFF Test 16](#_Toc442274301)

[2.1 External Component Test - FAN 16](#_Toc442274302)

[2.2 External Component Test - RTC 17](#_Toc442274303)

[2.3 External Component Test - LED 17](#_Toc442274304)

[2.4 Platform - Packets Path 19](#_Toc442274305)

[2.5 Platform – RPS 20](#_Toc442274306)

[2.6 Platform –CPU path 21](#_Toc442274307)

[2.7 Platform –USB 21](#_Toc442274308)

[2.8 Platform – Console 22](#_Toc442274309)

[2.9 Platform – Flash/SD 23](#_Toc442274310)

[2.10 Platform – Memory 23](#_Toc442274311)

[2.11 Platform – Stacking 24](#_Toc442274312)

[2.12 Platform – PoE 24](#_Toc442274313)

[2.13 Platform – Temperature 25](#_Toc442274314)

[2.14 Platform – Factory Reset 25](#_Toc442274315)

[2.15 Platform – Transceiver 26](#_Toc442274316)

[2.16 Platform – Link state 27](#_Toc442274317)

[2.17 Platform – Power 70](#_Toc442274318)

[2.18 Platform – LCD Panel 71](#_Toc442274319)

[3.1 Maximum Load Test 73](#_Toc442274320)

[3.2 Stress Test 73](#_Toc442274321)

[3.3 Parameter Test 74](#_Toc442274322)

[*Full loading test (Ether port) 74*](#_Toc442274323)

[*Full loading test (Fiber port) 78*](#_Toc442274324)

[*Full loading test (10G) 82*](#_Toc442274325)

[*Full loading test (40G) 88*](#_Toc442274326)

[Appendix A 94](#_Toc442274327)

**Hardware Features to be Tested**

* 1. **Endurance Test**

Equip.: IXIA ports, IxExplorer

Duration: 8 hours

Topology: Standalone

Objective: To ensure that the device can boot up normally under heavy broadcast traffic, generate traffic and work normally under a critical environment without hang.

1. Set the device to the Factory Default configuration.
2. Connect device's ports to IXIA.
3. Configure Transmit Setup (The number of device’s ports: n), and then start the traffic transmit.
4. Power on the device
5. Check the following items to ensure device's integrity:

Check all port's RX/TX function.

Check LED: Is there any LED lighting/flashing abnormally?

Check Switching Function: MAC learning works?

Check Statistics: CRC errors or other abnormal counters.

HW Table 1.1.1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **IXIA** | **DUT** | **Mode** | **Length** | **Background** | **Gap** | **VFD1** | **VFD2** | **Error** | **Result** |
| 100FC N-Way | N-Way | Continuous | 64 | All Zeros | 96 bits | 1<->n+1 | 000000000001 | Off |  |
| 100H/FC N-Way | N-Way | Continuous | Random | Random/IP | 96 bits | 1<->n+1 | 000000000002 | Off |
| 100H/FC N-Way | N-Way | Continuous | 1518 | Random/IP | 96 bits | 1<->n+1 | 000000000003 | Off |
| 100H | N-Way | Continuous | 1518 | Random/IP | 96 bits | 1<->n+1 | 000000000004 | Off |
| 100H | N-Way | Continuous | 64 | Random/IP | 96 bits | FFFFFFFFFFFF | 000000000005 | Off |
| 10 FC N-Way | N-Way | Continuous | Random | Random/IP | 96 bits | 1<->n+1 | 000000000006 | Off |
| 10 FC N-Way | N-Way | Continuous | 1518 | Random/IP | 96 bits | 1<->n+1 | 000000000007 | Off |
| 10H /FC N-Way | N-Way | Continuous | 1518 | Random/IP | 96 bits | 1<->n+1 | 000000000008 | Off |
| 10H | N-Way | Continuous | 64 | Random/IP | 96 bits | FFFFFFFFFFFF | 000000000009 | Off |
| N-Way | N-Way | Continuous | Random | Random/IP | 96 bits | Random | 00000000000a | Dribble |
| N-Way | N-Way | Continuous | 757 | Random/IP | 96 bits | Random | 00000000000b | CRC/Alig |
| N-Way FC | N-Way | Continuous | 1600/ Jumbo | Random/IP | 96 bits | Unknown | 00000000000c | Off |
| IEEE  default | N-Way | Continuous | 64 | All Zeros | 96 bits | 1<->n+1 | 000000000001 | Off |  |
| IEEE  Default/FC | N-Way | Continuous | Random | Random/IP | 96 bits | 1<->n+1 | 000000000002 | Off |
| IEEE  default | N-Way | Continuous | 1518 | Random/IP | 96 bits | 1<->n+1 | 000000000003 | Off |
| IEEE  Default/FC | N-Way | Continuous | 1518 | Random/IP | 96 bits | 1<->n+1 | 000000000004 | Off |
| IEEE  default | N-Way | Continuous | 64 | Random/IP | 96 bits | FFFFFFFFFFFF | 000000000005 | Off |
| IEEE  Default/FC | N-Way | Continuous | Random | Random/IP | 96 bits | 1<->n+1 | 000000000006 | Off |
| IEEE  default | N-Way | Continuous | 1518 | Random/IP | 96 bits | 1<->n+1 | 000000000007 | Off |
| IEEE  Default/FC | N-Way | Continuous | 1518 | Random/IP | 96 bits | 1<->n+1 | 000000000008 | Off |
| IEEE  default | N-Way | Continuous | 64 | Random/IP | 96 bits | FFFFFFFFFFFF | 000000000009 | Off |
| IEEE  Default/FC | N-Way | Continuous | Random | Random/IP | 96 bits | Random | 00000000000a | Dribble |
| IEEE  default | N-Way | Continuous | 757 | Random/IP | 96 bits | Random | 00000000000b | CRC/Alig |
| IEEE  Default/FC | N-Way | Continuous | 1600/ Jumbo | Random/IP | 96 bits | Unknown | 00000000000c | Off |

* 1. **Basic TX/RX Test**

Equip.: IXIA ports, IxExplorer

Objective: To ensure the integrity of TX/RX function.

Port Function Test - 10/100M/1000M/10G/40G Auto-negotiation

Equip: IXIA one port, IxExplorer

Duration: **3 minutes for each port**

Objective: Ensure each UTP port N-way function

1. Set the device to the Factory Default configuration and disable flow control.
2. Change the port in IXIA to the following state:
3. Use 40G AVAGO AFBR-79EIDZ and Fiber (3m) connect to device with then check the UTP mode and LED display in switch hub are correct.

HW Table 1.2.1

|  |  |  |  |
| --- | --- | --- | --- |
| Mode-IXIA | Rate | Half/Full | Result |
| Force | 10 | Half |  |
| 100 | Half |  |
| N-Way  (With flow control) | 10 | Half |  |
| 100 | Half |  |
| N-Way | 10 | Full |  |
| 100 | Full |  |
| 1000 | Full |  |
| 10G | Full |  |
| 40G | Full |  |
| N-Way  (With flow control) | 10 | Full |  |
| 100 | Full |  |
| 1000 | Full |  |
| 10G | Full |  |
| 40G | Full |  |
| Use IEEE defaults | 10G | Full |  |
| 40G | Full |  |
| Use IEEE defaults  (With flow control) | 10G | Full |  |
| 40G | Full |  |

Basic TX/RX Wire Speed Test

Equip.: IXIA ports, IxExplorer

Objective: Ensure TX/RX integrity

1. Set the device to the Factory Default configuration.
2. Set the IXIA with/without Flow Control
3. Configure Transmit Setup, pair-wise, the configuration as

pair-wise, transmit 10 sec. port configuration as following:

Packet size 64, 512, 1518, Jumbo IP unicast traffic. Load 100%

(Learning: Before each iteration, delay after Transmit =300 sec.)

1. Run the test; it fails if any of the following situations occurs.

1. Packet loss

2. Error packets received.

HW Table 1.2.2

|  |  |  |
| --- | --- | --- |
| Item | Port media | Result |
| Test 1. | 10H <-> 10H (N-Way/FC) |  |
| Test 2. | 10F <-> 10F (N-Way) |  |
| Test 3. | 10F <-> 10F (N-Way/FC) |  |
| Test 4. | 100H <-> 100H (N-Way/FC) |  |
| Test 5. | 100F <-> 100F (N-Way) |  |
| Test 6. | 100F <-> 100F (N-Way/FC) |  |
| Test 7. | 1000F <-> 1000F (N-Way) |  |
| Test 8. | 1000F <-> 1000F (N-Way/FC) |  |
| Test 9. | 10H <-> 100H (N-Way/FC) |  |
| Test 10 | 10F <-> 100F (N-Way/FC) |  |
| Test 11 | 100F <-> 1000F (N-Way/FC) |  |
| Test 12 | 10F <-> 1000F (N-Way/FC) |  |
| Test 13 | 10G <-> 10G |  |
| Test 14 | 10G <-> 40G |  |
| Test 15 | 40G<->40G |  |

Varying Cable length, Data Pattern, Packet Rate, Packet Size Test, Packet Type

Equipment ：IXIA 3 ports, IxExplorer

Duration：**3 Hours**

Objective： Ensure TX/RX integrity

a. Set the device to factory reset configuration.

b. Transmit and Receive Test, pair-wise except Packet Type one pair unicast with address table full (32765),

Third port send, broadcast and unknown 1k to check no error packets received and then execute next step.

c. Set IXIA port’s status as follow item in table1.2.3.1 and send packets, then receive packets from

TX ports, check no lose packets and error packets.

/\* unicast, broadcast and unknown should be sent separately

**10/100/1000TX**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cable Length | Mode | Packet Size | Data Pattern | I.F.G. | Packet Type |
| 3m | \* Single burst  10,000,000 | \*64 | \*Random | \*96 bit time | \*Unicast |
| 6m | \* Single burst  10,000,000 | 1518 | All One | Random | Broadcast |
| 6m | Single burst  10,000,000 | Random | All Zero | Random | Unknown(1k) |

**1000 SX/LX**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cable Length | Mode | Packet Size | Data Pattern | I.F.G. | Packet Type |
| 3m | \*Single burst  10,000,000 | \*64 | \*Random | \*96 bit time | \*Unicast |
| 6m | \*Single burst  10,000,000 | 1518 | All One | Random | Broadcast |
| 6m | \*Single burst  10,000,000 | Random | All Zero | Random | Unknown(1k) |

**10G**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cable Length | Mode | Packet Size | Data Pattern | I.F.G. | Packet Type |
| 3m | \*Single burst  100,000,000 | \*64 | \*Random | \*96 bit time | \*Unicast |
| 6m | \*Single burst  100,000,000 | 1518 | All One | Random | Broadcast |
| 6m | \*Single burst  100,000,000 | Random | All Zero | Random | Unknown(1k) |

**40G**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Cable Length | Mode | Packet Size | Data Pattern | Rate Control | Packet Type |
| 3m | \*Single burst  1,000,000,000 | \*64 | \*Random | \*100% | \*Unicast |
| 3m | \*Single burst  1,000,000,000 | 1518 | All One | \*100% | Broadcast |
| 3m | \*Single burst  1,000,000,000 | Random | All Zero | \*100% | Unknown(1k) |

“\*” mark as default setting. VFD3: 2 bytes of Type set to 0800 to avoid 8808 Pause frame

HW Table 1.2.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | | **Test Description** | **Results** | **Notes** |
| Cable Length | 1 | Varying Cable Length at 10Mbps Intra/Inter-RIC |  | 100m, other settings default |
|  | 6m, other settings default |
| 2 | Varying Cable Length at 100Mbps Intra/Inter-RIC |  | 100m, other settings default |
|  | 6m, other settings default |
| 3 | Varying Cable Length at 1000Mbps Intra/Inter-RIC(Module) |  | 100m, other settings default |
|  | 6m, other settings default |
| 4 | Varying Cable Length at 1000M bps Intra/Inter-RIC/ SX Module |  | 275/550m, other settings default |
|  | 3m, other settings default |
| 5 | Varying Cable Length at 1000M bps Intra/Inter-RIC / LX Module |  | 5Km(attenuator), other settings default |
|  | 3m, other settings default |
| 6 | Varying Cable Length at 10G/ Intra/Inter-RIC |  | 1m, other settings default |
|  | 3m, other settings default |
|  | 6m, other settings default |
| 7 | Varying Cable Length at 40G Intra/Inter-RIC (40G AVAGO AFBR-79EIDZ) |  | 3m, other settings default  /\* base on 40G table |
| Data pattern | 1 | Varying Data Pattern at 10M/100M/1000Mbps intra/Inter-RIC |  | All Zero, other settings default |
|  | Random, other settings default |
|  | All One, other settings default |
| 2 | Varying Data Pattern at 1000Mbps  Intra/Inter-RIC/ SX Module |  | All Zero, other settings default |
|  | Random, other settings default |
|  | All One, other settings default |
| 3 | Varying Data Pattern at 1000Mbps  Intra/Inter-RIC/ LX Module |  | All Zero, other settings default |
|  | Random, other settings default |
|  | All One, other settings default |
| 4 | Varying Data Pattern at 10GGbps intra/Inter-RIC |  | All Zero, other settings default |
|  | Random, other settings default |
|  | All One, other settings default |
| 5 | Varying Data Pattern at 40Gbps intra/Inter-RIC (40G AVAGO AFBR-79EIDZ) |  | All Zero, other settings default |
|  | Random, other settings default |
|  | All One, other settings default |
| Packet Size | 1 | Varying Packet Size at 10M/100M/1000Mbps intra/Inter-RIC |  | Random, other settings default |
|  | 64 bytes, other settings default |
|  | 1518 bytes, other settings default |
|  | Jumbo, other settings default |
| 2 | Varying Packet Size at 1000Mbps  Intra/Inter-RIC/ SX Module |  | Random, other settings default |
|  | 64 bytes, other settings default |
|  | 1518 bytes, other settings default |
|  | Jumbo, other settings default |
| 3 | Varying Packet Size at 1000Mbps  Intra/Inter-RIC/ LX Module |  | Random, other settings default |
|  | 64 bytes, other settings default |
|  | 1518 bytes, other settings default |
|  | Jumbo, other settings default |
| 4 | Varying Packet Size at 10G  intra/Inter-RIC |  | Random, other settings default |
|  | 64 bytes, other settings default |
|  | 1518 bytes, other settings default |
|  | Jumbo, other settings default |
| 5 | Varying Packet Size at 40Gbps intra/Inter-RIC (40G AVAGO AFBR-79EIDZ) |  | Random, other settings default |
|  | 64 bytes, other settings default |
|  | 1518 bytes, other settings default |
|  | Jumbo, other settings default |
| Packet Rate | 1 | Varying Packet Rate at 10M/100M/1000Mbps Intra/Inter-RIC |  | 96 bits time, other settings default |
|  | Random, other settings default |
| 2 | Varying Packet Rate at 1000M bps Intra/Inter-RIC/ SX Module |  | 96 bits time, other settings default |
|  | Random, other settings default |
| 3 | Varying Packet Rate at 1000M bps Intra/Inter-RIC / LX Module |  | 96 bits time, other settings default |
|  | Random, other settings default |
| 4 | Varying Packet Rate at 10G bps Intra/Inter-RIC |  | Wire-speed, other settings default |
| 5 | Varying Packet Rate at 40G bps Intra/Inter-RIC (40G adage afbr-79eidz) |  | Wire-speed, other settings default |
| Packet Type | 1 | Varying Packet Type at 10M/100M/1000Mbps  Intra/Inter-RIC |  | Unicast(address table full), other settings default |
|  | Broadcast , other settings default |
|  | Unknown(1k), other settings default  Type set to: 0x809b, 0x8137, 0x8863,  0x8864, 0x86DD |
| 2 | Varying Packet Type at 1000M bps Intra/Inter-RIC/ SX Module |  | Unicast, other settings default |
|  | Broadcast, other settings default |
|  | Unknown(1k), other settings default  Type set to: 0x809b, 0x8137, 0x8863,  0x8864, 0x86DD |
| 3 | Varying Packet Type at 1000M bps Intra/Inter-RIC / LX Module |  | Unicast, other settings default |
|  | Broadcast, other settings default |
|  | Unknown(1k), other settings default  Type set to: 0x809b, 0x8137, 0x8863,  0x8864, 0x86DD |
| 4 | Varying Packet Type at 10G bps  Intra/Inter-RIC |  | Unicast (address table depends on spec.), other settings default |
|  | Broadcast , other settings default |
|  | Unknown(1k), other settings default  Type set to: 0x809b, 0x8137, 0x8863,  0x8864, 0x86DD |
| 5 | Varying Packet Type at 40G bps  Intra/Inter-RIC (40G AVAGO AFBR-79EIDZ) |  | Unicast (address table depends on spec.), other settings default |
|  | Broadcast, other settings default |
|  | Unknown(1k), other settings default  Type set to: 0x809b, 0x8137, 0x8863,  0x8864, 0x86DD |

**Ethernet\_II**

**Ipv4 ->0x0800**

**IPv6->0x86DD**

**ARP->0x0806**

**RARP->0x8035**

**Appple Talk->0x809b**

**NetWare IPX/SPX->0x8137**

**PPPoE Discovery Stage->0x8863**

**PPPoE Session Stage->0x 8864**

* 1. **Switch function and performance Test**

Buffer Size

Equip.: IXIA 2 ports, IxExplorer

Duration: 20 **minutes**

Objective: Test how many packets can be sent before dropped, and the result should be confirmed as follow Table

Note: IXIA in Port 2 keep sending pause frames, IXIA in port1 send broadcast packets continuous with wire speed (Buffer size follow table packet length), and IXIA in port2 send pause frames with Pause Quanta 65535 and wire speed. Both IXIA ports inject packets at the same time for 3 seconds and then stop both IXIA ports traffic at the same time. Check how many Packets are received in port2.

**This test case is only for reference only.** /\*Enable FC on the DUT

**1000/100/10M UTP port Buffer size:**

HW Table 1.3.1

(A) Maximum Burst Size, Effective Buffer Size 100Mb to 10Mb

|  |  |
| --- | --- |
| **Packet Size** | **Buffer Size (Packets)** |
| 64 bytes |  |
| 512 bytes |  |
| 1024 bytes |  |
| 1518 bytes |  |

(B) Maximum Burst Size, Effective Buffer Size 1000Mb to 10Mb

|  |  |
| --- | --- |
| **Packet Size** | **Buffer Size (Packets)** |
| 64 bytes |  |
| 512 bytes |  |
| 1024 bytes |  |
| 1518 bytes |  |

(C) Maximum Burst Size, Effective Buffer Size 1000Mb to 100Mb

|  |  |
| --- | --- |
| **Packet Size** | **Buffer Size (Packets)** |
| 64 bytes |  |
| 512 bytes |  |
| 1024 bytes |  |
| 1518 bytes |  |

(D) Maximum Burst Size, Effective Buffer Size 10G to 1000Mb

|  |  |
| --- | --- |
| **Packet Size** | **Buffer Size (Packets)** |
| 64 bytes |  |
| 512 bytes |  |
| 1024 bytes |  |
| 1518 bytes |  |

(E) Maximum Burst Size, Effective Buffer Size 40G to 10G

|  |  |
| --- | --- |
| **Packet Size** | **Buffer Size (Packets)** |
| 64 bytes |  |
| 512 bytes |  |
| 1024 bytes |  |
| 1518 bytes |  |

Filtering Rate and Address Rate:

Equip.: 3 IXIA ports

Duration: **1 hour**

Objective: Test the device max. Filtering rate

Note: This test need to be tested at least twice (first trial and second trial), first trial is for DUT to learn the mac address, therefore, monitor port will received all packets. Once the DUT learned the mac address, sent the packets again to see the packets will flooding to the other ports or not (other ports should not receive the packets).

/\* the inject traffic speed rate = max mac address number / 2

(For example, if the max mac address number = 3000000, set single burst packet count = 15000 and send 15000 packets per second)

/\* the inject traffic speed rate of second trial is wire speed

1. Burst packets (packet count and rate control=59523810 frames/sec) with 40G wire speed for once

Burst packets (packet count and rate control =14881000 frames/sec) with 10G wire speed for once

Burst packets (packet count and rate control =1488100 frames/sec) with 1G wire speed for once

Burst packets (packet count and rate control =148810 frames/sec) with 100M wire speed for once

Burst packets (packet count and rate control =14880 frames/sec) with 10M wire speed for once

And DA=SA. (Increment max mac address table: depends on each device)

1. Burst packets (packet count and rate control =59523810 frames/sec) with 40G wire speed for once

Burst packets (packet count and rate control =14881000 frames/sec) with 10G wire speed for once

Burst packets (packet count and rate control =1488100 frames/sec) with 1G wire speed for once

Burst packets (packet count and rate control =148810 frames/sec) with 100M wire speed for once

Burst packets (packet count and rate control =14880 frames/sec) with 10M wire speed for once

And DA= previous SA, SA: 000000000002, DA: 000000000001(Increment max mac address table: depends on each device)

The device fails if 10/100M/1000M ports receive other ports Packets

(Except the first one learning pkt and it will be repeated)

1. Burst packets (=max mac address frames/sec) with 40G wire speed for once

Burst packets (=max mac address frames/sec) with 10G wire speed for once

Burst packets (=max mac address frames/sec) with 1G wire speed for once

Burst packets (=max mac address frames/sec) with 100M wire speed for once

Burst packets (=max mac address frames/sec) with 10M wire speed for once

And DA=SA. (Increment max mac address table: depends on each device)

HW Table 1.3.2

|  |  |  |
| --- | --- | --- |
| **Item** | **Test Description** | **Result** |
| 1.3.2.a | Filtering rate DA=SA |  |
| 1.3.2.b | Filtering rate DA= previous SA |  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Item** | **Test Description** | **1st time** | **2nd time** | **3rd time** | **4th time** | **5th time** | **6th time** | **7th time** | **8th time** | **9th time** | **10th time** |
| 1.3.2.c  40G | Filtering rate  DA=SA |  |  |  |  |  |  |  |  |  |  |
| 1.3.2.c  10G | Filtering rate  DA=SA |  |  |  |  |  |  |  |  |  |  |
| 1.3.2.c  1G | Filtering rate  DA=SA |  |  |  |  |  |  |  |  |  |  |

/\* For item b, need to minus the flooding packet and check the result is correct

Forwarding the Reserved Multicast Packets

Forwarding the BPDU packets

Equip.: Sniffer

Duration: 10 minutes

Objective: Test the device must forward the BPDU packets.

Connect the device’s port 1 to “Main Trunk”, and port 2 connects to Sniffer.

The device failure if the Sniffer cannot receive BPDU packets.

|  |  |  |
| --- | --- | --- |
| **Item** | **Test Description** | **Result** |
| 1.3.3 | Spanning tree disable |  |
| 1.3.4 | Spanning tree enable |  |

Forwarding the Reserved multicast packets

Equip.: IXIA

Duration: 30 minutes

Objective: Send increase DA (from 0180c2000000 to 0180c200003F) packets to the port1 of device.

I.F.G. set to 1 packets/sec, and 100% loading; set the configuration of the DUT to factory default first. (forward-unregistered)

Device should forward those multicast packets except the pause frame (01-80-c2-00-00-01).

Fill in the “Pass” or “Drop” as the test result.

HW Table 1.3.5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DA | Packet Size/ Data Pattern | Unmanaged Switch expectation  result | Smart switch expectation  result | Managed switch expectation  result | Testing result  Forward-All | Testing result  Forward-  unregistered | Testing result  Filter-  unregistered |
| 0180C2000000-Bridge Group Address | Random | Pass | Drop (\*1) | Drop (\*1) |  |  |  |
| 0180C2000001- PAUSE operation | Random | H/W process | H/W process | H/W process |  |  |  |
| 0180C2000002- 802.3ad Slow Protocol | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000003- 802.1x PAE Address | Random | Pass | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000004- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000005- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000006- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000007- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000008- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000009- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C200000A- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C200000B- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C200000C- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C200000D- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C200000E- LLDP IEEE Std 802.1ab Link Layer Discovery Protocol multicast Address | Random | Pass | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C200000F- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000010- All LANs STP  Group Address | Random | Drop | Drop(\*1) | Drop (\*1) |  |  |  |
| 0180c2000011~0180c2000013  Reserved | Random | Pass | Pass | Pass(\*1) |  |  |  |
| 0180c2000014~0180c2000015 | Random | Pass | Pass | Pass(\*1) |  |  |  |
| 0180c2000016~0180c200001F | Random | Pass | Pass | Pass(\*1) |  |  |  |
| 0180C2000020- GARP Address | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000021- GVRP Address | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000022- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000023- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000024- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000025- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000026- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000027- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000028- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000029- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C200002A- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C200002B- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C200002C- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C200002D- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C200002E- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C200002F- Reserved | Random | Drop | Drop(\*1) | Drop(\*1) |  |  |  |
| 0180C2000031- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C2000032- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C2000033- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C2000034- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C2000035- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C2000036- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C2000037- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C2000038- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C2000039- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C200003A- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C200003B- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C200003C- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C200003D- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C200003E- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |
| 0180C200003F- Reserved | Random | Pass | Pass(\*1) | Pass(\*1) |  |  |  |

1) Drop (\*1) means that this packet should be dropped by HW, but can be captured by CPU to process and re-send by

Software if needed (by command).

2) Pass (\*1) means that this packet should be passed, but if the switch support related function, it should be captured by CPU and dropped.

**Note:**

**Reserved Multicast addresses**

01-80-C2-00-00-00 --Bridge Group Address

01-80-C2-00-00-01 --IEEE Std. 802.3x Full Duplex PAUSE operation

01-80-C2-00-00-02 --IEEE Std. 802.3ad Slow\_Protocols\_Multicast address

01-80-C2-00-00-03 -- IEEE P802.1X PAE address

01-80-C2-00-00-04 ~ 01-80-C2-00-00-0F Reserved for future standardization

01-80-C2-00-00-0E -- LLDP

01-80-C2-00-00-10 -- All LANs Bridge Management Group Address

**GARP Application addresses**

01-80-C2-00-00-20 --GMRP

01-80-C2-00-00-21 -- GVRP

01-80-C2-00-00-22 –2F Reserved

802.1ag PDU CCM/LTM

01-80-C2-00-00-3y y:1-F

MAC addresses supported/Illegal MAC Test

Equip.: IXIA

Duration: 30 minutes

Objective: Check MAC addresses support and illegal MAC address cannot be learned.

1. Address table size, port 1 send increased SA to switch for learning, and send increased DA. MAC addresses packets from the other port of the switch, check the number of MAC address before broadcast. The device fails if the number < spec.

Result: Passed

1. Illegal Mac address cannot be learned as an SA (all zeros, all ones, broadcast/multicast); the device fails if any one of above MAC learnt.

Port1 (SA：all zeros/all ones/broadcast/multicast, DA：Broadcast)

Port2 (SA：000000000001, DA：all zeros/all ones/broadcast/multicast)

To see the frames have been filtered or not

Result: Passed

1. Illegal frame’s SA cannot be learned (with Oversize, Undersize, CRC, Alignment, and Symbol); the device fails if any one of above MAC learnt. The device fails if any one of above SA learned.

Port1 (SA：000000000001, DA： broadcast, with oversize/undersize length, CRC, alignment, symbol)

Port2 (SA： 000000000002, DA： 000000000001)

To see the frames have been flooding or not, SA not learned.

Result: Passed

/\* Alignment and Symbol need to connect to Smart bit 10/100 port to test if the RJ45 port can link up with 100M

HW Table 1.3.6

|  |  |  |
| --- | --- | --- |
| **Item** | **Test Description** | **Result** |
| 1.3.6.a | MAC address |  |
| 1.3.6.b | SA 000000000000 |  |
| 1.3.6.b | SA 111111111111 |  |
| 1.3.6.b | SA FFFFFFFFFFFF |  |
| 1.3.6.b | SA 010000000000 |  |
| 1.3.6.c | Oversize SA |  |
| 1.3.6.c | Undersize SA |  |
| 1.3.6.c | CRC SA |  |
| 1.3.6.c | Alignment SA |  |
| 1.3.6.c | Symbol SA |  |

Aging Time

Equip.: IXIA

Duration: 15 minutes

Objective: Check the Aging period

a) One MAC address learned.

b) And send unicast packets, and note the time when the packets are broadcast out.

Default time should (T) = 300secs or by product spec.

|  |  |  |
| --- | --- | --- |
| **Item** | **Test Description** | **Result** |
| 1.3.7 | Aging Time |  |

Overwrite the Forwarding Table

Equip.: IXIA

Duration: 30 minutes

Objective: Send increase SA to device port 1. MAC addresses packets more than address table support (30K) to the switch port for learning. Send new SA MAC address to device port 2.

Verify that the new MAC will flood if the forwarding table is full, and the old MAC entries still working till aging.

|  |  |  |
| --- | --- | --- |
| **Item** | **Test Description** | **Result** |
| 1.3.8 | MAC table full, new MACs will not affect the old MAC entries. |  |

Note: the overwrite characteristic depend on chip spec.

Update the Forwarding Table

Equip.: IXIA

Duration: 30 minutes

Objective: Test the duplicate SA MAC (000000000001) address in different port (1, 2).

Send to port1 first, then port2.

Send a packet with DA 000000000001 from port 3. The traffic will go to port2.

The device fails if the MAC address cannot be learned in new port (2).

|  |  |  |
| --- | --- | --- |
| **Item** | **Test Description** | **Result** |
| 1.3.9 | Update the Forwarding Table |  |

Learning Algorism Test

Equip.: IXIA

Duration: 10 minutes

Objective: Check addresses hash, send packets with DA 002000000000, SA. 001000000000 to port 1,

And send packets with DA 001000000000, SA 002000000000 to port 2.

/\* the inject traffic speed rate = max mac address number / 2

|  |  |  |
| --- | --- | --- |
| **Item** | **Test Description** | **Result** |
| 1.3.10 | Learning Algorism Test |  |

The device fails if other ports (not 1, 2) receive many flooded packets.

Filter Illegal Frames Test

Equip.: IXIA

Duration: 2 hours

Test Parameters:

HW Table 1.3.11

|  |  |  |  |
| --- | --- | --- | --- |
| **Frame Size/ Background** | **Load** | **Port Error** | **Result** |
| Random/Random | 100% | CRC |  |
| Random/Random | 100% | Alignment(i.e. the packet is not an integer number of bytes in length) |  |
| Random/Random | 100% | Symbol |  |
| Random/Random | 100% | Dribble Bit |  |
| Random/Random | 100% | Dribble Bit + CRC |  |
| Random/Random | 100% | Dribble Bit + Alignment |  |
| Random/Random | 100% | Dribble Bit + CRC + Alignment |  |
| 63 /Random | 100% | Undersize |  |
| Jumbo (length 9600 bytes) | 100% | Jumbo frames |  |
| Oversize 9601 and above | 100% | Oversize |  |
| Fragment | 100% | Collision |  |

a.) IXIA 2ports <-> device two ports

b.) Send error packets (Jabbers and fragments) to the switch.

c.) These error packets should not affect other port’s TX/RX in the switch

Jabber sends over 8K/CRC length packets, Fragment: use a hub to make the collision (2 ports

TX/RX the third port connects to DUT.

**Note:** 1.the legal packet size depends on chip spec.

2. Dribble bit and Alignment error packets only support at 10/100M bps.

E.g. Galileo switch chip configure legal frame size 64-1536 bytes.

/\* Alignment, Symbol and Dribble Bit error need connect to Smart bit 10/100 port to test it if the RJ45 port can link up with 100M

* 1. **Flow Control Tests**

Pause Frame in Flow Control Test

Equip.: IXIA.

Duration: **3 hours**

Objective: The objective of this test is to verify the ability of the DUT to operate with expected packet loss rates in environments where flow control is used.

For Fast Ethernet, 1 PAUSE frame every second will yield a reduction in throughput to approximately 66 Mbps.

MAC Control Parameter (pause Quanta) will be set to 0xFFFF (65535).

HW Table 1.4.1.1

|  |  |  |
| --- | --- | --- |
| **Test Description** | **P/F** | **Note** |
| **40GMb – 40GMb** Send 40 Pause frame/sec continuous (MAC Control Parameter: 0XFFFF) | | |
| 64 Byte packets transmitted |  | The rate must be **59523810**\*97.0% |
| 65 Byte packets transmitted |  | The rate must be **58823530**\*97.0% |
| 128 Byte packets transmitted |  | The rate must be **33783785**\*97.0% |
| 256 Byte packets transmitted |  | The rate must be **18115943**\*97.0% |
| 512 Byte packets transmitted |  | The rate must be **9398496**\*97.0% |
| 1024 Byte packets transmitted |  | The rate must be **4789272**\*97.0% |
| 1280 Byte packets transmitted |  | The rate must be **3846154**\*97.0% |
| 1518 Byte packets transmitted |  | The rate must be **3250973**\*97.0% |
| **10GMb – 10GMb** Send 10 Pause frame/sec continuous (MAC Control Parameter: 0XFFFF) | | |
| 64 Byte packets transmitted |  | The rate must be 14880952\*96.4% |
| 65 Byte packets transmitted |  | The rate must be 1470588\*96.4% |
| 128 Byte packets transmitted |  | The rate must be 844595\*96.4% |
| 256 Byte packets transmitted |  | The rate must be 452899\*96.4% |
| 512 Byte packets transmitted |  | The rate must be 234962\*96.4% |
| 1024 Byte packets transmitted |  | The rate must be119732\*96.4% |
| 1280 Byte packets transmitted |  | The rate must be 96154\*96.4% |
| 1518 Byte packets transmitted |  | The rate must be 81274\*96.4% |
| **1000Mb – 1000Mb** Send 1 Pause frame/sec continuous (MAC Control Parameter: 0XFFFF) | | |
| 64 Byte packets transmitted |  | The rate must be 1488095\*93.4% |
| 65 Byte packets transmitted |  | The rate must be 1470588\*93.4% |
| 128 Byte packets transmitted |  | The rate must be 844595\*93.4% |
| 256 Byte packets transmitted |  | The rate must be 452899\*93.4% |
| 512 Byte packets transmitted |  | The rate must be 234962\*93.4% |
| 1024 Byte packets transmitted |  | The rate must be 119732\*93.4% |
| 1280 Byte packets transmitted |  | The rate must be 96154\*93.4% |
| 1518 Byte packets transmitted |  | The rate must be 81274\*93.4% |
| Opcode not equal to 0x00-01 |  | 00 01  IXIA make a pause with opcode not equal to 0x0001,  1. DUT doesn’t deal the packet.  2. DUT doesn’t forward the packet. |
| Receive PAUSE frames of incorrect size |  | 64 bytes  IXIA make a pause with size<64,  1. DUT doesn’t forward the packet. |
| The device shall not forward pause frames. |  |  |
| The device can receive Pause frame with zero pause time and can work normal. |  | IXIA make a pause with slot time=0,  1. DUT doesn’t forward the packet. |
| The device can transmit valid 64 bytes Pause frames. |  |  |
| Pause frame SA(DUT’s MAC address) |  | DUT forward Pause frame |
| Pause frame reserved field (all zeroes) |  | Default is 0 |

Flow Control/Backpressure Tests

Equip.: IXIA

Duration: 30 minutes

Objective: To check the switch flow control ability.

Device with flow control enabled

A. Many to one: to see the flow control activity works.

Configure setup as:

IXIA, many to one, transmit single burst, port configuration as following:

Packet size 64, 1518, 9600 Load 100%

If packet loss checks other frame size: 128, 256, 512, 1024, and 1248

(Learning: Before each iteration, Delay after Transmit =300 sec.)

B. High speed to low speed :( 1000M->100M->10M)

High speed port transmit unicast/unknown/broadcast/multicast to low speed port in 64/1518/9600

Packet size, IXIA, high to low, transmit single burst 10 sec., port1 (broadcast/multicast/unknown/unicast-learnt), receiving ports link at high, middle and low speed.

Packet size 64, 1518, 9600 Load 100%

Check the flow control behavior.

C. One pair unicast, the third port transmits unknown/ broadcast/ multicast/pause frames.

In packet size 64/1518/9600bytes and check the flow controls behavior.

Priority Queues Test

Equip.: IXIA

Duration: **30 minutes**

Objective: Connect ports of device to IXIA (Flow control Enable/Disable).

A. Send packets (as table1.4.3.1) to the port9 of device, packet size 64, 1518, jumbo.

Check the IXIA counter.

Note: Number of queues supports depends on chip spec.

The device Pass if the receiving frames of port9 is following the spec.

HW Table1.4.1.3(a)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DA** | **SA** | **Packet Size** | **Priority in Tag** | **I.F.G.** | **Send mode** |
| 000000000009 | 000000000001 | Random/IP | 7/VID=2 | 96 bit time | Continue |
| 000000000009 | 000000000002 | Random/IP | 6/VID=2 | 96 bit time | Continue |
| 000000000009 | 000000000003 | Random/IP | 5/VID=2 | 96 bit time | Continue |
| 000000000009 | 000000000004 | Random/IP | 4/VID=2 | 96 bit time | Continue |
| 000000000009 | 000000000005 | Random/IP | 3/VID=2 | 96 bit time | Continue |
| 000000000009 | 000000000006 | Random/IP | 2/VID=2 | 96 bit time | Continue |
| 000000000009 | 000000000007 | Random/IP | 1/VID=2 | 96 bit time | Continue |
| 000000000009 | 000000000008 | Random/IP | 0/VID=2 | 96 bit time | Continue |
| FFFFFFFFFFFF | 000000000009 | Random/IP | VID=2 | 96 bit time | 1 frame/sec |

B. Connect 5 ports of device to IXIA.

(Port1-5 1G, Flow control Enable/Disable)

(Port 1-4 to generate four tag priority queues)

Send packets (as Table 1.4.3.2) to the port5 of device.

Check the IXIA counter.

Note: Number of queues support depends on chip spec.

HW Table 1.4.1.3(b)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DA** | **SA** | **Packet Size** | **Priority in Tag** | **I.F.G.** | **Send mode** |
| 000000000005 | 000000000001 | 4 IP streams | 7/VID=2  5/VID=2  3/VID=2  1/VID=2 | 96 bit time | Continue |
| 000000000005 | 00000000002 | 4 IP streams | 7/VID=2  5/VID=2  3/VID=2  1/VID=2 | 96 bit time | Continue |
| 000000000005 | 00000000003 | 4 IP streams | 7/VID=2  5/VID=2  3/VID=2  1/VID=2 | 96 bit time | Continue |
| 000000000005 | 00000000004 | 4 IP streams | 7/VID=2  5/VID=2  3/VID=2  1/VID=2 | 96 bit time | Continue |
| FFFFFFFFFFFF | 000000000005 | Random | Random | 96 bit time | 1 frame/sec |

Check the receiving frame in port 5 is following the spec. high queue first. Packet size 64, 1518, jumbo.

Result: PASS, port5 only received packets of “Priority in Tag =>7/VID=2”

Note: In this device support 4 queues and mode is strict, the mapping is:

Priority 1, 2 ->Class 0

Priority 0, 3 ->Class 1

Priority 4, 5 ->Class 2

Priority 6, 7 ->Class 3

Buffer Clean Test

Equip: IXIA

Duration: 30 minute

Objective: To ensure the device can be recovery from Pause Frames issued by loop or lots times of plug in/out cable in a very short time. /\* Make sure flow control is on and SPT is disabled

a.

Making two wires do port-pair transmission. One port is fixed and plug out/in the other port while the connection is established. Repeat this action to see if pause frame issued from port1 will stop the transmission of port2 or not.

b.

Making two ports do port pair transmission. Configure the third port do broadcast (DA= FF-FF-FF-FF-FF-FF), and take a cross-over wire to make a loop (Intra/Inter ASIC).

After free the loop. Check the device can restore RX/TX traffic or pause frames stop other abnormal symptom such as ports’ traffic.

|  |  |  |
| --- | --- | --- |
| **Item** | **Test Description** | **Result** |
| 1.4.1.5 | Buffer Clean Test |  |
| 1.4.1.6 | Buffer Clean Test |  |

10M/100M/1000M/10G/40G Hybrid Test

Equip: IXIA

Duration: 1 hour

Objective: Pair-wise with different speeds (10/100/1000M/10G/40G), packet 64, 1518, Jumbo.

|  |  |  |
| --- | --- | --- |
| **Item** | **Test Description** | **Result** |
| 1.4.1.7 | 10M/100M/1000M Hybrid Test |  |
| 1.4.1.8 | 10G/40G Hybrid Test |  |

Broadcast Mesh Test

Equip: IXIA

Duration: 1 hour

Objective: To ensure the broadcast/multicast can run mesh cases, Flow control can be work well in mesh condition.

1. Send full mesh of broadcast streams burst in 100,000,000 packets/load 100% in 1000M

Mode (FC on).( 1,000,000,000 packets/load 100% in 40G)

2. Send 10 low speed of broadcast streams burst in 1,000,000 packets/load 100%, to high

Speed port;

3. 10 high speed of broadcast streams burst in 1,000,000 packets/load 10% to other

Ports (same speed) flow control off.

|  |  |  |
| --- | --- | --- |
| **Item** | **Test Description** | **Result** |
| 1.4.1.9 | Broadcast mesh test |  |

* 1. **Power ON/OFF Test**

Test Equip.: IXIA

Duration: **12 hours**

Objective: To determine after stress power up/down, device still workable.

1 During the traffic load, power off/on the switch 100 times. Then check device can boot up/forwarding

packet/no LED abnormity.

2 Power off/on many times continuously within ultra-short time, then check device can boot up/forwarding

packet/no LED abnormity. Need to repeat this item 20 times.

Pass/fail criteria: No hung up happen or LED abnormity after the stress power on/off.

|  |  |  |
| --- | --- | --- |
| **Item** | **Test Description** | **Result** |
| 1.5.1 | Power off/on 100 times |  |
| 1.5.2 | Power off/on continuously within ultra-short time |  |

1. 1. **External Component Test - FAN**

Objective: To ensure DUT can display fan information correctly and system functions regarding fan can work normally.

Pass/fail criteria: Behavior should be the same as specified in spec.

**TC\_ID: 2.1.1**

|  |  |
| --- | --- |
| Purpose | Verify that system display correct information for fan. |
| Procedure | 1. Compare default display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.1.2**

|  |  |
| --- | --- |
| Purpose | Verify that system display correct information for fan after hot-remove it. |
| Procedure | 1. Hot-remove Fan. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.1.3**

|  |  |
| --- | --- |
| Purpose | Verify that system display correct information for fan after hot-insert it. |
| Procedure | 1. Hot-insert Fan. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.1.4**

|  |  |
| --- | --- |
| Purpose | Verify that system can record log for Fan correctly. |
| Procedure | 1. Make Log for Fan then check. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.1.5**

|  |  |
| --- | --- |
| Purpose | Verify that system can send trap for Fan correctly. |
| Procedure | 1. Make Trap for Fan then check. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.1.6**

|  |  |
| --- | --- |
| Purpose | Verify that system can send alarm for Fan correctly. |
| Procedure | 1. Make Alarm for Fan then check. |
| Expected Result | 1. Should implement same as spec design. |

* 1. **External Component Test - RTC**

Objective: To ensure RTC function can work correctly

**TC\_ID: 2.2.1**

|  |  |
| --- | --- |
| Purpose | Verify that system can display correctly info for RTC. |
| Procedure | 1. Compare default display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.2.2**

|  |  |
| --- | --- |
| Purpose | Verify that system can work normally after set /save/reboot. |
| Procedure | 1. Modify RTC value 2. Save configure. 3. Reboot DUT. 4. After boot up, check RTC value. |
| Expected Result | 1. System should work normally. |

* 1. **External Component Test - LED**

**TC\_ID: 2.3.1**

|  |  |
| --- | --- |
| Purpose | Verify that system display correctly LEDs for ALM LED. |
| Procedure | 1. Compare ALM LED with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.2**

|  |  |
| --- | --- |
| Purpose | Verify that system display correctly LEDs for SYS LED. |
| Procedure | 1. Compare SYS LED with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.3**

|  |  |
| --- | --- |
| Purpose | Verify that system display correctly LEDs for MST LED. |
| Procedure | 1. Compare MST LED with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.4**

|  |  |
| --- | --- |
| Purpose | SPD(Speed)  Verify that the LED blinking status matches with the speed that the interface is connected with. |
| Procedure | 1. Compare SPD(Speed) LED with spec. 2. Check when speed is 10Mbps. 3. Check when speed is 100Mbps. 4. Check when speed is 1000Mbps. 5. Check when speed is 10G 6. Check when speed is 40G |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.5**

|  |  |
| --- | --- |
| Purpose | DPX(Duplex)  Verify that the light display matches with the duplex mode (Full/Half) |
| Procedure | 1. Compare DPX(Duplex) LED with spec. 2. Check when DPX(Duplex) is "full duplex". 3. Check when DPX(Duplex) is "half duplex". |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.6**

|  |  |
| --- | --- |
| Purpose | ADM(Administrate)  Verify that the light display matches with the status (In use/Unused) |
| Procedure | 1. Compare ADM(Administrate) LED with spec. 2. Check when ADM(Administrate) is "In Use". 3. Check when ADM(Administrate) is "Unused". |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.7**

|  |  |
| --- | --- |
| Purpose | POE(Power over Ethernet)  Verify that the light display matches with the status  (Support PoE/Not Support PoE/Connected/Disconnected) |
| Procedure | 1. Compare POE (Power over Ethernet) LED with spec. 2. Check when the interface supports PoE and connect with PoE equipment. 3. Check when the interface supports PoE but doesn't connect with PoE equipment. 4. Check when the interface doesn't support PoE. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.8**

|  |  |
| --- | --- |
| Purpose | Link/Activity  Verify that the light display matches with the status  (Linked/Unlinked/With Traffic/Without Traffic) |
| Procedure | 1. Compare Link/Activity LED with spec. 2. Check Link/Activity LED when the interface is linked up. 3. Check Link/Activity LED when the interface is linked up and transmitting traffic. 4. Check Link/Activity LED when the interface is linked down. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.9**

|  |  |
| --- | --- |
| Purpose | Verify that when fans work fine, system LED lights display correctly. |
| Procedure | 1. When fan works, compare FAN LED with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.10**

|  |  |
| --- | --- |
| Purpose | Verify that when fans fail, system LED lights display correctly. |
| Procedure | 1. When fan fails, compare FAN LED with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.11**

|  |  |
| --- | --- |
| Purpose | Verify that system display correctly LEDs for Power LED. |
| Procedure | 1. Compare Power LED with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.12**

|  |  |
| --- | --- |
| Purpose | Verify that system display correctly LEDs for Console port LED. |
| Procedure | 1. Compare Console port LED with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.13**

|  |  |
| --- | --- |
| Purpose | Verify that system display correctly LEDs for management port LED. |
| Procedure | 1. Compare management port LED with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.14**

|  |  |
| --- | --- |
| Purpose | Verify that CPU, I/O and ASE Modules-Power LEDs work as spec |
| Procedure | 1. Power on DUT 2. Check CPU, I/O and ASE Modules-Power LED |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.15**

|  |  |
| --- | --- |
| Purpose | Verify that CPU, I/O and ASE Modules-System LEDs work as spec |
| Procedure | 1. Power on DUT 2. Check CPU, I/O and ASE Modules-System LED |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.16**

|  |  |
| --- | --- |
| Purpose | Verify that CPU Utilization LED works as spec |
| Procedure | 1. Power on DUT 2. Check CPU Utilization LED |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.17**

|  |  |
| --- | --- |
| Purpose | Verify that CPU Module-Master LED works as spec |
| Procedure | 1. Power on DUT 2. Check CPU Module-Master LED |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.18**

|  |  |
| --- | --- |
| Purpose | Verify that CPU Module-USB/RS-232 LEDs work as spec |
| Procedure | 1. Power on DUT 2. Check CPU Module-USB/ RS-232 LED |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.19**

|  |  |
| --- | --- |
| Purpose | Verify that CPU Module-Compact Flash LED work as spec |
| Procedure | 1. Power on DUT 2. Check CPU Module-Compact Flash LED |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.20**

|  |  |
| --- | --- |
| Purpose | Verify that LED for Fault/Locator works as spec |
| Procedure | 1. Power on DUT 2. Check LED for Fault/Locator |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.3.21**

|  |  |
| --- | --- |
| Purpose | Verify that LED for GBIC-Link works as spec |
| Procedure | 1. Power on DUT 2. Check LED for GBIC-Link |
| Expected Result | 1. Should implement same as spec design. |

* 1. **Platform - Packets Path**

**TC\_ID: 2.4.1**

|  |  |
| --- | --- |
| Purpose | Test the device max. Filtering rate |
| Procedure | 1. Send 1488100 frames/sec in 1000M; Send 148810 frames/sec in 100M; Send 14880 frames/sec in 10M; And DA=SA. 2. Send 1488100 frames/sec in 1000M; Send 148810 frames/sec in 100M; Send 14880 frames/sec in 10M; And DA= previous SA, 000000000002, 000000000001(increment) |
| Expected Result | 1. Pass/fail criteria : The device fails if 10/100M/1000M ports receive other ports Packets (Except the first one learning packet) |

**TC\_ID: 2.4.2**

|  |  |
| --- | --- |
| Purpose | Check MAC addresses support and illegal MAC address cannot be learnt. |
| Procedure | 1. Address table size, port 1 send increased SA to switch for learning, and send increased DA. MAC addresses packets from the other port of the switch, check the number of MAC address before broadcast. 2. Illegal Mac address cannot be learnt as an SA (all zeros, all ones, broadcast/multicast); 3. Illegal frame's SA cannot be learned (with Oversize, Undersize, CRC, Alignment, Symbol); |
| Expected Result | 1. The device fails if the number < spec.. 2. The device fails if any one of above MAC learnt. Port1 (SA： all zeros/all ones/broadcast/multicast, DA： broadcast) Port2 (SA：000000000001, DA： all zeros/all ones/broadcast/multicast) 3. The device fails if any one of above SA learnt. Port1 (SA：000000000001, DA： broadcast, with oversize/undersize length, CRC, alignment, symbol) Port2 (SA：000000000002, DA：000000000001) To see the frames have been flooding or not, SA not learnt. |

**TC\_ID: 2.4.3**

|  |  |
| --- | --- |
| Purpose | Check for the Filter Illegal Frames |
| Procedure | 1. IXIA two ports <-> device two ports. 2. Send error packets (Jabbers and fragments) to the switch. 3. These error packets should not affect other port's TX/RX in the switch   Jabber sends over 8K/CRC length packets, Fragment: use a hub to make the collision (2 ports TX/RX the third port connects to DUT. |
| Expected Result | 1. The legal packet size depends on chip spec.  (e.g. Galileo switch chip configure legal frame size 64-1536 bytes.) |

* 1. **Platform – RPS**

**TC\_ID: 2.5.1**

|  |  |
| --- | --- |
| Purpose | Verify that system can boot up success via RPS. |
| Procedure | 1. Connect minimum power with RPS (one power module) |
| Expected Result | 1. System should boot up success via RPS. |

**TC\_ID: 2.5.2**

|  |  |
| --- | --- |
| Purpose | Verify that system can boot up success via RPS. |
| Procedure | 1. Connect maximum power with RPS (two power module) |
| Expected Result | 1. System should boot up success via RPS. |

**TC\_ID: 2.5.3**

|  |  |
| --- | --- |
| Purpose | Verify that system can work normally after hot-swap master power via RPS. |
| Procedure | 1. Connect at least two powers with RPS. 2. Boot up DUT. 3. Redundancy master power with RPS. |
| Expected Result | 1. System can work normally after hot-swap master power via RPS. |

**TC\_ID: 2.5.4**

|  |  |
| --- | --- |
| Purpose | Verify that system can work normally after hot-swap slaver power via RPS. |
| Procedure | 1. Connect at least two power with RPS. 2. Boot up DUT. 3. Redundancy slaver power with RPS. |
| Expected Result | 1. System can work normally after hot-swap slaver power via RPS. |

* 1. **Platform –CPU path**

**TC\_ID: 2.6.1**

|  |  |
| --- | --- |
| Purpose | Verify that ICMP v4/v6 packet over 520 bytes/Fragmented should be forward |
| Procedure | 1. Two PC ping each other cross DUT. 2. Length of ping packet：64, 1518, 9KBytes. |
| Expected Result | 1. No packet loss, no error packets received. |

**TC\_ID: 2.6.2**

|  |  |
| --- | --- |
| Purpose | Verify that Source IP same as destination IP packets should be forward. |
| Procedure | 1. Two ports send packets DIP=SIP to each other, 2. DA: Unicast/Broadcast/Multicast/Unknown, 3. Length: 64, 1518, 9KBytes. |
| Expected Result | 1. No packet loss, no error packets received. |

**TC\_ID: 2.6.3**

|  |  |
| --- | --- |
| Purpose | Verify that ICMP packet over 1518 bytes cannot be fragmented into two packets. |
| Procedure | 1. Two IXIA ports connect to DUT, 2. IXIA#1 send ICMP packet over 1518 bytes 3. IXIA#2 receive packets. |
| Expected Result | 1. ICMP packet over 1518 bytes cannot be fragmented into two packets. |

**TC\_ID: 2.6.4**

|  |  |
| --- | --- |
| Purpose | Verify that Frist TCP/UDP fragment does not have the full TCP header/SPORT=DPORT/Spec define some bit will be forward. |
| Procedure | 1. Two IXIA ports link DUT, 2. IXIA#1 send TCP header/SPORT=DPORT/Spec defined. 3. IXIA#2 receive packets. |
| Expected Result | 1. Frist TCP/UDP fragment does not have the full TCP header/SPORT=DPORT/Spec define some bit will be forward. |

**TC\_ID: 2.6.5**

|  |  |
| --- | --- |
| Purpose | Verify that DUT drop packet by detecting source MAC address same as destination MAC address |
| Procedure | 1. Two IXIA ports link DUT, IXIA#1 send packets destination MAC = source MAC to DUT 2. IXIA#2 should not receive packets. |
| Expected Result | 1. DUT drop packet by detecting source MAC address same as destination MAC address |

* 1. **Platform –USB**

**TC\_ID: 2.7.1**

|  |  |
| --- | --- |
| Purpose | Verify that the device can boot up via USB successfully. |
| Procedure | 1. Connect USB to DUT. 2. Set boot up with USB then boot up |
| Expected Result | 1. System should boot up success via USB. |

**TC\_ID: 2.7.2**

|  |  |
| --- | --- |
| Purpose | Verify that the device can boot up successfully after put Multiple images on USB. |
| Procedure | 1. Put at least two images on USB. 2. Connect USB to DUT. 3. Set boot up with USB then boot up. |
| Expected Result | 1. System should boot up success via USB. |

**TC\_ID: 2.7.3**

|  |  |
| --- | --- |
| Purpose | Verify that firmware can be upgraded via USB successfully. |
| Procedure | 1. Connect USB to DUT. 2. Set boot up with USB then boot up. |
| Expected Result | 1. Firmware should be upgraded successfully. |

**TC\_ID: 2.7.4**

|  |  |
| --- | --- |
| Purpose | Verify that firmware can be downgraded via USB successfully. |
| Procedure | 1. Connect USB to DUT. 2. Set boot up with USB then boot up. |
| Expected Result | 1. Firmware should be downgraded successfully. |

* 1. **Platform – Console**

**TC\_ID: 2.8.1**

|  |  |
| --- | --- |
| Purpose | Check that the DUT can be controlled via console port. |
| Procedure | 1. Connect DUT via Console port. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.8.2**

|  |  |
| --- | --- |
| Purpose | Check that the DUT can be controlled via Mini-USB console port. |
| Procedure | 1. Connect DUT via Mini-USB Console port. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.8.3**

|  |  |
| --- | --- |
| Purpose | Check that the DUT can be controlled using Tool\_SecureCRT. |
| Procedure | 1. Connect DUT using Tool\_SecureCRT. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.8.4**

|  |  |
| --- | --- |
| Purpose | Check that the DUT can be controlled using Tool\_Tera Term. |
| Procedure | 1. Connect DUT using Tool\_Tera Term. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.8.5**

|  |  |
| --- | --- |
| Purpose | Check that the DUT can be controlled using Tool\_Procomm Plus Terminal. |
| Procedure | 1. Connect DUT using Tool\_Procomm Plus Terminal. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.8.6**

|  |  |
| --- | --- |
| Purpose | Verify that can control DUT via Baud Rate\_9600. |
| Procedure | 1. Connect DUT via Baud Rate\_9600. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.8.7**

|  |  |
| --- | --- |
| Purpose | Verify that can control DUT via Baud Rate\_19200. |
| Procedure | 1. Connect DUT via Baud Rate\_19200. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.8.8**

|  |  |
| --- | --- |
| Purpose | Verify that can control DUT via Baud Rate\_38400. |
| Procedure | 1. Connect DUT via Baud Rate\_38400. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.8.9**

|  |  |
| --- | --- |
| Purpose | Verify that can control DUT via Baud Rate\_152000. |
| Procedure | 1. Connect DUT via Baud Rate\_152000. |
| Expected Result | 1. Should implement same as spec design. |

* 1. **Platform – Flash/SD**

**TC\_ID: 2.9.1**

|  |  |
| --- | --- |
| Purpose | Verify that can boot up via FLASH/SD success. |
| Procedure | 1. Set boot up with FLASH/SD then boot up |
| Expected Result | 1. System should boot up via FLASH/SD successfully. |

**TC\_ID: 2.9.2**

|  |  |
| --- | --- |
| Purpose | Verify that can boot up successfully after put Multiple images on FLASH/SD. |
| Procedure | 1. Put at least two images on FLASH/SD. 2. Set boot up with FLASH/SD then boot up. |
| Expected Result | 1. System should boot up via FLASH/SD successfully. |

**TC\_ID: 2.9.3**

|  |  |
| --- | --- |
| Purpose | Verify that system can work normally if maximum images (according to spec.) are put on FLASH/SD. |
| Procedure | 1. Put Maximum images on FLASH/SD. (follow spec.) 2. Set boot up with FLASH/SD then boot up. |
| Expected Result | 1. System should boot up via FLASH/SD successfully. |

**TC\_ID: 2.9.4**

|  |  |
| --- | --- |
| Purpose | Verify that firmware can be upgraded via FLASH/SD successfully. |
| Procedure | 1. Set boot up with FLASH/SD then boot up. |
| Expected Result | 1. Firmware should be upgraded successfully. |

**TC\_ID: 2.9.5**

|  |  |
| --- | --- |
| Purpose | Verify that firmware can be downgraded via FLASH/SD successfully. |
| Procedure | 1. Set boot up with FLASH/SD then boot up. |
| Expected Result | 1. Firmware should be downgraded successfully. |

* 1. **Platform – Memory**

**TC\_ID: 2.10.1**

|  |  |
| --- | --- |
| Purpose | Verify that system can display memory information correctly. |
| Procedure | 1. Compare default display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.10.2**

|  |  |
| --- | --- |
| Purpose | Verify that system can work normally after boot up with multiple images.  Note : at least two images |
| Procedure | 1. Put at least two images on FLASH/SD. 2. Set boot up with FLASH/SD then boot up. |
| Expected Result | 1. System should work normally. |

**TC\_ID: 2.10.3**

|  |  |
| --- | --- |
| Purpose | Verify that the device can boot up successfully after being powered off/on many times continuously within ultra-short time.  Note: at least repeat this item 20 times. |
| Procedure | 1. Put at least two images on FLASH/SD. 2. Set boot up with FLASH/SD then boot up. 3. Power off/on many times continuously within ultra-short time. (at least repeat this item 20 times) |
| Expected Result | 1. System should boot up successfully and work normally. |

* 1. **Platform – Stacking**

**TC\_ID: 2.11.1**

|  |  |
| --- | --- |
| Purpose | Verify that system can boot up successfully with Stand along DUT. |
| Procedure | 1. Boot up DUT with Stand along DUT. |
| Expected Result | 1. System should boot up successfully and work normally. |

**TC\_ID: 2.11.2**

|  |  |
| --- | --- |
| Purpose | Verify that system can boot up successfully with two DUTs. |
| Procedure | 1. Boot up DUT with two DUTs. |
| Expected Result | 1. System should boot up successfully and work normally. |

**TC\_ID: 2.11.3**

|  |  |
| --- | --- |
| Purpose | Verify that system can boot up successfully with Maximum DUTs. |
| Procedure | 1. Boot up DUT with Maximum DUTs. |
| Expected Result | 1. System should boot up successfully and work normally. |

**TC\_ID: 2.11.4**

|  |  |
| --- | --- |
| Purpose | Verify that system cannot boot up successfully with Maximum DUTs. |
| Procedure | 1. Boot up DUT with over Maximum DUTs. |
| Expected Result | 1. System should not boot up successfully and work normally. (follow spec design) |

**TC\_ID: 2.11.5**

|  |  |
| --- | --- |
| Purpose | Verify that system can work normally after reboot master DUT. |
| Procedure | 1. Boot up DUT with at least two DUTs. 2. Reboot master DUT. |
| Expected Result | 1. System can work normally after reboot the master DUT. |

**TC\_ID: 2.11.6**

|  |  |
| --- | --- |
| Purpose | Verify that system can work normally after reboot the slaver DUT. |
| Procedure | 1. Boot up DUT with at least two DUTs. 2. Reboot the slaver DUT. |
| Expected Result | 1. System can work normally after reboot the slaver DUT. |

* 1. **Platform – PoE**

**TC\_ID: 2.12.1**

|  |  |
| --- | --- |
| Purpose | Verify that system can offer power via Ethernet for minimum PD. |
| Procedure | 1. Connect PD to PoE port. 2. Request minimum power via Ethernet via PD. |
| Expected Result | 1. System can offer power via Ethernet for minimum PD. |

**TC\_ID: 2.12.2**

|  |  |
| --- | --- |
| Purpose | Verify that system can offer power via Ethernet for Maximum PD. |
| Procedure | 1. Connect PD to PoE port. 2. Request Maximum power via Ethernet via PD. |
| Expected Result | 1. System can offer power via Ethernet for maximum PD. |

**TC\_ID: 2.12.3**

|  |  |
| --- | --- |
| Purpose | Verify that system can offer power via Ethernet for different PDs. |
| Procedure | 1. Connect different PD to PoE port. 2. Request power via Ethernet via different PDs. |
| Expected Result | 1. System can offer power via Ethernet for different PD. |

**TC\_ID: 2.12.4**

|  |  |
| --- | --- |
| Purpose | To ensure that the device can work normally and offer correct power without hang or crash after connect with PD for a long time (at least 12 hours). |
| Procedure | 1. Connect PD to PoE port. 2. Request power via Ethernet via PD for at least 12 hours. |
| Expected Result | 1. To ensure that the device can work normally and offer correct power without hang or crash after connect with PD for at least 12 hours. |

* 1. **Platform – Temperature**

**TC\_ID: 2.13.1**

|  |  |
| --- | --- |
| Purpose | Verify that system can display correct info for "default Temperature". |
| Procedure | 1. Compare default display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.13.2**

|  |  |
| --- | --- |
| Purpose | Verify that system can display correct info for "minimum/default/maximum" temperature value. |
| Procedure | 1. Modify "minimum/default/maximum" temperature value. 2. Check the display for "minimum/default/maximum" temperature value. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.13.3**

|  |  |
| --- | --- |
| Purpose | Verify that system can record correct log for temperature. |
| Procedure | 1. Make Log for temperature then check. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.13.4**

|  |  |
| --- | --- |
| Purpose | Verify that system can send correct trap for temperature. |
| Procedure | 1. Make trap for temperature then check. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.13.5**

|  |  |
| --- | --- |
| Purpose | Verify that system can send correct alarm for temperature. |
| Procedure | 1. Make alarm for temperature then check. |
| Expected Result | 1. Should implement same as spec design. |

* 1. **Platform – Factory Reset**

**TC\_ID: 2.14.1**

|  |  |
| --- | --- |
| Purpose | Verify that "Factory Reset" can work successfully via Loader. |
| Procedure | 1. "Factory Reset" via Loader. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.14.2**

|  |  |
| --- | --- |
| Purpose | Verify that "Factory Reset" can work successfully via U-boot. |
| Procedure | 1. "Factory Reset" via U-boot. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.14.3**

|  |  |
| --- | --- |
| Purpose | Verify that "Factory Reset" can work successfully via Junior OS. |
| Procedure | 1. "Factory Reset" via Junior OS. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.14.4**

|  |  |
| --- | --- |
| Purpose | Verify that "Factory Reset" can work successfully via Operational mode. |
| Procedure | 1. "Factory Reset" via Operational mode. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.14.5**

|  |  |
| --- | --- |
| Purpose | Verify that "Factory Reset" can work successfully via Configuration mode |
| Procedure | 1. "Factory Reset" via Configuration mode. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.14.6**

|  |  |
| --- | --- |
| Purpose | Verify that "Factory Reset" can work successfully via LCD panel. |
| Procedure | 1. "Factory Reset" via LCD panel. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.14.7**

|  |  |
| --- | --- |
| Purpose | Verify that "Factory Reset" can work successfully via reset button. |
| Procedure | 1. "Factory Reset" via reset button. |
| Expected Result | 1. Should implement same as spec design. |

* 1. **Platform – Transceiver**

**TC\_ID: 2.15.1**

|  |  |
| --- | --- |
| Purpose | Verify that 1G port interface can connect successfully via "EX-SFP-1GE-SX" SFP Transceivers. |
| Procedure | 1. Connect 1G port interface via "EX-SFP-1GE-SX" SFP Transceivers. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.15.2**

|  |  |
| --- | --- |
| Purpose | Verify that 1G port interface can connect successfully via "EXEX-SFP-1GE-LX” SFP Transceivers. |
| Procedure | 1. Connect 1G port interface via "EXEX-SFP-1GE-LX" SFP Transceivers. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.15.3**

|  |  |
| --- | --- |
| Purpose | Verify that 10G port interface can connect successfully via "EX-SFP-10GE-USR" SFP+ Transceivers. |
| Procedure | 1. Connect 10G port interface via "EX-SFP-10GE-USR" SFP+ Transceivers. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.15.4**

|  |  |
| --- | --- |
| Purpose | Verify that 10G port interface can connect successfully via "EX-SFP-10GE-SR" SFP+ Transceivers. |
| Procedure | 1. Connect 10G port interface via "EX-SFP-10GE-SR" SFP+ Transceivers. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.15.5**

|  |  |
| --- | --- |
| Purpose | Verify that 10G port interface can connect successfully via "EX-SFP-10GE-LRM" SFP+ Transceivers. |
| Procedure | 1. Connect 10G port interface via "EX-SFP-10GE-LRM" SFP+ Transceivers. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.15.6**

|  |  |
| --- | --- |
| Purpose | Verify that 10G port interface can connect successfully via "EX-SFP-10GE-LR" SFP+ Transceivers. |
| Procedure | 1. Connect 10G port interface via "EX-SFP-10GE-LR" SFP+ Transceivers. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.15.7**

|  |  |
| --- | --- |
| Purpose | Verify that 40G port interface can connect successfully via "QFX-QSFP-40GE-SR4" QSFP+ Transceivers. |
| Procedure | 1. Connect 40G port interface via "QFX-QSFP-40GE-SR4" QSFP+ Transceivers. |
| Expected Result | 1. Should implement same as spec design. |

* 1. **Platform – Link state**

**TC\_ID: Copper\_straight 2.16.1.1**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay) |
| Procedure | Configured as listed parameters:   1. NWay <--> NWay(All) 2. NWay <--> NWay(10-100-F) 3. NWay <--> NWay(1000-F) 4. NWay <--> NWay(100-F) 5. NWay <--> NWay(10-F) 6. NWay <--> NWay(10-100-H) 7. NWay <--> NWay(100-H) 8. NWay <--> NWay(10-H) 9. NWay <--> 100-F 10. NWay <--> 10-F 11. NWay <--> 100-H 12. NWay <--> 10-H 13. NWay <--> 1000\_Master-Auto 14. NWay <--> 1000\_Slave-Auto 15. NWay <--> 1000\_Master-F 16. NWay <--> 1000\_Slave-F 17. NWay <--> 1000\_Master-H 18. NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.2**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay-F) |
| Procedure | Configured as listed parameters:   1. NWay-F <--> NWay(All) 2. NWay-F <--> NWay(10-100-F) 3. NWay-F <--> NWay(1000-F) 4. NWay-F <--> NWay(100-F) 5. NWay-F <--> NWay(10-F) 6. NWay-F <--> NWay(10-100-H) 7. NWay-F <--> NWay(100-H) 8. NWay-F <--> NWay(10-H) 9. NWay-F <--> 100-F 10. NWay-F <--> 10-F 11. NWay-F <--> 100-H 12. NWay-F <--> 10-H 13. NWay-F <--> 1000\_Master-Auto 14. NWay-F <--> 1000\_Slave-Auto 15. NWay-F <--> 1000\_Master-F 16. NWay-F <--> 1000\_Slave-F 17. NWay-F <--> 1000\_Master-H 18. NWay-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.3**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay-H) |
| Procedure | Configured as listed parameters:   1. NWay-H <--> NWay(All) 2. NWay-H <--> NWay(10-100-F) 3. NWay-H <--> NWay(1000-F) 4. NWay-H <--> NWay(100-F) 5. NWay-H <--> NWay(10-F) 6. NWay-H <--> NWay(10-100-H) 7. NWay-H <--> NWay(100-H) 8. NWay-H <--> NWay(10-H) 9. NWay-H <--> 100-F 10. NWay-H <--> 10-F 11. NWay-H <--> 100-H 12. NWay-H <--> 10-H 13. NWay-H <--> 1000\_Master-Auto 14. NWay-H <--> 1000\_Slave-Auto 15. NWay-H <--> 1000\_Master-F 16. NWay-H <--> 1000\_Slave-F 17. NWay-H <--> 1000\_Master-H 18. NWay-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.4**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay-10-100) |
| Procedure | Configured as listed parameters:   1. NWay-10-100 <--> NWay(All) 2. NWay-10-100 <--> NWay(10-100-F) 3. NWay-10-100 <--> NWay(1000-F) 4. NWay-10-100 <--> NWay(100-F) 5. NWay-10-100 <--> NWay(10-F) 6. NWay-10-100 <--> NWay(10-100-H) 7. NWay-10-100 <--> NWay(100-H) 8. NWay-10-100 <--> NWay(10-H) 9. NWay-10-100 <--> 100-F 10. NWay-10-100 <--> 10-F 11. NWay-10-100 <--> 100-H 12. NWay-10-100 <--> 10-H 13. NWay-10-100 <--> 1000\_Master-NWay 14. NWay-10-100 <--> 1000\_Slave-NWay 15. NWay-10-100 <--> 1000\_Master-F 16. NWay-10-100 <--> 1000\_Slave-F 17. NWay-10-100 <--> 1000\_Master-H 18. NWay-10-100 <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.5**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10-NWay) |
| Procedure | Configured as listed parameters:   1. 10-NWay <--> NWay(All) 2. 10-NWay <--> NWay(10-100-F) 3. 10-NWay <--> NWay(1000-F) 4. 10-NWay <--> NWay(100-F) 5. 10-NWay <--> NWay(10-F) 6. 10-NWay <--> NWay(10-100-H) 7. 10-NWay <--> NWay(100-H) 8. 10-NWay <--> NWay(10-H) 9. 10-NWay <--> 100-F 10. 10-NWay <--> 10-F 11. 10-NWay <--> 100-H 12. 10-NWay <--> 10-H 13. 10-NWay <--> 1000\_Master-NWay 14. 10-NWay <--> 1000\_Slave-NWay 15. 10-NWay <--> 1000\_Master-F 16. 10-NWay <--> 1000\_Slave-F 17. 10-NWay <--> 1000\_Master-H 18. 10-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.6**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10-F) |
| Procedure | Configured as listed parameters:   1. 10-F <--> NWay(All) 2. 10-F <--> NWay(10-100-F) 3. 10-F <--> NWay(1000-F) 4. 10-F <--> NWay(100-F) 5. 10-F <--> NWay(10-F) 6. 10-F <--> NWay(10-100-H) 7. 10-F <--> NWay(100-H) 8. 10-F <--> NWay(10-H) 9. 10-F <--> 100-F 10. 10-F <--> 10-F 11. 10-F <--> 100-H 12. 10-F <--> 10-H 13. 10-F <--> 1000\_Master-NWay 14. 10-F <--> 1000\_Slave-NWay 15. 10-F <--> 1000\_Master-F 16. 10-F <--> 1000\_Slave-F 17. 10-F <--> 1000\_Master-H 18. 10-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.7**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10-H) |
| Procedure | Configured as listed parameters:   1. 10-H <--> NWay(All) 2. 10-H <--> NWay(10-100-F) 3. 10-H <--> NWay(1000-F) 4. 10-H <--> NWay(100-F) 5. 10-H <--> NWay(10-F) 6. 10-H <--> NWay(10-100-H) 7. 10-H <--> NWay(100-H) 8. 10-H <--> NWay(10-H) 9. 10-H <--> 100-F 10. 10-H <--> 10-F 11. 10-H <--> 100-H 12. 10-H <--> 10-H 13. 10-H <--> 1000\_Master-NWay 14. 10-H <--> 1000\_Slave-NWay 15. 10-H <--> 1000\_Master-F 16. 10-H <--> 1000\_Slave-F 17. 10-H <--> 1000\_Master-H 18. 10-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.8**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (100-NWay) |
| Procedure | Configured as listed parameters:   1. 100-NWay <--> NWay(All) 2. 100-NWay <--> NWay(10-100-F) 3. 100-NWay <--> NWay(1000-F) 4. 100-NWay <--> NWay(100-F) 5. 100-NWay <--> NWay(10-F) 6. 100-NWay <--> NWay(10-100-H) 7. 100-NWay <--> NWay(100-H) 8. 100-NWay <--> NWay(10-H) 9. 100-NWay <--> 100-F 10. 100-NWay <--> 10-F 11. 100-NWay <--> 100-H 12. 100-NWay <--> 10-H 13. 100-NWay <--> 1000\_Master-NWay 14. 100-NWay <--> 1000\_Slave-NWay 15. 100-NWay <--> 1000\_Master-F 16. 100-NWay <--> 1000\_Slave-F 17. 100-NWay <--> 1000\_Master-H 18. 100-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.9**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (100-F) |
| Procedure | Configured as listed parameters:   1. 100-F <--> NWay(All) 2. 100-F <--> NWay(10-100-F) 3. 100-F <--> NWay(1000-F) 4. 100-F <--> NWay(100-F) 5. 100-F <--> NWay(10-F) 6. 100-F <--> NWay(10-100-H) 7. 100-F <--> NWay(100-H) 8. 100-F <--> NWay(10-H) 9. 100-F <--> 100-F 10. 100-F <--> 10-F 11. 100-F <--> 100-H 12. 100-F <--> 10-H 13. 100-F <--> 1000\_Master-NWay 14. 100-F <--> 1000\_Slave-NWay 15. 100-F <--> 1000\_Master-F 16. 100-F <--> 1000\_Slave-F 17. 100-F <--> 1000\_Master-H 18. 100-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.10**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (100-H) |
| Procedure | Configured as listed parameters:   1. 100-H <--> NWay(All) 2. 100-H <--> NWay(10-100-F) 3. 100-H <--> NWay(1000-F) 4. 100-H <--> NWay(100-F) 5. 100-H <--> NWay(10-F) 6. 100-H <--> NWay(10-100-H) 7. 100-H <--> NWay(100-H) 8. 100-H <--> NWay(10-H) 9. 100-H <--> 100-F 10. 100-H <--> 10-F 11. 100-H <--> 100-H 12. 100-H <--> 10-H 13. 100-H <--> 1000\_Master-NWay 14. 100-H <--> 1000\_Slave-NWay 15. 100-H <--> 1000\_Master-F 16. 100-H <--> 1000\_Slave-F 17. 100-H <--> 1000\_Master-H 18. 100-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.11**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Master-NWay <--> NWay(All) 2. 1000\_Master-NWay <--> NWay(10-100-F) 3. 1000\_Master-NWay <--> NWay(1000-F) 4. 1000\_Master-NWay <--> NWay(100-F) 5. 1000\_Master-NWay <--> NWay(10-F) 6. 1000\_Master-NWay <--> NWay(10-100-H) 7. 1000\_Master-NWay <--> NWay(100-H) 8. 1000\_Master-NWay <--> NWay(10-H) 9. 1000\_Master-NWay <--> 100-F 10. 1000\_Master-NWay <--> 10-F 11. 1000\_Master-NWay <--> 100-H 12. 1000\_Master-NWay <--> 10-H 13. 1000\_Master-NWay <--> 1000\_Master-NWay 14. 1000\_Master-NWay <--> 1000\_Slave-NWay 15. 1000\_Master-NWay <--> 1000\_Master-F 16. 1000\_Master-NWay <--> 1000\_Slave-F 17. 1000\_Master-NWay <--> 1000\_Master-H 18. 1000\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.12**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave-NWay <--> NWay(All) 2. 1000\_Slave-NWay <--> NWay(10-100-F) 3. 1000\_Slave-NWay <--> NWay(1000-F) 4. 1000\_Slave-NWay <--> NWay(100-F) 5. 1000\_Slave-NWay <--> NWay(10-F) 6. 1000\_Slave-NWay <--> NWay(10-100-H) 7. 1000\_Slave-NWay <--> NWay(100-H) 8. 1000\_Slave-NWay <--> NWay(10-H) 9. 1000\_Slave-NWay <--> 100-F 10. 1000\_Slave-NWay <--> 10-F 11. 1000\_Slave-NWay <--> 100-H 12. 1000\_Slave-NWay <--> 10-H 13. 1000\_Slave-NWay <--> 1000\_Master-NWay 14. 1000\_Slave-NWay <--> 1000\_Slave-NWay 15. 1000\_Slave-NWay <--> 1000\_Master-F 16. 1000\_Slave-NWay <--> 1000\_Slave-F 17. 1000\_Slave-NWay <--> 1000\_Master-H 18. 1000\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.13**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Master\_F <--> NWay(All) 2. 1000\_Master\_F <--> NWay(10-100-F) 3. 1000\_Master\_F <--> NWay(1000-F) 4. 1000\_Master\_F <--> NWay(100-F) 5. 1000\_Master\_F <--> NWay(10-F) 6. 1000\_Master\_F <--> NWay(10-100-H) 7. 1000\_Master\_F <--> NWay(100-H) 8. 1000\_Master\_F <--> NWay(10-H) 9. 1000\_Master\_F <--> 100-F 10. 1000\_Master\_F <--> 100-H 11. 1000\_Master\_F <--> 10-F 12. 1000\_Master\_F <--> 10-H 13. 1000\_Master\_F <--> 1000\_Master-NWay 14. 1000\_Master\_F <--> 1000\_Slave-NWay 15. 1000\_Master\_F <--> 1000\_Master-F 16. 1000\_Master\_F <--> 1000\_Slave-F 17. 1000\_Master\_F <--> 1000\_Master-H 18. 1000\_Master\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.14**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave\_F <--> NWay(All) 2. 1000\_Slave\_F <--> NWay(10-100-F) 3. 1000\_Slave\_F <--> NWay(1000-F) 4. 1000\_Slave\_F <--> NWay(100-F) 5. 1000\_Slave\_F <--> NWay(10-F) 6. 1000\_Slave\_F <--> NWay(10-100-H) 7. 1000\_Slave\_F <--> NWay(100-H) 8. 1000\_Slave\_F <--> NWay(10-H) 9. 1000\_Slave\_F <--> 100-F 10. 1000\_Slave\_F <--> 100-H 11. 1000\_Slave\_F <--> 10-F 12. 1000\_Slave\_F <--> 10-H 13. 1000\_Slave\_F <--> 1000\_Master-NWay 14. 1000\_Slave\_F <--> 1000\_Slave-NWay 15. 1000\_Slave\_F <--> 1000\_Master-F 16. 1000\_Slave\_F <--> 1000\_Slave-F 17. 1000\_Slave\_F <--> 1000\_Master-H 18. 1000\_Slave\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.15**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Master\_H <--> NWay(All) 2. 1000\_Master\_H <--> NWay(10-100-F) 3. 1000\_Master\_H <--> NWay(1000-F) 4. 1000\_Master\_H <--> NWay(100-F) 5. 1000\_Master\_H <--> NWay(10-F) 6. 1000\_Master\_H <--> NWay(10-100-H) 7. 1000\_Master\_H <--> NWay(100-H) 8. 1000\_Master\_H <--> NWay(10-H) 9. 1000\_Master\_H <--> 100-F 10. 1000\_Master\_H <--> 100-H 11. 1000\_Master\_H <--> 10-F 12. 1000\_Master\_H <--> 10-H 13. 1000\_Master\_H <--> 1000\_Master-NWay 14. 1000\_Master\_H <--> 1000\_Slave-NWay 15. 1000\_Master\_H <--> 1000\_Master-F 16. 1000\_Master\_H <--> 1000\_Slave-F 17. 1000\_Master\_H <--> 1000\_Master-H 18. 1000\_Master\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_straight 2.16.1.16**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave\_H <--> NWay(All) 2. 1000\_Slave\_H <--> NWay(10-100-F) 3. 1000\_Slave\_H <--> NWay(1000-F) 4. 1000\_Slave\_H <--> NWay(100-F) 5. 1000\_Slave\_H <--> NWay(10-F) 6. 1000\_Slave\_H <--> NWay(10-100-H) 7. 1000\_Slave\_H <--> NWay(100-H) 8. 1000\_Slave\_H <--> NWay(10-H) 9. 1000\_Slave\_H <--> 100-F 10. 1000\_Slave\_H <--> 100-H 11. 1000\_Slave\_H <--> 10-F 12. 1000\_Slave\_H <--> 10-H 13. 1000\_Slave\_H <--> 1000\_Master-NWay 14. 1000\_Slave\_H <--> 1000\_Slave-NWay 15. 1000\_Slave\_H <--> 1000\_Master-F 16. 1000\_Slave\_H <--> 1000\_Slave-F 17. 1000\_Slave\_H <--> 1000\_Master-H 18. 1000\_Slave\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.1**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay) |
| Procedure | Configured as listed parameters:   1. NWay <--> NWay(All) 2. NWay <--> NWay(10-100-F) 3. NWay <--> NWay(1000-F) 4. NWay <--> NWay(100-F) 5. NWay <--> NWay(10-F) 6. NWay <--> NWay(10-100-H) 7. NWay <--> NWay(100-H) 8. NWay <--> NWay(10-H) 9. NWay <--> 100-F 10. NWay <--> 10-F 11. NWay <--> 100-H 12. NWay <--> 10-H 13. NWay <--> 1000\_Master-Auto 14. NWay <--> 1000\_Slave-Auto 15. NWay <--> 1000\_Master-F 16. NWay <--> 1000\_Slave-F 17. NWay <--> 1000\_Master-H 18. NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.2**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay-F) |
| Procedure | Configured as listed parameters:   1. NWay-F <--> NWay(All) 2. NWay-F <--> NWay(10-100-F) 3. NWay-F <--> NWay(1000-F) 4. NWay-F <--> NWay(100-F) 5. NWay-F <--> NWay(10-F) 6. NWay-F <--> NWay(10-100-H) 7. NWay-F <--> NWay(100-H) 8. NWay-F <--> NWay(10-H) 9. NWay-F <--> 100-F 10. NWay-F <--> 10-F 11. NWay-F <--> 100-H 12. NWay-F <--> 10-H 13. NWay-F <--> 1000\_Master-Auto 14. NWay-F <--> 1000\_Slave-Auto 15. NWay-F <--> 1000\_Master-F 16. NWay-F <--> 1000\_Slave-F 17. NWay-F <--> 1000\_Master-H 18. NWay-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.3**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay-H) |
| Procedure | Configured as listed parameters:   1. NWay-H <--> NWay(All) 2. NWay-H <--> NWay(10-100-F) 3. NWay-H <--> NWay(1000-F) 4. NWay-H <--> NWay(100-F) 5. NWay-H <--> NWay(10-F) 6. NWay-H <--> NWay(10-100-H) 7. NWay-H <--> NWay(100-H) 8. NWay-H <--> NWay(10-H) 9. NWay-H <--> 100-F 10. NWay-H <--> 10-F 11. NWay-H <--> 100-H 12. NWay-H <--> 10-H 13. NWay-H <--> 1000\_Master-Auto 14. NWay-H <--> 1000\_Slave-Auto 15. NWay-H <--> 1000\_Master-F 16. NWay-H <--> 1000\_Slave-F 17. NWay-H <--> 1000\_Master-H 18. NWay-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.4**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay-10-100) |
| Procedure | Configured as listed parameters:   1. NWay-10-100 <--> NWay(All) 2. NWay-10-100 <--> NWay(10-100-F) 3. NWay-10-100 <--> NWay(1000-F) 4. NWay-10-100 <--> NWay(100-F) 5. NWay-10-100 <--> NWay(10-F) 6. NWay-10-100 <--> NWay(10-100-H) 7. NWay-10-100 <--> NWay(100-H) 8. NWay-10-100 <--> NWay(10-H) 9. NWay-10-100 <--> 100-F 10. NWay-10-100 <--> 10-F 11. NWay-10-100 <--> 100-H 12. NWay-10-100 <--> 10-H 13. NWay-10-100 <--> 1000\_Master-NWay 14. NWay-10-100 <--> 1000\_Slave-NWay 15. NWay-10-100 <--> 1000\_Master-F 16. NWay-10-100 <--> 1000\_Slave-F 17. NWay-10-100 <--> 1000\_Master-H 18. NWay-10-100 <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.5**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10-NWay) |
| Procedure | Configured as listed parameters:   1. 10-NWay <--> NWay(All) 2. 10-NWay <--> NWay(10-100-F) 3. 10-NWay <--> NWay(1000-F) 4. 10-NWay <--> NWay(100-F) 5. 10-NWay <--> NWay(10-F) 6. 10-NWay <--> NWay(10-100-H) 7. 10-NWay <--> NWay(100-H) 8. 10-NWay <--> NWay(10-H) 9. 10-NWay <--> 100-F 10. 10-NWay <--> 10-F 11. 10-NWay <--> 100-H 12. 10-NWay <--> 10-H 13. 10-NWay <--> 1000\_Master-NWay 14. 10-NWay <--> 1000\_Slave-NWay 15. 10-NWay <--> 1000\_Master-F 16. 10-NWay <--> 1000\_Slave-F 17. 10-NWay <--> 1000\_Master-H 18. 10-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.6**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10-F) |
| Procedure | Configured as listed parameters:   1. 10-F <--> NWay(All) 2. 10-F <--> NWay(10-100-F) 3. 10-F <--> NWay(1000-F) 4. 10-F <--> NWay(100-F) 5. 10-F <--> NWay(10-F) 6. 10-F <--> NWay(10-100-H) 7. 10-F <--> NWay(100-H) 8. 10-F <--> NWay(10-H) 9. 10-F <--> 100-F 10. 10-F <--> 10-F 11. 10-F <--> 100-H 12. 10-F <--> 10-H 13. 10-F <--> 1000\_Master-NWay 14. 10-F <--> 1000\_Slave-NWay 15. 10-F <--> 1000\_Master-F 16. 10-F <--> 1000\_Slave-F 17. 10-F <--> 1000\_Master-H 18. 10-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.7**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10-H) |
| Procedure | Configured as listed parameters:   1. 10-H <--> NWay(All) 2. 10-H <--> NWay(10-100-F) 3. 10-H <--> NWay(1000-F) 4. 10-H <--> NWay(100-F) 5. 10-H <--> NWay(10-F) 6. 10-H <--> NWay(10-100-H) 7. 10-H <--> NWay(100-H) 8. 10-H <--> NWay(10-H) 9. 10-H <--> 100-F 10. 10-H <--> 10-F 11. 10-H <--> 100-H 12. 10-H <--> 10-H 13. 10-H <--> 1000\_Master-NWay 14. 10-H <--> 1000\_Slave-NWay 15. 10-H <--> 1000\_Master-F 16. 10-H <--> 1000\_Slave-F 17. 10-H <--> 1000\_Master-H 18. 10-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.8**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (100-NWay) |
| Procedure | Configured as listed parameters:   1. 100-NWay <--> NWay(All) 2. 100-NWay <--> NWay(10-100-F) 3. 100-NWay <--> NWay(1000-F) 4. 100-NWay <--> NWay(100-F) 5. 100-NWay <--> NWay(10-F) 6. 100-NWay <--> NWay(10-100-H) 7. 100-NWay <--> NWay(100-H) 8. 100-NWay <--> NWay(10-H) 9. 100-NWay <--> 100-F 10. 100-NWay <--> 10-F 11. 100-NWay <--> 100-H 12. 100-NWay <--> 10-H 13. 100-NWay <--> 1000\_Master-NWay 14. 100-NWay <--> 1000\_Slave-NWay 15. 100-NWay <--> 1000\_Master-F 16. 100-NWay <--> 1000\_Slave-F 17. 100-NWay <--> 1000\_Master-H 18. 100-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.9**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (100-F) |
| Procedure | Configured as listed parameters:   1. 100-F <--> NWay(All) 2. 100-F <--> NWay(10-100-F) 3. 100-F <--> NWay(1000-F) 4. 100-F <--> NWay(100-F) 5. 100-F <--> NWay(10-F) 6. 100-F <--> NWay(10-100-H) 7. 100-F <--> NWay(100-H) 8. 100-F <--> NWay(10-H) 9. 100-F <--> 100-F 10. 100-F <--> 10-F 11. 100-F <--> 100-H 12. 100-F <--> 10-H 13. 100-F <--> 1000\_Master-NWay 14. 100-F <--> 1000\_Slave-NWay 15. 100-F <--> 1000\_Master-F 16. 100-F <--> 1000\_Slave-F 17. 100-F <--> 1000\_Master-H 18. 100-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.10**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (100-H) |
| Procedure | Configured as listed parameters:   1. 100-H <--> NWay(All) 2. 100-H <--> NWay(10-100-F) 3. 100-H <--> NWay(1000-F) 4. 100-H <--> NWay(100-F) 5. 100-H <--> NWay(10-F) 6. 100-H <--> NWay(10-100-H) 7. 100-H <--> NWay(100-H) 8. 100-H <--> NWay(10-H) 9. 100-H <--> 100-F 10. 100-H <--> 10-F 11. 100-H <--> 100-H 12. 100-H <--> 10-H 13. 100-H <--> 1000\_Master-NWay 14. 100-H <--> 1000\_Slave-NWay 15. 100-H <--> 1000\_Master-F 16. 100-H <--> 1000\_Slave-F 17. 100-H <--> 1000\_Master-H 18. 100-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.11**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Master-NWay <--> NWay(All) 2. 1000\_Master-NWay <--> NWay(10-100-F) 3. 1000\_Master-NWay <--> NWay(1000-F) 4. 1000\_Master-NWay <--> NWay(100-F) 5. 1000\_Master-NWay <--> NWay(10-F) 6. 1000\_Master-NWay <--> NWay(10-100-H) 7. 1000\_Master-NWay <--> NWay(100-H) 8. 1000\_Master-NWay <--> NWay(10-H) 9. 1000\_Master-NWay <--> 100-F 10. 1000\_Master-NWay <--> 10-F 11. 1000\_Master-NWay <--> 100-H 12. 1000\_Master-NWay <--> 10-H 13. 1000\_Master-NWay <--> 1000\_Master-NWay 14. 1000\_Master-NWay <--> 1000\_Slave-NWay 15. 1000\_Master-NWay <--> 1000\_Master-F 16. 1000\_Master-NWay <--> 1000\_Slave-F 17. 1000\_Master-NWay <--> 1000\_Master-H 18. 1000\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.12**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave-NWay <--> NWay(All) 2. 1000\_Slave-NWay <--> NWay(10-100-F) 3. 1000\_Slave-NWay <--> NWay(1000-F) 4. 1000\_Slave-NWay <--> NWay(100-F) 5. 1000\_Slave-NWay <--> NWay(10-F) 6. 1000\_Slave-NWay <--> NWay(10-100-H) 7. 1000\_Slave-NWay <--> NWay(100-H) 8. 1000\_Slave-NWay <--> NWay(10-H) 9. 1000\_Slave-NWay <--> 100-F 10. 1000\_Slave-NWay <--> 10-F 11. 1000\_Slave-NWay <--> 100-H 12. 1000\_Slave-NWay <--> 10-H 13. 1000\_Slave-NWay <--> 1000\_Master-NWay 14. 1000\_Slave-NWay <--> 1000\_Slave-NWay 15. 1000\_Slave-NWay <--> 1000\_Master-F 16. 1000\_Slave-NWay <--> 1000\_Slave-F 17. 1000\_Slave-NWay <--> 1000\_Master-H 18. 1000\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.13**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Master\_F <--> NWay(All) 2. 1000\_Master\_F <--> NWay(10-100-F) 3. 1000\_Master\_F <--> NWay(1000-F) 4. 1000\_Master\_F <--> NWay(100-F) 5. 1000\_Master\_F <--> NWay(10-F) 6. 1000\_Master\_F <--> NWay(10-100-H) 7. 1000\_Master\_F <--> NWay(100-H) 8. 1000\_Master\_F <--> NWay(10-H) 9. 1000\_Master\_F <--> 100-F 10. 1000\_Master\_F <--> 100-H 11. 1000\_Master\_F <--> 10-F 12. 1000\_Master\_F <--> 10-H 13. 1000\_Master\_F <--> 1000\_Master-NWay 14. 1000\_Master\_F <--> 1000\_Slave-NWay 15. 1000\_Master\_F <--> 1000\_Master-F 16. 1000\_Master\_F <--> 1000\_Slave-F 17. 1000\_Master\_F <--> 1000\_Master-H 18. 1000\_Master\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.14**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave\_F <--> NWay(All) 2. 1000\_Slave\_F <--> NWay(10-100-F) 3. 1000\_Slave\_F <--> NWay(1000-F) 4. 1000\_Slave\_F <--> NWay(100-F) 5. 1000\_Slave\_F <--> NWay(10-F) 6. 1000\_Slave\_F <--> NWay(10-100-H) 7. 1000\_Slave\_F <--> NWay(100-H) 8. 1000\_Slave\_F <--> NWay(10-H) 9. 1000\_Slave\_F <--> 100-F 10. 1000\_Slave\_F <--> 100-H 11. 1000\_Slave\_F <--> 10-F 12. 1000\_Slave\_F <--> 10-H 13. 1000\_Slave\_F <--> 1000\_Master-NWay 14. 1000\_Slave\_F <--> 1000\_Slave-NWay 15. 1000\_Slave\_F <--> 1000\_Master-F 16. 1000\_Slave\_F <--> 1000\_Slave-F 17. 1000\_Slave\_F <--> 1000\_Master-H 18. 1000\_Slave\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.15**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Master\_H <--> NWay(All) 2. 1000\_Master\_H <--> NWay(10-100-F) 3. 1000\_Master\_H <--> NWay(1000-F) 4. 1000\_Master\_H <--> NWay(100-F) 5. 1000\_Master\_H <--> NWay(10-F) 6. 1000\_Master\_H <--> NWay(10-100-H) 7. 1000\_Master\_H <--> NWay(100-H) 8. 1000\_Master\_H <--> NWay(10-H) 9. 1000\_Master\_H <--> 100-F 10. 1000\_Master\_H <--> 100-H 11. 1000\_Master\_H <--> 10-F 12. 1000\_Master\_H <--> 10-H 13. 1000\_Master\_H <--> 1000\_Master-NWay 14. 1000\_Master\_H <--> 1000\_Slave-NWay 15. 1000\_Master\_H <--> 1000\_Master-F 16. 1000\_Master\_H <--> 1000\_Slave-F 17. 1000\_Master\_H <--> 1000\_Master-H 18. 1000\_Master\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Copper\_Crossover 2.16.2.16**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave\_H <--> NWay(All) 2. 1000\_Slave\_H <--> NWay(10-100-F) 3. 1000\_Slave\_H <--> NWay(1000-F) 4. 1000\_Slave\_H <--> NWay(100-F) 5. 1000\_Slave\_H <--> NWay(10-F) 6. 1000\_Slave\_H <--> NWay(10-100-H) 7. 1000\_Slave\_H <--> NWay(100-H) 8. 1000\_Slave\_H <--> NWay(10-H) 9. 1000\_Slave\_H <--> 100-F 10. 1000\_Slave\_H <--> 100-H 11. 1000\_Slave\_H <--> 10-F 12. 1000\_Slave\_H <--> 10-H 13. 1000\_Slave\_H <--> 1000\_Master-NWay 14. 1000\_Slave\_H <--> 1000\_Slave-NWay 15. 1000\_Slave\_H <--> 1000\_Master-F 16. 1000\_Slave\_H <--> 1000\_Slave-F 17. 1000\_Slave\_H <--> 1000\_Master-H 18. 1000\_Slave\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.1**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay) |
| Procedure | Configured as listed parameters:   1. NWay <--> NWay(All) 2. NWay <--> NWay(10-100-F) 3. NWay <--> NWay(1000-F) 4. NWay <--> NWay(100-F) 5. NWay <--> NWay(10-F) 6. NWay <--> NWay(10-100-H) 7. NWay <--> NWay(100-H) 8. NWay <--> NWay(10-H) 9. NWay <--> 100-F 10. NWay <--> 10-F 11. NWay <--> 100-H 12. NWay <--> 10-H 13. NWay <--> 1000\_Master-Auto 14. NWay <--> 1000\_Slave-Auto 15. NWay <--> 1000\_Master-F 16. NWay <--> 1000\_Slave-F 17. NWay <--> 1000\_Master-H 18. NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.2**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay-F) |
| Procedure | Configured as listed parameters:   1. NWay-F <--> NWay(All) 2. NWay-F <--> NWay(10-100-F) 3. NWay-F <--> NWay(1000-F) 4. NWay-F <--> NWay(100-F) 5. NWay-F <--> NWay(10-F) 6. NWay-F <--> NWay(10-100-H) 7. NWay-F <--> NWay(100-H) 8. NWay-F <--> NWay(10-H) 9. NWay-F <--> 100-F 10. NWay-F <--> 10-F 11. NWay-F <--> 100-H 12. NWay-F <--> 10-H 13. NWay-F <--> 1000\_Master-Auto 14. NWay-F <--> 1000\_Slave-Auto 15. NWay-F <--> 1000\_Master-F 16. NWay-F <--> 1000\_Slave-F 17. NWay-F <--> 1000\_Master-H 18. NWay-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.3**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay-H) |
| Procedure | Configured as listed parameters:   1. NWay-H <--> NWay(All) 2. NWay-H <--> NWay(10-100-F) 3. NWay-H <--> NWay(1000-F) 4. NWay-H <--> NWay(100-F) 5. NWay-H <--> NWay(10-F) 6. NWay-H <--> NWay(10-100-H) 7. NWay-H <--> NWay(100-H) 8. NWay-H <--> NWay(10-H) 9. NWay-H <--> 100-F 10. NWay-H <--> 10-F 11. NWay-H <--> 100-H 12. NWay-H <--> 10-H 13. NWay-H <--> 1000\_Master-Auto 14. NWay-H <--> 1000\_Slave-Auto 15. NWay-H <--> 1000\_Master-F 16. NWay-H <--> 1000\_Slave-F 17. NWay-H <--> 1000\_Master-H 18. NWay-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.4**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay-10-100) |
| Procedure | Configured as listed parameters:   1. NWay-10-100 <--> NWay(All) 2. NWay-10-100 <--> NWay(10-100-F) 3. NWay-10-100 <--> NWay(1000-F) 4. NWay-10-100 <--> NWay(100-F) 5. NWay-10-100 <--> NWay(10-F) 6. NWay-10-100 <--> NWay(10-100-H) 7. NWay-10-100 <--> NWay(100-H) 8. NWay-10-100 <--> NWay(10-H) 9. NWay-10-100 <--> 100-F 10. NWay-10-100 <--> 10-F 11. NWay-10-100 <--> 100-H 12. NWay-10-100 <--> 10-H 13. NWay-10-100 <--> 1000\_Master-NWay 14. NWay-10-100 <--> 1000\_Slave-NWay 15. NWay-10-100 <--> 1000\_Master-F 16. NWay-10-100 <--> 1000\_Slave-F 17. NWay-10-100 <--> 1000\_Master-H 18. NWay-10-100 <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.5**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10-NWay) |
| Procedure | Configured as listed parameters:   1. 10-NWay <--> NWay(All) 2. 10-NWay <--> NWay(10-100-F) 3. 10-NWay <--> NWay(1000-F) 4. 10-NWay <--> NWay(100-F) 5. 10-NWay <--> NWay(10-F) 6. 10-NWay <--> NWay(10-100-H) 7. 10-NWay <--> NWay(100-H) 8. 10-NWay <--> NWay(10-H) 9. 10-NWay <--> 100-F 10. 10-NWay <--> 10-F 11. 10-NWay <--> 100-H 12. 10-NWay <--> 10-H 13. 10-NWay <--> 1000\_Master-NWay 14. 10-NWay <--> 1000\_Slave-NWay 15. 10-NWay <--> 1000\_Master-F 16. 10-NWay <--> 1000\_Slave-F 17. 10-NWay <--> 1000\_Master-H 18. 10-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.6**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10-F) |
| Procedure | Configured as listed parameters:   1. 10-F <--> NWay(All) 2. 10-F <--> NWay(10-100-F) 3. 10-F <--> NWay(1000-F) 4. 10-F <--> NWay(100-F) 5. 10-F <--> NWay(10-F) 6. 10-F <--> NWay(10-100-H) 7. 10-F <--> NWay(100-H) 8. 10-F <--> NWay(10-H) 9. 10-F <--> 100-F 10. 10-F <--> 10-F 11. 10-F <--> 100-H 12. 10-F <--> 10-H 13. 10-F <--> 1000\_Master-NWay 14. 10-F <--> 1000\_Slave-NWay 15. 10-F <--> 1000\_Master-F 16. 10-F <--> 1000\_Slave-F 17. 10-F <--> 1000\_Master-H 18. 10-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.7**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10-H) |
| Procedure | Configured as listed parameters:   1. 10-H <--> NWay(All) 2. 10-H <--> NWay(10-100-F) 3. 10-H <--> NWay(1000-F) 4. 10-H <--> NWay(100-F) 5. 10-H <--> NWay(10-F) 6. 10-H <--> NWay(10-100-H) 7. 10-H <--> NWay(100-H) 8. 10-H <--> NWay(10-H) 9. 10-H <--> 100-F 10. 10-H <--> 10-F 11. 10-H <--> 100-H 12. 10-H <--> 10-H 13. 10-H <--> 1000\_Master-NWay 14. 10-H <--> 1000\_Slave-NWay 15. 10-H <--> 1000\_Master-F 16. 10-H <--> 1000\_Slave-F 17. 10-H <--> 1000\_Master-H 18. 10-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.8**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (100-NWay) |
| Procedure | Configured as listed parameters:   1. 100-NWay <--> NWay(All) 2. 100-NWay <--> NWay(10-100-F) 3. 100-NWay <--> NWay(1000-F) 4. 100-NWay <--> NWay(100-F) 5. 100-NWay <--> NWay(10-F) 6. 100-NWay <--> NWay(10-100-H) 7. 100-NWay <--> NWay(100-H) 8. 100-NWay <--> NWay(10-H) 9. 100-NWay <--> 100-F 10. 100-NWay <--> 10-F 11. 100-NWay <--> 100-H 12. 100-NWay <--> 10-H 13. 100-NWay <--> 1000\_Master-NWay 14. 100-NWay <--> 1000\_Slave-NWay 15. 100-NWay <--> 1000\_Master-F 16. 100-NWay <--> 1000\_Slave-F 17. 100-NWay <--> 1000\_Master-H 18. 100-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.9**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (100-F) |
| Procedure | Configured as listed parameters:   1. 100-F <--> NWay(All) 2. 100-F <--> NWay(10-100-F) 3. 100-F <--> NWay(1000-F) 4. 100-F <--> NWay(100-F) 5. 100-F <--> NWay(10-F) 6. 100-F <--> NWay(10-100-H) 7. 100-F <--> NWay(100-H) 8. 100-F <--> NWay(10-H) 9. 100-F <--> 100-F 10. 100-F <--> 10-F 11. 100-F <--> 100-H 12. 100-F <--> 10-H 13. 100-F <--> 1000\_Master-NWay 14. 100-F <--> 1000\_Slave-NWay 15. 100-F <--> 1000\_Master-F 16. 100-F <--> 1000\_Slave-F 17. 100-F <--> 1000\_Master-H 18. 100-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.10**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (100-H) |
| Procedure | Configured as listed parameters:   1. 100-H <--> NWay(All) 2. 100-H <--> NWay(10-100-F) 3. 100-H <--> NWay(1000-F) 4. 100-H <--> NWay(100-F) 5. 100-H <--> NWay(10-F) 6. 100-H <--> NWay(10-100-H) 7. 100-H <--> NWay(100-H) 8. 100-H <--> NWay(10-H) 9. 100-H <--> 100-F 10. 100-H <--> 10-F 11. 100-H <--> 100-H 12. 100-H <--> 10-H 13. 100-H <--> 1000\_Master-NWay 14. 100-H <--> 1000\_Slave-NWay 15. 100-H <--> 1000\_Master-F 16. 100-H <--> 1000\_Slave-F 17. 100-H <--> 1000\_Master-H 18. 100-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.11**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Master-NWay <--> NWay(All) 2. 1000\_Master-NWay <--> NWay(10-100-F) 3. 1000\_Master-NWay <--> NWay(1000-F) 4. 1000\_Master-NWay <--> NWay(100-F) 5. 1000\_Master-NWay <--> NWay(10-F) 6. 1000\_Master-NWay <--> NWay(10-100-H) 7. 1000\_Master-NWay <--> NWay(100-H) 8. 1000\_Master-NWay <--> NWay(10-H) 9. 1000\_Master-NWay <--> 100-F 10. 1000\_Master-NWay <--> 10-F 11. 1000\_Master-NWay <--> 100-H 12. 1000\_Master-NWay <--> 10-H 13. 1000\_Master-NWay <--> 1000\_Master-NWay 14. 1000\_Master-NWay <--> 1000\_Slave-NWay 15. 1000\_Master-NWay <--> 1000\_Master-F 16. 1000\_Master-NWay <--> 1000\_Slave-F 17. 1000\_Master-NWay <--> 1000\_Master-H 18. 1000\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.12**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave-NWay <--> NWay(All) 2. 1000\_Slave-NWay <--> NWay(10-100-F) 3. 1000\_Slave-NWay <--> NWay(1000-F) 4. 1000\_Slave-NWay <--> NWay(100-F) 5. 1000\_Slave-NWay <--> NWay(10-F) 6. 1000\_Slave-NWay <--> NWay(10-100-H) 7. 1000\_Slave-NWay <--> NWay(100-H) 8. 1000\_Slave-NWay <--> NWay(10-H) 9. 1000\_Slave-NWay <--> 100-F 10. 1000\_Slave-NWay <--> 10-F 11. 1000\_Slave-NWay <--> 100-H 12. 1000\_Slave-NWay <--> 10-H 13. 1000\_Slave-NWay <--> 1000\_Master-NWay 14. 1000\_Slave-NWay <--> 1000\_Slave-NWay 15. 1000\_Slave-NWay <--> 1000\_Master-F 16. 1000\_Slave-NWay <--> 1000\_Slave-F 17. 1000\_Slave-NWay <--> 1000\_Master-H 18. 1000\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.13**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Master\_F <--> NWay(All) 2. 1000\_Master\_F <--> NWay(10-100-F) 3. 1000\_Master\_F <--> NWay(1000-F) 4. 1000\_Master\_F <--> NWay(100-F) 5. 1000\_Master\_F <--> NWay(10-F) 6. 1000\_Master\_F <--> NWay(10-100-H) 7. 1000\_Master\_F <--> NWay(100-H) 8. 1000\_Master\_F <--> NWay(10-H) 9. 1000\_Master\_F <--> 100-F 10. 1000\_Master\_F <--> 100-H 11. 1000\_Master\_F <--> 10-F 12. 1000\_Master\_F <--> 10-H 13. 1000\_Master\_F <--> 1000\_Master-NWay 14. 1000\_Master\_F <--> 1000\_Slave-NWay 15. 1000\_Master\_F <--> 1000\_Master-F 16. 1000\_Master\_F <--> 1000\_Slave-F 17. 1000\_Master\_F <--> 1000\_Master-H 18. 1000\_Master\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.14**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave\_F <--> NWay(All) 2. 1000\_Slave\_F <--> NWay(10-100-F) 3. 1000\_Slave\_F <--> NWay(1000-F) 4. 1000\_Slave\_F <--> NWay(100-F) 5. 1000\_Slave\_F <--> NWay(10-F) 6. 1000\_Slave\_F <--> NWay(10-100-H) 7. 1000\_Slave\_F <--> NWay(100-H) 8. 1000\_Slave\_F <--> NWay(10-H) 9. 1000\_Slave\_F <--> 100-F 10. 1000\_Slave\_F <--> 100-H 11. 1000\_Slave\_F <--> 10-F 12. 1000\_Slave\_F <--> 10-H 13. 1000\_Slave\_F <--> 1000\_Master-NWay 14. 1000\_Slave\_F <--> 1000\_Slave-NWay 15. 1000\_Slave\_F <--> 1000\_Master-F 16. 1000\_Slave\_F <--> 1000\_Slave-F 17. 1000\_Slave\_F <--> 1000\_Master-H 18. 1000\_Slave\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.15**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Master\_H <--> NWay(All) 2. 1000\_Master\_H <--> NWay(10-100-F) 3. 1000\_Master\_H <--> NWay(1000-F) 4. 1000\_Master\_H <--> NWay(100-F) 5. 1000\_Master\_H <--> NWay(10-F) 6. 1000\_Master\_H <--> NWay(10-100-H) 7. 1000\_Master\_H <--> NWay(100-H) 8. 1000\_Master\_H <--> NWay(10-H) 9. 1000\_Master\_H <--> 100-F 10. 1000\_Master\_H <--> 100-H 11. 1000\_Master\_H <--> 10-F 12. 1000\_Master\_H <--> 10-H 13. 1000\_Master\_H <--> 1000\_Master-NWay 14. 1000\_Master\_H <--> 1000\_Slave-NWay 15. 1000\_Master\_H <--> 1000\_Master-F 16. 1000\_Master\_H <--> 1000\_Slave-F 17. 1000\_Master\_H <--> 1000\_Master-H 18. 1000\_Master\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Fiber 2.16.3.16**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave\_H <--> NWay(All) 2. 1000\_Slave\_H <--> NWay(10-100-F) 3. 1000\_Slave\_H <--> NWay(1000-F) 4. 1000\_Slave\_H <--> NWay(100-F) 5. 1000\_Slave\_H <--> NWay(10-F) 6. 1000\_Slave\_H <--> NWay(10-100-H) 7. 1000\_Slave\_H <--> NWay(100-H) 8. 1000\_Slave\_H <--> NWay(10-H) 9. 1000\_Slave\_H <--> 100-F 10. 1000\_Slave\_H <--> 100-H 11. 1000\_Slave\_H <--> 10-F 12. 1000\_Slave\_H <--> 10-H 13. 1000\_Slave\_H <--> 1000\_Master-NWay 14. 1000\_Slave\_H <--> 1000\_Slave-NWay 15. 1000\_Slave\_H <--> 1000\_Master-F 16. 1000\_Slave\_H <--> 1000\_Slave-F 17. 1000\_Slave\_H <--> 1000\_Master-H 18. 1000\_Slave\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.1**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay) |
| Procedure | Configured as listed parameters:   1. NWay <--> NWay(All) 2. NWay <--> NWay(10-100-F) 3. NWay <--> NWay(1000-F) 4. NWay <--> NWay(100-F) 5. NWay <--> NWay(10-F) 6. NWay <--> NWay(10-100-H) 7. NWay <--> NWay(100-H) 8. NWay <--> NWay(10-H) 9. NWay <--> 100-F 10. NWay <--> 10-F 11. NWay <--> 100-H 12. NWay <--> 10-H 13. NWay <--> 1000\_Master-Auto 14. NWay <--> 1000\_Slave-Auto 15. NWay <--> 1000\_Master-F 16. NWay <--> 1000\_Slave-F 17. NWay <--> 1000\_Master-H 18. NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.2**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay-F) |
| Procedure | Configured as listed parameters:   1. NWay-F <--> NWay(All) 2. NWay-F <--> NWay(10-100-F) 3. NWay-F <--> NWay(1000-F) 4. NWay-F <--> NWay(100-F) 5. NWay-F <--> NWay(10-F) 6. NWay-F <--> NWay(10-100-H) 7. NWay-F <--> NWay(100-H) 8. NWay-F <--> NWay(10-H) 9. NWay-F <--> 100-F 10. NWay-F <--> 10-F 11. NWay-F <--> 100-H 12. NWay-F <--> 10-H 13. NWay-F <--> 1000\_Master-Auto 14. NWay-F <--> 1000\_Slave-Auto 15. NWay-F <--> 1000\_Master-F 16. NWay-F <--> 1000\_Slave-F 17. NWay-F <--> 1000\_Master-H 18. NWay-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.3**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay-H) |
| Procedure | Configured as listed parameters:   1. NWay-H <--> NWay(All) 2. NWay-H <--> NWay(10-100-F) 3. NWay-H <--> NWay(1000-F) 4. NWay-H <--> NWay(100-F) 5. NWay-H <--> NWay(10-F) 6. NWay-H <--> NWay(10-100-H) 7. NWay-H <--> NWay(100-H) 8. NWay-H <--> NWay(10-H) 9. NWay-H <--> 100-F 10. NWay-H <--> 10-F 11. NWay-H <--> 100-H 12. NWay-H <--> 10-H 13. NWay-H <--> 1000\_Master-Auto 14. NWay-H <--> 1000\_Slave-Auto 15. NWay-H <--> 1000\_Master-F 16. NWay-H <--> 1000\_Slave-F 17. NWay-H <--> 1000\_Master-H 18. NWay-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.4**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (NWay-10-100) |
| Procedure | Configured as listed parameters:   1. NWay-10-100 <--> NWay(All) 2. NWay-10-100 <--> NWay(10-100-F) 3. NWay-10-100 <--> NWay(1000-F) 4. NWay-10-100 <--> NWay(100-F) 5. NWay-10-100 <--> NWay(10-F) 6. NWay-10-100 <--> NWay(10-100-H) 7. NWay-10-100 <--> NWay(100-H) 8. NWay-10-100 <--> NWay(10-H) 9. NWay-10-100 <--> 100-F 10. NWay-10-100 <--> 10-F 11. NWay-10-100 <--> 100-H 12. NWay-10-100 <--> 10-H 13. NWay-10-100 <--> 1000\_Master-NWay 14. NWay-10-100 <--> 1000\_Slave-NWay 15. NWay-10-100 <--> 1000\_Master-F 16. NWay-10-100 <--> 1000\_Slave-F 17. NWay-10-100 <--> 1000\_Master-H 18. NWay-10-100 <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.5**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10-NWay) |
| Procedure | Configured as listed parameters:   1. 10-NWay <--> NWay(All) 2. 10-NWay <--> NWay(10-100-F) 3. 10-NWay <--> NWay(1000-F) 4. 10-NWay <--> NWay(100-F) 5. 10-NWay <--> NWay(10-F) 6. 10-NWay <--> NWay(10-100-H) 7. 10-NWay <--> NWay(100-H) 8. 10-NWay <--> NWay(10-H) 9. 10-NWay <--> 100-F 10. 10-NWay <--> 10-F 11. 10-NWay <--> 100-H 12. 10-NWay <--> 10-H 13. 10-NWay <--> 1000\_Master-NWay 14. 10-NWay <--> 1000\_Slave-NWay 15. 10-NWay <--> 1000\_Master-F 16. 10-NWay <--> 1000\_Slave-F 17. 10-NWay <--> 1000\_Master-H 18. 10-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.6**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10-F) |
| Procedure | Configured as listed parameters:   1. 10-F <--> NWay(All) 2. 10-F <--> NWay(10-100-F) 3. 10-F <--> NWay(1000-F) 4. 10-F <--> NWay(100-F) 5. 10-F <--> NWay(10-F) 6. 10-F <--> NWay(10-100-H) 7. 10-F <--> NWay(100-H) 8. 10-F <--> NWay(10-H) 9. 10-F <--> 100-F 10. 10-F <--> 10-F 11. 10-F <--> 100-H 12. 10-F <--> 10-H 13. 10-F <--> 1000\_Master-NWay 14. 10-F <--> 1000\_Slave-NWay 15. 10-F <--> 1000\_Master-F 16. 10-F <--> 1000\_Slave-F 17. 10-F <--> 1000\_Master-H 18. 10-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.7**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10-H) |
| Procedure | Configured as listed parameters:   1. 10-H <--> NWay(All) 2. 10-H <--> NWay(10-100-F) 3. 10-H <--> NWay(1000-F) 4. 10-H <--> NWay(100-F) 5. 10-H <--> NWay(10-F) 6. 10-H <--> NWay(10-100-H) 7. 10-H <--> NWay(100-H) 8. 10-H <--> NWay(10-H) 9. 10-H <--> 100-F 10. 10-H <--> 10-F 11. 10-H <--> 100-H 12. 10-H <--> 10-H 13. 10-H <--> 1000\_Master-NWay 14. 10-H <--> 1000\_Slave-NWay 15. 10-H <--> 1000\_Master-F 16. 10-H <--> 1000\_Slave-F 17. 10-H <--> 1000\_Master-H 18. 10-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.8**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (100-NWay) |
| Procedure | Configured as listed parameters:   1. 100-NWay <--> NWay(All) 2. 100-NWay <--> NWay(10-100-F) 3. 100-NWay <--> NWay(1000-F) 4. 100-NWay <--> NWay(100-F) 5. 100-NWay <--> NWay(10-F) 6. 100-NWay <--> NWay(10-100-H) 7. 100-NWay <--> NWay(100-H) 8. 100-NWay <--> NWay(10-H) 9. 100-NWay <--> 100-F 10. 100-NWay <--> 10-F 11. 100-NWay <--> 100-H 12. 100-NWay <--> 10-H 13. 100-NWay <--> 1000\_Master-NWay 14. 100-NWay <--> 1000\_Slave-NWay 15. 100-NWay <--> 1000\_Master-F 16. 100-NWay <--> 1000\_Slave-F 17. 100-NWay <--> 1000\_Master-H 18. 100-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.9**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (100-F) |
| Procedure | Configured as listed parameters:   1. 100-F <--> NWay(All) 2. 100-F <--> NWay(10-100-F) 3. 100-F <--> NWay(1000-F) 4. 100-F <--> NWay(100-F) 5. 100-F <--> NWay(10-F) 6. 100-F <--> NWay(10-100-H) 7. 100-F <--> NWay(100-H) 8. 100-F <--> NWay(10-H) 9. 100-F <--> 100-F 10. 100-F <--> 10-F 11. 100-F <--> 100-H 12. 100-F <--> 10-H 13. 100-F <--> 1000\_Master-NWay 14. 100-F <--> 1000\_Slave-NWay 15. 100-F <--> 1000\_Master-F 16. 100-F <--> 1000\_Slave-F 17. 100-F <--> 1000\_Master-H 18. 100-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.10**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (100-H) |
| Procedure | Configured as listed parameters:   1. 100-H <--> NWay(All) 2. 100-H <--> NWay(10-100-F) 3. 100-H <--> NWay(1000-F) 4. 100-H <--> NWay(100-F) 5. 100-H <--> NWay(10-F) 6. 100-H <--> NWay(10-100-H) 7. 100-H <--> NWay(100-H) 8. 100-H <--> NWay(10-H) 9. 100-H <--> 100-F 10. 100-H <--> 10-F 11. 100-H <--> 100-H 12. 100-H <--> 10-H 13. 100-H <--> 1000\_Master-NWay 14. 100-H <--> 1000\_Slave-NWay 15. 100-H <--> 1000\_Master-F 16. 100-H <--> 1000\_Slave-F 17. 100-H <--> 1000\_Master-H 18. 100-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.11**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Master-NWay <--> NWay(All) 2. 1000\_Master-NWay <--> NWay(10-100-F) 3. 1000\_Master-NWay <--> NWay(1000-F) 4. 1000\_Master-NWay <--> NWay(100-F) 5. 1000\_Master-NWay <--> NWay(10-F) 6. 1000\_Master-NWay <--> NWay(10-100-H) 7. 1000\_Master-NWay <--> NWay(100-H) 8. 1000\_Master-NWay <--> NWay(10-H) 9. 1000\_Master-NWay <--> 100-F 10. 1000\_Master-NWay <--> 10-F 11. 1000\_Master-NWay <--> 100-H 12. 1000\_Master-NWay <--> 10-H 13. 1000\_Master-NWay <--> 1000\_Master-NWay 14. 1000\_Master-NWay <--> 1000\_Slave-NWay 15. 1000\_Master-NWay <--> 1000\_Master-F 16. 1000\_Master-NWay <--> 1000\_Slave-F 17. 1000\_Master-NWay <--> 1000\_Master-H 18. 1000\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.12**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave-NWay <--> NWay(All) 2. 1000\_Slave-NWay <--> NWay(10-100-F) 3. 1000\_Slave-NWay <--> NWay(1000-F) 4. 1000\_Slave-NWay <--> NWay(100-F) 5. 1000\_Slave-NWay <--> NWay(10-F) 6. 1000\_Slave-NWay <--> NWay(10-100-H) 7. 1000\_Slave-NWay <--> NWay(100-H) 8. 1000\_Slave-NWay <--> NWay(10-H) 9. 1000\_Slave-NWay <--> 100-F 10. 1000\_Slave-NWay <--> 10-F 11. 1000\_Slave-NWay <--> 100-H 12. 1000\_Slave-NWay <--> 10-H 13. 1000\_Slave-NWay <--> 1000\_Master-NWay 14. 1000\_Slave-NWay <--> 1000\_Slave-NWay 15. 1000\_Slave-NWay <--> 1000\_Master-F 16. 1000\_Slave-NWay <--> 1000\_Slave-F 17. 1000\_Slave-NWay <--> 1000\_Master-H 18. 1000\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.13**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Master\_F <--> NWay(All) 2. 1000\_Master\_F <--> NWay(10-100-F) 3. 1000\_Master\_F <--> NWay(1000-F) 4. 1000\_Master\_F <--> NWay(100-F) 5. 1000\_Master\_F <--> NWay(10-F) 6. 1000\_Master\_F <--> NWay(10-100-H) 7. 1000\_Master\_F <--> NWay(100-H) 8. 1000\_Master\_F <--> NWay(10-H) 9. 1000\_Master\_F <--> 100-F 10. 1000\_Master\_F <--> 100-H 11. 1000\_Master\_F <--> 10-F 12. 1000\_Master\_F <--> 10-H 13. 1000\_Master\_F <--> 1000\_Master-NWay 14. 1000\_Master\_F <--> 1000\_Slave-NWay 15. 1000\_Master\_F <--> 1000\_Master-F 16. 1000\_Master\_F <--> 1000\_Slave-F 17. 1000\_Master\_F <--> 1000\_Master-H 18. 1000\_Master\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.14**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave\_F <--> NWay(All) 2. 1000\_Slave\_F <--> NWay(10-100-F) 3. 1000\_Slave\_F <--> NWay(1000-F) 4. 1000\_Slave\_F <--> NWay(100-F) 5. 1000\_Slave\_F <--> NWay(10-F) 6. 1000\_Slave\_F <--> NWay(10-100-H) 7. 1000\_Slave\_F <--> NWay(100-H) 8. 1000\_Slave\_F <--> NWay(10-H) 9. 1000\_Slave\_F <--> 100-F 10. 1000\_Slave\_F <--> 100-H 11. 1000\_Slave\_F <--> 10-F 12. 1000\_Slave\_F <--> 10-H 13. 1000\_Slave\_F <--> 1000\_Master-NWay 14. 1000\_Slave\_F <--> 1000\_Slave-NWay 15. 1000\_Slave\_F <--> 1000\_Master-F 16. 1000\_Slave\_F <--> 1000\_Slave-F 17. 1000\_Slave\_F <--> 1000\_Master-H 18. 1000\_Slave\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.15**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Master\_H <--> NWay(All) 2. 1000\_Master\_H <--> NWay(10-100-F) 3. 1000\_Master\_H <--> NWay(1000-F) 4. 1000\_Master\_H <--> NWay(100-F) 5. 1000\_Master\_H <--> NWay(10-F) 6. 1000\_Master\_H <--> NWay(10-100-H) 7. 1000\_Master\_H <--> NWay(100-H) 8. 1000\_Master\_H <--> NWay(10-H) 9. 1000\_Master\_H <--> 100-F 10. 1000\_Master\_H <--> 100-H 11. 1000\_Master\_H <--> 10-F 12. 1000\_Master\_H <--> 10-H 13. 1000\_Master\_H <--> 1000\_Master-NWay 14. 1000\_Master\_H <--> 1000\_Slave-NWay 15. 1000\_Master\_H <--> 1000\_Master-F 16. 1000\_Master\_H <--> 1000\_Slave-F 17. 1000\_Master\_H <--> 1000\_Master-H 18. 1000\_Master\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: Fiber\_Cat 2.16.4.16**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave\_H <--> NWay(All) 2. 1000\_Slave\_H <--> NWay(10-100-F) 3. 1000\_Slave\_H <--> NWay(1000-F) 4. 1000\_Slave\_H <--> NWay(100-F) 5. 1000\_Slave\_H <--> NWay(10-F) 6. 1000\_Slave\_H <--> NWay(10-100-H) 7. 1000\_Slave\_H <--> NWay(100-H) 8. 1000\_Slave\_H <--> NWay(10-H) 9. 1000\_Slave\_H <--> 100-F 10. 1000\_Slave\_H <--> 100-H 11. 1000\_Slave\_H <--> 10-F 12. 1000\_Slave\_H <--> 10-H 13. 1000\_Slave\_H <--> 1000\_Master-NWay 14. 1000\_Slave\_H <--> 1000\_Slave-NWay 15. 1000\_Slave\_H <--> 1000\_Master-F 16. 1000\_Slave\_H <--> 1000\_Slave-F 17. 1000\_Slave\_H <--> 1000\_Master-H 18. 1000\_Slave\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Fiber 2.16.5.1**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Master-NWay) |
| Procedure | Configured as listed parameters:   1. 10G\_Master-NWay <--> NWay(All) 2. 10G\_Master-NWay <--> 10G\_Master-NWay 3. 10G\_Master-NWay <--> 10G\_Slave-NWay 4. 10G\_Master-NWay <--> 10G\_Master-F 5. 10G\_Master-NWay <--> 10G\_Slave-F 6. 10G\_Master-NWay <--> 10G\_Master-H 7. 10G\_Master-NWay <--> 10G\_Slave-H 8. 10G\_Master-NWay <--> 1000\_Master-NWay 9. 10G\_Master-NWay <--> 1000\_Slave-NWay 10. 10G\_Master-NWay <--> 1000\_Master-F 11. 10G\_Master-NWay <--> 1000\_Slave-F 12. 10G\_Master-NWay <--> 1000\_Master-H 13. 10G\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Fiber 2.16.5.2**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 10G\_Slave-NWay <--> NWay(All) 2. 10G\_Slave-NWay <--> 10G\_Master-NWay 3. 10G\_Slave-NWay <--> 10G\_Slave-NWay 4. 10G\_Slave-NWay <--> 10G\_Master-F 5. 10G\_Slave-NWay <--> 10G\_Slave-F 6. 10G\_Slave-NWay <--> 10G\_Master-H 7. 10G\_Slave-NWay <--> 10G\_Slave-H 8. 10G\_Slave-NWay <--> 1000\_Master-NWay 9. 10G\_Slave-NWay <--> 1000\_Slave-NWay 10. 10G\_Slave-NWay <--> 1000\_Master-F 11. 10G\_Slave-NWay <--> 1000\_Slave-F 12. 10G\_Slave-NWay <--> 1000\_Master-H 13. 10G\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Fiber 2.16.5.3**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Master-F) |
| Procedure | Configured as listed parameters:   1. 10G\_Master\_F <--> NWay(All) 2. 10G\_Master\_F <--> 10G\_Master-NWay 3. 10G\_Master\_F <--> 10G\_Slave-NWay 4. 10G\_Master\_F <--> 10G\_Master-F 5. 10G\_Master\_F <--> 10G\_Slave-F 6. 10G\_Master\_F <--> 10G\_Master-H 7. 10G\_Master\_F <--> 10G\_Slave-H 8. 10G\_Master\_F <--> 1000\_Master-NWay 9. 10G\_Master\_F <--> 1000\_Slave-NWay 10. 10G\_Master\_F <--> 1000\_Master-F 11. 10G\_Master\_F <--> 1000\_Slave-F 12. 10G\_Master\_F <--> 1000\_Master-H 13. 10G\_Master\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Fiber 2.16.5.4**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Slave-F) |
| Procedure | Configured as listed parameters:   1. 10G\_Slave\_F <--> NWay(All) 2. 10G\_Slave\_F <--> 10G\_Master-NWay 3. 10G\_Slave\_F <--> 10G\_Slave-NWay 4. 10G\_Slave\_F <--> 10G\_Master-F 5. 10G\_Slave\_F <--> 10G\_Slave-F 6. 10G\_Slave\_F <--> 10G\_Master-H 7. 10G\_Slave\_F <--> 10G\_Slave-H 8. 10G\_Slave\_F <--> 1000\_Master-NWay 9. 10G\_Slave\_F <--> 1000\_Slave-NWay 10. 10G\_Slave\_F <--> 1000\_Master-F 11. 10G\_Slave\_F <--> 1000\_Slave-F 12. 10G\_Slave\_F <--> 1000\_Master-H 13. 10G\_Slave\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Fiber 2.16.5.5**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Master-H) |
| Procedure | Configured as listed parameters:   1. 10G\_Master\_H <--> NWay(All) 2. 10G\_Master\_H <--> 10G\_Master-NWay 3. 10G\_Master\_H <--> 10G\_Slave-NWay 4. 10G\_Master\_H <--> 10G\_Master-F 5. 10G\_Master\_H <--> 10G\_Slave-F 6. 10G\_Master\_H <--> 10G\_Master-H 7. 10G\_Master\_H <--> 10G\_Slave-H 8. 10G\_Master\_H <--> 1000\_Master-NWay 9. 10G\_Master\_H <--> 1000\_Slave-NWay 10. 10G\_Master\_H <--> 1000\_Master-F 11. 10G\_Master\_H <--> 1000\_Slave-F 12. 10G\_Master\_H <--> 1000\_Master-H 13. 10G\_Master\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Fiber 2.16.5.6**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Slave-H) |
| Procedure | Configured as listed parameters:   1. 10G\_Slave\_H <--> NWay(All) 2. 10G\_Slave\_H <--> 10G\_Master-NWay 3. 10G\_Slave\_H <--> 10G\_Slave-NWay 4. 10G\_Slave\_H <--> 10G\_Master-F 5. 10G\_Slave\_H <--> 10G\_Slave-F 6. 10G\_Slave\_H <--> 10G\_Master-H 7. 10G\_Slave\_H <--> 10G\_Slave-H 8. 10G\_Slave\_H <--> 1000\_Master-NWay 9. 10G\_Slave\_H <--> 1000\_Slave-NWay 10. 10G\_Slave\_H <--> 1000\_Master-F 11. 10G\_Slave\_H <--> 1000\_Slave-F 12. 10G\_Slave\_H <--> 1000\_Master-H 13. 10G\_Slave\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Fiber 2.16.5.7**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Master-NWay <--> NWay(All) 2. 1000\_Master-NWay <--> 10G\_Master-NWay 3. 1000\_Master-NWay <--> 10G\_Slave-NWay 4. 1000\_Master-NWay <--> 10G\_Master-F 5. 1000\_Master-NWay <--> 10G\_Slave-F 6. 1000\_Master-NWay <--> 10G\_Master-H 7. 1000\_Master-NWay <--> 10G\_Slave-H 8. 1000\_Master-NWay <--> 1000\_Master-NWay 9. 1000\_Master-NWay <--> 1000\_Slave-NWay 10. 1000\_Master-NWay <--> 1000\_Master-F 11. 1000\_Master-NWay <--> 1000\_Slave-F 12. 1000\_Master-NWay <--> 1000\_Master-H 13. 1000\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Fiber 2.16.5.8**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave-NWay <--> NWay(All) 2. 1000\_Slave-NWay <--> 10G\_Master-NWay 3. 1000\_Slave-NWay <--> 10G\_Slave-NWay 4. 1000\_Slave-NWay <--> 10G\_Master-F 5. 1000\_Slave-NWay <--> 10G\_Slave-F 6. 1000\_Slave-NWay <--> 10G\_Master-H 7. 1000\_Slave-NWay <--> 10G\_Slave-H 8. 1000\_Slave-NWay <--> 1000\_Master-NWay 9. 1000\_Slave-NWay <--> 1000\_Slave-NWay 10. 1000\_Slave-NWay <--> 1000\_Master-F 11. 1000\_Slave-NWay <--> 1000\_Slave-F 12. 1000\_Slave-NWay <--> 1000\_Master-H 13. 1000\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Fiber 2.16.5.9**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Master\_F <--> NWay(All) 2. 1000\_Master\_F <--> 10G\_Master-NWay 3. 1000\_Master\_F <--> 10G\_Slave-NWay 4. 1000\_Master\_F <--> 10G\_Master-F 5. 1000\_Master\_F <--> 10G\_Slave-F 6. 1000\_Master\_F <--> 10G\_Master-H 7. 1000\_Master\_F <--> 10G\_Slave-H 8. 1000\_Master\_F <--> 1000\_Master-NWay 9. 1000\_Master\_F <--> 1000\_Slave-NWay 10. 1000\_Master\_F <--> 1000\_Master-F 11. 1000\_Master\_F <--> 1000\_Slave-F 12. 1000\_Master\_F <--> 1000\_Master-H 13. 1000\_Master\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Fiber 2.16.5.10**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave-F <--> NWay(All) 2. 1000\_Slave-F <--> 40G\_Master-NWay 3. 1000\_Slave-F <--> 40G\_Slave-NWay 4. 1000\_Slave-F <--> 40G\_Master-F 5. 1000\_Slave-F <--> 40G\_Slave-F 6. 1000\_Slave-F <--> 40G\_Master-H 7. 1000\_Slave-F <--> 40G\_Slave-H 8. 1000\_Slave-F <--> 10G\_Master-NWay 9. 1000\_Slave-F <--> 10G\_Slave-NWay 10. 1000\_Slave-F <--> 10G\_Master-F 11. 1000\_Slave-F <--> 10G\_Slave-F 12. 1000\_Slave-F <--> 10G\_Master-H 13. 1000\_Slave-F <--> 10G\_Slave-H 14. 1000\_Slave-F <--> 1000\_Master-NWay 15. 1000\_Slave-F <--> 1000\_Slave-NWay 16. 1000\_Slave-F <--> 1000\_Master-F 17. 1000\_Slave-F <--> 1000\_Slave-F 18. 1000\_Slave-F <--> 1000\_Master-H 19. 1000\_Slave-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Fiber 2.16.5.11**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Master\_H <--> NWay(All) 2. 1000\_Master\_H <--> 10G\_Master-NWay 3. 1000\_Master\_H <--> 10G\_Slave-NWay 4. 1000\_Master\_H <--> 10G\_Master-F 5. 1000\_Master\_H <--> 10G\_Slave-F 6. 1000\_Master\_H <--> 10G\_Master-H 7. 1000\_Master\_H <--> 10G\_Slave-H 8. 1000\_Master\_H <--> 1000\_Master-NWay 9. 1000\_Master\_H <--> 1000\_Slave-NWay 10. 1000\_Master\_H <--> 1000\_Master-F 11. 1000\_Master\_H <--> 1000\_Slave-F 12. 1000\_Master\_H <--> 1000\_Master-H 13. 1000\_Master\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Fiber 2.16.5.12**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave\_H <--> NWay(All) 2. 1000\_Slave\_H <--> 10G\_Master-NWay 3. 1000\_Slave\_H <--> 10G\_Slave-NWay 4. 1000\_Slave\_H <--> 10G\_Master-F 5. 1000\_Slave\_H <--> 10G\_Slave-F 6. 1000\_Slave\_H <--> 10G\_Master-H 7. 1000\_Slave\_H <--> 10G\_Slave-H 8. 1000\_Slave\_H <--> 1000\_Master-NWay 9. 1000\_Slave\_H <--> 1000\_Slave-NWay 10. 1000\_Slave\_H <--> 1000\_Master-F 11. 1000\_Slave\_H <--> 1000\_Slave-F 12. 1000\_Slave\_H <--> 1000\_Master-H 13. 1000\_Slave\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Cat 2.16.6.1**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Master-NWay) |
| Procedure | Configured as listed parameters:   1. 10G\_Master-NWay <--> NWay(All) 2. 10G\_Master-NWay <--> 10G\_Master-NWay 3. 10G\_Master-NWay <--> 10G\_Slave-NWay 4. 10G\_Master-NWay <--> 10G\_Master-F 5. 10G\_Master-NWay <--> 10G\_Slave-F 6. 10G\_Master-NWay <--> 10G\_Master-H 7. 10G\_Master-NWay <--> 10G\_Slave-H 8. 10G\_Master-NWay <--> 1000\_Master-NWay 9. 10G\_Master-NWay <--> 1000\_Slave-NWay 10. 10G\_Master-NWay <--> 1000\_Master-F 11. 10G\_Master-NWay <--> 1000\_Slave-F 12. 10G\_Master-NWay <--> 1000\_Master-H 13. 10G\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Cat 2.16.6.2**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 10G\_Slave-NWay <--> NWay(All) 2. 10G\_Slave-NWay <--> 10G\_Master-NWay 3. 10G\_Slave-NWay <--> 10G\_Slave-NWay 4. 10G\_Slave-NWay <--> 10G\_Master-F 5. 10G\_Slave-NWay <--> 10G\_Slave-F 6. 10G\_Slave-NWay <--> 10G\_Master-H 7. 10G\_Slave-NWay <--> 10G\_Slave-H 8. 10G\_Slave-NWay <--> 1000\_Master-NWay 9. 10G\_Slave-NWay <--> 1000\_Slave-NWay 10. 10G\_Slave-NWay <--> 1000\_Master-F 11. 10G\_Slave-NWay <--> 1000\_Slave-F 12. 10G\_Slave-NWay <--> 1000\_Master-H 13. 10G\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Cat 2.16.6.3**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Master-F) |
| Procedure | Configured as listed parameters:   1. 10G\_Master\_F <--> NWay(All) 2. 10G\_Master\_F <--> 10G\_Master-NWay 3. 10G\_Master\_F <--> 10G\_Slave-NWay 4. 10G\_Master\_F <--> 10G\_Master-F 5. 10G\_Master\_F <--> 10G\_Slave-F 6. 10G\_Master\_F <--> 10G\_Master-H 7. 10G\_Master\_F <--> 10G\_Slave-H 8. 10G\_Master\_F <--> 1000\_Master-NWay 9. 10G\_Master\_F <--> 1000\_Slave-NWay 10. 10G\_Master\_F <--> 1000\_Master-F 11. 10G\_Master\_F <--> 1000\_Slave-F 12. 10G\_Master\_F <--> 1000\_Master-H 13. 10G\_Master\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Cat 2.16.6.4**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Slave-F) |
| Procedure | Configured as listed parameters:   1. 10G\_Slave\_F <--> NWay(All) 2. 10G\_Slave\_F <--> 10G\_Master-NWay 3. 10G\_Slave\_F <--> 10G\_Slave-NWay 4. 10G\_Slave\_F <--> 10G\_Master-F 5. 10G\_Slave\_F <--> 10G\_Slave-F 6. 10G\_Slave\_F <--> 10G\_Master-H 7. 10G\_Slave\_F <--> 10G\_Slave-H 8. 10G\_Slave\_F <--> 1000\_Master-NWay 9. 10G\_Slave\_F <--> 1000\_Slave-NWay 10. 10G\_Slave\_F <--> 1000\_Master-F 11. 10G\_Slave\_F <--> 1000\_Slave-F 12. 10G\_Slave\_F <--> 1000\_Master-H 13. 10G\_Slave\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Cat 2.16.6.5**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Master-H) |
| Procedure | Configured as listed parameters:   1. 10G\_Master\_H <--> NWay(All) 2. 10G\_Master\_H <--> 10G\_Master-NWay 3. 10G\_Master\_H <--> 10G\_Slave-NWay 4. 10G\_Master\_H <--> 10G\_Master-F 5. 10G\_Master\_H <--> 10G\_Slave-F 6. 10G\_Master\_H <--> 10G\_Master-H 7. 10G\_Master\_H <--> 10G\_Slave-H 8. 10G\_Master\_H <--> 1000\_Master-NWay 9. 10G\_Master\_H <--> 1000\_Slave-NWay 10. 10G\_Master\_H <--> 1000\_Master-F 11. 10G\_Master\_H <--> 1000\_Slave-F 12. 10G\_Master\_H <--> 1000\_Master-H 13. 10G\_Master\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Cat 2.16.6.6**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Slave-H) |
| Procedure | Configured as listed parameters:   1. 10G\_Slave\_H <--> NWay(All) 2. 10G\_Slave\_H <--> 10G\_Master-NWay 3. 10G\_Slave\_H <--> 10G\_Slave-NWay 4. 10G\_Slave\_H <--> 10G\_Master-F 5. 10G\_Slave\_H <--> 10G\_Slave-F 6. 10G\_Slave\_H <--> 10G\_Master-H 7. 10G\_Slave\_H <--> 10G\_Slave-H 8. 10G\_Slave\_H <--> 1000\_Master-NWay 9. 10G\_Slave\_H <--> 1000\_Slave-NWay 10. 10G\_Slave\_H <--> 1000\_Master-F 11. 10G\_Slave\_H <--> 1000\_Slave-F 12. 10G\_Slave\_H <--> 1000\_Master-H 13. 10G\_Slave\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Cat 2.16.6.7**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Master-NWay <--> NWay(All) 2. 1000\_Master-NWay <--> 10G\_Master-NWay 3. 1000\_Master-NWay <--> 10G\_Slave-NWay 4. 1000\_Master-NWay <--> 10G\_Master-F 5. 1000\_Master-NWay <--> 10G\_Slave-F 6. 1000\_Master-NWay <--> 10G\_Master-H 7. 1000\_Master-NWay <--> 10G\_Slave-H 8. 1000\_Master-NWay <--> 1000\_Master-NWay 9. 1000\_Master-NWay <--> 1000\_Slave-NWay 10. 1000\_Master-NWay <--> 1000\_Master-F 11. 1000\_Master-NWay <--> 1000\_Slave-F 12. 1000\_Master-NWay <--> 1000\_Master-H 13. 1000\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Cat 2.16.6.8**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave-NWay <--> NWay(All) 2. 1000\_Slave-NWay <--> 10G\_Master-NWay 3. 1000\_Slave-NWay <--> 10G\_Slave-NWay 4. 1000\_Slave-NWay <--> 10G\_Master-F 5. 1000\_Slave-NWay <--> 10G\_Slave-F 6. 1000\_Slave-NWay <--> 10G\_Master-H 7. 1000\_Slave-NWay <--> 10G\_Slave-H 8. 1000\_Slave-NWay <--> 1000\_Master-NWay 9. 1000\_Slave-NWay <--> 1000\_Slave-NWay 10. 1000\_Slave-NWay <--> 1000\_Master-F 11. 1000\_Slave-NWay <--> 1000\_Slave-F 12. 1000\_Slave-NWay <--> 1000\_Master-H 13. 1000\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Cat 2.16.6.9**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Master\_F <--> NWay(All) 2. 1000\_Master\_F <--> 10G\_Master-NWay 3. 1000\_Master\_F <--> 10G\_Slave-NWay 4. 1000\_Master\_F <--> 10G\_Master-F 5. 1000\_Master\_F <--> 10G\_Slave-F 6. 1000\_Master\_F <--> 10G\_Master-H 7. 1000\_Master\_F <--> 10G\_Slave-H 8. 1000\_Master\_F <--> 1000\_Master-NWay 9. 1000\_Master\_F <--> 1000\_Slave-NWay 10. 1000\_Master\_F <--> 1000\_Master-F 11. 1000\_Master\_F <--> 1000\_Slave-F 12. 1000\_Master\_F <--> 1000\_Master-H 13. 1000\_Master\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Cat 2.16.6.10**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave-F <--> NWay(All) 2. 1000\_Slave-F <--> 40G\_Master-NWay 3. 1000\_Slave-F <--> 40G\_Slave-NWay 4. 1000\_Slave-F <--> 40G\_Master-F 5. 1000\_Slave-F <--> 40G\_Slave-F 6. 1000\_Slave-F <--> 40G\_Master-H 7. 1000\_Slave-F <--> 40G\_Slave-H 8. 1000\_Slave-F <--> 10G\_Master-NWay 9. 1000\_Slave-F <--> 10G\_Slave-NWay 10. 1000\_Slave-F <--> 10G\_Master-F 11. 1000\_Slave-F <--> 10G\_Slave-F 12. 1000\_Slave-F <--> 10G\_Master-H 13. 1000\_Slave-F <--> 10G\_Slave-H 14. 1000\_Slave-F <--> 1000\_Master-NWay 15. 1000\_Slave-F <--> 1000\_Slave-NWay 16. 1000\_Slave-F <--> 1000\_Master-F 17. 1000\_Slave-F <--> 1000\_Slave-F 18. 1000\_Slave-F <--> 1000\_Master-H 19. 1000\_Slave-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Cat 2.16.6.11**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Master\_H <--> NWay(All) 2. 1000\_Master\_H <--> 10G\_Master-NWay 3. 1000\_Master\_H <--> 10G\_Slave-NWay 4. 1000\_Master\_H <--> 10G\_Master-F 5. 1000\_Master\_H <--> 10G\_Slave-F 6. 1000\_Master\_H <--> 10G\_Master-H 7. 1000\_Master\_H <--> 10G\_Slave-H 8. 1000\_Master\_H <--> 1000\_Master-NWay 9. 1000\_Master\_H <--> 1000\_Slave-NWay 10. 1000\_Master\_H <--> 1000\_Master-F 11. 1000\_Master\_H <--> 1000\_Slave-F 12. 1000\_Master\_H <--> 1000\_Master-H 13. 1000\_Master\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 10G\_Fiber\_Cat 2.16.6.12**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave\_H <--> NWay(All) 2. 1000\_Slave\_H <--> 10G\_Master-NWay 3. 1000\_Slave\_H <--> 10G\_Slave-NWay 4. 1000\_Slave\_H <--> 10G\_Master-F 5. 1000\_Slave\_H <--> 10G\_Slave-F 6. 1000\_Slave\_H <--> 10G\_Master-H 7. 1000\_Slave\_H <--> 10G\_Slave-H 8. 1000\_Slave\_H <--> 1000\_Master-NWay 9. 1000\_Slave\_H <--> 1000\_Slave-NWay 10. 1000\_Slave\_H <--> 1000\_Master-F 11. 1000\_Slave\_H <--> 1000\_Slave-F 12. 1000\_Slave\_H <--> 1000\_Master-H 13. 1000\_Slave\_H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.1**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (40G-Master-NWay) |
| Procedure | Configured as listed parameters:   1. 40G\_Master-NWay <--> NWay(All) 2. 40G\_Master-NWay <--> 40G\_Master-NWay 3. 40G\_Master-NWay <--> 40G\_Slave-NWay 4. 40G\_Master-NWay <--> 40G\_Master-F 5. 40G\_Master-NWay <--> 40G\_Slave-F 6. 40G\_Master-NWay <--> 40G\_Master-H 7. 40G\_Master-NWay <--> 40G\_Slave-H 8. 40G\_Master-NWay <--> 10G\_Master-NWay 9. 40G\_Master-NWay <--> 10G\_Slave-NWay 10. 40G\_Master-NWay <--> 10G\_Master-F 11. 40G\_Master-NWay <--> 10G\_Slave-F 12. 40G\_Master-NWay <--> 10G\_Master-H 13. 40G\_Master-NWay <--> 10G\_Slave-H 14. 40G\_Master-NWay <--> 1000\_Master-NWay 15. 40G\_Master-NWay <--> 1000\_Slave-NWay 16. 40G\_Master-NWay <--> 1000\_Master-F 17. 40G\_Master-NWay <--> 1000\_Slave-F 18. 40G\_Master-NWay <--> 1000\_Master-H 19. 40G\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.2**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (40G-Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 40G\_Slave-NWay <--> NWay(All) 2. 40G\_Slave-NWay <--> 40G\_Master-NWay 3. 40G\_Slave-NWay <--> 40G\_Slave-NWay 4. 40G\_Slave-NWay <--> 40G\_Master-F 5. 40G\_Slave-NWay <--> 40G\_Slave-F 6. 40G\_Slave-NWay <--> 40G\_Master-H 7. 40G\_Slave-NWay <--> 40G\_Slave-H 8. 40G\_Slave-NWay <--> 10G\_Master-NWay 9. 40G\_Slave-NWay <--> 10G\_Slave-NWay 10. 40G\_Slave-NWay <--> 10G\_Master-F 11. 40G\_Slave-NWay <--> 10G\_Slave-F 12. 40G\_Slave-NWay <--> 10G\_Master-H 13. 40G\_Slave-NWay <--> 10G\_Slave-H 14. 40G\_Slave-NWay <--> 1000\_Master-NWay 15. 40G\_Slave-NWay <--> 1000\_Slave-NWay 16. 40G\_Slave-NWay <--> 1000\_Master-F 17. 40G\_Slave-NWay <--> 1000\_Slave-F 18. 40G\_Slave-NWay <--> 1000\_Master-H 19. 40G\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.3**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (40G-Master-F) |
| Procedure | Configured as listed parameters:   1. 40G\_Master-F <--> NWay(All) 2. 40G\_Master-F <--> 40G\_Master-NWay 3. 40G\_Master-F <--> 40G\_Slave-NWay 4. 40G\_Master-F <--> 40G\_Master-F 5. 40G\_Master-F <--> 40G\_Slave-F 6. 40G\_Master-F <--> 40G\_Master-H 7. 40G\_Master-F <--> 40G\_Slave-H 8. 40G\_Master-F <--> 10G\_Master-NWay 9. 40G\_Master-F <--> 10G\_Slave-NWay 10. 40G\_Master-F <--> 10G\_Master-F 11. 40G\_Master-F <--> 10G\_Slave-F 12. 40G\_Master-F <--> 10G\_Master-H 13. 40G\_Master-F <--> 10G\_Slave-H 14. 40G\_Master-F <--> 1000\_Master-NWay 15. 40G\_Master-F <--> 1000\_Slave-NWay 16. 40G\_Master-F <--> 1000\_Master-F 17. 40G\_Master-F <--> 1000\_Slave-F 18. 40G\_Master-F <--> 1000\_Master-H 19. 40G\_Master-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.4**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (40G-Slave-F) |
| Procedure | Configured as listed parameters:   1. 40G\_Slave-F <--> NWay(All) 2. 40G\_Slave-F <--> 40G\_Master-NWay 3. 40G\_Slave-F <--> 40G\_Slave-NWay 4. 40G\_Slave-F <--> 40G\_Master-F 5. 40G\_Slave-F <--> 40G\_Slave-F 6. 40G\_Slave-F <--> 40G\_Master-H 7. 40G\_Slave-F <--> 40G\_Slave-H 8. 40G\_Slave-F <--> 10G\_Master-NWay 9. 40G\_Slave-F <--> 10G\_Slave-NWay 10. 40G\_Slave-F <--> 10G\_Master-F 11. 40G\_Slave-F <--> 10G\_Slave-F 12. 40G\_Slave-F <--> 10G\_Master-H 13. 40G\_Slave-F <--> 10G\_Slave-H 14. 40G\_Slave-F <--> 1000\_Master-NWay 15. 40G\_Slave-F <--> 1000\_Slave-NWay 16. 40G\_Slave-F <--> 1000\_Master-F 17. 40G\_Slave-F <--> 1000\_Slave-F 18. 40G\_Slave-F <--> 1000\_Master-H 19. 40G\_Slave-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.5**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (40G-Master-H) |
| Procedure | Configured as listed parameters:   1. 40G\_Master-H <--> NWay(All) 2. 40G\_Master-H <--> 40G\_Master-NWay 3. 40G\_Master-H <--> 40G\_Slave-NWay 4. 40G\_Master-H <--> 40G\_Master-F 5. 40G\_Master-H <--> 40G\_Slave-F 6. 40G\_Master-H <--> 40G\_Master-H 7. 40G\_Master-H <--> 40G\_Slave-H 8. 40G\_Master-H <--> 10G\_Master-NWay 9. 40G\_Master-H <--> 10G\_Slave-NWay 10. 40G\_Master-H <--> 10G\_Master-F 11. 40G\_Master-H <--> 10G\_Slave-F 12. 40G\_Master-H <--> 10G\_Master-H 13. 40G\_Master-H <--> 10G\_Slave-H 14. 40G\_Master-H <--> 1000\_Master-NWay 15. 40G\_Master-H <--> 1000\_Slave-NWay 16. 40G\_Master-H <--> 1000\_Master-F 17. 40G\_Master-H <--> 1000\_Slave-F 18. 40G\_Master-H <--> 1000\_Master-H 19. 40G\_Master-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.6**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (40G-Slave-H) |
| Procedure | Configured as listed parameters:   1. 40G\_Slave-H <--> NWay(All) 2. 40G\_Slave-H <--> 40G\_Master-NWay 3. 40G\_Slave-H <--> 40G\_Slave-NWay 4. 40G\_Slave-H <--> 40G\_Master-F 5. 40G\_Slave-H <--> 40G\_Slave-F 6. 40G\_Slave-H <--> 40G\_Master-H 7. 40G\_Slave-H <--> 40G\_Slave-H 8. 40G\_Slave-H <--> 10G\_Master-NWay 9. 40G\_Slave-H <--> 10G\_Slave-NWay 10. 40G\_Slave-H <--> 10G\_Master-F 11. 40G\_Slave-H <--> 10G\_Slave-F 12. 40G\_Slave-H <--> 10G\_Master-H 13. 40G\_Slave-H <--> 10G\_Slave-H 14. 40G\_Slave-H <--> 1000\_Master-NWay 15. 40G\_Slave-H <--> 1000\_Slave-NWay 16. 40G\_Slave-H <--> 1000\_Master-F 17. 40G\_Slave-H <--> 1000\_Slave-F 18. 40G\_Slave-H <--> 1000\_Master-H 19. 40G\_Slave-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.7**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Master-NWay) |
| Procedure | Configured as listed parameters:   1. 10G\_Master-NWay <--> NWay(All) 2. 10G\_Master-NWay <--> 40G\_Master-NWay 3. 10G\_Master-NWay <--> 40G\_Slave-NWay 4. 10G\_Master-NWay <--> 40G\_Master-F 5. 10G\_Master-NWay <--> 40G\_Slave-F 6. 10G\_Master-NWay <--> 40G\_Master-H 7. 10G\_Master-NWay <--> 40G\_Slave-H 8. 10G\_Master-NWay <--> 10G\_Master-NWay 9. 10G\_Master-NWay <--> 10G\_Slave-NWay 10. 10G\_Master-NWay <--> 10G\_Master-F 11. 10G\_Master-NWay <--> 10G\_Slave-F 12. 10G\_Master-NWay <--> 10G\_Master-H 13. 10G\_Master-NWay <--> 10G\_Slave-H 14. 10G\_Master-NWay <--> 1000\_Master-NWay 15. 10G\_Master-NWay <--> 1000\_Slave-NWay 16. 10G\_Master-NWay <--> 1000\_Master-F 17. 10G\_Master-NWay <--> 1000\_Slave-F 18. 10G\_Master-NWay <--> 1000\_Master-H 19. 10G\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.8**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 10G\_Slave-NWay <--> NWay(All) 2. 10G\_Slave-NWay <--> 40G\_Master-NWay 3. 10G\_Slave-NWay <--> 40G\_Slave-NWay 4. 10G\_Slave-NWay <--> 40G\_Master-F 5. 10G\_Slave-NWay <--> 40G\_Slave-F 6. 10G\_Slave-NWay <--> 40G\_Master-H 7. 10G\_Slave-NWay <--> 40G\_Slave-H 8. 10G\_Slave-NWay <--> 10G\_Master-NWay 9. 10G\_Slave-NWay <--> 10G\_Slave-NWay 10. 10G\_Slave-NWay <--> 10G\_Master-F 11. 10G\_Slave-NWay <--> 10G\_Slave-F 12. 10G\_Slave-NWay <--> 10G\_Master-H 13. 10G\_Slave-NWay <--> 10G\_Slave-H 14. 10G\_Slave-NWay <--> 1000\_Master-NWay 15. 10G\_Slave-NWay <--> 1000\_Slave-NWay 16. 10G\_Slave-NWay <--> 1000\_Master-F 17. 10G\_Slave-NWay <--> 1000\_Slave-F 18. 10G\_Slave-NWay <--> 1000\_Master-H 19. 10G\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.9**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Master-F) |
| Procedure | Configured as listed parameters:   1. 10G\_Master-F <--> NWay(All) 2. 10G\_Master-F <--> 40G\_Master-NWay 3. 10G\_Master-F <--> 40G\_Slave-NWay 4. 10G\_Master-F <--> 40G\_Master-F 5. 10G\_Master-F <--> 40G\_Slave-F 6. 10G\_Master-F <--> 40G\_Master-H 7. 10G\_Master-F <--> 40G\_Slave-H 8. 10G\_Master-F <--> 10G\_Master-NWay 9. 10G\_Master-F <--> 10G\_Slave-NWay 10. 10G\_Master-F <--> 10G\_Master-F 11. 10G\_Master-F <--> 10G\_Slave-F 12. 10G\_Master-F <--> 10G\_Master-H 13. 10G\_Master-F <--> 10G\_Slave-H 14. 10G\_Master-F <--> 1000\_Master-NWay 15. 10G\_Master-F <--> 1000\_Slave-NWay 16. 10G\_Master-F <--> 1000\_Master-F 17. 10G\_Master-F <--> 1000\_Slave-F 18. 10G\_Master-F <--> 1000\_Master-H 19. 10G\_Master-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.10**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Slave-F) |
| Procedure | Configured as listed parameters:   1. 10G\_Slave-F <--> NWay(All) 2. 10G\_Slave-F <--> 40G\_Master-NWay 3. 10G\_Slave-F <--> 40G\_Slave-NWay 4. 10G\_Slave-F <--> 40G\_Master-F 5. 10G\_Slave-F <--> 40G\_Slave-F 6. 10G\_Slave-F <--> 40G\_Master-H 7. 10G\_Slave-F <--> 40G\_Slave-H 8. 10G\_Slave-F <--> 10G\_Master-NWay 9. 10G\_Slave-F <--> 10G\_Slave-NWay 10. 10G\_Slave-F <--> 10G\_Master-F 11. 10G\_Slave-F <--> 10G\_Slave-F 12. 10G\_Slave-F <--> 10G\_Master-H 13. 10G\_Slave-F <--> 10G\_Slave-H 14. 10G\_Slave-F <--> 1000\_Master-NWay 15. 10G\_Slave-F <--> 1000\_Slave-NWay 16. 10G\_Slave-F <--> 1000\_Master-F 17. 10G\_Slave-F <--> 1000\_Slave-F 18. 10G\_Slave-F <--> 1000\_Master-H 19. 10G\_Slave-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.11**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Master-H) |
| Procedure | Configured as listed parameters:   1. 10G\_Master-H <--> NWay(All) 2. 10G\_Master-H <--> 40G\_Master-NWay 3. 10G\_Master-H <--> 40G\_Slave-NWay 4. 10G\_Master-H <--> 40G\_Master-F 5. 10G\_Master-H <--> 40G\_Slave-F 6. 10G\_Master-H <--> 40G\_Master-H 7. 10G\_Master-H <--> 40G\_Slave-H 8. 10G\_Master-H <--> 10G\_Master-NWay 9. 10G\_Master-H <--> 10G\_Slave-NWay 10. 10G\_Master-H <--> 10G\_Master-F 11. 10G\_Master-H <--> 10G\_Slave-F 12. 10G\_Master-H <--> 10G\_Master-H 13. 10G\_Master-H <--> 10G\_Slave-H 14. 10G\_Master-H <--> 1000\_Master-NWay 15. 10G\_Master-H <--> 1000\_Slave-NWay 16. 10G\_Master-H <--> 1000\_Master-F 17. 10G\_Master-H <--> 1000\_Slave-F 18. 10G\_Master-H <--> 1000\_Master-H 19. 10G\_Master-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.12**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Slave-H) |
| Procedure | Configured as listed parameters:   1. 10G\_Slave-H <--> NWay(All) 2. 10G\_Slave-H <--> 40G\_Master-NWay 3. 10G\_Slave-H <--> 40G\_Slave-NWay 4. 10G\_Slave-H <--> 40G\_Master-F 5. 10G\_Slave-H <--> 40G\_Slave-F 6. 10G\_Slave-H <--> 40G\_Master-H 7. 10G\_Slave-H <--> 40G\_Slave-H 8. 10G\_Slave-H <--> 10G\_Master-NWay 9. 10G\_Slave-H <--> 10G\_Slave-NWay 10. 10G\_Slave-H <--> 10G\_Master-F 11. 10G\_Slave-H <--> 10G\_Slave-F 12. 10G\_Slave-H <--> 10G\_Master-H 13. 10G\_Slave-H <--> 10G\_Slave-H 14. 10G\_Slave-H <--> 1000\_Master-NWay 15. 10G\_Slave-H <--> 1000\_Slave-NWay 16. 10G\_Slave-H <--> 1000\_Master-F 17. 10G\_Slave-H <--> 1000\_Slave-F 18. 10G\_Slave-H <--> 1000\_Master-H 19. 10G\_Slave-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.13**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Master-NWay <--> NWay(All) 2. 1000\_Master-NWay <--> 40G\_Master-NWay 3. 1000\_Master-NWay <--> 40G\_Slave-NWay 4. 1000\_Master-NWay <--> 40G\_Master-F 5. 1000\_Master-NWay <--> 40G\_Slave-F 6. 1000\_Master-NWay <--> 40G\_Master-H 7. 1000\_Master-NWay <--> 40G\_Slave-H 8. 1000\_Master-NWay <--> 10G\_Master-NWay 9. 1000\_Master-NWay <--> 10G\_Slave-NWay 10. 1000\_Master-NWay <--> 10G\_Master-F 11. 1000\_Master-NWay <--> 10G\_Slave-F 12. 1000\_Master-NWay <--> 10G\_Master-H 13. 1000\_Master-NWay <--> 10G\_Slave-H 14. 1000\_Master-NWay <--> 1000\_Master-NWay 15. 1000\_Master-NWay <--> 1000\_Slave-NWay 16. 1000\_Master-NWay <--> 1000\_Master-F 17. 1000\_Master-NWay <--> 1000\_Slave-F 18. 1000\_Master-NWay <--> 1000\_Master-H 19. 1000\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.14**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave-NWay <--> NWay(All) 2. 1000\_Slave-NWay <--> 40G\_Master-NWay 3. 1000\_Slave-NWay <--> 40G\_Slave-NWay 4. 1000\_Slave-NWay <--> 40G\_Master-F 5. 1000\_Slave-NWay <--> 40G\_Slave-F 6. 1000\_Slave-NWay <--> 40G\_Master-H 7. 1000\_Slave-NWay <--> 40G\_Slave-H 8. 1000\_Slave-NWay <--> 10G\_Master-NWay 9. 1000\_Slave-NWay <--> 10G\_Slave-NWay 10. 1000\_Slave-NWay <--> 10G\_Master-F 11. 1000\_Slave-NWay <--> 10G\_Slave-F 12. 1000\_Slave-NWay <--> 10G\_Master-H 13. 1000\_Slave-NWay <--> 10G\_Slave-H 14. 1000\_Slave-NWay <--> 1000\_Master-NWay 15. 1000\_Slave-NWay <--> 1000\_Slave-NWay 16. 1000\_Slave-NWay <--> 1000\_Master-F 17. 1000\_Slave-NWay <--> 1000\_Slave-F 18. 1000\_Slave-NWay <--> 1000\_Master-H 19. 1000\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.15**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Master-F <--> NWay(All) 2. 1000\_Master-F <--> 40G\_Master-NWay 3. 1000\_Master-F <--> 40G\_Slave-NWay 4. 1000\_Master-F <--> 40G\_Master-F 5. 1000\_Master-F <--> 40G\_Slave-F 6. 1000\_Master-F <--> 40G\_Master-H 7. 1000\_Master-F <--> 40G\_Slave-H 8. 1000\_Master-F <--> 10G\_Master-NWay 9. 1000\_Master-F <--> 10G\_Slave-NWay 10. 1000\_Master-F <--> 10G\_Master-F 11. 1000\_Master-F <--> 10G\_Slave-F 12. 1000\_Master-F <--> 10G\_Master-H 13. 1000\_Master-F <--> 10G\_Slave-H 14. 1000\_Master-F <--> 1000\_Master-NWay 15. 1000\_Master-F <--> 1000\_Slave-NWay 16. 1000\_Master-F <--> 1000\_Master-F 17. 1000\_Master-F <--> 1000\_Slave-F 18. 1000\_Master-F <--> 1000\_Master-H 19. 1000\_Master-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.16**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave\_F <--> NWay(All) 2. 1000\_Slave\_F <--> 10G\_Master-NWay 3. 1000\_Slave\_F <--> 10G\_Slave-NWay 4. 1000\_Slave\_F <--> 10G\_Master-F 5. 1000\_Slave\_F <--> 10G\_Slave-F 6. 1000\_Slave\_F <--> 10G\_Master-H 7. 1000\_Slave\_F <--> 10G\_Slave-H 8. 1000\_Slave\_F <--> 1000\_Master-NWay 9. 1000\_Slave\_F <--> 1000\_Slave-NWay 10. 1000\_Slave\_F <--> 1000\_Master-F 11. 1000\_Slave\_F <--> 1000\_Slave-F 12. 1000\_Slave\_F <--> 1000\_Master-H 13. 1000\_Slave\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.17**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Master-H <--> NWay(All) 2. 1000\_Master-H <--> 40G\_Master-NWay 3. 1000\_Master-H <--> 40G\_Slave-NWay 4. 1000\_Master-H <--> 40G\_Master-F 5. 1000\_Master-H <--> 40G\_Slave-F 6. 1000\_Master-H <--> 40G\_Master-H 7. 1000\_Master-H <--> 40G\_Slave-H 8. 1000\_Master-H <--> 10G\_Master-NWay 9. 1000\_Master-H <--> 10G\_Slave-NWay 10. 1000\_Master-H <--> 10G\_Master-F 11. 1000\_Master-H <--> 10G\_Slave-F 12. 1000\_Master-H <--> 10G\_Master-H 13. 1000\_Master-H <--> 10G\_Slave-H 14. 1000\_Master-H <--> 1000\_Master-NWay 15. 1000\_Master-H <--> 1000\_Slave-NWay 16. 1000\_Master-H <--> 1000\_Master-F 17. 1000\_Master-H <--> 1000\_Slave-F 18. 1000\_Master-H <--> 1000\_Master-H 19. 1000\_Master-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Fiber 2.16.7.18**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave-H <--> NWay(All) 2. 1000\_Slave-H <--> 40G\_Master-NWay 3. 1000\_Slave-H <--> 40G\_Slave-NWay 4. 1000\_Slave-H <--> 40G\_Master-F 5. 1000\_Slave-H <--> 40G\_Slave-F 6. 1000\_Slave-H <--> 40G\_Master-H 7. 1000\_Slave-H <--> 40G\_Slave-H 8. 1000\_Slave-H <--> 10G\_Master-NWay 9. 1000\_Slave-H <--> 10G\_Slave-NWay 10. 1000\_Slave-H <--> 10G\_Master-F 11. 1000\_Slave-H <--> 10G\_Slave-F 12. 1000\_Slave-H <--> 10G\_Master-H 13. 1000\_Slave-H <--> 10G\_Slave-H 14. 1000\_Slave-H <--> 1000\_Master-NWay 15. 1000\_Slave-H <--> 1000\_Slave-NWay 16. 1000\_Slave-H <--> 1000\_Master-F 17. 1000\_Slave-H <--> 1000\_Slave-F 18. 1000\_Slave-H <--> 1000\_Master-H 19. 1000\_Slave-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.1**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (40G-Master-NWay) |
| Procedure | Configured as listed parameters:   1. 40G\_Master-NWay <--> NWay(All) 2. 40G\_Master-NWay <--> 40G\_Master-NWay 3. 40G\_Master-NWay <--> 40G\_Slave-NWay 4. 40G\_Master-NWay <--> 40G\_Master-F 5. 40G\_Master-NWay <--> 40G\_Slave-F 6. 40G\_Master-NWay <--> 40G\_Master-H 7. 40G\_Master-NWay <--> 40G\_Slave-H 8. 40G\_Master-NWay <--> 10G\_Master-NWay 9. 40G\_Master-NWay <--> 10G\_Slave-NWay 10. 40G\_Master-NWay <--> 10G\_Master-F 11. 40G\_Master-NWay <--> 10G\_Slave-F 12. 40G\_Master-NWay <--> 10G\_Master-H 13. 40G\_Master-NWay <--> 10G\_Slave-H 14. 40G\_Master-NWay <--> 1000\_Master-NWay 15. 40G\_Master-NWay <--> 1000\_Slave-NWay 16. 40G\_Master-NWay <--> 1000\_Master-F 17. 40G\_Master-NWay <--> 1000\_Slave-F 18. 40G\_Master-NWay <--> 1000\_Master-H 19. 40G\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.2**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (40G-Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 40G\_Slave-NWay <--> NWay(All) 2. 40G\_Slave-NWay <--> 40G\_Master-NWay 3. 40G\_Slave-NWay <--> 40G\_Slave-NWay 4. 40G\_Slave-NWay <--> 40G\_Master-F 5. 40G\_Slave-NWay <--> 40G\_Slave-F 6. 40G\_Slave-NWay <--> 40G\_Master-H 7. 40G\_Slave-NWay <--> 40G\_Slave-H 8. 40G\_Slave-NWay <--> 10G\_Master-NWay 9. 40G\_Slave-NWay <--> 10G\_Slave-NWay 10. 40G\_Slave-NWay <--> 10G\_Master-F 11. 40G\_Slave-NWay <--> 10G\_Slave-F 12. 40G\_Slave-NWay <--> 10G\_Master-H 13. 40G\_Slave-NWay <--> 10G\_Slave-H 14. 40G\_Slave-NWay <--> 1000\_Master-NWay 15. 40G\_Slave-NWay <--> 1000\_Slave-NWay 16. 40G\_Slave-NWay <--> 1000\_Master-F 17. 40G\_Slave-NWay <--> 1000\_Slave-F 18. 40G\_Slave-NWay <--> 1000\_Master-H 19. 40G\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.3**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (40G-Master-F) |
| Procedure | Configured as listed parameters:   1. 40G\_Master-F <--> NWay(All) 2. 40G\_Master-F <--> 40G\_Master-NWay 3. 40G\_Master-F <--> 40G\_Slave-NWay 4. 40G\_Master-F <--> 40G\_Master-F 5. 40G\_Master-F <--> 40G\_Slave-F 6. 40G\_Master-F <--> 40G\_Master-H 7. 40G\_Master-F <--> 40G\_Slave-H 8. 40G\_Master-F <--> 10G\_Master-NWay 9. 40G\_Master-F <--> 10G\_Slave-NWay 10. 40G\_Master-F <--> 10G\_Master-F 11. 40G\_Master-F <--> 10G\_Slave-F 12. 40G\_Master-F <--> 10G\_Master-H 13. 40G\_Master-F <--> 10G\_Slave-H 14. 40G\_Master-F <--> 1000\_Master-NWay 15. 40G\_Master-F <--> 1000\_Slave-NWay 16. 40G\_Master-F <--> 1000\_Master-F 17. 40G\_Master-F <--> 1000\_Slave-F 18. 40G\_Master-F <--> 1000\_Master-H 19. 40G\_Master-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.4**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (40G-Slave-F) |
| Procedure | Configured as listed parameters:   1. 40G\_Slave-F <--> NWay(All) 2. 40G\_Slave-F <--> 40G\_Master-NWay 3. 40G\_Slave-F <--> 40G\_Slave-NWay 4. 40G\_Slave-F <--> 40G\_Master-F 5. 40G\_Slave-F <--> 40G\_Slave-F 6. 40G\_Slave-F <--> 40G\_Master-H 7. 40G\_Slave-F <--> 40G\_Slave-H 8. 40G\_Slave-F <--> 10G\_Master-NWay 9. 40G\_Slave-F <--> 10G\_Slave-NWay 10. 40G\_Slave-F <--> 10G\_Master-F 11. 40G\_Slave-F <--> 10G\_Slave-F 12. 40G\_Slave-F <--> 10G\_Master-H 13. 40G\_Slave-F <--> 10G\_Slave-H 14. 40G\_Slave-F <--> 1000\_Master-NWay 15. 40G\_Slave-F <--> 1000\_Slave-NWay 16. 40G\_Slave-F <--> 1000\_Master-F 17. 40G\_Slave-F <--> 1000\_Slave-F 18. 40G\_Slave-F <--> 1000\_Master-H 19. 40G\_Slave-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.5**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (40G-Master-H) |
| Procedure | Configured as listed parameters:   1. 40G\_Master-H <--> NWay(All) 2. 40G\_Master-H <--> 40G\_Master-NWay 3. 40G\_Master-H <--> 40G\_Slave-NWay 4. 40G\_Master-H <--> 40G\_Master-F 5. 40G\_Master-H <--> 40G\_Slave-F 6. 40G\_Master-H <--> 40G\_Master-H 7. 40G\_Master-H <--> 40G\_Slave-H 8. 40G\_Master-H <--> 10G\_Master-NWay 9. 40G\_Master-H <--> 10G\_Slave-NWay 10. 40G\_Master-H <--> 10G\_Master-F 11. 40G\_Master-H <--> 10G\_Slave-F 12. 40G\_Master-H <--> 10G\_Master-H 13. 40G\_Master-H <--> 10G\_Slave-H 14. 40G\_Master-H <--> 1000\_Master-NWay 15. 40G\_Master-H <--> 1000\_Slave-NWay 16. 40G\_Master-H <--> 1000\_Master-F 17. 40G\_Master-H <--> 1000\_Slave-F 18. 40G\_Master-H <--> 1000\_Master-H 19. 40G\_Master-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.6**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (40G-Slave-H) |
| Procedure | Configured as listed parameters:   1. 40G\_Slave-H <--> NWay(All) 2. 40G\_Slave-H <--> 40G\_Master-NWay 3. 40G\_Slave-H <--> 40G\_Slave-NWay 4. 40G\_Slave-H <--> 40G\_Master-F 5. 40G\_Slave-H <--> 40G\_Slave-F 6. 40G\_Slave-H <--> 40G\_Master-H 7. 40G\_Slave-H <--> 40G\_Slave-H 8. 40G\_Slave-H <--> 10G\_Master-NWay 9. 40G\_Slave-H <--> 10G\_Slave-NWay 10. 40G\_Slave-H <--> 10G\_Master-F 11. 40G\_Slave-H <--> 10G\_Slave-F 12. 40G\_Slave-H <--> 10G\_Master-H 13. 40G\_Slave-H <--> 10G\_Slave-H 14. 40G\_Slave-H <--> 1000\_Master-NWay 15. 40G\_Slave-H <--> 1000\_Slave-NWay 16. 40G\_Slave-H <--> 1000\_Master-F 17. 40G\_Slave-H <--> 1000\_Slave-F 18. 40G\_Slave-H <--> 1000\_Master-H 19. 40G\_Slave-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.7**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Master-NWay) |
| Procedure | Configured as listed parameters:   1. 10G\_Master-NWay <--> NWay(All) 2. 10G\_Master-NWay <--> 40G\_Master-NWay 3. 10G\_Master-NWay <--> 40G\_Slave-NWay 4. 10G\_Master-NWay <--> 40G\_Master-F 5. 10G\_Master-NWay <--> 40G\_Slave-F 6. 10G\_Master-NWay <--> 40G\_Master-H 7. 10G\_Master-NWay <--> 40G\_Slave-H 8. 10G\_Master-NWay <--> 10G\_Master-NWay 9. 10G\_Master-NWay <--> 10G\_Slave-NWay 10. 10G\_Master-NWay <--> 10G\_Master-F 11. 10G\_Master-NWay <--> 10G\_Slave-F 12. 10G\_Master-NWay <--> 10G\_Master-H 13. 10G\_Master-NWay <--> 10G\_Slave-H 14. 10G\_Master-NWay <--> 1000\_Master-NWay 15. 10G\_Master-NWay <--> 1000\_Slave-NWay 16. 10G\_Master-NWay <--> 1000\_Master-F 17. 10G\_Master-NWay <--> 1000\_Slave-F 18. 10G\_Master-NWay <--> 1000\_Master-H 19. 10G\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.8**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 10G\_Slave-NWay <--> NWay(All) 2. 10G\_Slave-NWay <--> 40G\_Master-NWay 3. 10G\_Slave-NWay <--> 40G\_Slave-NWay 4. 10G\_Slave-NWay <--> 40G\_Master-F 5. 10G\_Slave-NWay <--> 40G\_Slave-F 6. 10G\_Slave-NWay <--> 40G\_Master-H 7. 10G\_Slave-NWay <--> 40G\_Slave-H 8. 10G\_Slave-NWay <--> 10G\_Master-NWay 9. 10G\_Slave-NWay <--> 10G\_Slave-NWay 10. 10G\_Slave-NWay <--> 10G\_Master-F 11. 10G\_Slave-NWay <--> 10G\_Slave-F 12. 10G\_Slave-NWay <--> 10G\_Master-H 13. 10G\_Slave-NWay <--> 10G\_Slave-H 14. 10G\_Slave-NWay <--> 1000\_Master-NWay 15. 10G\_Slave-NWay <--> 1000\_Slave-NWay 16. 10G\_Slave-NWay <--> 1000\_Master-F 17. 10G\_Slave-NWay <--> 1000\_Slave-F 18. 10G\_Slave-NWay <--> 1000\_Master-H 19. 10G\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.9**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Master-F) |
| Procedure | Configured as listed parameters:   1. 10G\_Master-F <--> NWay(All) 2. 10G\_Master-F <--> 40G\_Master-NWay 3. 10G\_Master-F <--> 40G\_Slave-NWay 4. 10G\_Master-F <--> 40G\_Master-F 5. 10G\_Master-F <--> 40G\_Slave-F 6. 10G\_Master-F <--> 40G\_Master-H 7. 10G\_Master-F <--> 40G\_Slave-H 8. 10G\_Master-F <--> 10G\_Master-NWay 9. 10G\_Master-F <--> 10G\_Slave-NWay 10. 10G\_Master-F <--> 10G\_Master-F 11. 10G\_Master-F <--> 10G\_Slave-F 12. 10G\_Master-F <--> 10G\_Master-H 13. 10G\_Master-F <--> 10G\_Slave-H 14. 10G\_Master-F <--> 1000\_Master-NWay 15. 10G\_Master-F <--> 1000\_Slave-NWay 16. 10G\_Master-F <--> 1000\_Master-F 17. 10G\_Master-F <--> 1000\_Slave-F 18. 10G\_Master-F <--> 1000\_Master-H 19. 10G\_Master-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.10**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Slave-F) |
| Procedure | Configured as listed parameters:   1. 10G\_Slave-F <--> NWay(All) 2. 10G\_Slave-F <--> 40G\_Master-NWay 3. 10G\_Slave-F <--> 40G\_Slave-NWay 4. 10G\_Slave-F <--> 40G\_Master-F 5. 10G\_Slave-F <--> 40G\_Slave-F 6. 10G\_Slave-F <--> 40G\_Master-H 7. 10G\_Slave-F <--> 40G\_Slave-H 8. 10G\_Slave-F <--> 10G\_Master-NWay 9. 10G\_Slave-F <--> 10G\_Slave-NWay 10. 10G\_Slave-F <--> 10G\_Master-F 11. 10G\_Slave-F <--> 10G\_Slave-F 12. 10G\_Slave-F <--> 10G\_Master-H 13. 10G\_Slave-F <--> 10G\_Slave-H 14. 10G\_Slave-F <--> 1000\_Master-NWay 15. 10G\_Slave-F <--> 1000\_Slave-NWay 16. 10G\_Slave-F <--> 1000\_Master-F 17. 10G\_Slave-F <--> 1000\_Slave-F 18. 10G\_Slave-F <--> 1000\_Master-H 19. 10G\_Slave-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.11**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Master-H) |
| Procedure | Configured as listed parameters:   1. 10G\_Master-H <--> NWay(All) 2. 10G\_Master-H <--> 40G\_Master-NWay 3. 10G\_Master-H <--> 40G\_Slave-NWay 4. 10G\_Master-H <--> 40G\_Master-F 5. 10G\_Master-H <--> 40G\_Slave-F 6. 10G\_Master-H <--> 40G\_Master-H 7. 10G\_Master-H <--> 40G\_Slave-H 8. 10G\_Master-H <--> 10G\_Master-NWay 9. 10G\_Master-H <--> 10G\_Slave-NWay 10. 10G\_Master-H <--> 10G\_Master-F 11. 10G\_Master-H <--> 10G\_Slave-F 12. 10G\_Master-H <--> 10G\_Master-H 13. 10G\_Master-H <--> 10G\_Slave-H 14. 10G\_Master-H <--> 1000\_Master-NWay 15. 10G\_Master-H <--> 1000\_Slave-NWay 16. 10G\_Master-H <--> 1000\_Master-F 17. 10G\_Master-H <--> 1000\_Slave-F 18. 10G\_Master-H <--> 1000\_Master-H 19. 10G\_Master-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.12**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (10G\_Slave-H) |
| Procedure | Configured as listed parameters:   1. 10G\_Slave-H <--> NWay(All) 2. 10G\_Slave-H <--> 40G\_Master-NWay 3. 10G\_Slave-H <--> 40G\_Slave-NWay 4. 10G\_Slave-H <--> 40G\_Master-F 5. 10G\_Slave-H <--> 40G\_Slave-F 6. 10G\_Slave-H <--> 40G\_Master-H 7. 10G\_Slave-H <--> 40G\_Slave-H 8. 10G\_Slave-H <--> 10G\_Master-NWay 9. 10G\_Slave-H <--> 10G\_Slave-NWay 10. 10G\_Slave-H <--> 10G\_Master-F 11. 10G\_Slave-H <--> 10G\_Slave-F 12. 10G\_Slave-H <--> 10G\_Master-H 13. 10G\_Slave-H <--> 10G\_Slave-H 14. 10G\_Slave-H <--> 1000\_Master-NWay 15. 10G\_Slave-H <--> 1000\_Slave-NWay 16. 10G\_Slave-H <--> 1000\_Master-F 17. 10G\_Slave-H <--> 1000\_Slave-F 18. 10G\_Slave-H <--> 1000\_Master-H 19. 10G\_Slave-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.13**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Master-NWay <--> NWay(All) 2. 1000\_Master-NWay <--> 40G\_Master-NWay 3. 1000\_Master-NWay <--> 40G\_Slave-NWay 4. 1000\_Master-NWay <--> 40G\_Master-F 5. 1000\_Master-NWay <--> 40G\_Slave-F 6. 1000\_Master-NWay <--> 40G\_Master-H 7. 1000\_Master-NWay <--> 40G\_Slave-H 8. 1000\_Master-NWay <--> 10G\_Master-NWay 9. 1000\_Master-NWay <--> 10G\_Slave-NWay 10. 1000\_Master-NWay <--> 10G\_Master-F 11. 1000\_Master-NWay <--> 10G\_Slave-F 12. 1000\_Master-NWay <--> 10G\_Master-H 13. 1000\_Master-NWay <--> 10G\_Slave-H 14. 1000\_Master-NWay <--> 1000\_Master-NWay 15. 1000\_Master-NWay <--> 1000\_Slave-NWay 16. 1000\_Master-NWay <--> 1000\_Master-F 17. 1000\_Master-NWay <--> 1000\_Slave-F 18. 1000\_Master-NWay <--> 1000\_Master-H 19. 1000\_Master-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.14**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-NWay) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave-NWay <--> NWay(All) 2. 1000\_Slave-NWay <--> 40G\_Master-NWay 3. 1000\_Slave-NWay <--> 40G\_Slave-NWay 4. 1000\_Slave-NWay <--> 40G\_Master-F 5. 1000\_Slave-NWay <--> 40G\_Slave-F 6. 1000\_Slave-NWay <--> 40G\_Master-H 7. 1000\_Slave-NWay <--> 40G\_Slave-H 8. 1000\_Slave-NWay <--> 10G\_Master-NWay 9. 1000\_Slave-NWay <--> 10G\_Slave-NWay 10. 1000\_Slave-NWay <--> 10G\_Master-F 11. 1000\_Slave-NWay <--> 10G\_Slave-F 12. 1000\_Slave-NWay <--> 10G\_Master-H 13. 1000\_Slave-NWay <--> 10G\_Slave-H 14. 1000\_Slave-NWay <--> 1000\_Master-NWay 15. 1000\_Slave-NWay <--> 1000\_Slave-NWay 16. 1000\_Slave-NWay <--> 1000\_Master-F 17. 1000\_Slave-NWay <--> 1000\_Slave-F 18. 1000\_Slave-NWay <--> 1000\_Master-H 19. 1000\_Slave-NWay <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.15**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Master-F <--> NWay(All) 2. 1000\_Master-F <--> 40G\_Master-NWay 3. 1000\_Master-F <--> 40G\_Slave-NWay 4. 1000\_Master-F <--> 40G\_Master-F 5. 1000\_Master-F <--> 40G\_Slave-F 6. 1000\_Master-F <--> 40G\_Master-H 7. 1000\_Master-F <--> 40G\_Slave-H 8. 1000\_Master-F <--> 10G\_Master-NWay 9. 1000\_Master-F <--> 10G\_Slave-NWay 10. 1000\_Master-F <--> 10G\_Master-F 11. 1000\_Master-F <--> 10G\_Slave-F 12. 1000\_Master-F <--> 10G\_Master-H 13. 1000\_Master-F <--> 10G\_Slave-H 14. 1000\_Master-F <--> 1000\_Master-NWay 15. 1000\_Master-F <--> 1000\_Slave-NWay 16. 1000\_Master-F <--> 1000\_Master-F 17. 1000\_Master-F <--> 1000\_Slave-F 18. 1000\_Master-F <--> 1000\_Master-H 19. 1000\_Master-F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.16**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-F) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave\_F <--> NWay(All) 2. 1000\_Slave\_F <--> 10G\_Master-NWay 3. 1000\_Slave\_F <--> 10G\_Slave-NWay 4. 1000\_Slave\_F <--> 10G\_Master-F 5. 1000\_Slave\_F <--> 10G\_Slave-F 6. 1000\_Slave\_F <--> 10G\_Master-H 7. 1000\_Slave\_F <--> 10G\_Slave-H 8. 1000\_Slave\_F <--> 1000\_Master-NWay 9. 1000\_Slave\_F <--> 1000\_Slave-NWay 10. 1000\_Slave\_F <--> 1000\_Master-F 11. 1000\_Slave\_F <--> 1000\_Slave-F 12. 1000\_Slave\_F <--> 1000\_Master-H 13. 1000\_Slave\_F <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.17**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Master-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Master-H <--> NWay(All) 2. 1000\_Master-H <--> 40G\_Master-NWay 3. 1000\_Master-H <--> 40G\_Slave-NWay 4. 1000\_Master-H <--> 40G\_Master-F 5. 1000\_Master-H <--> 40G\_Slave-F 6. 1000\_Master-H <--> 40G\_Master-H 7. 1000\_Master-H <--> 40G\_Slave-H 8. 1000\_Master-H <--> 10G\_Master-NWay 9. 1000\_Master-H <--> 10G\_Slave-NWay 10. 1000\_Master-H <--> 10G\_Master-F 11. 1000\_Master-H <--> 10G\_Slave-F 12. 1000\_Master-H <--> 10G\_Master-H 13. 1000\_Master-H <--> 10G\_Slave-H 14. 1000\_Master-H <--> 1000\_Master-NWay 15. 1000\_Master-H <--> 1000\_Slave-NWay 16. 1000\_Master-H <--> 1000\_Master-F 17. 1000\_Master-H <--> 1000\_Slave-F 18. 1000\_Master-H <--> 1000\_Master-H 19. 1000\_Master-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

**TC\_ID: 40G\_Fiber\_Cat 2.16.8.18**

|  |  |
| --- | --- |
| Purpose | Verify that the listed parameters can be configured and take effect normally (1000\_Slave-H) |
| Procedure | Configured as listed parameters:   1. 1000\_Slave-H <--> NWay(All) 2. 1000\_Slave-H <--> 40G\_Master-NWay 3. 1000\_Slave-H <--> 40G\_Slave-NWay 4. 1000\_Slave-H <--> 40G\_Master-F 5. 1000\_Slave-H <--> 40G\_Slave-F 6. 1000\_Slave-H <--> 40G\_Master-H 7. 1000\_Slave-H <--> 40G\_Slave-H 8. 1000\_Slave-H <--> 10G\_Master-NWay 9. 1000\_Slave-H <--> 10G\_Slave-NWay 10. 1000\_Slave-H <--> 10G\_Master-F 11. 1000\_Slave-H <--> 10G\_Slave-F 12. 1000\_Slave-H <--> 10G\_Master-H 13. 1000\_Slave-H <--> 10G\_Slave-H 14. 1000\_Slave-H <--> 1000\_Master-NWay 15. 1000\_Slave-H <--> 1000\_Slave-NWay 16. 1000\_Slave-H <--> 1000\_Master-F 17. 1000\_Slave-H <--> 1000\_Slave-F 18. 1000\_Slave-H <--> 1000\_Master-H 19. 1000\_Slave-H <--> 1000\_Slave-H |
| Expected Result | 1. The parameters can be configured and take effect normally |

* 1. **Platform – Power**

**TC\_ID: 2.17.1**

|  |  |
| --- | --- |
| Purpose | Verify that DUT can boot up successfully and display correct info when insert AC power with AIR IN (AFI) label. |
| Procedure | 1. Insert AC power with AIR IN (AFI) label to DUT. |
| Expected Result | 1. DUT should boot up successfully and display correct info |

**TC\_ID: 2.17.2**

|  |  |
| --- | --- |
| Purpose | Verify that DUT can boot up successfully and display correct info when insert AC power with AIR OUT (AFO) label. |
| Procedure | 1. Insert AC power with AIR OUT (AFO) label to DUT. |
| Expected Result | 1. DUT should boot up successfully and display correct info |

**TC\_ID: 2.17.3**

|  |  |
| --- | --- |
| Purpose | Verify that system can record correct log for AC power. |
| Procedure | 1. Make Log for AC power then check. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.17.4**

|  |  |
| --- | --- |
| Purpose | Verify that system can send correct trap for AC power. |
| Procedure | 1. Make Trap for AC power then check. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.17.5**

|  |  |
| --- | --- |
| Purpose | Verify that system can send correct alarm for AC power. |
| Procedure | 1. Make Alarm for AC power then check. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.17.6**

|  |  |
| --- | --- |
| Purpose | Verify that DUT can boot up success and display correct info when insert DC power with AIR IN (AFI) label. |
| Procedure | 1. Insert DC power with AIR IN (AFI) label to DUT. |
| Expected Result | 1. DUT should boot up successfully and display correct info |

**TC\_ID: 2.17.7**

|  |  |
| --- | --- |
| Purpose | Verify that DUT can boot up successfully and display correct info when insert DC power with AIR OUT (AFO) label. |
| Procedure | 1. Insert DC power with AIR OUT (AFO) label to DUT. |
| Expected Result | 1. DUT should boot up successfully and display correct info |

**TC\_ID: 2.17.8**

|  |  |
| --- | --- |
| Purpose | Verify that system can record correct log for DC power. |
| Procedure | 1. Make Log for DC power then check. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.17.9**

|  |  |
| --- | --- |
| Purpose | Verify that system can send correct trap for DC power. |
| Procedure | 1. Make trap for DC power then check. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.17.10**

|  |  |
| --- | --- |
| Purpose | Verify that system can send correct alarm for DC power. |
| Procedure | 1. Make alarm for DC power then check. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.17.11**

|  |  |
| --- | --- |
| Purpose | Verify that AC and DC power supplies cannot mix in the same chassis. |
| Procedure | 1. Insert AC power to DUT. 2. Insert DC power to DUT. 3. Boot up DUT. |
| Expected Result | 1. AC and DC power supplies cannot mix in the same chassis. |

**TC\_ID: 2.17.12**

|  |  |
| --- | --- |
| Purpose | Verify that power supplies with different airflow labels (AIR IN (AFI) and AIR OUT (AFO)) can not mix in the same chassis. |
| Procedure | 1. Insert AIR IN (AFI) power to DUT. 2. Insert AIR OUT (AFO) power to DUT. 3. Boot up DUT. |
| Expected Result | 1. Power supplies with different airflow labels (AIR IN (AFI) and AIR OUT (AFO)) can not mix in the same chassis. |

**TC\_ID: 2.17.13**

|  |  |
| --- | --- |
| Purpose | Verify that system can boot up successfully with maximum number of power inserted. |
| Procedure | 1. Insert maximum number of power into DUT. 2. Boot up DUT. |
| Expected Result | 1. System should boot up successfully and work normally. |

* 1. **Platform – LCD Panel**

**TC\_ID: 2.18.1**

|  |  |
| --- | --- |
| Purpose | Verify that system displays LED/ALARM correctly on the page. |
| Procedure | 1. Compare LED/ALARM page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.2**

|  |  |
| --- | --- |
| Purpose | Verify that system displays STATUS MENU correctly on the page and can enter successfully. |
| Procedure | 1. Compare STATUS MENU page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.3**

|  |  |
| --- | --- |
| Purpose | Verify that system displays STATUS MENU\_VCP/0/1 correctly on the page. |
| Procedure | 1. Compare STATUS MENU\_VCP/0/1 page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.4**

|  |  |
| --- | --- |
| Purpose | Verify that system displays STATUS MENU\_VCP/2/3 correctly on the page. |
| Procedure | 1. Compare STATUS MENU\_VCP/2/3 page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.5**

|  |  |
| --- | --- |
| Purpose | Verify that system displays STATUS MENU\_PWR0/1 ABS correctly on the page. |
| Procedure | 1. Compare STATUS MENU\_PWR0/1 ABS page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.6**

|  |  |
| --- | --- |
| Purpose | Verify that system displays STATUS MENU\_FANS/TEMP correctly on the page. |
| Procedure | 1. Compare STATUS MENU\_FANS/TEMP page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.7**

|  |  |
| --- | --- |
| Purpose | Verify that system displays STATUS MENU\_JUNOS VERSION correctly on the page. |
| Procedure | 1. Compare STATUS MENU\_JUNOS VERSION page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.8**

|  |  |
| --- | --- |
| Purpose | Verify that system displays STATUS MENU\_EXI STAT NEMU correctly on the page and can exist successfully on the page. |
| Procedure | 1. Compare STATUS MENU\_EXI STAT NEMU page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.9**

|  |  |
| --- | --- |
| Purpose | Verify that system displays MAINTENANCE MENU correctly on the page and can enter successfully on the page. |
| Procedure | 1. Compare MAINTENANCE MENU page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.10**

|  |  |
| --- | --- |
| Purpose | Verify that system can "SYSTEM HALT" successfully on the page. |
| Procedure | 1. Compare MAINTENANCE MENU\_SYSTEM HALT page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.11**

|  |  |
| --- | --- |
| Purpose | Verify that system can "SYSTEM REBOOT" successfully on the page. |
| Procedure | 1. Compare MAINTENANCE MENU\_SYSTEM REBOOT page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.12**

|  |  |
| --- | --- |
| Purpose | Verify that system can "LOAD RESCUE" successfully on the page. |
| Procedure | 1. Compare MAINTENANCE MENU\_LOAD RESCUE page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.13**

|  |  |
| --- | --- |
| Purpose | Verify that system can "REQUEST VC PORT" successfully on the page. |
| Procedure | 1. Compare MAINTENANCE MENU\_REQUEST VC PORT page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.14**

|  |  |
| --- | --- |
| Purpose | Verify that system can "FACTORY DEFAUCLT" successfully on the page. |
| Procedure | 1. Compare MAINTENANCE MENU\_FACTORY DEFAUCLT page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.15**

|  |  |
| --- | --- |
| Purpose | Verify that system can "FACTORY DEFAUCLT" successfully on the page. |
| Procedure | 1. Compare MAINTENANCE MENU\_FACTORY DEFAUCLT page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

**TC\_ID: 2.18.16**

|  |  |
| --- | --- |
| Purpose | Verify that system displays MAINTENANCE MENU\_EXI MAIN NEMU correctly on the page and can exist successfully on the page. |
| Procedure | 1. Compare MAINTENANCE MENU\_EXI MAIN NEMU page display with spec. |
| Expected Result | 1. Should implement same as spec design. |

* 1. **Maximum Load Test**

Test Equip.: IXIA

Duration: **30 minutes**

Objective: To determine if the device get stress traffic after 30 minutes, it is still workable.

Transmit/Receive from all port at the fastest speed and the following IXIA setting.

DA ff-ff-ff-ff-ff-ff SA 00-19-62-05-19-0\* (Port \*)

Number of Packets: Continuous

Length [bytes]: Random

Background: Random

Protocol (VFD3): Custom [08 00]

Flow Control: ON and OFF

Pass/fail criteria: No any errors or hung up after 30 minutes

|  |  |  |
| --- | --- | --- |
| **Item** | **Test Description** | **Result** |
| 3.1 | Maximum Load Test |  |

* 1. **Stress Test**

Test Equip.: IXIA

Duration: **24 hours**

Objective: To determine if the device can be handle traffic as a ring.

One ring: configure port1->port2->port3->…->port24->port 25-> port 26->port1

a. same speed, b. two speeds

Two ring: configure port1->port2->port3->…->port24->port1; Port 25-> port 26

Mode: Single Burst 7000,000,000 packets for 1G Full duplex, Single Burst 700,000,000 packets for 100M Full duplex

- Length: Random (64~1518 Bytes), jumbo frame

- Background: AAAA

- IPG: 100% rate

- Flow control enable/disable

Pass/Fail criteria:

* + 1. Only allow 1 packet lost per 10,000,000 packets.
    2. CRC error must be less than or equal to: For 10M speed, per 100,000,000 allow loss one packet.

For 100M speed, per 10,000,000,000 allow loss one packet. For 1000M speed, per 1,000,000,000,000 allow loss one packet.

|  |  |  |
| --- | --- | --- |
| **Item** | **Description** | **Result** |
| 3.2.1.1 | One ring same speed; flow control enable |  |
| 3.2.1.2 | One ring same speed; flow control disable |  |
| 3.2.1.3 | One ring two speed; flow control enable |  |
| 3.2.1.4 | One ring two speed; flow control disable |  |
| 3.2.2.1 | Two ring two speed; flow control enable |  |
| 3.2.2.2 | Two ring two speed; flow control disable |  |

* 1. **Parameter Test**(See Appendix A for the topology set up.)
     1. **Full loading test (Ether port) - Burst 10 minutes for each tested item**

**TC\_ID: 3.3.1.1**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior( On ether port )   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packet length = 64 bytes 7. Transmission Mode = Single Burst 8. Packet Count = 900,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.2**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior( On ether port )   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packets length = 1518 bytes 7. Transmission Mode = Single Burst 8. Packet Count = 50,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.3**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send multi burst packets successfully without losing packets. |
| Procedure | Basic Behavior( On ether port )   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packets length = 1518 bytes 7. Transmission Mode = Multi Burst 8. Multi Burst Count = 24 9. Packet per Burst = 2,080,000 (total = 2,080,000\*24 = 49,920,000)   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.4**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send multi burst packets successfully without losing packets. |
| Procedure | Basic Behavior( On ether port )   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packets length = 64 bytes 7. Transmission Mode = Multi Burst 8. Multi Burst Count = 432 9. Packet per Burst = 2,080,000 (total = 2,080,000\*432 = 898,560,000)   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.5**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all ones. 7. Packets length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.6**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all ones. 7. Packets length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 50,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.7**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all zeros. 7. Packets length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.8**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all zeros. 7. Packets length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 50,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.9**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Alvin packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packets length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.10**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Alvin packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packets length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 50,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.11**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Killer packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packets length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.12**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Killer packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packets length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 50,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.13**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Steven packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packets length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.14**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Steven packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packets length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 50,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.15**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Random packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set random. 7. Packets length = random 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.16**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Random packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set random. 7. Packets length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.1.17**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Random packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set random. 7. Packets length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 50,000,000   [IXIA#1(100-Full) and IXIA#2(100-Full) send unicast packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

* + 1. **Full loading test (Fiber port) - Burst 10 minutes for each tested item**

**TC\_ID: 3.3.2.1**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior(On Giga port )   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packet length = 64 bytes 7. Transmission Mode = Single Burst 8. Packet Count = 900,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.2**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior(On Giga Port )   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packet length = 1518 bytes 7. Transmission Mode = Single Burst 8. Packet Count = 50,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.3**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior(On Giga Port )   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packet length = 1518 bytes 7. Transmission Mode = Multi Burst 8. Multi Burst Count = 24 9. Packet per Burst = 2,080,000 (total = 2,080,000\*24 = 49,920,000)   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.4**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior(On Giga Port )   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packet length = 64 bytes 7. Transmission Mode = Multi Burst 8. Multi Burst Count = 432 9. Packet per Burst = 2,080,000 (total = 2,080,000\*432 = 898,560,000)   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.5**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all ones. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.6**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all ones. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 50,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.7**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all zeros. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.8**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all zeros. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 50,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.9**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Alvin packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.10**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Alvin packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 50,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.11**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Killer packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.12**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Killer packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 50,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.13**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Steven packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.14**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Steven packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 50,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.15**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Random packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set Random. 7. Packet length = Random 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.16**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Random packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set Random. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 900,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.2.17**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Random packets behavior :   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set Random. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 50,000,000   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

* + 1. **Full loading test (10G) - Burst 10 minutes for each tested item**

**TC\_ID: 3.3.3.1**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packet length = 64 bytes 7. Transmission Mode = Single Burst 8. Packet Count = 9,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.2**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packet length = 1518 bytes 7. Transmission Mode = Single Burst 8. Packet Count = 500,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.3**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packet length = 1518 bytes 7. Transmission Mode = Multi Burst 8. Multi Burst Count = 24 9. Packet per Burst = 20,800,000 (total = 20,800,000\*24 = 499,200,000)   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.4**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packet length = 64 bytes 7. Transmission Mode = Multi Burst 8. Multi Burst Count = 432 9. Packet per Burst = 20,800,000 (total = 20,800,000\*432 = 8,985,600,000)   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.5**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all ones. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 9,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.6**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all ones. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 500,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.7**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all ones. 7. Packet length = Random 8. Transmission Mode = Single Burst 9. Packet Count = 1,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.8**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all zeros. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 9,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.9**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all zeros. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 500,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.10**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all zeros. 7. Packet length = Random 8. Transmission Mode = Single Burst 9. Packet Count = 1,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.11**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Alvin packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 9,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.12**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Alvin packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 500,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.13**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Alvin packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = Random 8. Transmission Mode = Single Burst 9. Packet Count = 1,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.14**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Killer packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 9,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.15**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Killer packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 500,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.16**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Killer packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = Random 8. Transmission Mode = Single Burst 9. Packet Count = 1,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.17**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Steven packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 9,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.18**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Steven packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 500,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.19**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Steven packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = Random 8. Transmission Mode = Single Burst 9. Packet Count = 1,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.20**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Random packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set random. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 9,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.21**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Random packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set random. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 500,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.3.22**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Random packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set random. 7. Packet length = Random bytes 8. Transmission Mode = Single Burst 9. Packet Count = 1,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

* + 1. **Full loading test (40G) - Burst 10 minutes for each tested item**

**TC\_ID: 3.3.4.1**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packet length = 64 bytes 7. Transmission Mode = Single Burst 8. Packet Count = 36,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.2**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packet length = 1518 bytes 7. Transmission Mode = Single Burst 8. Packet Count = 2,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.3**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packet length = 1518 bytes 7. Transmission Mode = Multi Burst 8. Multi Burst Count = 24 9. Packet per Burst = 83,200,000 (total = 83,200,000\*24 = 1,996,800,000)   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.4**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Basic Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Packet length = 64 bytes 7. Transmission Mode = Multi Burst 8. Multi Burst Count = 432 9. Packet per Burst = 83,200,000 (total = 83,200,000\*432 = 35,942,400,000)   [IXIA#1(1 Giga) and IXIA#2(1 Giga) use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.5**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all ones. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 36,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.6**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all ones. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 2,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.7**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all ones. 7. Packet length = Random 8. Transmission Mode = Single Burst 9. Packet Count = 4,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.8**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all zeros. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 36,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.9**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all zeros. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 2,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.10**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Specific packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set all zeros. 7. Packet length = Random 8. Transmission Mode = Single Burst 9. Packet Count = 4,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.11**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Alvin packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 36,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.12**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Alvin packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 2,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.13**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Alvin packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = Random 8. Transmission Mode = Single Burst 9. Packet Count = 4,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.14**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Killer packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 36,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.15**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Killer packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 2,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.16**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Killer packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = Random 8. Transmission Mode = Single Burst 9. Packet Count = 4,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.17**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Steven packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 36,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.18**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Steven packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 2,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.19**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Steven packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set custom. 7. Packet length = Random 8. Transmission Mode = Single Burst 9. Packet Count = 4,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.20**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Random packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set random. 7. Packet length = 64 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 36,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.21**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Random packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set random. 7. Packet length = 1518 bytes 8. Transmission Mode = Single Burst 9. Packet Count = 2,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**TC\_ID: 3.3.4.22**

|  |  |
| --- | --- |
| Purpose | Verify that the device can send packets successfully without losing packets. |
| Procedure | Random packets Behavior   1. Connect IXIA#1 to port(1), IXIA#2 to port(n). 2. Connect DUT port2 to port3, port4 to port5 …… and port(n-2) to port(n-1) 3. SA:00:00:00:11:22:33/DA:00:00:00:33:22:11 at wire speed from IXIA#1 4. SA:00:00:00:33:22:11/DA:00:00:00:11:22:33 at wire speed from IXIA#2 5. Rate Control set at 100% 6. Background set random. 7. Packet length = Random bytes 8. Transmission Mode = Single Burst 9. Packet Count = 4,000,000,000   [IXIA#1 and IXIA#2 use unicast to send packet.] |
| Expected Result | 1. DUT should forward packets without packets lost |

**Appendix A**

Topology: Snake One ring

DUT-----IXIA

Ixia -> port 1, port 2 -> port 3,port 4 -> port 5,port 6 -> port 7,………………….,port 46 -> port 47,port 48-> Ixia

port 1 & 2 same vlan, port 3 & 4 same vlan, port 5 & 6 same vlan,……………………..

