N9500-64OC Smoke Test Report

AI Datacenter 64*800G SONiC Switch

NADDOD Pte.Ltd.

Contents

一、Overall Test Conclusion

The N9500-64OC device demonstrated excellent performance across all functional tests within the SONiC system environment. All test items were successfully completed without any major defects identified that could affect normal operation. The device fully meets design specifications in critical domains such as L2/L3 network functionalities, QoS mechanisms, and system management, exhibiting robust stability, reliability, and performance metrics that align with industry standards and practical application requirements.

二、Functional Test Conclusion

2.1 L2 Functional Tests

Link Aggregation (802.3ad)：The device supports multi-link aggregation, enabling seamless traffic failover upon link disconnection. The load balancing mechanism of aggregated ports operates normally without packet loss, verifying the redundancy and bandwidth expansion capabilities.

VLAN Functionality： It accurately processes VLAN tagging as configured, forwards untagged/tagged packets correctly, and ensures error-free cross-port VLAN communication, compliant with IEEE 802.1q standards.

MAC Address Table Specifications：The MAC address table can learn up to 8,191 entries, meeting the specified requirements. The address learning and aging mechanisms function properly.

Spanning Tree Protocol (STP)：The device achieves rapid convergence to eliminate network loops, with port state transitions complying with standards to ensure no loop risks.

2.2 L3 Functional Tests

ARP Protocol：It correctly handles ARP requests/responses, supports invalid MAC address filtering, and the ARP table can accommodate up to 16,383 entries, meeting large-scale network

requirements.

BGP Routing： It stably establishes connections with external neighbors, learns and forwards routes, and supports IPv4/IPv6 routing table specification tests (capable of learning 500,000 and 610,000 routes respectively). The routing convergence speed meets expectations.

ECMP Load Balancing： Multipath forwarding functions normally, distributing traffic proportionally across links without significant packet loss or skew.

IPv6 Support： It fully supports IPv6 address configuration, neighbor discovery, and routing forwarding, with good compatibility with the IPv4 protocol stack. Bidirectional communication remains error-free in tests.

2.3 QoS Mechanism Tests

Traffic Classification & Marking： Classification and marking based on COS/DSCP are accurate, mapping packets to specified queues. The egress marking modification function meets design requirements.

Priority Flow Control (PFC)： Supports symmetric/asymmetric PFC to pause traffic by priority. Watchdog and Watermark mechanisms prevent congestion effectively, ensuring lossless transmission.

Queue Scheduling & Shaping： The WRR scheduling algorithm allocates bandwidth by weight, and port/queue shaping precisely limits traffic rates, compliant with QoS service level agreements (SLAs).

Congestion Management (WRED/ECN)： Dynamically drops packets based on queue depth. The ECN marking function correctly feedbacks to senders during congestion, optimizing network transmission efficiency.

2.4 System Management Tests

SNMP & Traps： Supports SNMPv2c for remote device status acquisition. Trap alarms are real-time, accurately reporting events like port state changes.

Logging & Remote Management： Rsyslog functions normally for real-time log transmission to remote servers. SSH/SCP/SFTP services are stable, enabling secure remote login and file transfer.

Mirroring： Copies traffic from specified ports to monitoring ports, supporting multi-source port mirroring with complete data capture for network monitoring scenarios.

Dynamic Port Breakout (DPB)： Supports breaking 800G ports into 2x200G/4x200G modes. Post-breakout, port rates and forwarding functions remain normal, meeting flexible networking needs.

2.5 High Availability Tests

COPP Protection： Defends against CPU-directed malicious traffic, maintaining CPU utilization <20% under high load to ensure stable system operation.

Zero Touch Provisioning (ZTP)： Supports automatic configuration acquisition via DHCP+TFTP, enabling rapid system initialization and configuration loading after reboot to reduce deployment complexity.

2.6 Performance Metrics Conclusion

RFC-2544 Testing： The device's metrics including throughput, latency, and packet loss rate comply with requirements. It achieves line-rate forwarding without packet loss, demonstrating excellent performance.

Large-Scale Specification Testing： Both IPv4/IPv6 routing tables and ARP/MAC address entries reach the design upper limits, with no significant decrease in query and forwarding efficiency.

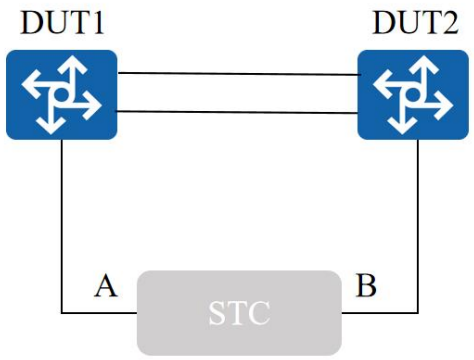
2.7 Testing Environment

The switch testing environment consists of the Device Under Test (DUT), testing instruments, servers, routers, and configuration terminals. Most test items are conducted between the DUT and instruments. Servers are used to verify the DUT's support for functions such as RADIUS and

SYSLOG. For specific test items and environments, refer to the detailed descriptions in each test section.

三、L2

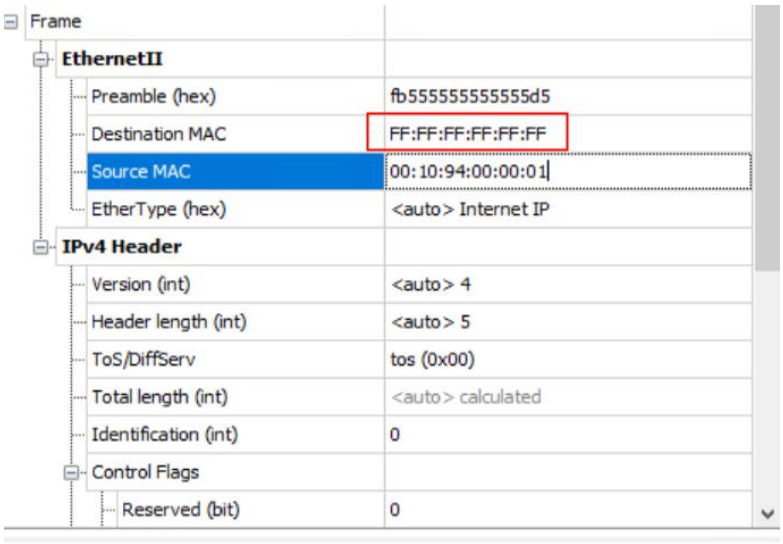
3.1 Link Aggregation Function Test

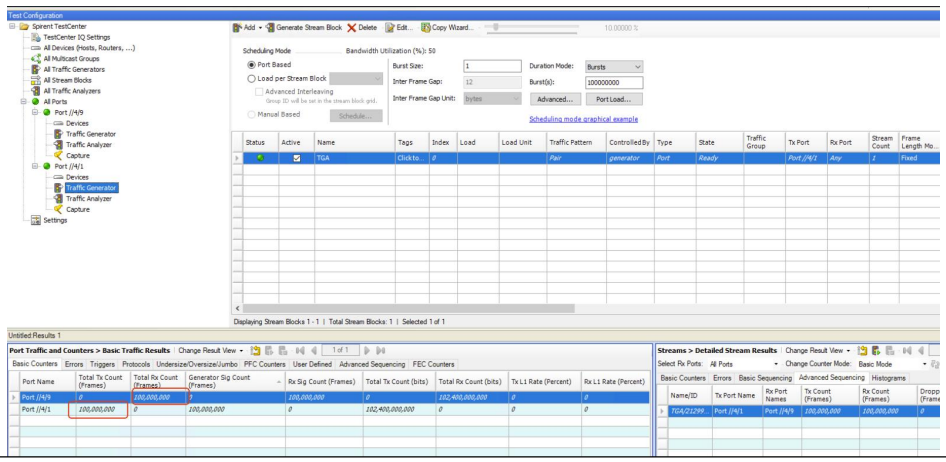
Test Items	Link Aggregation Function Test
Test Content	Verify the Accuracy of DUT Link Aggregation Function
Test Topology	
Test Steps	1、Connect devices as per the test environment. 2、Set the ports connected between DUT1 and DUT2 to be aggregated into one group. 3、The test instrument's Port A sends a broadcast (with MAC address all "F") to Port B. 4、Disconnect one of the links. 5、The test instrument's Port A sends N unicast flows to Port B (where N is greater than the number of aggregated links), and check whether Port B can receive the packets sent by Port A ,Check1. 6、Disconnect one of the links. 7、The test instrument's Port A sends two unicast data flows with the same priority to Port B. The traffic of Data Flow 1 exceeds the line rate of the aggregated port, and the traffic of Data Flow 2 is lower than the line rate of the aggregated port. Observe whether Port B can receive the two data flows sent by Port A, Check2.
Expected Results	Check1: The test instrument's Port B should be able to receive the packets sent from Port A. Check2: The test instrument's Port B should be able to receive the two data flows sent from Port A.
Actual Test Results	config portchannel add PortChannel12 config portchannel member add PortChannel12 Ethernet4 config portchannel member add PortChannel12 Ethernet8 config vlan add 100 config vlan member add 100 PortChannel12 -u config vlan member add 100 Ethernet16 -u

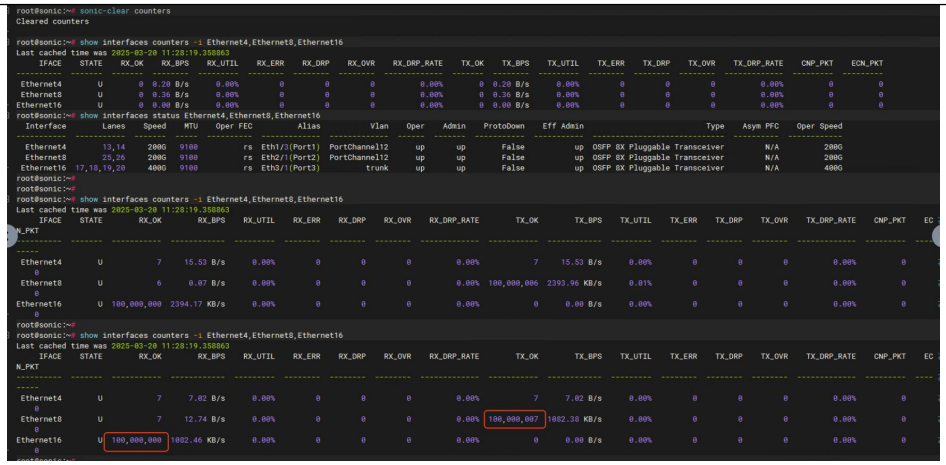
```
root@sonic:~# config portchannel add PortChannel12
root@sonic:~#
root@sonic:~# config portchannel member add PortChannel12 Ethernet4
root@sonic:~# config portchannel member add PortChannel12 Ethernet8
root@sonic:~#
root@sonic:~# show interfaces portchannel
Flags: A - active, I - inactive, Up - up, Dw - Down, N/A - not available,
       S - selected, D - deselected, * - not synced,
       M - mixed speed
  No.  Team Dev       Protocol     Ports                    Oper Key  Admin Key   Fast Rate   System ID   System Priority   Dev Number   Max Ports
-----  -------------  -----------  -----------------------  --------  ----------  ----------  ----------  ----------------  -----------  ----------
   12  PortChannel12  LACP(A)(Up)  Ethernet8(S) Ethernet4(S)      112  auto        false                                            1          64
root@sonic:~#
root@sonic:~# show vlan brief
+-----------+--------------+-----------+---------+---------+----------------+-----------------------------+
| VLAN ID   | IP Address   | Ports     | Port    | Proxy   | DHCP Helper    | DHCP Relay Configuration    |
|           |              |           | Tagging | ARP     | Address        |                             |
+===========+==============+===========+=========+=========+================+=============================+
|         2 | 192.168.1.1/24 | Ethernet384 | untagged | disabled |              | Source Interface:           |
|           |              |           |         |         |                | Link Selection:             |
|           |              |           |         |         |                | Server Vrf:                 |
|           |              |           |         |         |                | Server ID Override:         |
+-----------+--------------+-----------+---------+---------+----------------+-----------------------------+
|         3 | 192.168.2.1/24 | Ethernet386 | untagged | disabled |              | Source Interface:           |
|           |              |           |         |         |                | Link Selection:             |
|           |              |           |         |         |                | Server Vrf:                 |
|           |              |           |         |         |                | Server ID Override:         |
+-----------+--------------+-----------+---------+---------+----------------+-----------------------------+
root@sonic:~# config vlan add 100
root@sonic:~# config vlan member add 100 PortChannel12 -u
root@sonic:~# config vlan member add 100 Ethernet16 -u
root@sonic:~#
root@sonic:~# show vlan brief
+-----------+--------------+-----------+---------+---------+----------------+-----------------------------+
| VLAN ID   | IP Address   | Ports     | Port    | Proxy   | DHCP Helper    | DHCP Relay Configuration    |
|           |              |           | Tagging | ARP     | Address        |                             |
+===========+==============+===========+=========+=========+================+=============================+
|         2 | 192.168.1.1/24 | Ethernet384 | untagged | disabled |              | Source Interface:           |
|           |              |           |         |         |                | Link Selection:             |
|           |              |           |         |         |                | Server Vrf:                 |
|           |              |           |         |         |                | Server ID Override:         |
+-----------+--------------+-----------+---------+---------+----------------+-----------------------------+
|         3 | 192.168.2.1/24 | Ethernet386 | untagged | disabled |              | Source Interface:           |
|           |              |           |         |         |                | Link Selection:             |
|           |              |           |         |         |                | Server Vrf:                 |
|           |              |           |         |         |                | Server ID Override:         |
+-----------+--------------+-----------+---------+---------+----------------+-----------------------------+
|       100 |              | Ethernet16  | untagged | disabled |              | Source Interface:           |
|           |              | PortChannel12 | untagged |       |              | Link Selection:             |
|           |              |           |         |         |                | Server Vrf:                 |
|           |              |           |         |         |                | Server ID Override:         |
+-----------+--------------+-----------+---------+---------+----------------+-----------------------------+
```

Port A sends a broadcast packet to Port B.

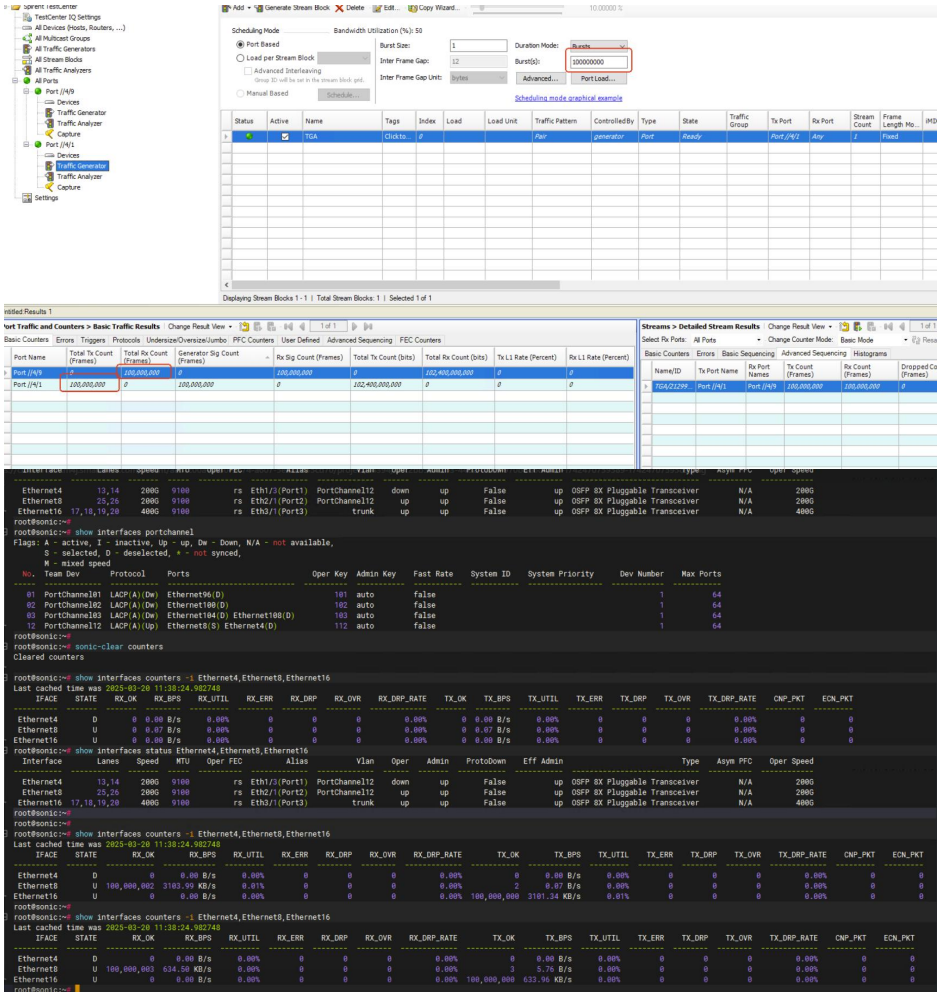
Port A



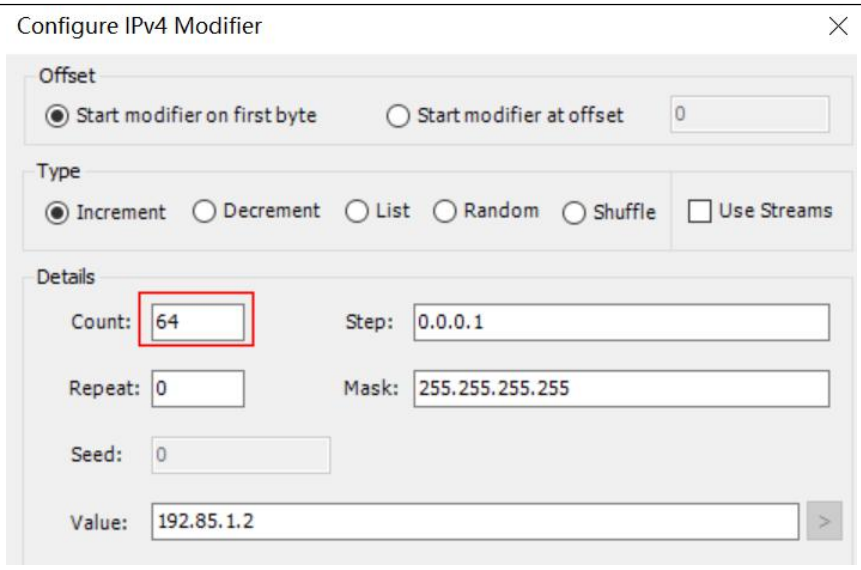


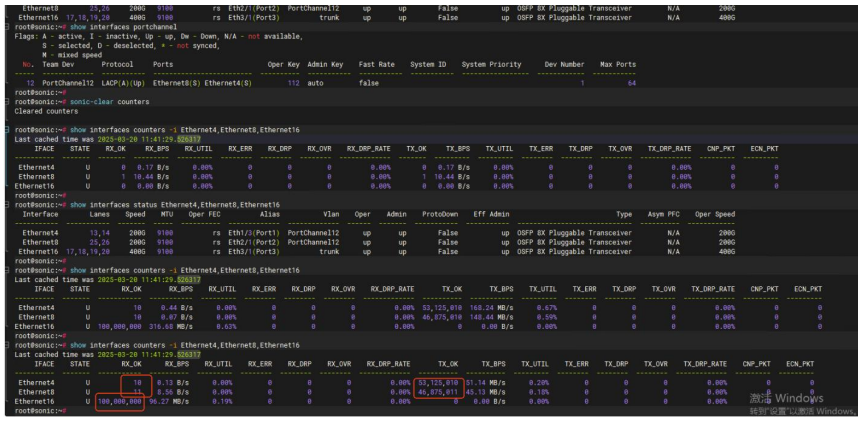


Disconnect one link

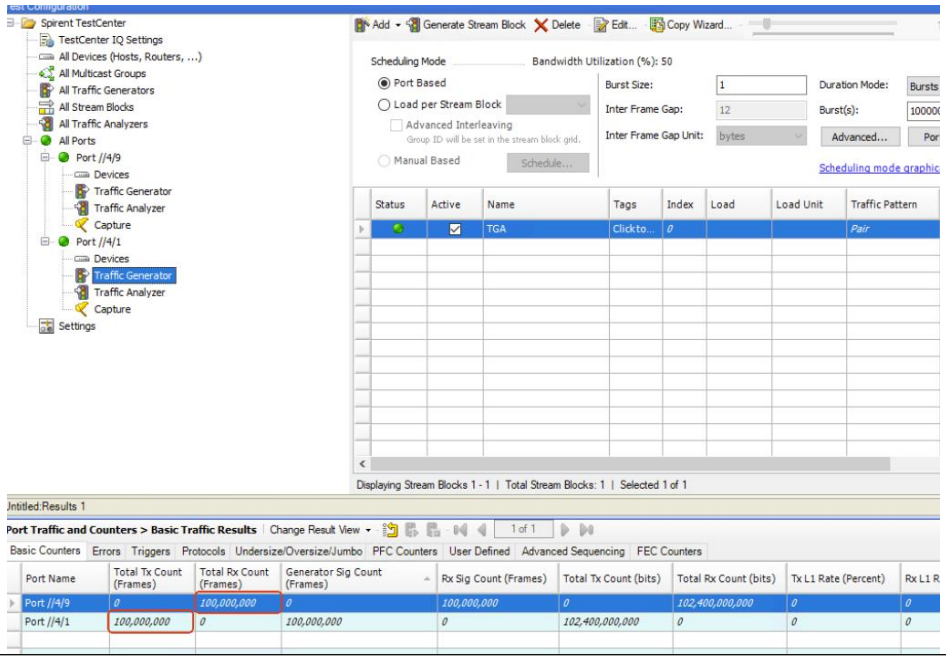


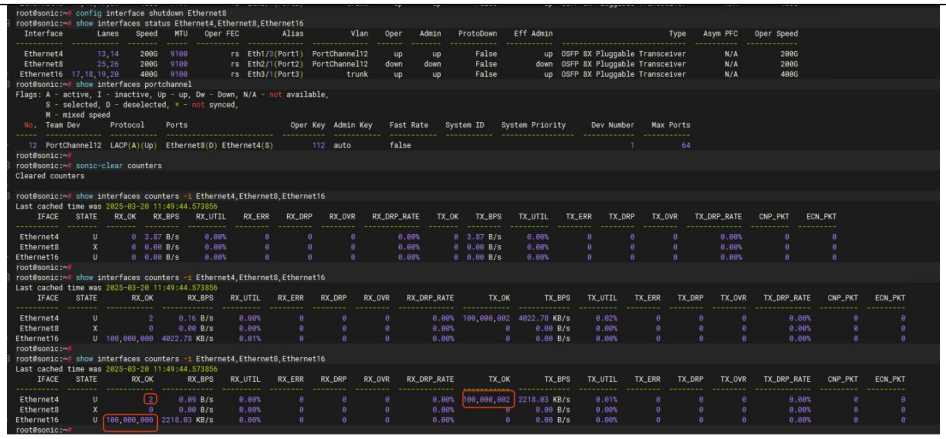
Port A sends unicast flows with incrementing source IPs to Port B.





Disconnect one link



	
Test results	PASS
Remarks	

3.2 LLDP Function Test

Test Items	LLDP Test
Test Content	The test device can normally send and receive LLDP and parse it.
Test Topology	
Test Steps	1、Set up the test environment as per the diagram. 2、Enable LLDP on the device and third-party devices. 3、View the LLDP neighbor information of the device, Check1. 4、View the LLDP neighbor information of the peer third-party device, Check2. 5、Delete the relevant configurations.
Expected Results	Check1: The device can correctly process the LLDP packets from the third-party device and display normal information. Check2: The LLDP packets sent by the device can be parsed by the third-party device, and the content is normal.
Actual Test Results	root@sonic:/home/admin# show lldp table Capability codes: (R) Router, (B) Bridge, (O) Other LocalPort RemoteDevice RemotePortID Capability RemotePortDescr ---------- ---------- ---------- ---------- ---------- Ethernet40 sonic Eth6/1(Port6) BR Ethernet40 Ethernet44 sonic Eth6/2(Port6) BR Ethernet44 Ethernet232 sonic Eth30(Port30) BR Ethernet232 eth0 68:21:5f:34:14:a2 BR Ethernet Port on unit 1, port 15 --- Total entries displayed: 4

root@sonic:/home/admin# show lldp table
Capability codes: (R) Router, (B) Bridge, (O) Other
LocalPort RemoteDevice RemotePortID Capability RemotePortDescr
---
Ethernet40 sonic Eth6/1(Port6) BR Ethernet40
Ethernet44 sonic Eth6/2(Port6) BR Ethernet44
Ethernet232 sonic Eth30(Port30) BR Ethernet232
eth0 68:21:5f:34:14:a2 BR Ethernet Port on unit 1, port 15

Total entries displayed: 4
root@sonic:/home/admin# show lldp global

LLDP Global configuration:

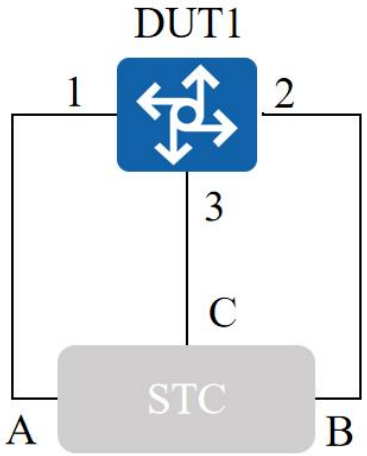
HELLO TIMER 30
SYSTEM NAME sonic
SYSTEM DESCRIPTION SONiC Software Version: SONiC.202111.30.6.2.251 - HwSku: Naddod-N9500-64OC - Distribution: Debian 11.11 - Kernel: 5.10.0-8-2-amd64
---
root@sonic:/home/admin# show lldp neighbors

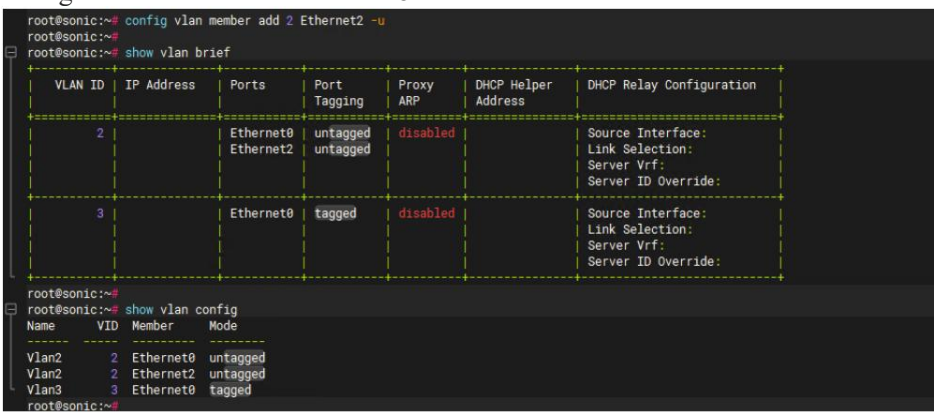
LLDP neighbors:

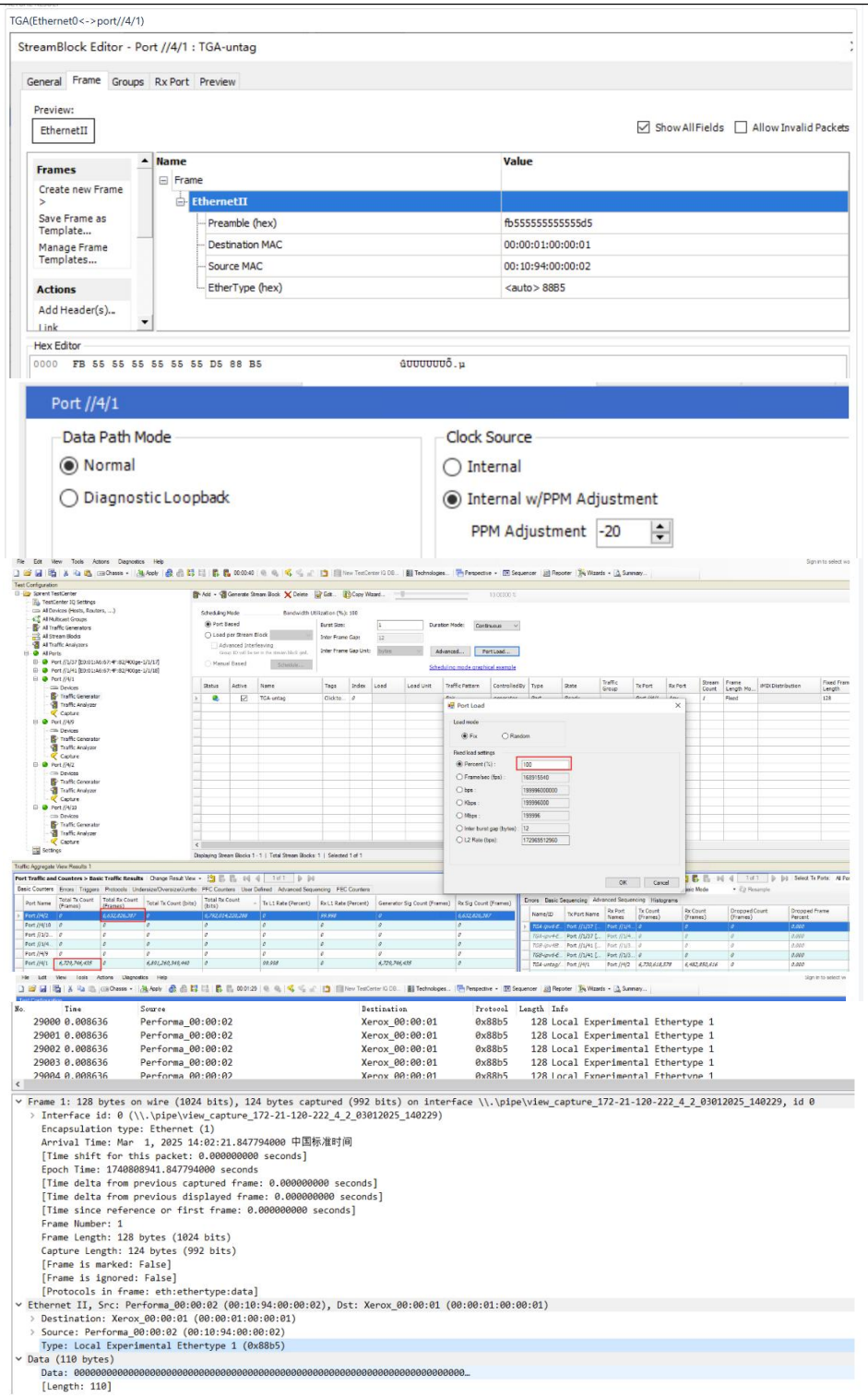
Interface: eth0, via: LLDP, RID: 2, Time: 0 day, 00:00:15
Chassis:
ChassisID: mac 68:21:5f:34:14:93
SysDescr: ECS5520-18T
Capability: Bridge, on
Capability: Router, on
Port:
PortID: mac 68:21:5f:34:14:a2
PortDescr: Ethernet Port on unit 1, port 15
TTL: 120
MFS: 1522
PMD autoneg: supported: yes, enabled: yes
Adv: 10Base-T, HD: yes, FD: yes
Adv: 100Base-TX, HD: yes, FD: yes
Adv: 1000Base-T, HD: no, FD: yes
MAU oper type: 1000BaseTFD - Four-pair Category 5 UTP, full duplex mode
VLAN: 120, pvid: yes
PPVID: supported: yes, enabled: no
PI: 00 27 42 42 03 00 00 02
PI: 88 09 01
PI: 88 09 03
PI: 88 8e
LLDP-MED:
Device Type: Network Connectivity Device
Capability: Capabilities, yes
Capability: Policy, yes
Capability: Location, yes
Capability: Inventory, yes
LLDP-MED Location Identification: Type: Civic address
Country: TW
Inventory:
Hardware Revision: R0A

	Firmware Revision: 10.27.14.15 Serial Number: EC1948000847 Manufacturer: edgecore Model: ECS5520-18T
Test results	**PASS**
Remarks	

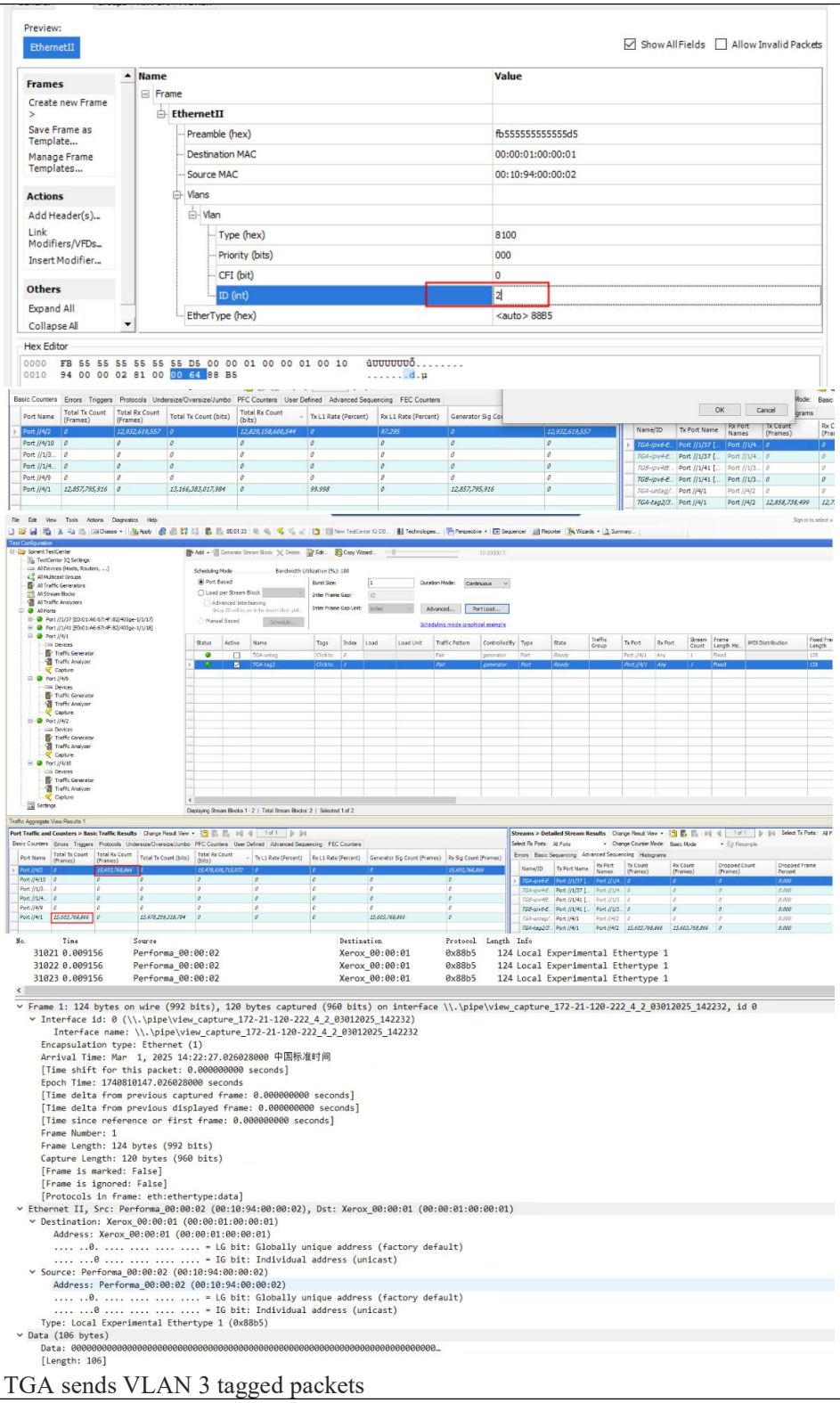
3.3 VLAN Function Test

Test Items	VLAN Function Test
Test Content	The test device can forward according to the set VLAN.
Test Topology	
Test Steps	1、Set up the test environment as per the diagram. 2、Create VLAN 2 and VLAN 3. 3、Configure DUTP1 to join VLAN 2 as untagged and VLAN 3 as tagged. 4、Configure DUTP2 to join VLAN 2 as untagged. 5、Configure DUTP3 to join VLAN 3 as tagged, and check whether the configuration is successful ,Check1. 6、The TGA sends untagged packets, and view the packet reception status on the instrument ,Check2. 7、The TGA sends VLAN 2 tagged packets, and view the packet reception status on the instrument ,Check3. 8、The TGA sends VLAN 3 tagged packets, and view the packet reception status on the instrument ,Check4. 9、The TGA sends VLAN 4 tagged packets, and view the packet reception status on the instrument ,Check5.
Expected Results	Check1: Configuration and binding are successful. Check2: Only TGB receives the packets, and the packets are untagged. Check3: Only TGB receives the packets, and the packets are untagged. Check4: Only TGC receives the packets, and the packets are VLAN 3 tagged.

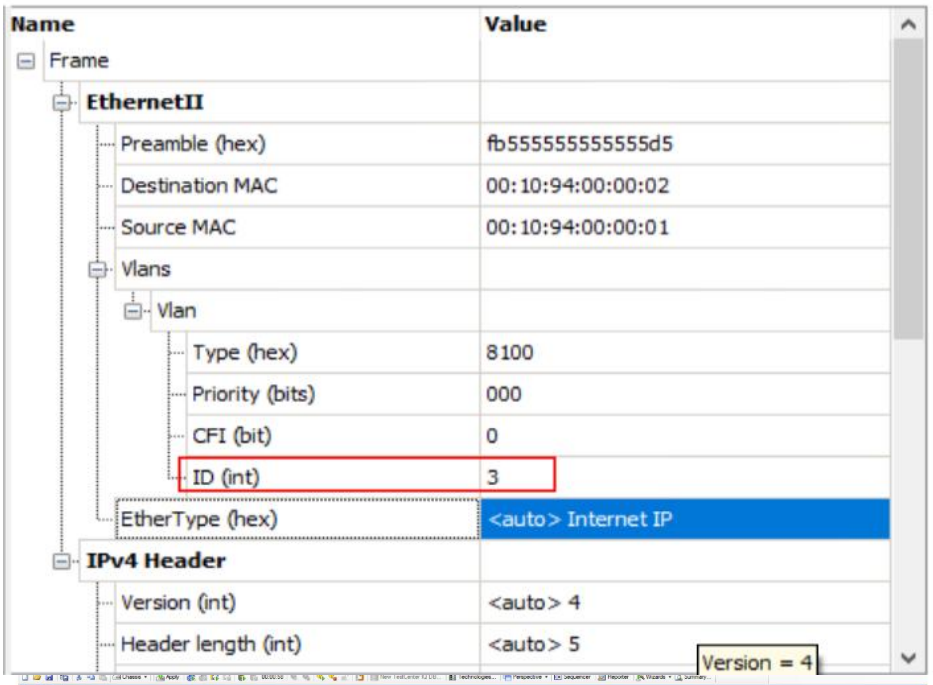
	Check5: The packets are discarded, and no port receives the packets.
Actual Test Results	Configure DUTP1 to join VLAN 2 as untagged and VLAN 3 as tagged. config vlan member add 2 Ethernet0 -u config vlan member add 3 Ethernet0  config vlan member add 2 Ethernet8 -u  config vlan member add 3 Ethernet16  TGA sends untagged packets.

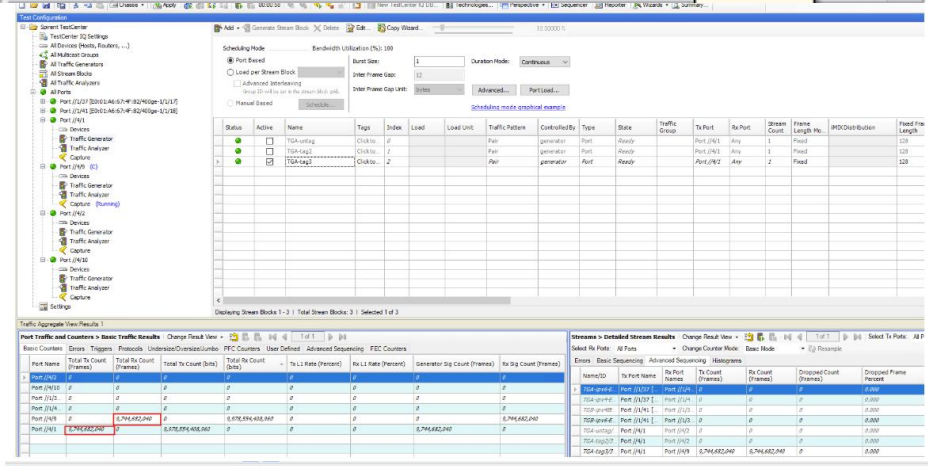


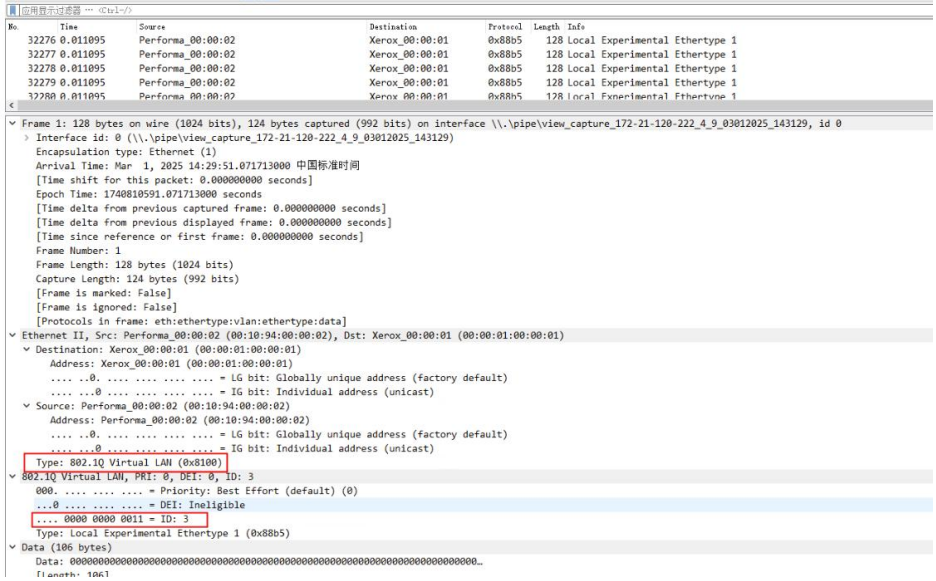
TGA sends VLAN 2 tagged packets



TGA sends VLAN 3 tagged packets

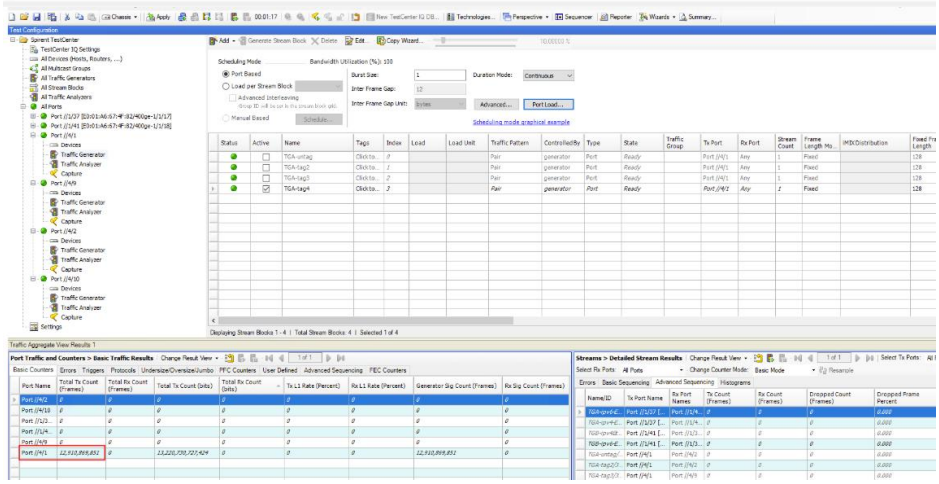




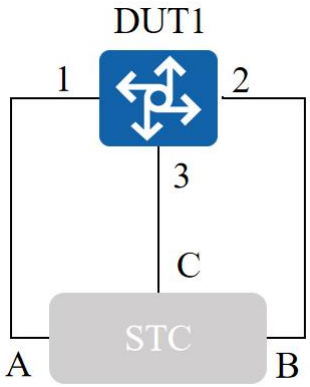
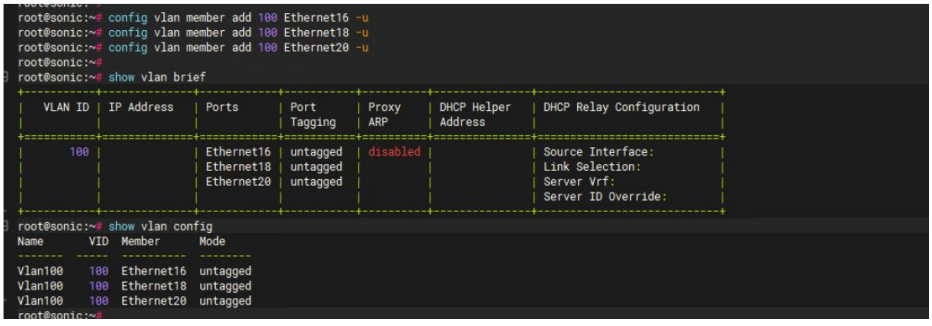


No.	Time	Source	Destination	Protocol	Length	Info
32276	0.011095	Performa_00:00:02	Xerox_00:00:01	0x88b5	128	Local Experimental Ethertype 1
32277	0.011095	Performa_00:00:02	Xerox_00:00:01	0x88b5	128	Local Experimental Ethertype 1
32278	0.011095	Performa_00:00:02	Xerox_00:00:01	0x88b5	128	Local Experimental Ethertype 1
32279	0.011095	Performa_00:00:02	Xerox_00:00:01	0x88b5	128	Local Experimental Ethertype 1
32280	0.011095	Performa_00:00:02	Xerox_00:00:01	0x88b5	128	Local Experimental Ethertype 1

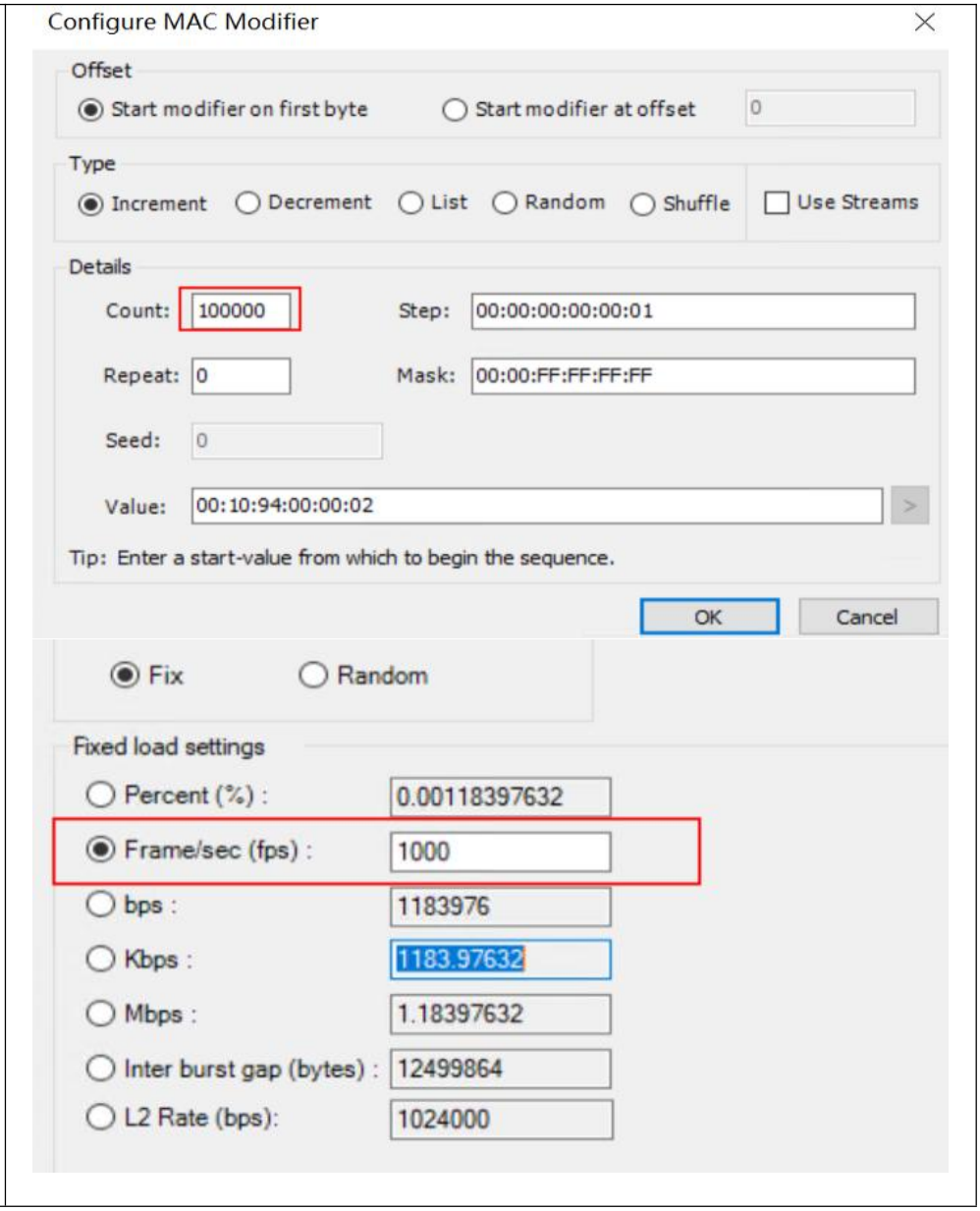
TGA sends VLAN 2 tagged packets

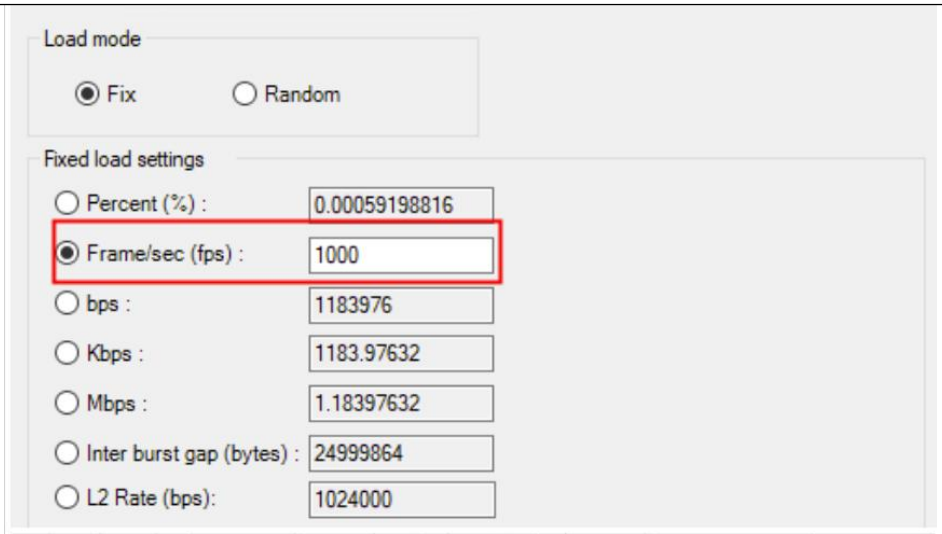
	Frame EthernetII Preamble (hex): fb555555555555d5 Destination MAC: 00:10:94:00:00:02 Source MAC: 00:10:94:00:00:01 Vlans Vlan Type (hex): 8100 Priority (bits): 000 CFI (bit): 0 ID (int): 4 EtherType (hex): <auto> Internet IP IPv4 Header Version (int): <auto> 4 Header length (int): <auto> 5 
Test results	**PASS**
Remarks	

3.4 Mac Address Table Entry Specification Test

Test Items	Mac Address Table Entry Specification Test
Test Content	Verification of Device MAC Address Table Entry Specifications
Test Topology	
Test Steps	1、set up the test environment as shown in the diagram. 2、Create VLAN 100. 3、Configure ports DUTP1-DUTP3 to join VLAN 100 as untagged members. 4、The test instrument TGA sends 100,000 packets with varying source MAC addresses (total 100,000 unique SMACs) at a rate of 1,000 pps. 5、Clear the packet reception statistics on the instrument. 6、TGB injects 100,000 packets with DMAC set to the SMAC of TGA, then perform Check 1.
Expected Results	Check1: The number of packets received by TGA minus the number of packets received by TGC equals the number of MAC addresses learned by the DUT.
Actual Test Results	config vlan add 100 config vlan member add 100 Ethernet16 -u config vlan member add 100 Ethernet18 -u config vlan member add 100 Ethernet20 -u  TGA: DST MAC: 00:00:02:00:00:01 SRC MAC: 00:10:94:00:00:02 – Increment (100000 个 MAC) Loading: 1000pps

```
root@sonic:~# config vlan member add 100 Ethernet16 -u
root@sonic:~# config vlan member add 100 Ethernet18 -u
root@sonic:~# config vlan member add 100 Ethernet20 -u
root@sonic:~#
root@sonic:~# show vlan brief
+-----------+-------------+------------+----------+---------+----------------+--------------------------+
| VLAN ID   | IP Address  | Ports      | Port     | Proxy   | DHCP Helper    | DHCP Relay Configuration |
|           |             |            | Tagging  | ARP     | Address        |                          |
+===========+=============+============+==========+=========+================+==========================+
|       100 |             | Ethernet16 | untagged | disabled|                | Source Interface:        |
|           |             | Ethernet18 | untagged |         |                | Link Selection:          |
|           |             | Ethernet20 | untagged |         |                | Server Vrf:              |
|           |             |            |          |         |                | Server ID Override:      |
+-----------+-------------+------------+----------+---------+----------------+--------------------------+
root@sonic:~# show vlan config
Name       VID  Member      Mode
-------  -----  ----------  --------
Vlan100    100  Ethernet16  untagged
Vlan100    100  Ethernet18  untagged
Vlan100    100  Ethernet20  untagged
root@sonic:~#
```

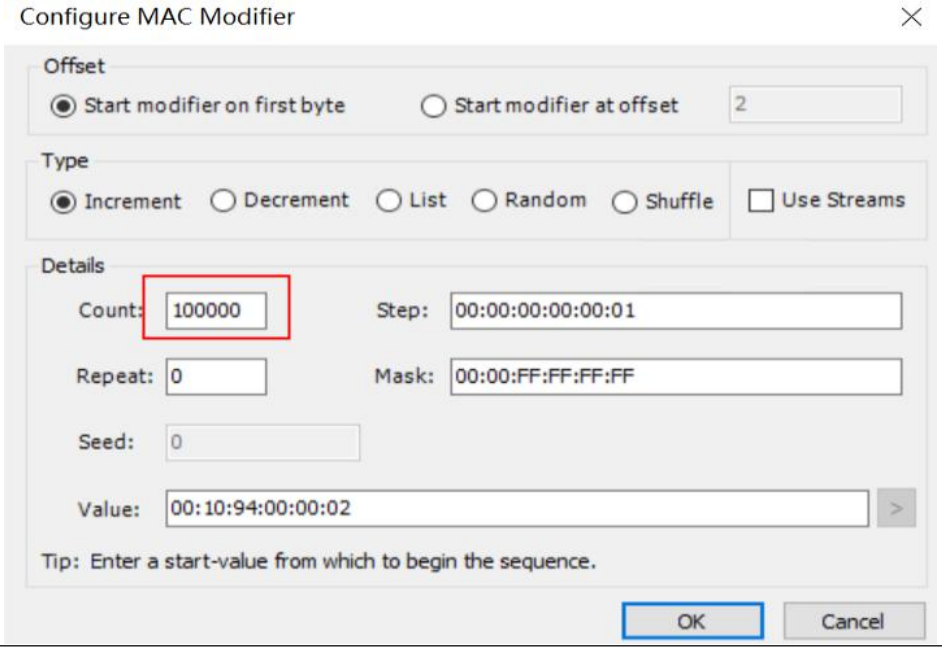

Configure MAC Modifier
Offset
Start modifier on first byte
Start modifier at offset
0
Type
Increment
Decrement
List
Random
Shuffle
Use Streams
Details
Count:
100000
Step:
00:00:00:00:00:01
Repeat:
0
Mask:
00:00:FF:FF:FF:FF
Seed:
0
Value:
00:10:94:00:00:02
Tip: Enter a start-value from which to begin the sequence.
OK
Cancel
Fix
Random
Fixed load settings
Percent (%) :
0.00118397632
Frame/sec (fps) :
1000
bps :
1183976
Kbps :
1183.97632
Mbps :
1.18397632
Inter burst gap (bytes) :
12499864
L2 Rate (bps):
1024000

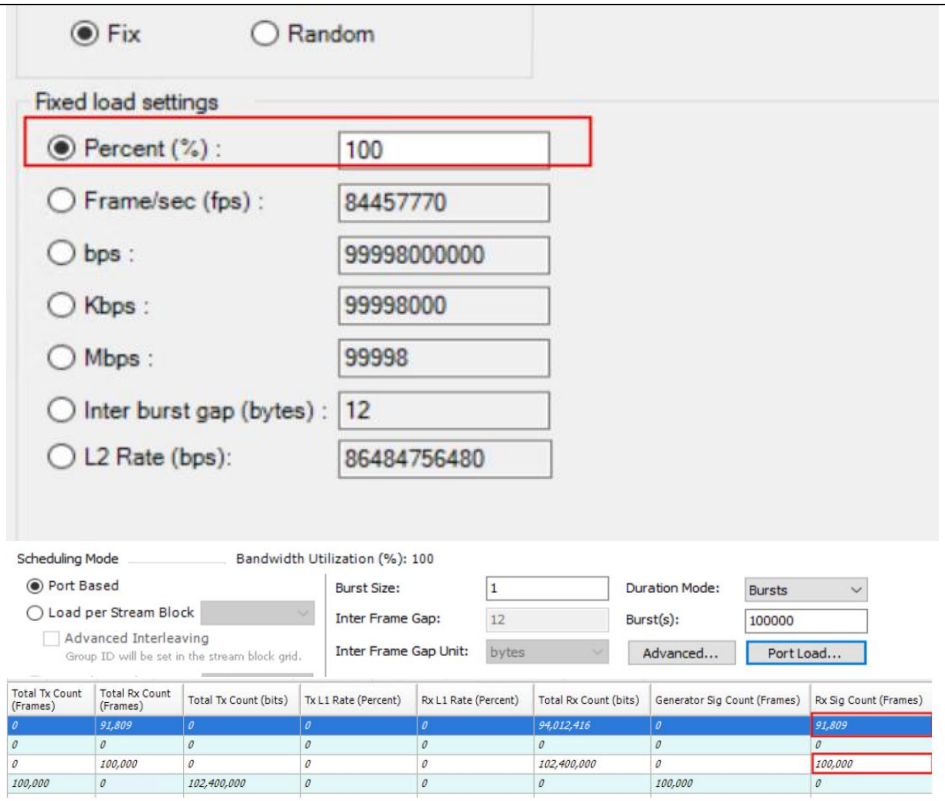


Total Tx Count (Frames)	Total Rx Count (Frames)	Total Tx Count (bits)	Tx L1 Rate (Percent)	Rx L1 Rate (Percent)	Total Rx Count (bits)	Generator Sig Count (Frames)	Rx Sig Count (Frames)
0	100,000	0	0	0	102,400,000	0	100,000
0	0	0	0	0	0	0	0
100,000	0	102,400,000	0	0	0	100,000	0
0	100,000	0	0	0	102,400,000	0	100,000

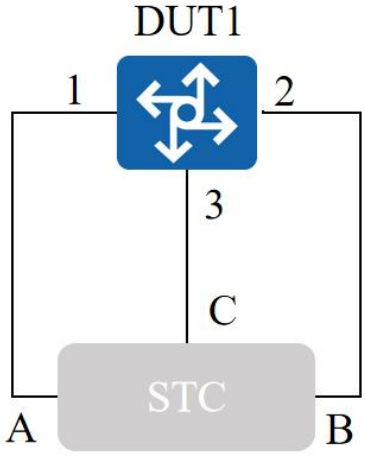
```
  8185     100  00:10:94:00:10:A3  Ethernet20  Dynamic
  8186     100  00:10:94:00:14:C8  Ethernet20  Dynamic
  8187     100  00:10:94:00:0E:5C  Ethernet20  Dynamic
  8188     100  00:10:94:00:11:5F  Ethernet20  Dynamic
  8189     100  00:10:94:00:02:9E  Ethernet20  Dynamic
  8190     100  00:10:94:00:1D:EE  Ethernet20  Dynamic
  8191     100  00:10:94:00:1D:13  Ethernet20  Dynamic
Total number of entries 8191
root@sonic:~#
root@sonic:~# show mac -c
Total number of entries 8191
root@sonic:~#
```

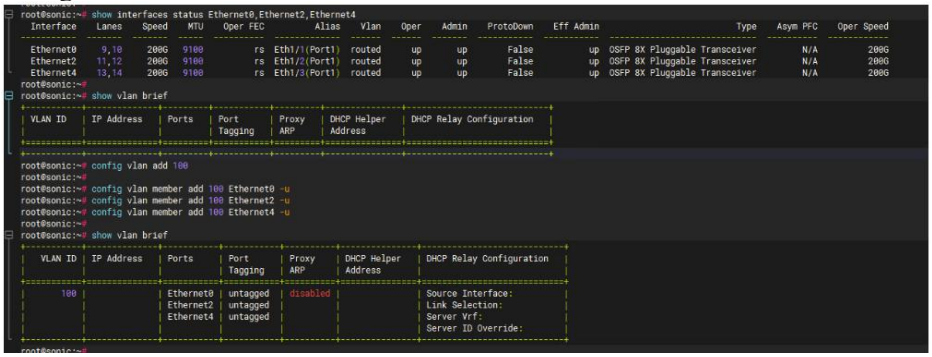
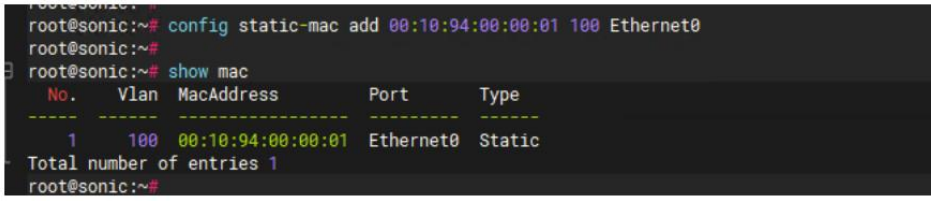
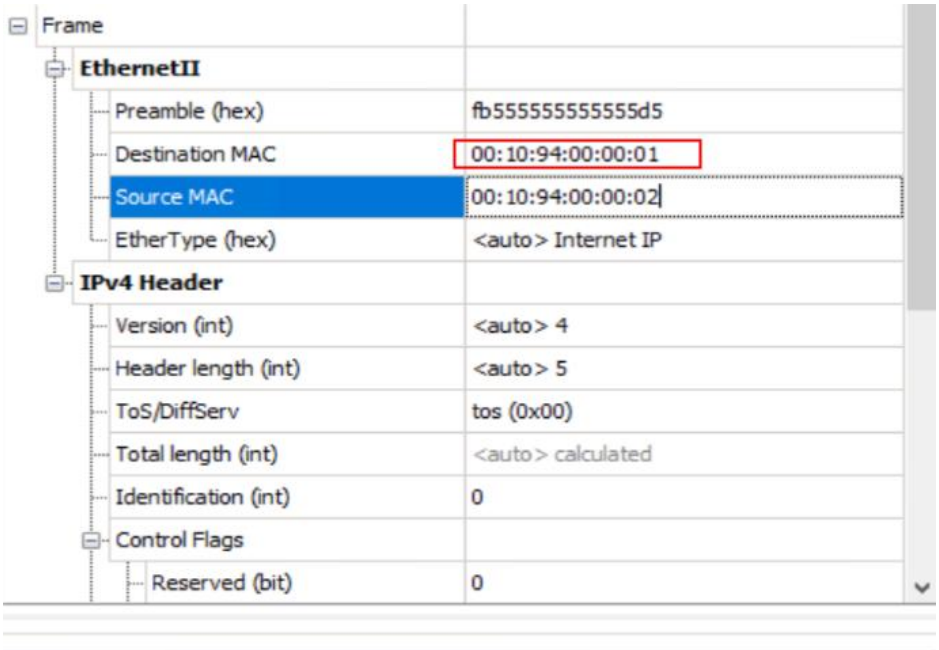
TGB:
Dst MAC: 00:10:94:00:00:02 – Increment (100000 个 MAC)
SRC MAC: 00:00:02:00:00:01
Loading: 100%

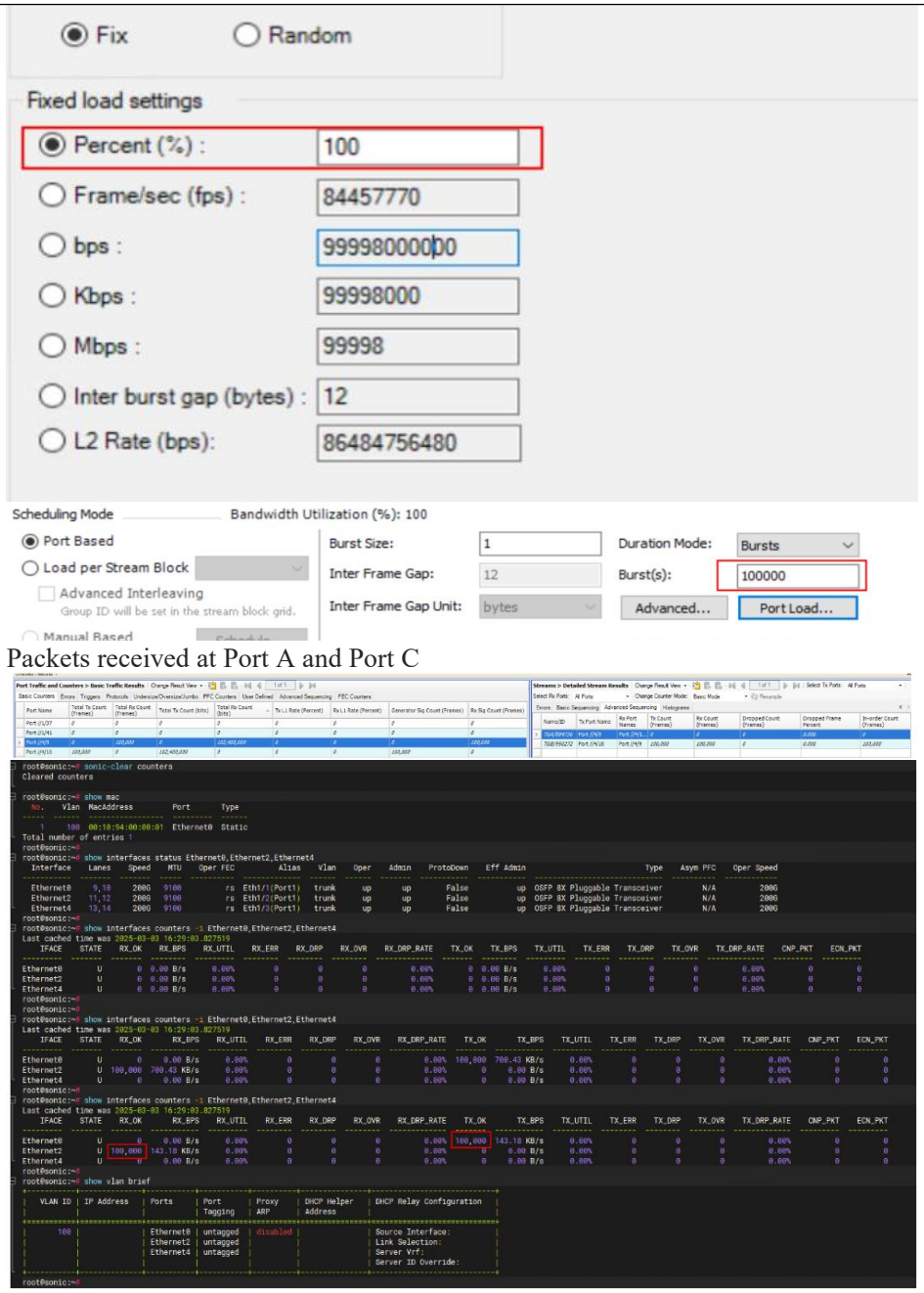


	
Test results	**PASS**
Remarks	Due to the relationship of MAC hash, it is necessary to inject more packets than the number of MAC address table entries.

3.5 Static FDB Function Test

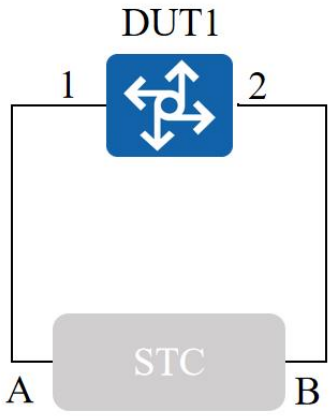
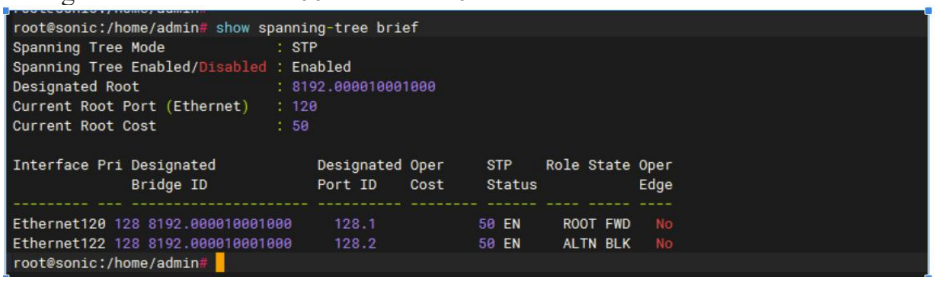
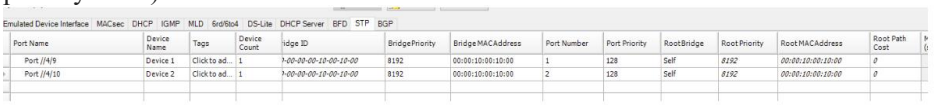
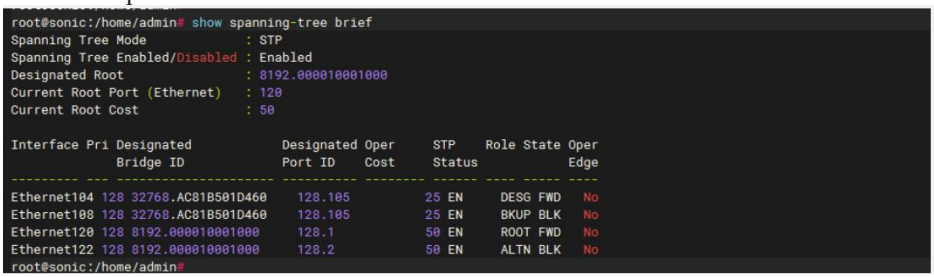
Test Items	Static FDB Test
Test Content	Verify that the device's Static FDB is functioning properly.
Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2、Create VLAN 100.

	3、Configure ports DUTP1-DUTP3 to join VLAN 100 as untagged members. 4、Configure a static MAC address for VLAN 100 on DUTP1. 5、TGB injects 100,000 packets with DMAC set to the static MAC address. 6、configured for VLAN 100 on DUTP1, then observe the traffic transmission and reception status，Check 1.
Expected Results	Check1: TGA receives all 100,000 packets sent by TGB, while TGC receives no packets.
Actual Test Results	config vlan add 100 config vlan member add 100 Ethernet0 -u config vlan member add 100 Ethernet2 -u config vlan member add 100 Ethernet4 -u  config static-mac add 00:10:94:00:00:01 100 Ethernet0  Port B 

	 Packets received at Port A and Port C
Test results	**PASS**
Remarks	

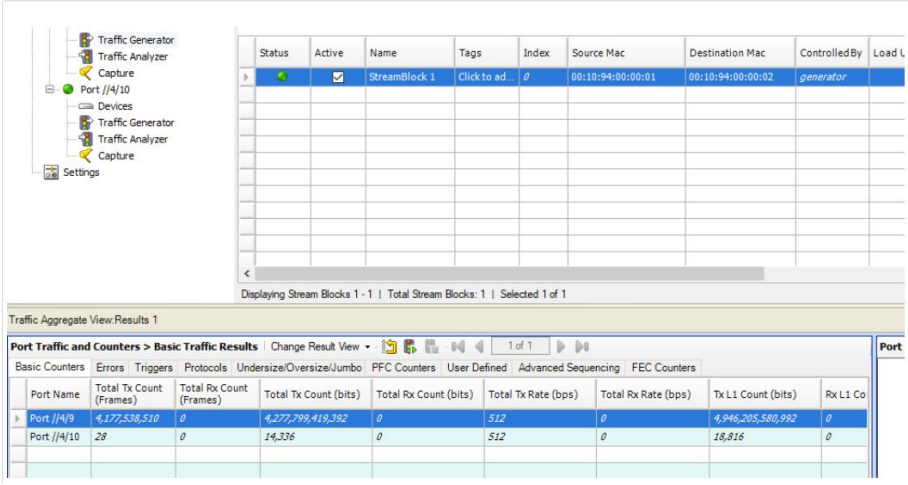
3.6 STP Function Test

Test Items	STP Function Test
Test Content	Test that the device's STP is working properly.

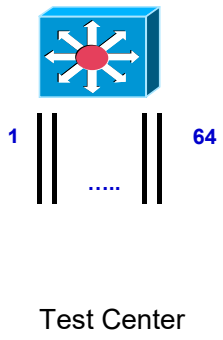
Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2、Configure VLANs. 3、Configure STP. 4、TGA and TGB simulate two ports of the same device with STP enabled (root priority 8192). 5、Display the STP port status ,check 1. 6、TGA sends data traffic to confirm whether a loop exists, and observe the traffic transmission and reception status,check2.
Expected Results	Check1: No network loop exists. Check2: TGB does not receive any packets from TGA.
Actual Test Results	config spanning-tree enable stp config vlan add 100 config vlan member add 100 Ethernet122 -u config vlan member add 100 Ethernet120 -u  TGA and TGB simulate two ports of the same device with STP enabled (root priority 8192).  View STP port status  No network loop exists.

```
root@sonic:/home/admin# show spanning-tree brief
Spanning Tree Mode            : STP
Spanning Tree Enabled/Disabled : Enabled
Designated Root               : 8192.000010001000
Current Root Port (Ethernet)  : 120
Current Root Cost             : 50

Interface Pri Designated           Designated Oper     STP    Role State Oper
              Bridge ID            Port ID    Cost     Status           Edge
--------- --- -------------------- ---------- -------- ------ ---- ----- ----
Ethernet120 128 8192.000010001000     128.1            50 EN     ROOT FWD   No
Ethernet122 128 8192.000010001000     128.2            50 EN     ALTN BLK   No
root@sonic:/home/admin#
```

```
root@sonic:/home/admin# show spanning-tree brief
Spanning Tree Mode            : STP
Spanning Tree Enabled/Disabled : Enabled
Designated Root               : 8192.000010001000
Current Root Port (Ethernet)  : 120
Current Root Cost             : 50

Interface Pri Designated           Designated Oper     STP    Role State Oper
              Bridge ID            Port ID    Cost     Status           Edge
--------- --- -------------------- ---------- -------- ------ ---- ----- ----
Ethernet104 128 32768.AC81B501D460    128.105          25 EN     DESG FWD   No
Ethernet108 128 32768.AC81B501D460    128.105          25 EN     BKUP BLK   No
Ethernet120 128 8192.000010001000     128.1            50 EN     ROOT FWD   No
Ethernet122 128 8192.000010001000     128.2            50 EN     ALTN BLK   No
root@sonic:/home/admin#
```

	
Test results	**PASS**
Remarks	

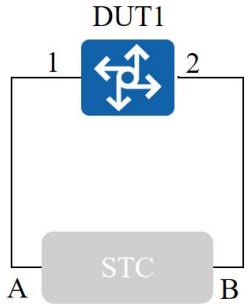
3.7 RFC-2544 Function Test

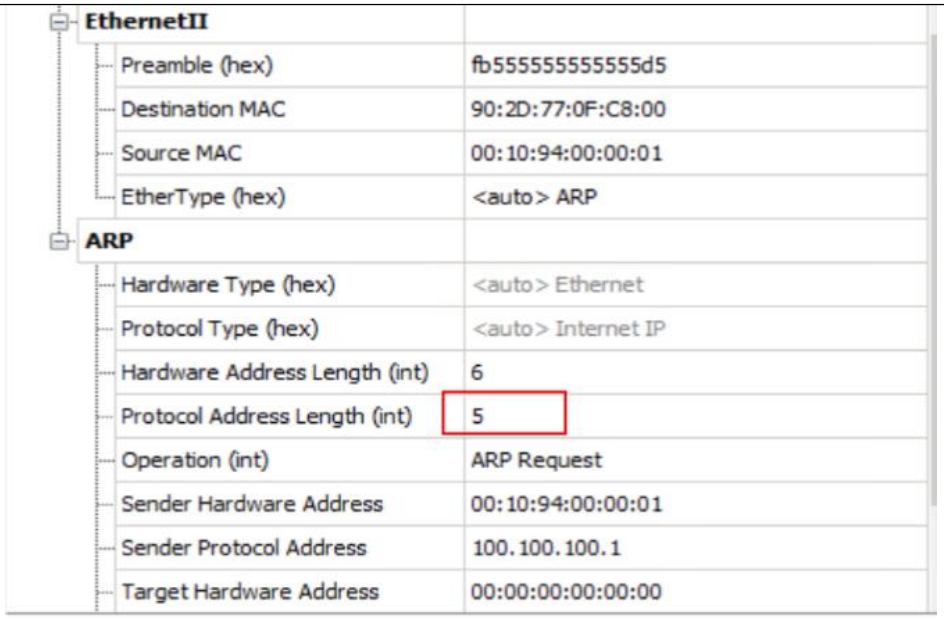
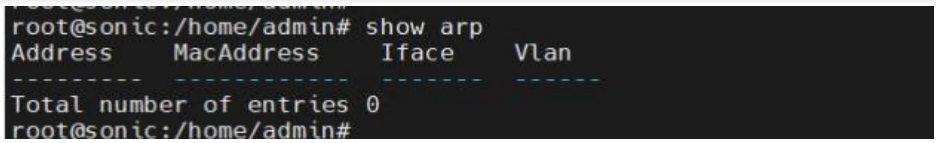
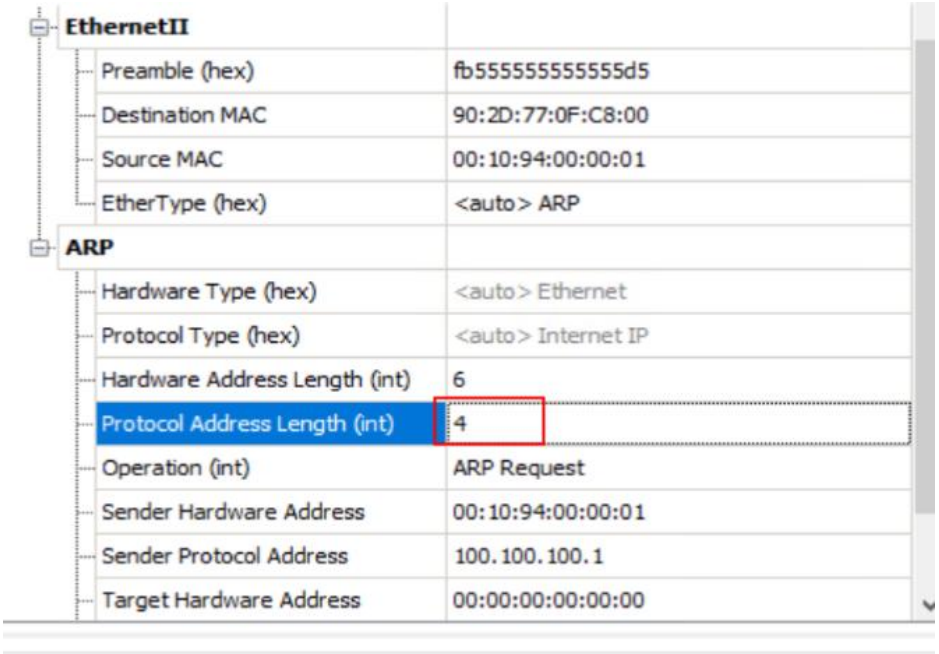
Test Items	RFC2544 Test
Test Content	The device is connected to a tester for RFC2544 testing.
Test Topology	
Test Steps	1. Set up the test environment as shown in the diagram. 2. Configure VLAN. 3. Use the tester to perform RFC2544 testing. 4. Delete the relevant configurations.
Expected Results	The RFC2544 test results meet expectations.
Actual Test Results	config vlan add 100 config vlan member add 100 Ethernet440 -u config vlan member add 100 Ethernet8 -u

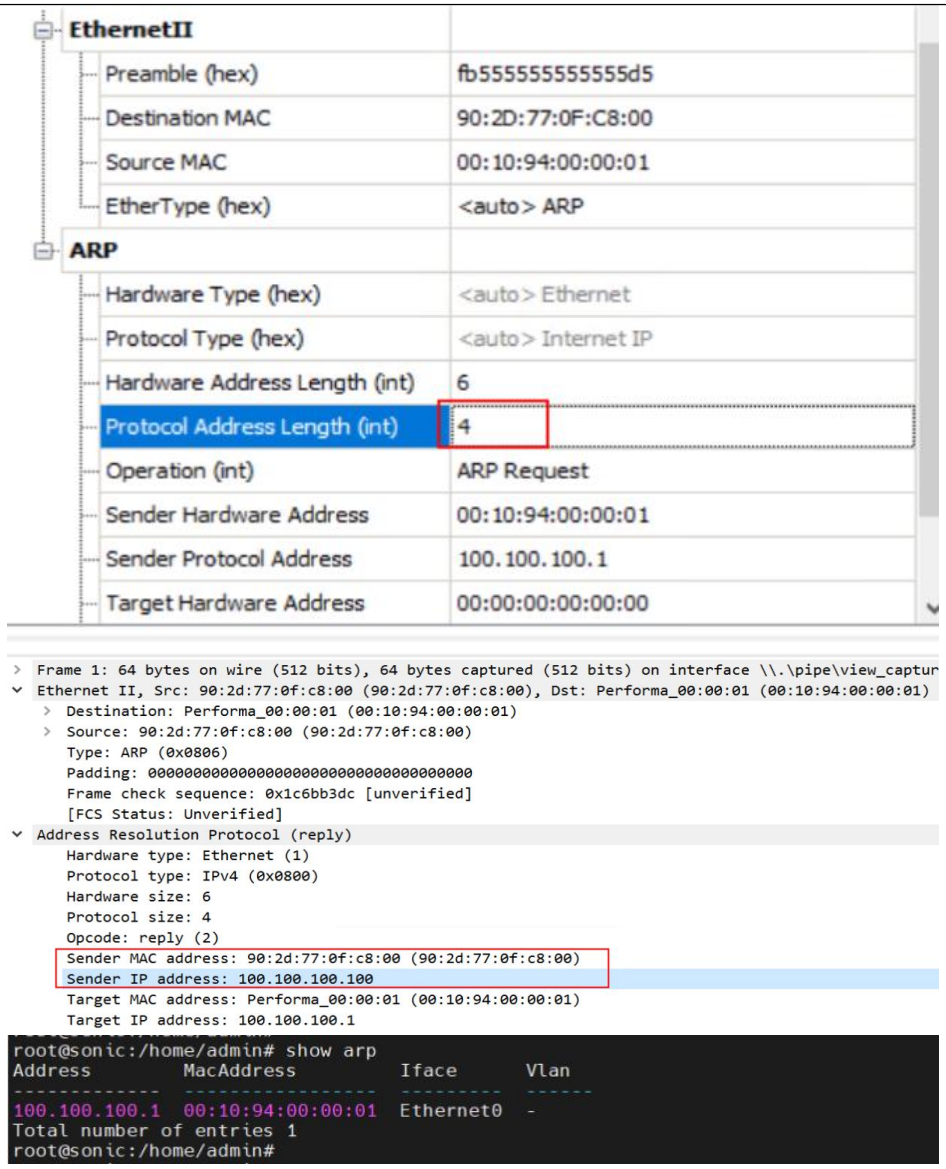
	Close lldp systemctl stop lldp.service Throughput report： rfc2544_topology_two_400G_port
Test results	**PASS**
Remarks	

四、L3

4.1 ARP Protocol Test

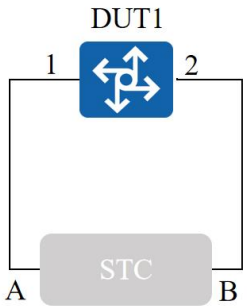
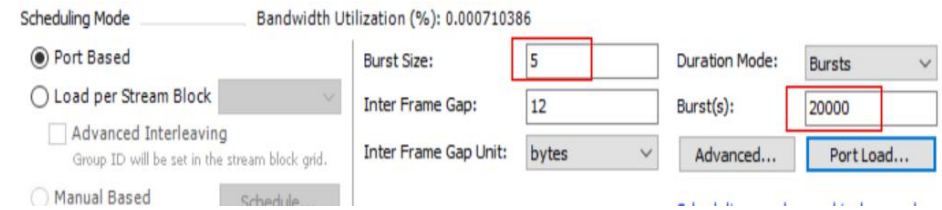
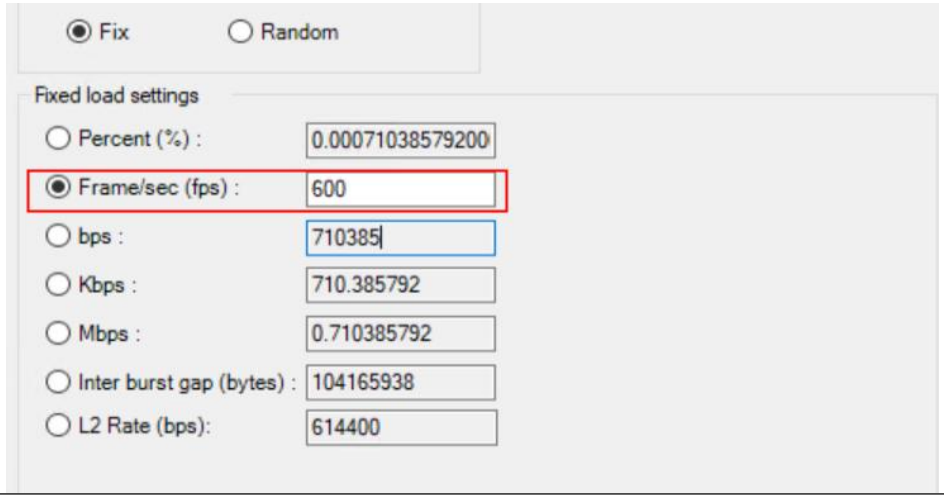
Test Items	ARP Protocol Test
Test Content	Test that the device can correctly process ARP packets.
Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2、Configure DUTP1 IP as 100.100.100.100/24. 3、TGA sends an incorrect ARP request to 100.100.100.100 with 100.100.100.1 as the source MAC for Check 1. 4、TGA sends a correct ARP request to 100.100.100.100 with 100.100.100.1 as the source MAC for Check 2. 5、Delete the relevant configurations.
Expected Results	Check1: The DUT does not learn 100.100.100.1. Check2: The DUT learns 100.100.100.1.
Actual Test Results	Configure the IP address of device port 1 as 100.100.100.100/24. config interface ip add Ethernet0 100.100.100.100/24 Port A

	  Port A 

	
Test results	**PASS**
Remarks	

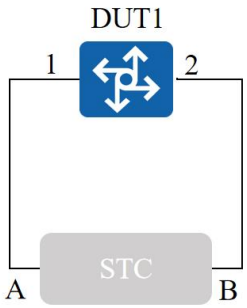
4.2 ARP Specification Test

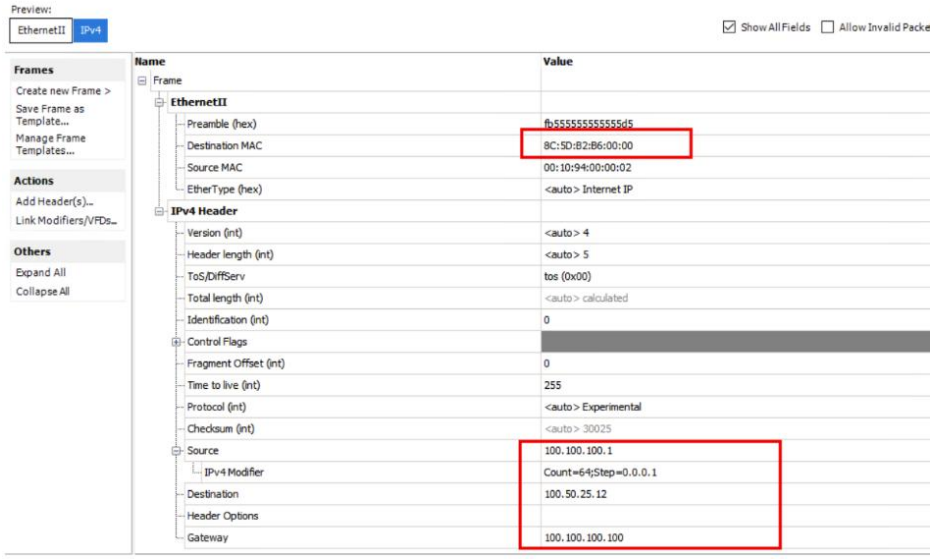
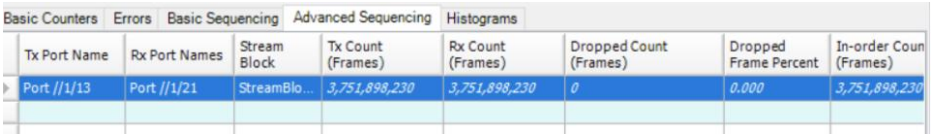
Test Items	ARP Specification Test
Test Content	Test the ARP specifications of the device

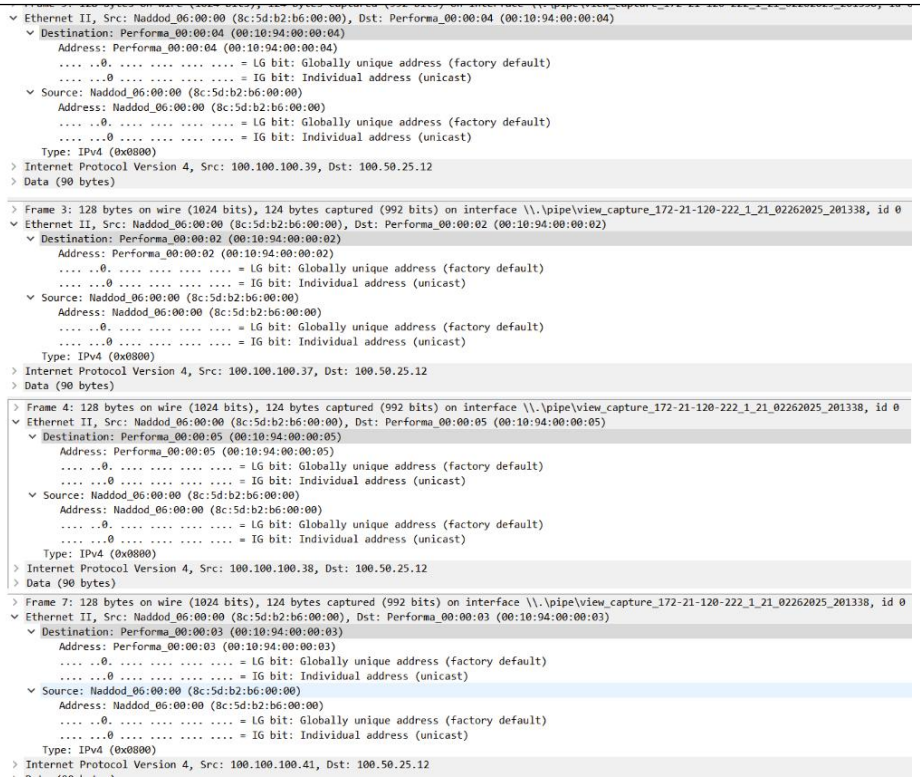
Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2、Configure the IP address of DUT port P1 as 100.0.0.1/8. 3、TGA starts sending incremental ARP requests from 100.0.0.2 to 100.0.0.1，Check 1. 4、Delete the relevant configurations.
Expected Results	Check1: The ARP entries learned by the DUT comply with the specifications.
Actual Test Results	Configure the IP address of device port 1 as 100.0.0.1/8. config interface ip add Ethernet0 100.0.0.1/8 Port A  

	ACTUAL RESULT 192.168.64.246 00:10:94:00:3f:f5 - 154 192.168.64.247 00:10:94:00:3f:f6 - 154 192.168.64.248 00:10:94:00:3f:f7 - 154 192.168.64.249 00:10:94:00:3f:f8 - 154 192.168.64.250 00:10:94:00:3f:f9 - 154 192.168.64.251 00:10:94:00:3f:fa - 154 192.168.64.252 00:10:94:00:3f:fb - 154 192.168.64.253 00:10:94:00:3f:fc - 154 Total number of entries 16383 root@sonic:/home/admin#
Test results	**PASS**
Remarks	

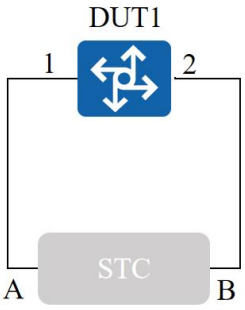
4.3 ECMP

Test Items	ECMP Function Test
Test Content	Test that the device can learn routes via BGP and forward packets accordingly
Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2、Configure the interface IP address, Check 1. 3、The tester TGA creates an IP interface and performs ARP interaction with device Port 1. 4、The tester TGB creates multiple IP interfaces and performs ARP interaction with device P2. 5、Configure equal-cost static routes, Check 2. 6、TGA sends L3 packets to the destination IP specified by the equal-cost routes, Check 3. 7、Delete the relevant configurations.
Expected Results	Check1: The IP address is configured successfully. Check2: The configuration is successful. Check3: The packet forwarding is normal without packet loss, and load balancing is achieved via ECMP.

Actual Test Results	Configure the interface IP address.
	config interface ip add Ethernet0 100.100.100.100/24
	config interface ip add Ethernet8 200.200.200.200/24
	The tester TGA creates an interface with IP 100.100.100.1 and performs ARP interaction with the device IP 100.100.100.100.
	The tester TGB creates interfaces with IPs 200.200.200.1 to 200.200.200.4 and performs ARP interaction with the device IP 200.200.200.200.
	Configure equal-cost static routes.
	vtysh
	configure terminal
	ip route 100.50.25.12/32 200.200.200.1
	ip route 100.50.25.12/32 200.200.200.2
	ip route 100.50.25.12/32 200.200.200.3
	ip route 100.50.25.12/32 200.200.200.4
	`sonic# show ip route static`
	`Codes: K - kernel route, C - connected, S - static, R - RIP,`
	`O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,`
	`T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,`
	`f - OpenFabric,`
	`> - selected route, * - FIB route, q - queued, r - rejected, b - backup`
	`t - trapped, o - offload failure`
	`S>* 100.50.25.12/32 [1/0] via 200.200.200.1, Ethernet416, weight 1, 00:00:44`
	`* via 200.200.200.2, Ethernet416, weight 1, 00:00:44`
	`* via 200.200.200.3, Ethernet416, weight 1, 00:00:44`
	`* via 200.200.200.4, Ethernet416, weight 1, 00:00:44`
	`sonic#`
	The tester TGA continuously sends packets with incremental source IPs from 100.100.100.1 to 100.100.100.64, and the destination IP is 100.50.25.12.
	Port A
	 
	Port B
	 
	The tester TGB captures packets and checks the packet content.

	
Test results	**PASS**
Remarks	

4.4 BGP

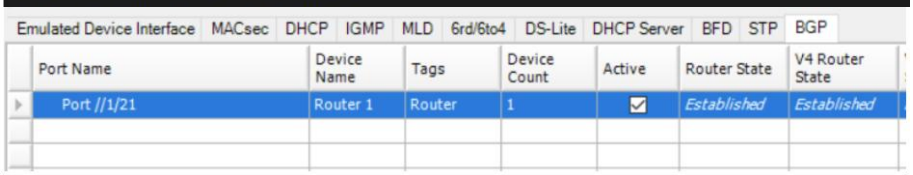
Test Items	BGP Function Test
Test Content	Test that the device can learn routes via BGP and forward packets accordingly
Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2、Configure DUT port P1 with IP 100.100.100.100/24 and port P2 with IP 200.200.200.200/24, Check 1. 3、Configure DUT BGP AS100; advertise networks 100.100.100.100/24 and 200.200.200.200/24; set neighbor 200.200.200.1 (AS200), Check 2. 4、TGA (100.100.100.1/24) performs ARP interaction with the device. 5、TGB (200.200.200.1/24) establishes an eBGP neighbor with the device and advertises route 2.2.2.0/24, Check 3.

	6、TGA sends packets with DA=DUT MAC, SIP=100.100.100.1, DIP=2.2.2.100, Check 4. 7、Delete the relevant configurations.
Expected Results	Check1: The IP address configuration is successful. Check2: The configuration is successful. Check3: The neighbor establishment is successful, and the DUT learns the route 2.2.2.0/24. Check4: The packet forwarding is normal without packet loss.

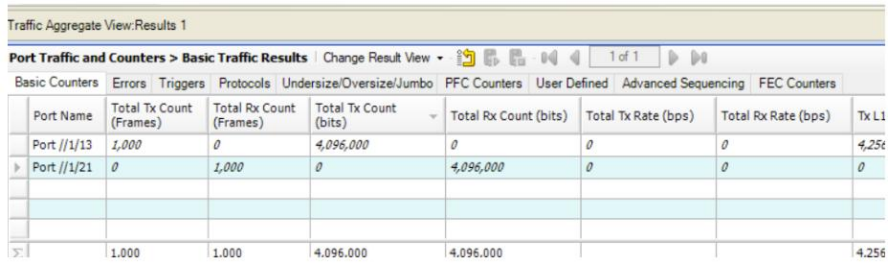
Actual Test Results	Configure the IP address of device port 1 as 100.100.100.100/24 and port 2 as 200.200.200.200/24.
	config interface ip add Ethernet0 100.100.100.100/24
	config interface ip add Ethernet8 200.200.200.200/24
	`root@sonic:~# show ip interfaces`
	`Interface Master IPv4 address/mask Admin/Oper BGP Neighbor Neighbor IP`
	`Ethernet384 100.100.100.100/24 up/up N/A N/A`
	`Ethernet416 200.200.200.200/24 up/up N/A N/A`
	Configure DUT BGP
	vtysh
	configure terminal
	router bgp 100
	network 100.100.100.0/24
	network 200.200.200.0/24
	neighbor 200.200.200.1 remote-as 200
	no bgp ebgp-requires-policy
	`router bgp 100`
	`no bgp ebgp-requires-policy`
	`neighbor 200.200.200.1 remote-as 200`
	`!`
	`address-family ipv4 unicast`
	`network 100.100.100.0/24`
	`network 200.200.200.0/24`
	`exit-address-family`
	`exit`
	TGA (100.100.100.1/24) performs ARP interaction with the device.
	Ping Options
	Frame Count: 1
	Time Interval(Sec): 1
	Gateway/Destination IP
	Same As Gateway: 100.100.100.100
	Custom:
	Ping Status:
	Ping
	Stop
	PING 100.100.100.100 from 100.100.100.1: 56 data bytes
	64 bytes from 100.100.100.100: icmp_seq=1 ttl=64 time=0.692 ms
	--- ping statistics ---
	1 packets transmitted, 1 packets received, 0.0% packet loss
	rtt min/avg/max/stddev = 0.692/0.692/0.692/0.000 ms
	Neighbor status and routes

<table>
<tr>
<td></td>
<td>
<pre>
BGP neighbor is 200.200.200.1, remote AS 200, local AS 100, external link
  BGP version 4, remote router ID 192.0.0.5, local router ID 200.200.200.200
  BGP state = Established, up for 00:45:14
  Last read 00:00:26, Last write 00:00:14
  Hold time is 90, keepalive interval is 30 seconds
  Neighbor capabilities:
    4 Byte AS: advertised
    Extended Message: advertised
    AddPath:
      IPv4 Unicast: RX advertised
    Route refresh: advertised
    Enhanced Route Refresh: advertised
    Address Family IPv4 Unicast: advertised and received
    Hostname Capability: advertised (name: sonic,domain name: n/a) not received
    Graceful Restart Capability: advertised
  Graceful restart information:
    Local GR Mode: Helper*
    Remote GR Mode: Disable
    R bit: False
    Timers:
      Configured Restart Time(sec): 120
      Received Restart Time(sec): 0
  Message statistics:
    Inq depth is 0
    Outq depth is 0
                         Sent   Rcvd
    Opens: 1   1
    Notifications: 0   0
    Updates: 2   1
    Keepalives: 91   91
    Route Refresh: 0   0
    Capability: 0   0
    Total: 94   93
  Minimum time between advertisement runs is 0 seconds

 For address family: IPv4 Unicast
  Update group 1, subgroup 1
  Packet Queue length 0
  Community attribute sent to this neighbor(all)
  2 accepted prefixes

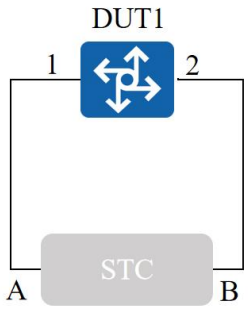
  Connections established 1; dropped 0
  Last reset 00:46:51, No AFI/SAFI activated for peer
Local host: 200.200.200.200, Local port: 179
Foreign host: 200.200.200.1, Foreign port: 53096
Nexthop: 200.200.200.200
Nexthop global: fe80::8e5d:b2ff:feb6:0
Nexthop local: fe80::8e5d:b2ff:feb6:0
BGP connection: shared network
BGP Connect Retry Timer in Seconds: 120
Read thread: on Write thread: on FD used: 27
</pre>


<pre>
sonic# show ip route bgp
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure

B>* 2.2.2.0/24 [20/0] via 200.200.200.1, Ethernet416, weight 1, 00:16:19
B>* 2.2.3.0/24 [20/0] via 200.200.200.1, Ethernet416, weight 1, 00:16:19
sonic#
</pre>
Packet forwarding result


</td>
</tr>
<tr>
<td>Test results</td>
<td>PASS</td>
</tr>
<tr>
<td>Remarks</td>
<td></td>
</tr>
</table>

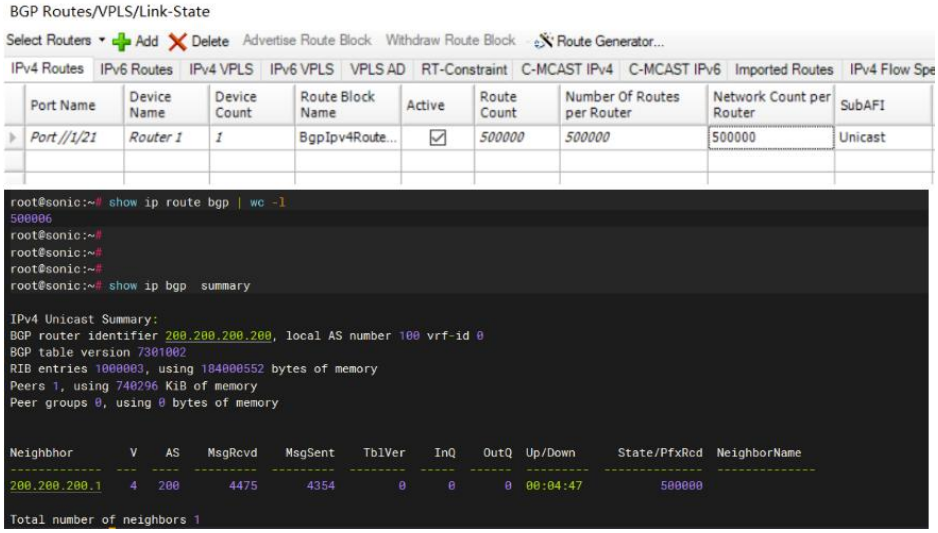
4.5 IPv4 Route table specifications

Test Items	IPv4 Route table specifications
Test Content	IPv4 Route table specifications Test
Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2、Configure DUT port P1 with IP 100.100.100.100/24 and port P2 with IP 200.200.200.200/24. 3、Configure DUT BGP AS100; advertise networks 100.100.100.100/24 and 200.200.200.200/24; set neighbor 200.200.200.1 (AS200). 4、TGB (200.200.200.1/24) establishes an eBGP neighbor with the device and advertises 1,100,000 routes, Check 1.
Expected Results	Check 1: View the maximum number of directly connected routes that BGP can learn.
Actual Test Results	Configure device Port 1 with IP address 100.100.100.100/24 and Port 2 with IP address 200.200.200.200/24. config interface ip add Ethernet0 100.100.100.100/24 config interface ip add Ethernet8 200.200.200.200/24 root@sonic:~# show ip interfaces Interface Master IPv4 address/mask Admin/Oper BGP Neighbor Neighbor IP Ethernet384 100.100.100.100/24 up/up N/A N/A Ethernet416 200.200.200.200/24 up/up N/A N/A Configure DUT BGP vtysh configure terminal router bgp 100 network 100.100.100.0/24 network 200.200.200.0/24 neighbor 200.200.200.1 remote-as 200 no bgp ebgp-requires-policy

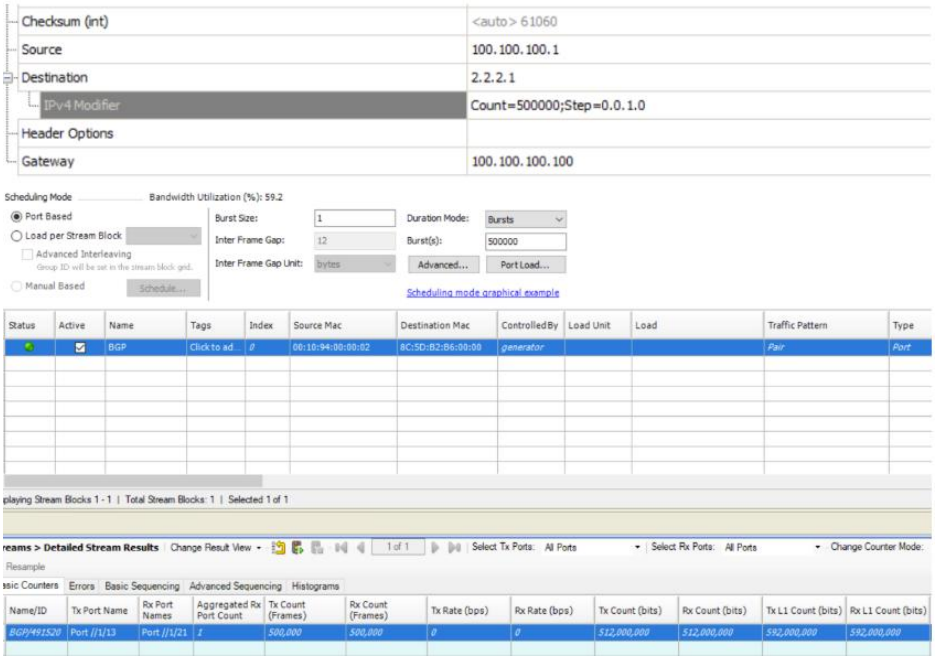
```
router bgp 100
 no bgp ebgp-requires-policy
 neighbor 200.200.200.1 remote-as 200
 !
 address-family ipv4 unicast
  network 100.100.100.0/24
  network 200.200.200.0/24
 exit-address-family
exit
```

BGP route learning count

Test by bursting 1,100,000 packets

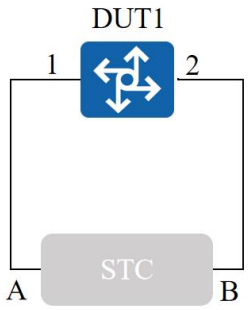


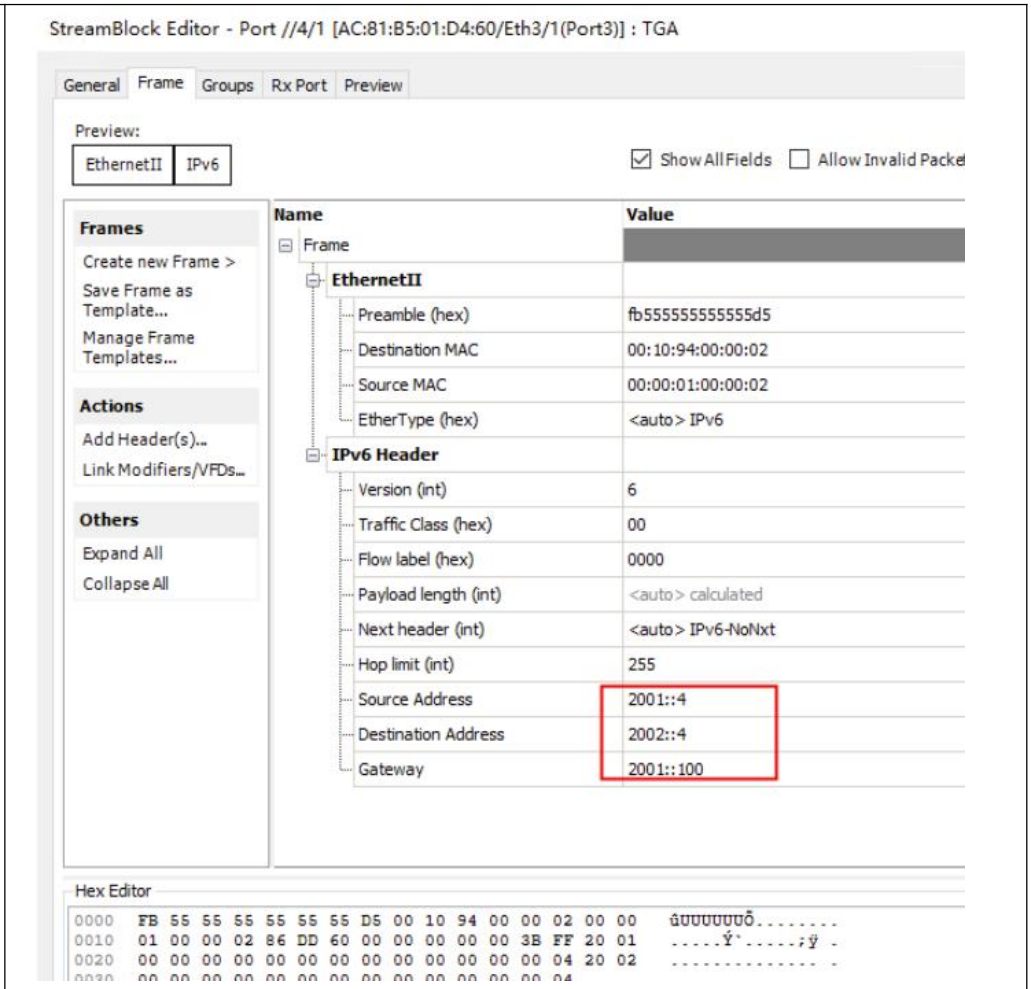

```
root@sonic:~# show ip route bgp | wc -l
500006
root@sonic:~#
root@sonic:~#
root@sonic:~#
root@sonic:~# show ip bgp  summary

IPv4 Unicast Summary:
BGP router identifier 200.200.200.200, local AS number 100 vrf-id 0
BGP table version 7301002
RIB entries 1000003, using 184000552 bytes of memory
Peers 1, using 740296 KiB of memory
Peer groups 0, using 0 bytes of memory

Neighbhor        V    AS    MsgRcvd    MsgSent    TblVer    InQ    OutQ  Up/Down      State/PfxRcd  NeighborName
---------------  ---  ----  ---------  ---------  --------  -----  ------  ---------  -------------  -------------
200.200.200.1    4    200       4475       4354         0      0       0  00:04:47           500000

Total number of neighbors 1
```



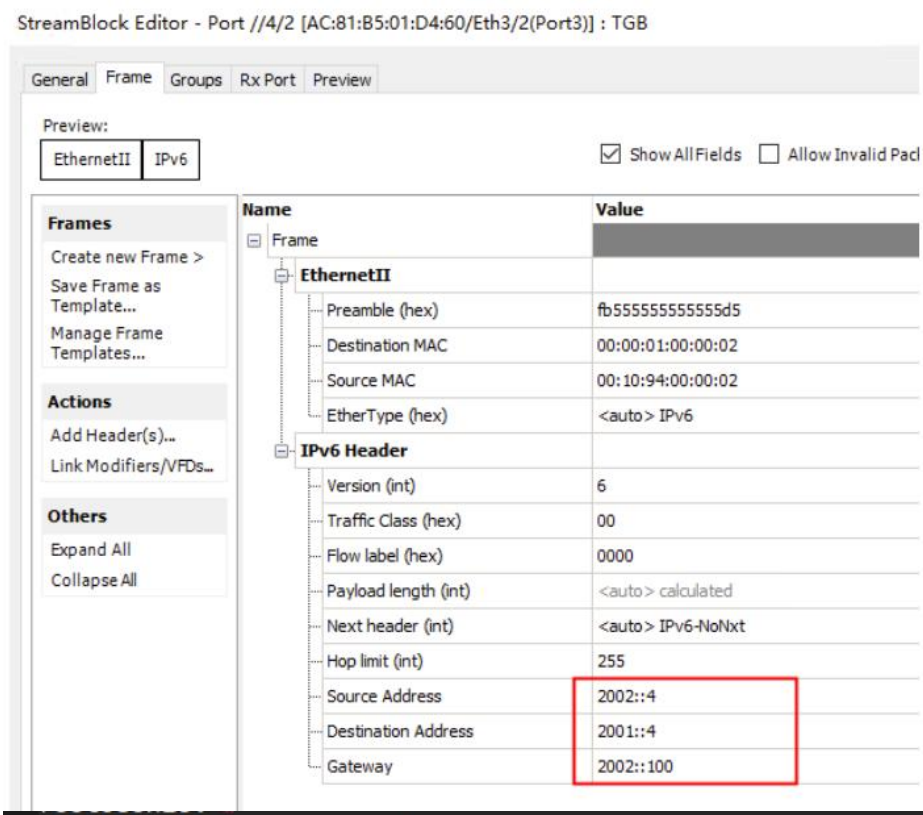
Test results	**PASS**
Remarks	The actual number of test-passed packets is 500,000.

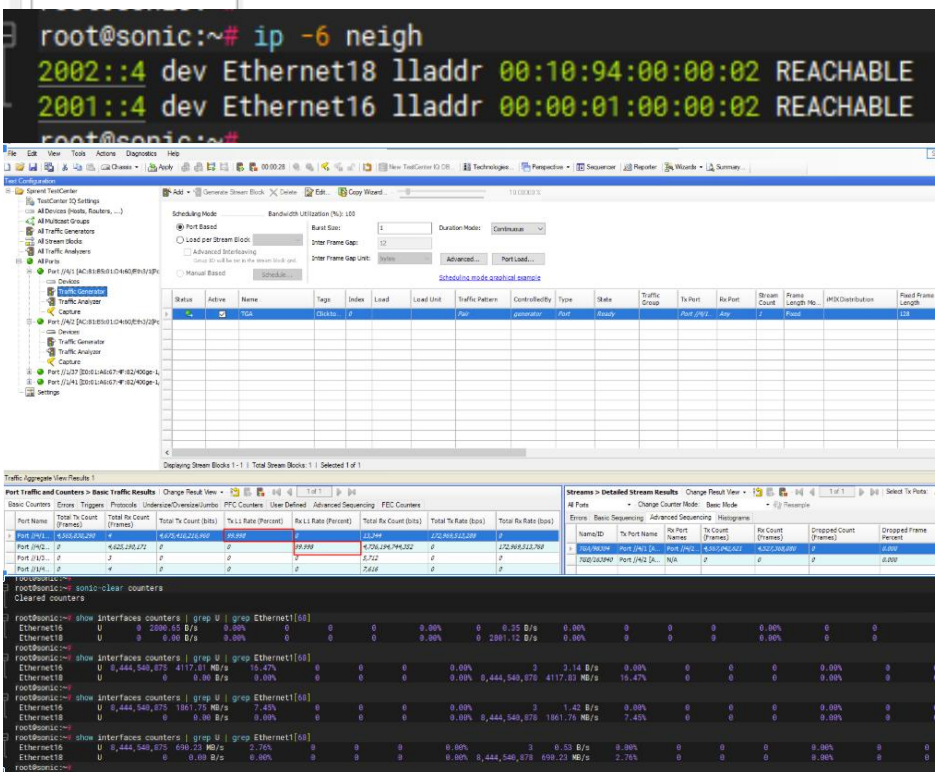
4.6 IPv6

Test Items	IPv6 Test
Test Content	Test that the device can support IPv6.
Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2、Configure IPv6 addresses on the interfaces，check1； 3、Tester TGA creates an IPv6 interface and interacts with device port P1. 4、Tester TGB creates an IPv6 interface and interacts with device port P2. 5、TGA sends L3 IPv6 packets to TGB，check5； 6、Delete the relevant configurations.
Expected Results	1、Query IPv6 information. 2、Data is forwarded correctly.
Actual Test Results	Configure IPv6 addresses config interface ip add Ethernet0 2001::100/64 config interface ip add Ethernet8 2002::100/64 show ipv6 interface ip -6 neigh  Port A


StreamBlock Editor - Port //4/1 [AC:81:B5:01:D4:60/Eth3/1(Port3)] : TGA
General
Frame
Groups
Rx Port
Preview
Preview:
EthernetII
IPv6
Show All Fields
Allow Invalid Packe
Frames
Create new Frame >
Save Frame as Template...
Manage Frame Templates...
Actions
Add Header(s)...
Link Modifiers/VFDs...
Others
Expand All
Collapse All
Name
Value
Frame
EthernetII
Preamble (hex)
fb5555555555555d5
Destination MAC
00:10:94:00:00:02
Source MAC
00:00:01:00:00:02
EtherType (hex)
<auto> IPv6
IPv6 Header
Version (int)
6
Traffic Class (hex)
00
Flow label (hex)
0000
Payload length (int)
<auto> calculated
Next header (int)
<auto> IPv6-NoNxt
Hop limit (int)
255
Source Address
2001::4
Destination Address
2002::4
Gateway
2001::100
Hex Editor
0000 FB 55 55 55 55 55 55 D5 00 10 94 00 00 02 00 00
0010 01 00 00 02 86 DD 60 00 00 00 00 00 3B FF 20 01
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 20 02

Port B

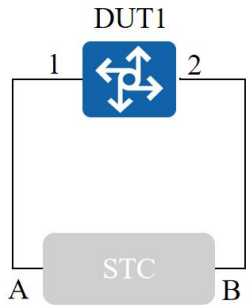




Test results	**PASS**

Remarks	

4.7 IPv6 Route table specifications

Test Items	IPv6 Route table specifications
Test Content	IPv6 Route table specifications Test
Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2 、 Configure DUT port 1 with IPv6 address 2001::100/64 and port 2 with 2002::100/64. 3、Configure DUT BGP AS100; set neighbor as 2002::1/64 (AS200). 4 、 TGB (2002::1/64) establishes an eBGP neighbor with the device and advertises 610,000 routes，check1； 5、Delete the relevant configurations.
Expected Results	Check1：View the maximum number of routes that BGP can learn.
Actual Test Results	Configure IP address config interface ip add Ethernet0 2001::100/64 config interface ip add Ethernet8 2002::200/64 Configure to accept IPv6 link-local addresses as neighbors. config interface ipv6 enable use-link-local-only Ethernet8 config interface ipv6 enable use-link-local-only Ethernet16 Configure DUT BGP vtysh configure terminal router bgp 100 neighbor 2002::1 remote-as 200 no bgp ebgp-requires-policy address-family ipv6 unicast neighbor 2002::1 activate network 2002::200/64

```
root@sonic:~#
root@sonic:~# ip -6 route show | wc -l
610029
root@sonic:~# show ipv6 bgp  summary

IPv6 Unicast Summary:
BGP router identifier 172.21.120.7, local AS number 100 vrf-id 0
BGP table version 610001
RIB entries 1220001, using 224480184 bytes of memory
Peers 1, using 740296 KiB of memory
Peer groups 0, using 0 bytes of memory


Neighbhor        V    AS    MsgRcvd    MsgSent    TblVer    InQ    OutQ  Up/Down      State/PfxRcd  NeighborName
-----------  ---  ----  ---------  ---------  --------  -----  -----  ---------  --------------  --------------
2002::1          4   200       2657       2600         0      0      0  00:00:46          610000

Total number of neighbors 1
root@sonic:~#
```

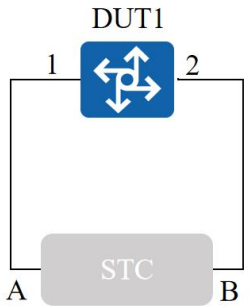
Send data flow to the learned route table entries.

Preamble (hex)	fb555555555555d5
Destination MAC	8C:5D:B2:B6:00:00
Source MAC	00:10:94:00:00:02
EtherType (hex)	<auto> IPv6
IPv6 Header	
Version (int)	6
Traffic Class (hex)	00
Flow label (hex)	0000
Payload length (int)	<auto> calculated
Next header (int)	<auto> IPv6-NoNxt
Hop limit (int)	255
Source Address	2001::2
Destination Address	3000::1
IPv6 Modifier	Count=610000;Step=::1
Gateway	::0

Test results	**Pass**
Remarks	Send 610,000 packets and receive 609,999 packets.

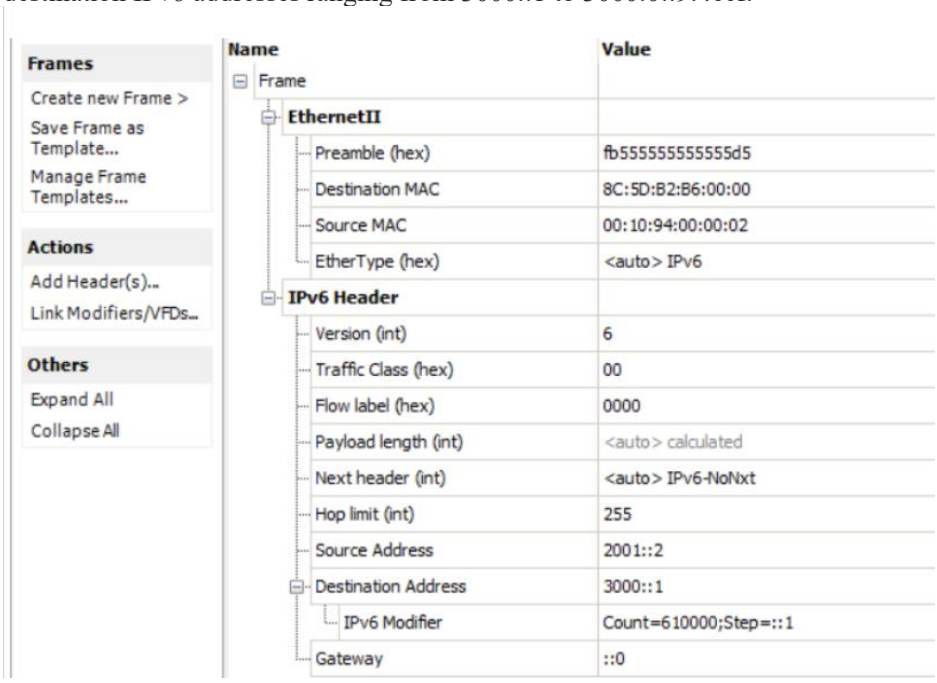
4.8 IPv6 Route table specifications-Global

Test Items	IPv6 Route table specifications
Test Content	IPv6 IPv6 Route table specifications Test

Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2、Configure DUT port P1 with IPv6 address 2001::100/64 and port P2 with 2002::100/64. 3、Configure the route map. 4、Configure DUT BGP AS100; advertise network 2002::1/64 (AS200) and bind the route map. 5、TGB (2002::1/64) establishes an eBGP neighbor with the device and advertises 610,000 routes. Check 1.
Expected Results	Check 1: View the maximum number of routes that BGP can learn.
Actual Test Results	Configure IP address config interface ip add Ethernet0 2001::100/64 config interface ip add Ethernet8 2002::200/64 Configure Route map vtysh configure terminal route-map FROM_BGP_PEER_V6 permit 1 on-match next set ipv6 next-hop prefer-global Configure BGP vtysh configure terminal router bgp 100 neighbor 2002::1 remote-as 200 no bgp ebgp-requires-policy no bgp default ipv4-unicast address-family ipv6 unicast neighbor 2002::1 activate neighbor 2002::1 soft-reconfiguration inbound neighbor 2002::1 route-map FROM_BGP_PEER_V6 in network 2002::200/64

```
root@sonic:~# show ipv6 bgp  summary

IPv6 Unicast Summary:
BGP router identifier 172.21.120.7, local AS number 100 vrf-id 0
BGP table version 610001
RIB entries 1220001, using 224480184 bytes of memory
Peers 1, using 740296 KiB of memory
Peer groups 0, using 0 bytes of memory

Neighbhor        V    AS   MsgRcvd   MsgSent   TblVer  InQ  OutQ  Up/Down      State/PfxRcd  NeighborName
2002::1          4   200      2658      2599        0    0     0  00:01:06           610000

Total number of neighbors 1
root@sonic:~#
root@sonic:~# ip -6 route show | wc -l
610029
root@sonic:~#
```

TGA sends 610,000 Layer 3 IPv6 packets to the eBGP routes advertised by TGB, with destination IPv6 addresses ranging from 3000::1 to 3000:0::9:4ecf.



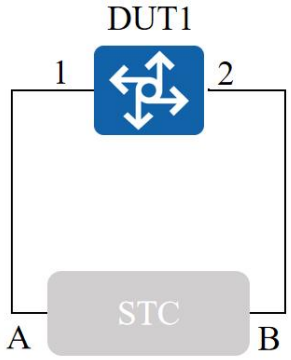
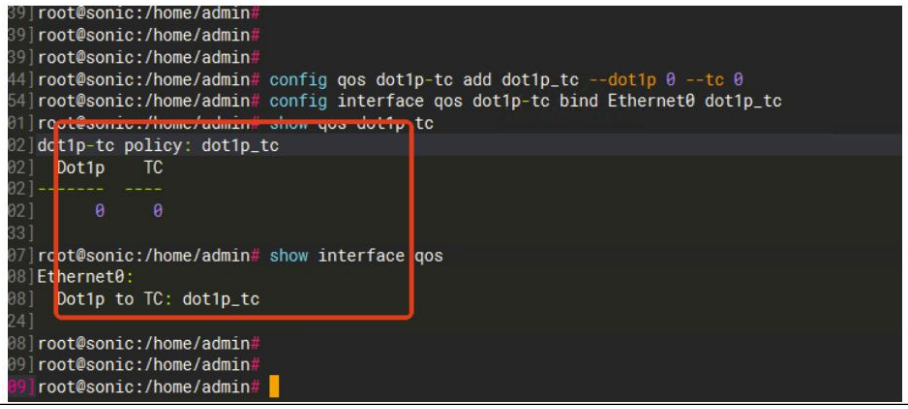
Packet reception

Test results	**PASS**
Remarks	ip -6 route show displays that 610,0014 routes are learned, and the traffic flow can be established.

五、QoS

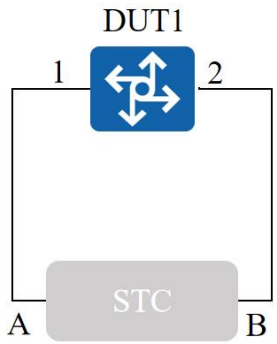
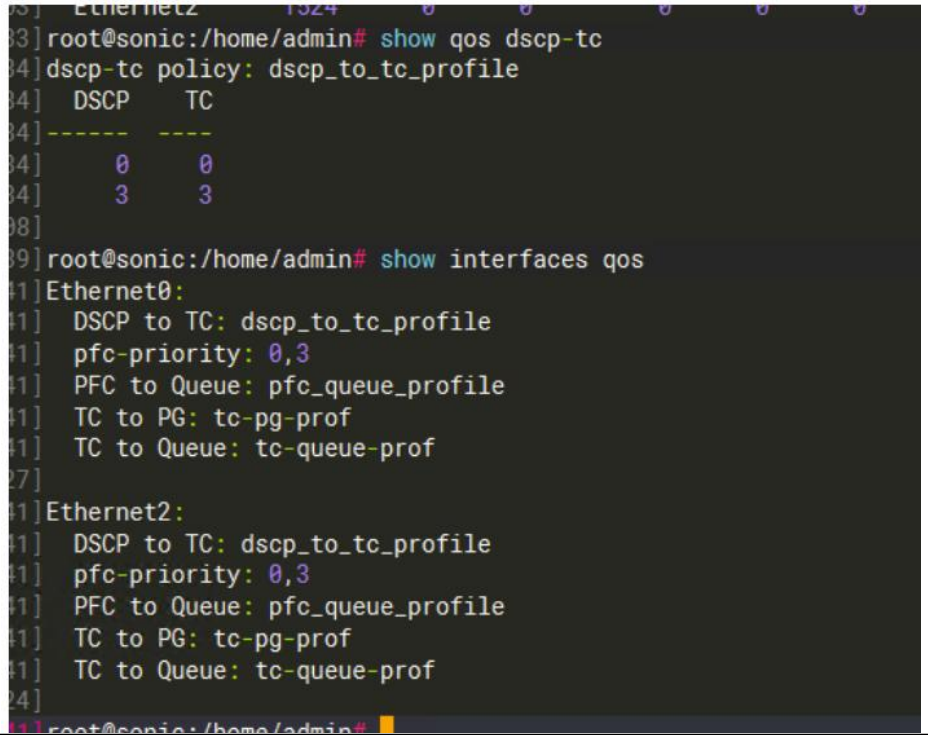
5.1 Classification-COS

Test Items	Classfication COS Test

Test Content	The testing device can classify according to the set Classfication COS and bind it to the ingress port.
Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2、Create a dot1p-tc and map dot1p to TC. 3、Bind the created dot1p-tc to the port, check1; 4、View the created dot1p-tc and its binding, check2.
Expected Results	Check1: Configuration is successful. Check2: The displayed content is consistent with the creation.
Actual Test Results	Configure dot1p-tc mapping and bind it to the interface. config qos dot1p-tc add dot1p_tc --dot1p 0 --tc 0 config interface qos dot1p-tc bind Ethernet0 dot1p_tc show qos dot1p-tc show interface qos 
Test results	**PASS**
Remarks	

5.2 Classification-DSCP

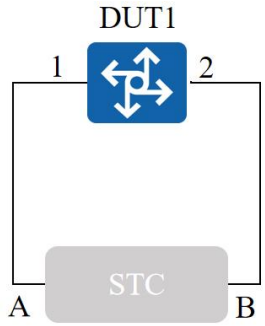
Test Items	Classfication DSCP Test
Test Content	The testing device can classify according to the set Classfication DSCP and bind it to

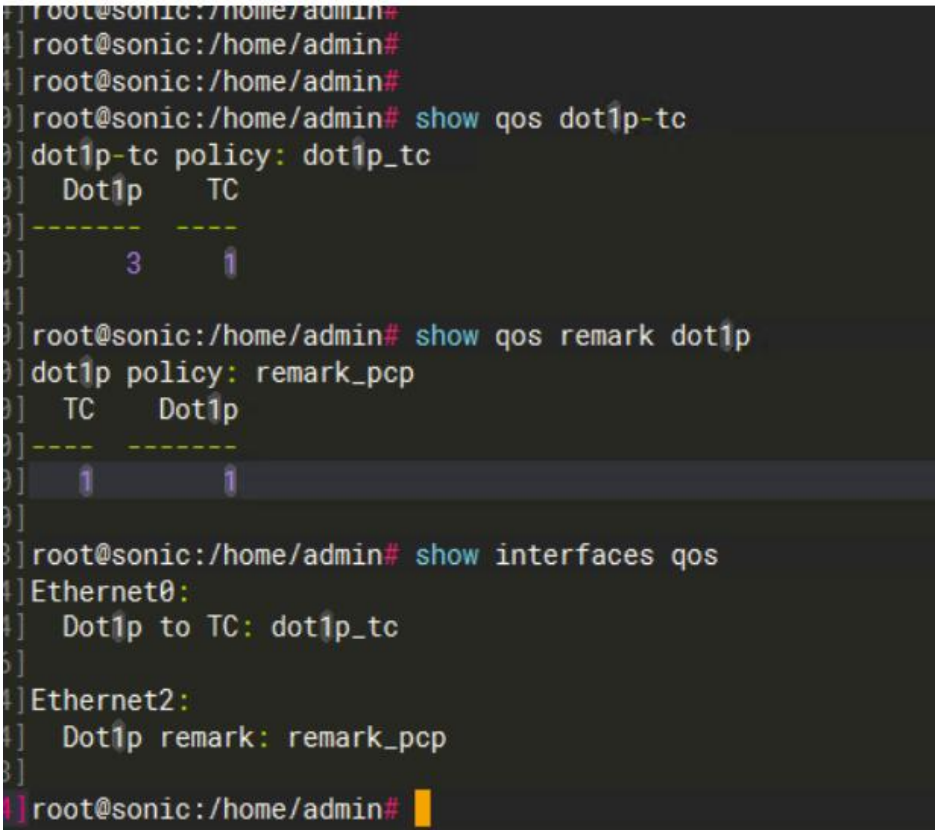
	the ingress port.
Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2、Create a dscp-tc mapping and map DSCP values to Traffic Classes (TC). 3、Bind the created dscp-tc mapping to the port, check1; 4、View the created dscp-tc mapping and its binding, check2.
Expected Results	Check1: Configuration successful. Check2: Displayed content matches the configuration.
Actual Test Results	config qos dscp-tc add dscp_to_tc_profile --dscp 0 --tc 1 config interface qos dscp-tc bind Ethernet0 dscp_to_tc_profile show qos dscp-tc show interface qos 
Test results	**PASS**

```
root@sonic:/home/admin# show qos dscp-tc
dscp-tc policy: dscp_to_tc_profile
  DSCP    TC
------  ----
     0     0
     3     3

root@sonic:/home/admin# show interfaces qos
Ethernet0:
  DSCP to TC: dscp_to_tc_profile
  pfc-priority: 0,3
  PFC to Queue: pfc_queue_profile
  TC to PG: tc-pg-prof
  TC to Queue: tc-queue-prof

Ethernet2:
  DSCP to TC: dscp_to_tc_profile
  pfc-priority: 0,3
  PFC to Queue: pfc_queue_profile
  TC to PG: tc-pg-prof
  TC to Queue: tc-queue-prof
```

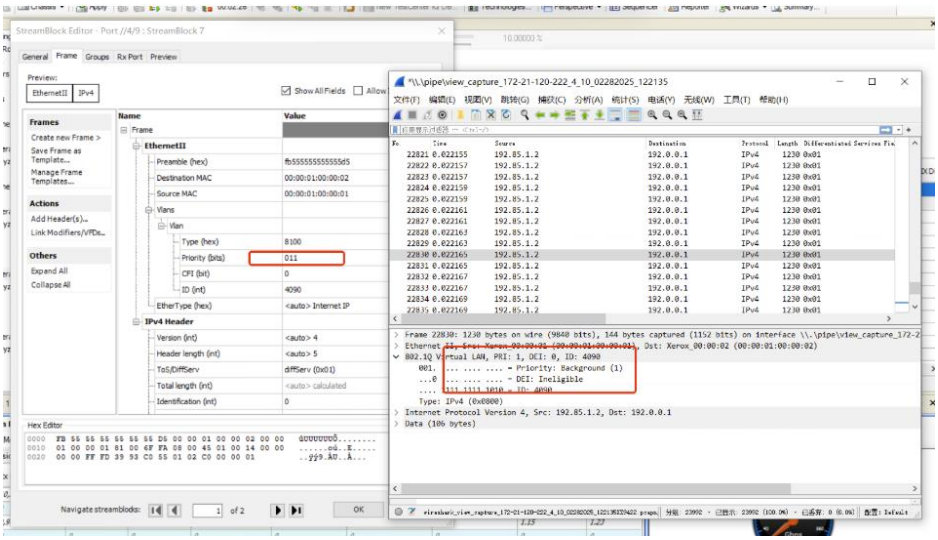
Remarks	

5.3 Marking-COS

Test Items	Marking COS Test
Test Content	The testing equipment can classify traffic according to the preset ingress Classification COS and directly translate it to the egress Marking COS.
Test Topology	
Test Steps	1、Set up the test environment according to the diagram. 2、Create a dot1p-tc mapping that maps dot1p 3 to TC 1. 3、Bind the created dot1p-tc mapping to port DUTP1. 4、Create a remark rule to remark TC 1 packets with dot1p 1. 5、Bind the created remark rule to port DUTP2 , check1. 6、View the created QoS rules and their bindings ,check2. 7、Create VLAN 100 and add ports DUTP1 and DUTP2 as tagged members. 8、 Send packets with VLAN Priority 3 from TGA.Capture packets at TGB to check the VLAN Priority value ,check3.
Expected Results	Check1: Configuration and binding are successful. Check2: The displayed QoS configuration matches the settings. Check3: The VLAN priority value in the packets received by TGB is changed to 1.
Actual Test Results	config qos dot1p-tc add dot1p_tc --dot1p 3 --tc 1 config interface qos dot1p-tc bind Ethernet0 dot1p_tc config qos remark dot1p add remark_pcp --tc 1 --dot1p 1 config interface qos remark dot1p bind Ethernet8 remark_pcp show qos dot1p-tc show qos remark dot1p show interface qos

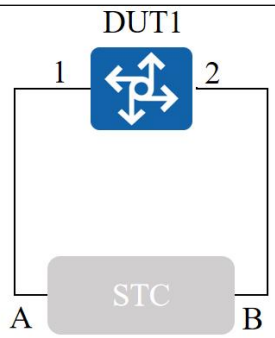


config vlan add 4090
config vlan member add 4090 Ethernet0
config vlan member add 4090 Ethernet2



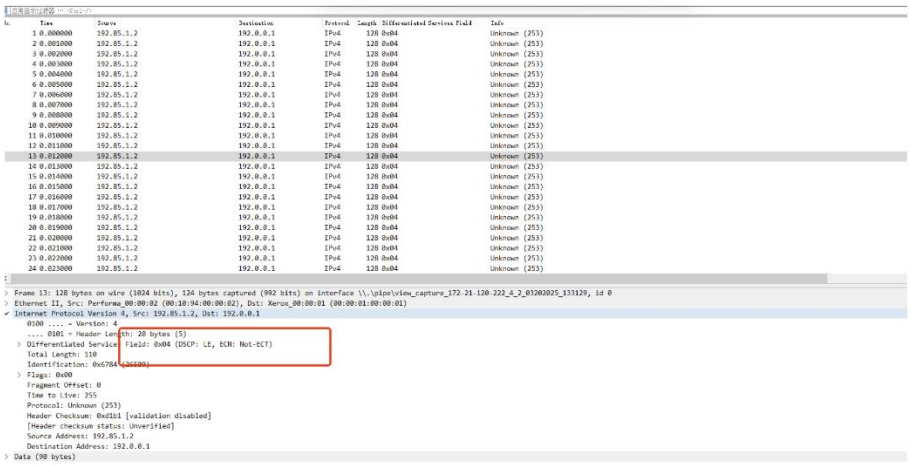
Test results	**PASS**
Remarks	

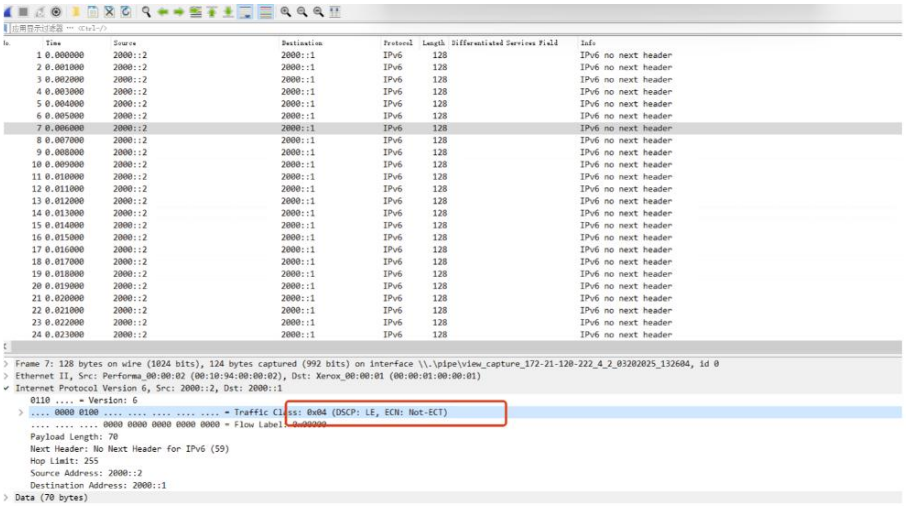
5.4 Marking-DSCP

Test Items	Marking DSCP Test
Test Content	The testing equipment can classify traffic according to the preset ingress Classification DSCP and directly translate it to the egress Marking DSCP.
Test Topology	
Test Steps	1、Set up the test environment according to the diagram. 2、Create a dscp-tc mapping that maps DSCP 0 to TC 3. 3、Bind the created dscp-tc mapping to port DUTP1. 4、Create a remark rule to remark TC 3 packets with DSCP 1. 5、Bind the created remark rule to port DUTP2， check1. 6、View the created QoS rules and their bindings ， check2. 7、Create VLAN 100 and add ports DUTP1 and DUTP2 as untagged members. 8、Send IPv4 packets with DSCP 0 from TGA. Capture packets at TGB to check the DSCP value. 9、Send IPv6 packets with Traffic Class 0 from TGA. Capture packets at TGB to check the Traffic Class value ,check3.
Expected Results	Check1: Configuration and binding are successful. Check2: The displayed QoS configuration matches the settings. Check3: The DSCP value of IPv4 packets and the Traffic Class value of IPv6 packets received by TGB are both changed to 0x04.
Actual Test Results	config qos dscp-tc add dscp_to_tc_profile --dscp 0 --tc 3 config interface qos dscp-tc bind Ethernet0 dscp_to_tc_profile config qos remark dscp add remark_dscp --tc 3 --dscp 1 config interface qos remark dscp bind Ethernet2 remark_dscp show qos dscp-tc show qos remark dscp show interface qos

```
root@sonic:/home/admin#
root@sonic:/home/admin# show qos dscp-tc
dscp-tc policy: dscp_to_tc_profile
  DSCP    TC
------  ----
     0     3

root@sonic:/home/admin# show qos remark dscp
dscp policy: remark_dscp
  TC    DSCP
----  ------
   3       1

root@sonic:/home/admin# show interface qos
Ethernet2:
  IP DSCP remark: remark_dscp

root@sonic:/home/admin#
```

config vlan add 100
config vlan member add 100 Ethernet0 -u
config vlan member add 100 Ethernet2 -u

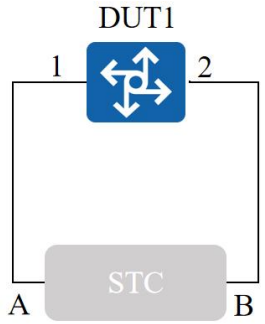




Test results	**PASS**

Remarks	

5.5 DiffServ

Test Items	Diffserv Functional Test
Test Content	The testing equipment can forward packets according to the DiffServ configuration.
Test Topology	
Test Steps	1、Set up the test environment according to the diagram. 2、Create VLAN 100 and add ports DUT P1 and DUTP2 as untagged members. 3、Create a diffserv-type ACL to match DSCP values and enable policer actions. 4、 Bind the previously created diffserv-type ACL to the ingress direction of the TGA port. 5、 Create a policer rule to modify DSCP values for packets of different colors (e.g., green, yellow, red). 6、Add the policer rule to the ACL ,Check1. 7、Send DSCP-matching traffic from TGA at a rate exceeding the PIR (Peak Information Rate). Verify behavior ,Check2. 8、Capture packets at TGB and check the DSCP values.
Expected Results	Check1: Configuration is successful. Check2: Packets are forwarded successfully, and DSCP values are modified according to the DiffServ settings.
Actual Test Results	config vlan add 100 config vlan member add 100 Ethernet0 -u config vlan member add 100 Ethernet8 -u Create an ACL of type diffserv-type that matches DSCP values and supports policer actions. config acl add table-type diffserv-type --match-dscp --action-set-policer show acl table-type root@sonic:/home/admin# config acl add table-type diffserv-type --match-dscp --action-set-policer root@sonic:/home/admin# show acl table-type Name Bind Point Match Policer ---------- ---------- ------- --------- diffserv-type PORT DSCP Yes

IP type
root@sonic:/home/admin# config acl add table diffserv-table diffserv-type -s ingress -p Ethernet68
root@sonic:/home/admin# show acl table
Name Type Binding Description Stage Policer
---------- ---------- ---------- ---------- ------- ---------
diffserv-table diffserv-type Ethernet68 diffserv-table ingress Yes

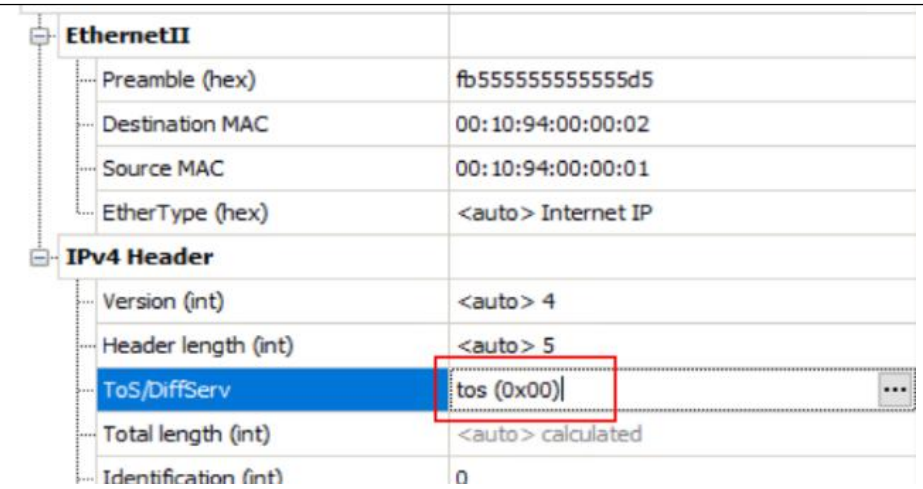
root@sonic:/home/admin# config policer add pol tr_tcm packets blind --cir 100 --cbs 200 --pir 300 --pbs 400 --green_set_dscp 0 --yellow_set_dscp 20 --red_set_dscp 30
root@sonic:/home/admin# show policer
Name Type Mode Color Source CIR CBS PIR PBS Green Packet Action Yellow Packet Action Red Packet Action
------ ------- ------ ---------- ----- ----- ----- ----- ---------- ---------- ----------
pol packets tr_tcm blind 100 200 300 400 DSCP: 0 DSCP: 20 DSCP: 30

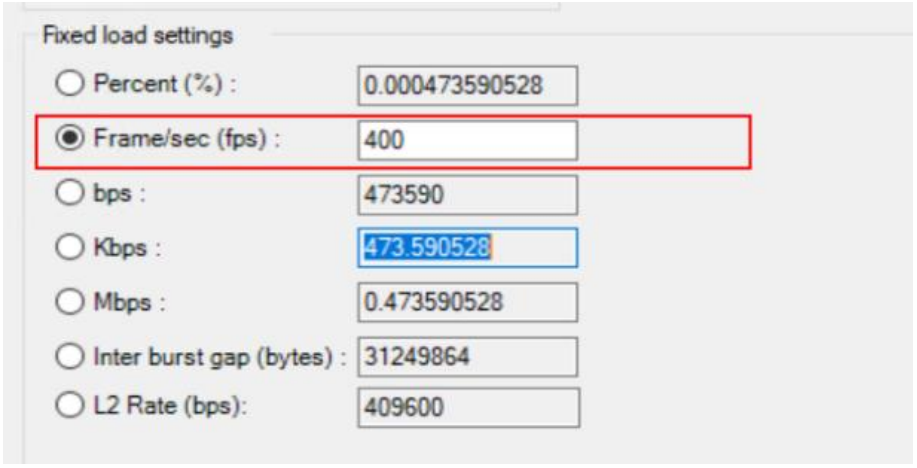
root@sonic:/home/admin# config acl add rule diffserv-table permit --priority 10000 --dscp 0 --policer pol
root@sonic:/home/admin# show policer
Name Type Mode Color Source CIR CBS PIR PBS Green Packet Action Yellow Packet Action Red Packet Action
------ ------- ------ ---------- ----- ----- ----- ----- ---------- ---------- ----------
pol packets tr_tcm blind 100 200 300 400 DSCP: 0 DSCP: 20 DSCP: 30

root@sonic:/home/admin# show acl rule
Table Rule Priority Action Match Status
---------- ------ ---------- ---------- ---------- --------
diffserv-table 10000 10000 FORWARD DSCP: 0 Active
POLICER: pol IP_TYPE: IP
root@sonic:/home/admin# show vlan brief
+-----------+-------------+------------+-----------+----------+--------------+-------------------+
| VLAN ID | IP Address | Ports | Port | Proxy | DHCP Helper | DHCP Relay Configuration |
| | | | Tagging | ARP | Address | |
+===========+=============+============+===========+==========+==============+===================+
100		Ethernet68	untagged	disabled		Source Interface:
		Ethernet70	untagged			Link Selection:
						Server Vrf:
						Server ID Override:
+-----------+-------------+------------+-----------+----------+--------------+-------------------+

root@sonic:/home/admin# show acl table
Name Type Binding Description Stage Policer
---------- ---------- ---------- ---------- ------- ---------
diffserv-table diffserv-type Ethernet68 diffserv-table ingress Yes
config policer add pol tr_tcm packets blind --cir 100 --cbs 200 --pir 300 --pbs 400 --green_set_dscp 0 --yellow_set_dscp 20 --red_set_dscp 30
config acl add rule diffserv-table permit --priority 10000 --dscp 0 --policer pol
show policer

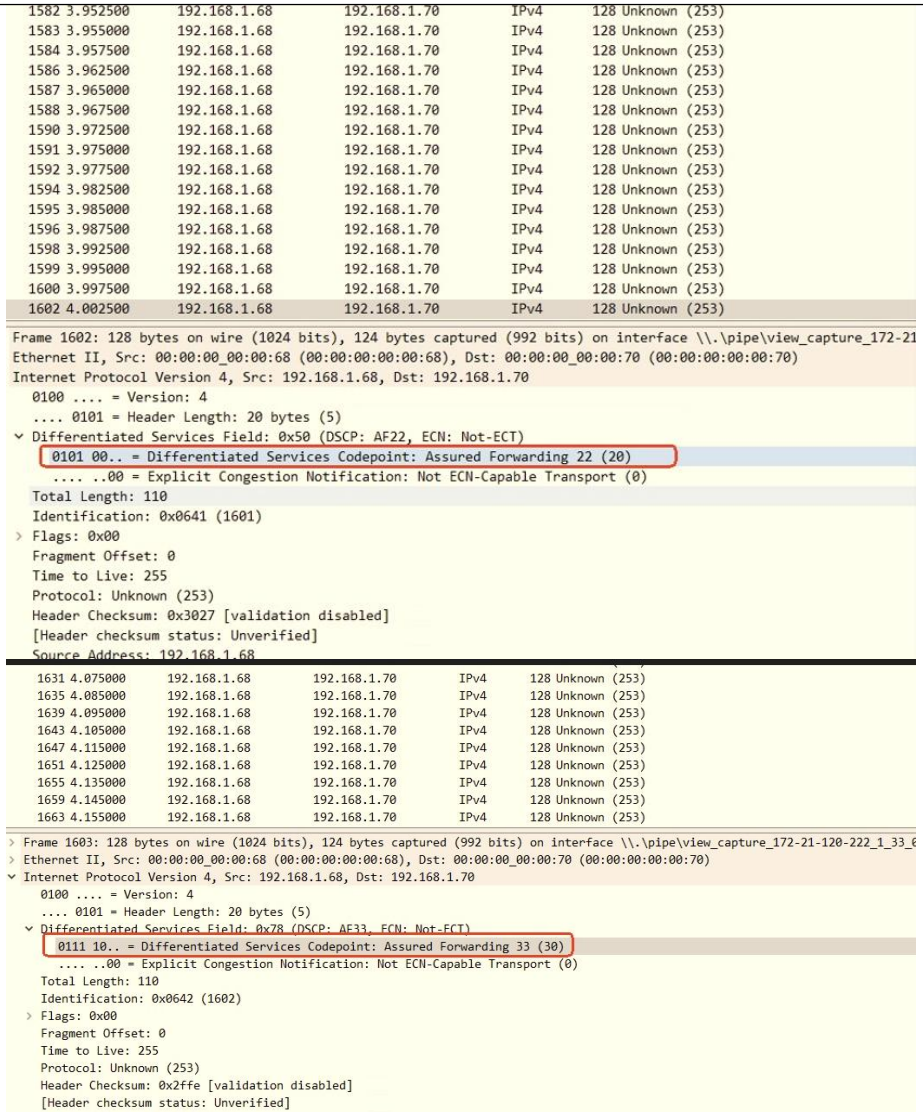
TGA sends packets with DSCP 0 at a rate of 400 packets per second.





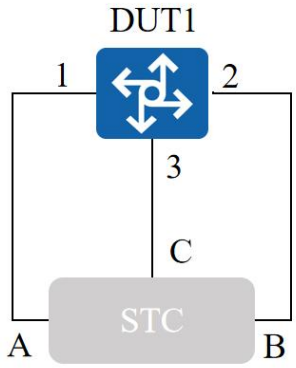
TGB captures the received packets and checks the DSCP settings:
Green packets: DSCP 0
Yellow packets: DSCP 20
Red packets: DSCP 30

```
Frame 299: 128 bytes on wire (1024 bits), 124 bytes captured (992 bits) on interface \\.\pipe\view_c
Ethernet II, Src: Performa_00:00:01 (00:10:94:00:00:01), Dst: Performa_00:00:02 (00:10:94:00:00:02)
Internet Protocol Version 4, Src: 192.85.1.2, Dst: 192.0.0.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 110
```

	
Test results	PASS
Remarks	

5.6 PFC (Priority Flow Control)

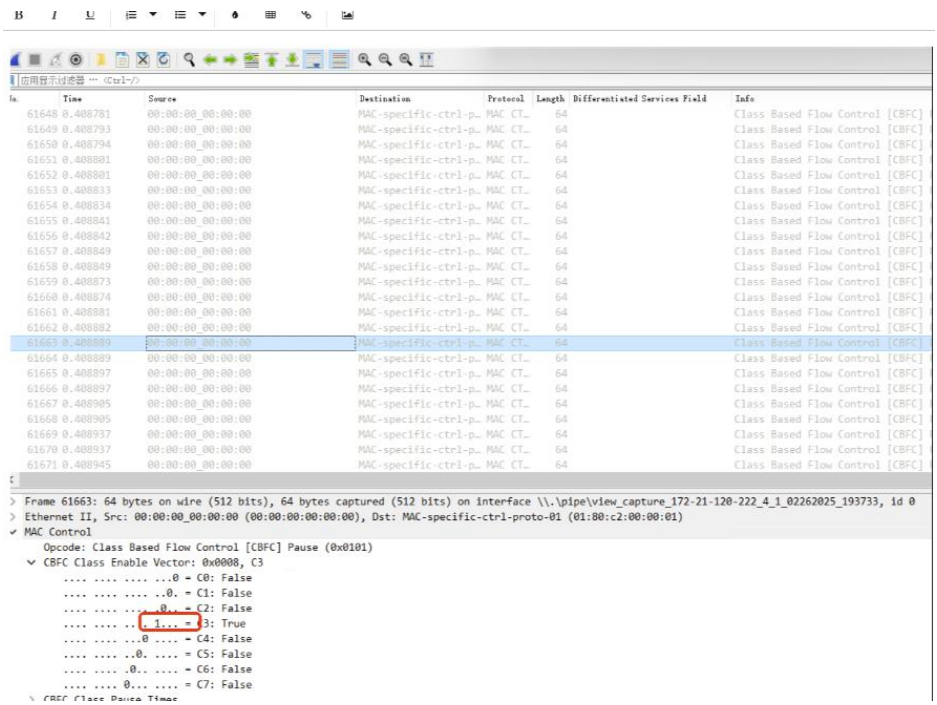
Test Items	PFC Functional Test
Test Content	The testing equipment's PFC (Priority Flow Control) can trigger PFC packets and stop traffic based on PFC priorities.
Test Topology	

	
Test Steps	1、Set up the test environment according to the diagram. 2、Map dot1p priority 3 to tc3 and pg3. 3、Bind the QoS mapping to the port. 4、Enable PFC priority 3 on DUTP1 and DUTP2. 5、Create VLAN100 and add DUT P1-P3 as tagged members. 6、TGB and TGC send bidirectional VLAN tagged traffic, with TGC sending at 100% rate to TGB. 7、TGA sends line-rate VLAN100 packets with priority 3 to TGB (check1). 8、TGA captures packets and confirms PFC priority 3 packets sent by the device (check2). 9、Stop traffic transmission on all ports. 10、Create pfc-priority-queue to map PFC priority 3 to queue3. 11、Bind the created pfc-priority-queue to TGB. 12、TGA sends line-rate VLAN100 packets with priority 3 to TGB. 13、TGB sends PFC priority 3 packets (check3).
Expected Results	Check1: Configuration is successful. Check2: The device sends PFC priority 3 packets. Check3: TGB stops receiving VLAN100 packets with priority 3 sent from TGA.
Actual Test Results	Configure QOS config qos dot1p-tc add dot1p_to_tc_profile --dot1p 3 --tc 3 config interface qos dot1p-tc bind Ethernet0 dot1p_to_tc_profile config interface qos dot1p-tc bind Ethernet8 dot1p_to_tc_profile config interface qos dot1p-tc bind Ethernet16 dot1p_to_tc_profile config qos tc-pg add tc-pg-prof --tc 3 --pg 3 config interface qos tc-pg bind Ethernet0 tc-pg-prof config interface qos tc-pg bind Ethernet8 tc-pg-prof config interface qos tc-pg bind Ethernet16 tc-pg-prof config qos tc-queue add tc-queue-prof --tc 3 --queue 3 config interface qos tc-queue bind Ethernet0 tc-queue-prof config interface qos tc-queue bind Ethernet8 tc-queue-prof config interface qos tc-queue bind Ethernet16 tc-queue-prof config interface pfc priority Ethernet0 3 on config interface pfc priority Ethernet8 3 on Configure vlan

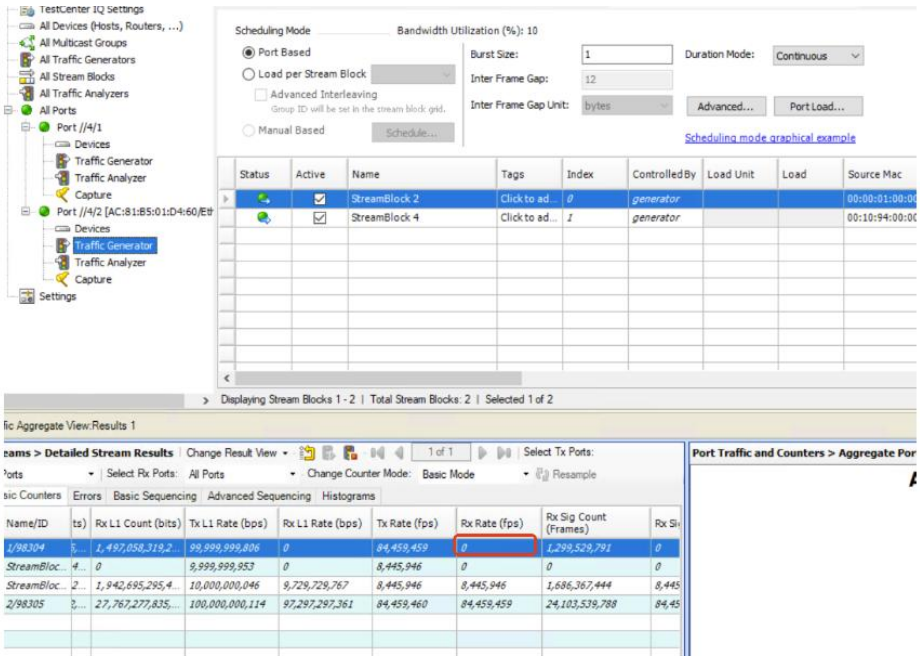
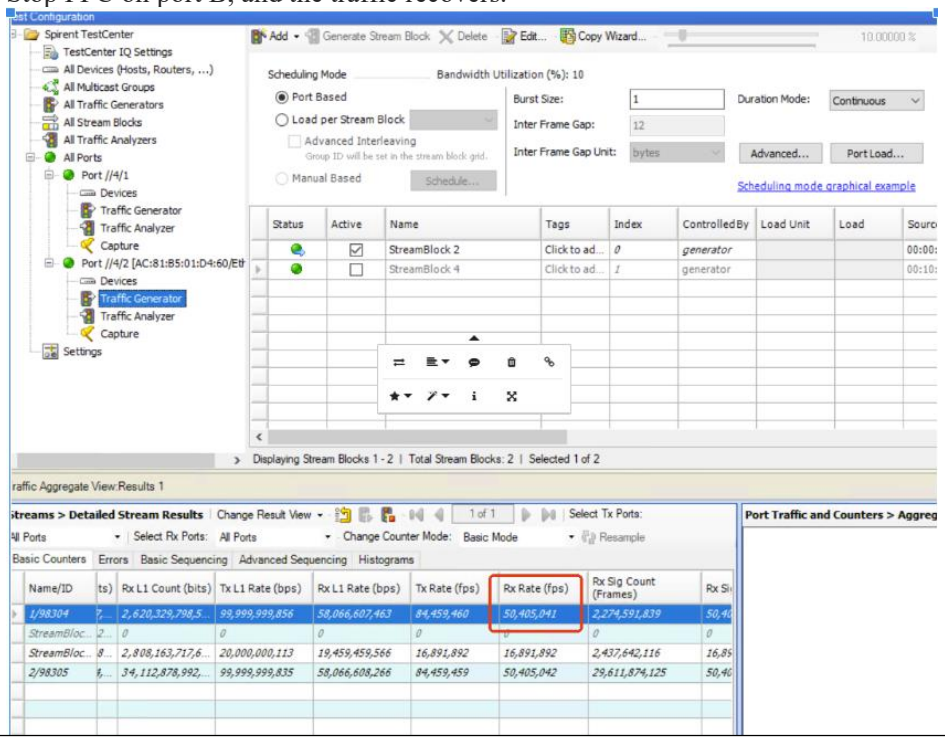
config vlan add 100
config vlan member add 100 Ethernet0
config vlan member add 100 Ethernet8
config vlan member add 100 Ethernet16

```
oot@sonic:/home/admin# config qos reload

unning command: /usr/local/bin/sonic-cfggen  -d --write-to-db -t /usr/share/sonic/device/x86_64-naddod_n9500_64oc
addod-N9500-64OC/qos.json.j2,config-db -y /etc/sonic/sonic_version.yml
onfig interface buffer bind priority-group Ethernet20 0 ingress_lossless_profile
oot@sonic:/home/admin# config interface buffer bind priority-group Ethernet22 0 ingress_lossless_profile
oot@sonic:/home/admin#
oot@sonic:/home/admin# config interface buffer bind queue Ethernet20 0 egress_lossless_profile
oot@sonic:/home/admin# config interface buffer bind queue Ethernet22 0 egress_lossless_profile
oot@sonic:/home/admin#
oot@sonic:/home/admin#
oot@sonic:/home/admin# config interface buffer bind priority-group Ethernet20 3 ingress_lossless_profile
oot@sonic:/home/admin# config interface buffer bind priority-group Ethernet22 3 ingress_lossless_profile
oot@sonic:/home/admin#
oot@sonic:/home/admin# config interface buffer bind queue Ethernet20 3 egress_lossless_profile
oot@sonic:/home/admin# config interface buffer bind queue Ethernet22 3 egress_lossless_profile
oot@sonic:/home/admin#
oot@sonic:/home/admin#
oot@sonic:/home/admin# config qos dot1p-tc add dot1p_to_tc_profile --dot1p 3 --tc 3
oot@sonic:/home/admin# config qos tc-pg add tc-pg-prof --tc 3 --pg 3
oot@sonic:/home/admin# config qos tc-queue add tc-queue-prof --tc 3 --queue 3
oot@sonic:/home/admin#
oot@sonic:/home/admin#
oot@sonic:/home/admin# config interface qos dot1p-tc bind Ethernet20 dot1p_to_tc_profile
oot@sonic:/home/admin# config interface qos dot1p-tc bind Ethernet22 dot1p_to_tc_profile
oot@sonic:/home/admin#
oot@sonic:/home/admin# config interface qos tc-pg bind Ethernet20 tc-pg-prof
oot@sonic:/home/admin# config interface qos tc-pg bind Ethernet22 tc-pg-prof
oot@sonic:/home/admin#
oot@sonic:/home/admin# config interface qos tc-queue bind Ethernet20 tc-queue-prof
oot@sonic:/home/admin# config interface qos tc-queue bind Ethernet22 tc-queue-prof
```

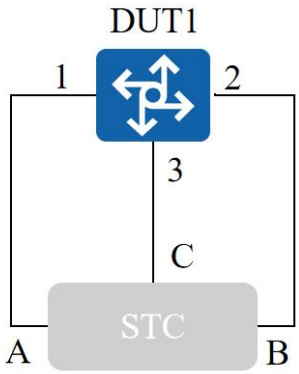
TGB and TGC send bidirectional VLAN tagged data flows, with TGC sending traffic to TGB at a rate of 100%.TGA sends line-rate data packets of VLAN 100 with priority 3 to TGB.TGA captures packets and can see the PFC priority 3 packets sent by the device.



pfc-priority-queue
config qos pfc-priority-queue add pfc_queue_profile --pfc-priority 3 --queue 3
config interface qos pfc-priority-queue bind Ethernet8 pfc_queue_profile

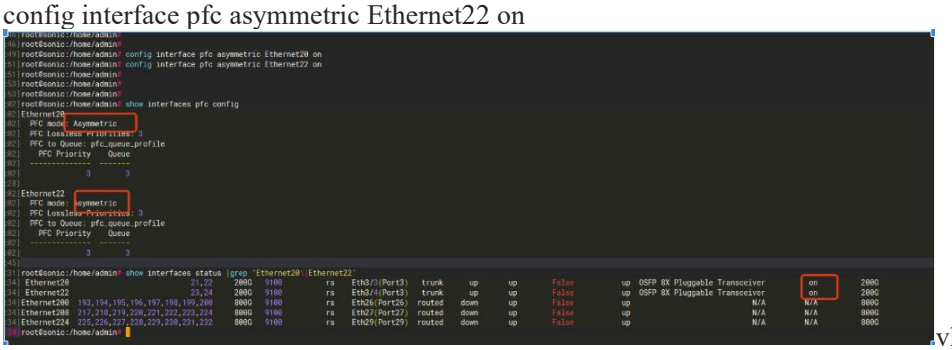
	Stop packet transmission on all ports.TGA sends line-rate VLAN 100 packets with priority 3 to TGB.TGB sends destination learning packets and PFC priority 3 packets.  Stop PFC on port B, and the traffic recovers. 
Test results	**PASS**
Remarks	

5.7 Asym PFC

Test Items	Asym PFC Functional Test
Test Content	The testing equipment's Asym PFC (Asymmetric Priority Flow Control) can correctly trigger PFC packets and stop traffic based on PFC priorities.
Test Topology	
Test Steps	1、Set up the test environment according to the diagram. 2、Map dot1p priority 3 to tc3 and pg3. 3、Bind the QoS mapping to the port. 4、Enable PFC priority 3 and Asym PFC on DUTP1 and DUTP2. 5、Create VLAN100 and add DUT P1-P3 as tagged members. 6、TGB and TGC send bidirectional VLAN tagged traffic, with TGC sending at 100% rate to TGB ， check1. 7、TGA sends line-rate VLAN100 packets with priority 3 to TGB. 8、TGA captures packets and confirms PFC priority 3 packets sent by the device ， check2. 9、Stop traffic transmission on all ports. 10、Create pfc-priority-queue to map PFC priority 3 to queue3. 11、Bind the created pfc-priority-queue to TGB. 12、TGA sends line-rate VLAN100 packets with priority 3 to TGB. 13、TGB sends PFC priority 3 packets ,check3.
Expected Results	Check1: Configuration is successful. Check2: The device sends PFC priority 3 packets. Check3: TGB stops receiving VLAN100 priority 3 packets sent from TGA.
Actual Test Results	Configure QOS config qos dot1p-tc add dot1p_to_tc_profile --dot1p 3 --tc 3 config interface qos dot1p-tc bind Ethernet0 dot1p_to_tc_profile config interface qos dot1p-tc bind Ethernet8 dot1p_to_tc_profile config interface qos dot1p-tc bind Ethernet16 dot1p_to_tc_profile config qos tc-pg add tc-pg-prof --tc 3 --pg 3

config interface qos tc-pg bind Ethernet0 tc-pg-prof
config interface qos tc-pg bind Ethernet8 tc-pg-prof
config interface qos tc-pg bind Ethernet16 tc-pg-prof
config qos tc-queue add tc-queue-prof --tc 3 --queue 3
config interface qos tc-queue bind Ethernet0 tc-queue-prof
config interface qos tc-queue bind Ethernet8 tc-queue-prof
config interface qos tc-queue bind Ethernet16 tc-queue-prof

config interface pfc priority Ethernet20 3 on
config interface pfc priority Ethernet22 3 on
config interface pfc asymmetric Ethernet20 on
config interface pfc asymmetric Ethernet22 on

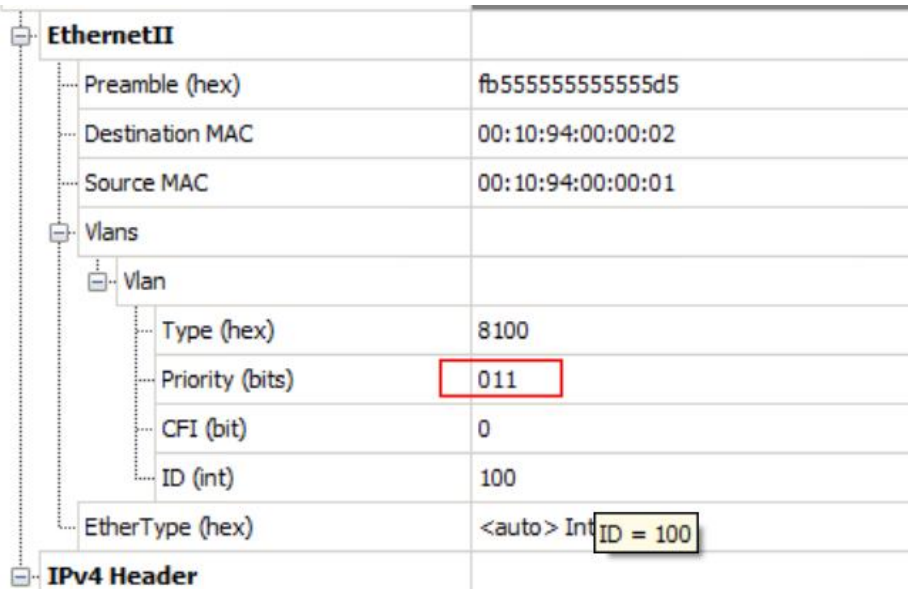


vlan

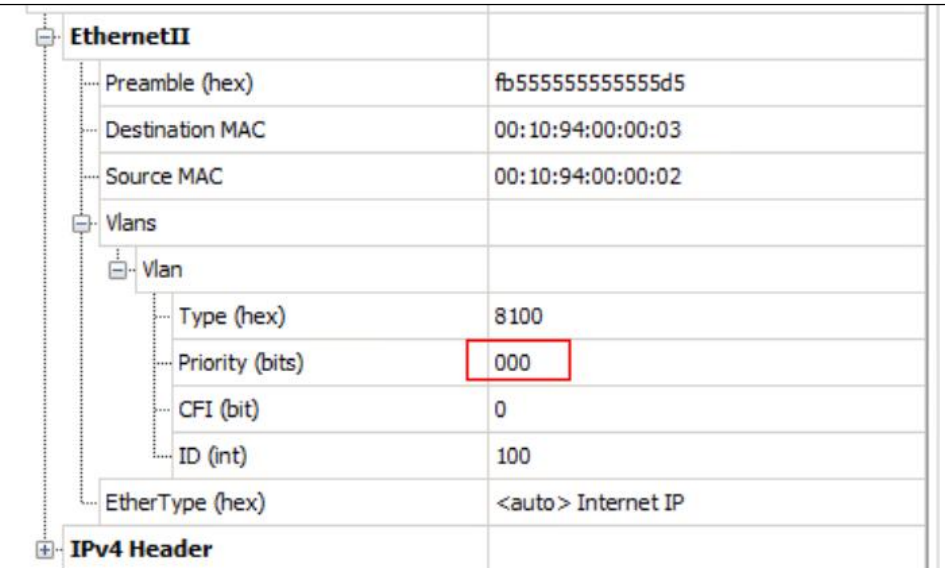
config vlan add 100
config vlan member add 100 Ethernet20
config vlan member add 100 Ethernet22

Rate limit port 22.
TGB and TGC send bidirectional VLAN tagged traffic, with TGC sending traffic to TGB at 100% rate.
TGA sends line-rate VLAN 100 packets with priority 3 to TGB.
TGA captures packets and can see the PFC priority 3 packets sent by the device.

Port A

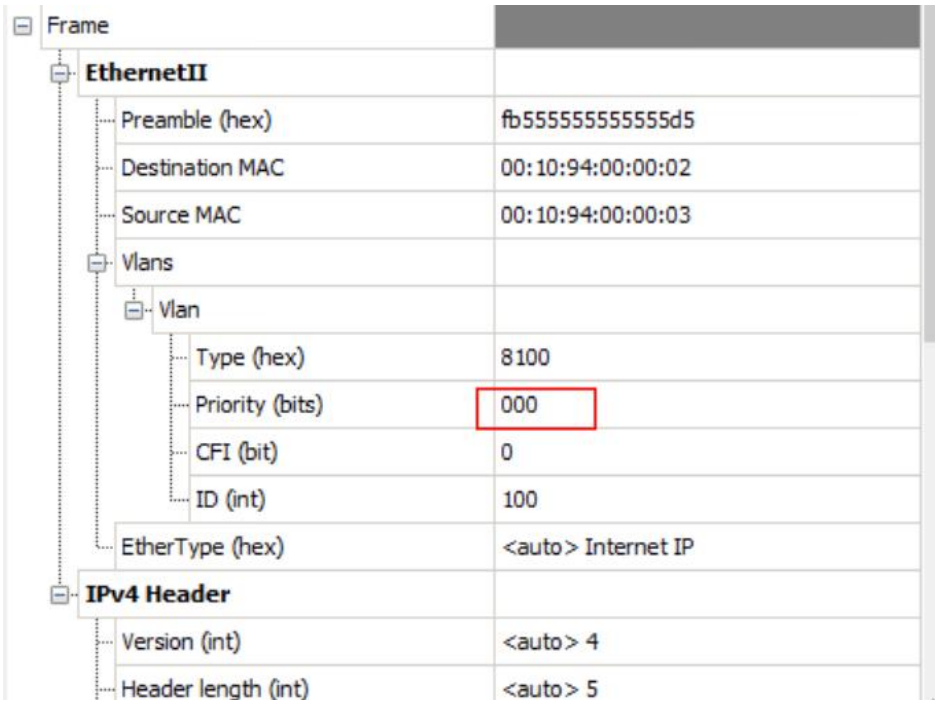


Port B

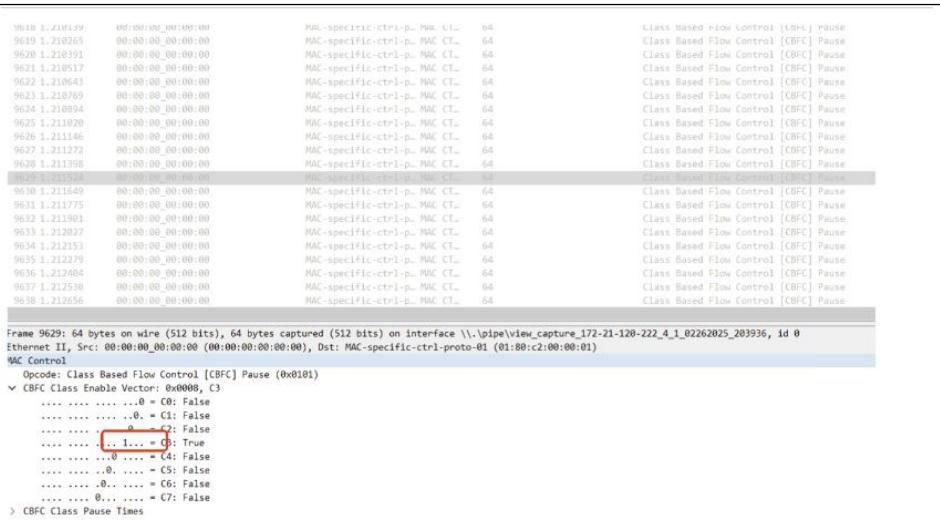


EthernetII	
Preamble (hex)	fb555555555555d5
Destination MAC	00:10:94:00:00:03
Source MAC	00:10:94:00:00:02
Vlans	
Vlan	
Type (hex)	8100
Priority (bits)	000
CFI (bit)	0
ID (int)	100
EtherType (hex)	<auto> Internet IP
IPv4 Header	

Port C



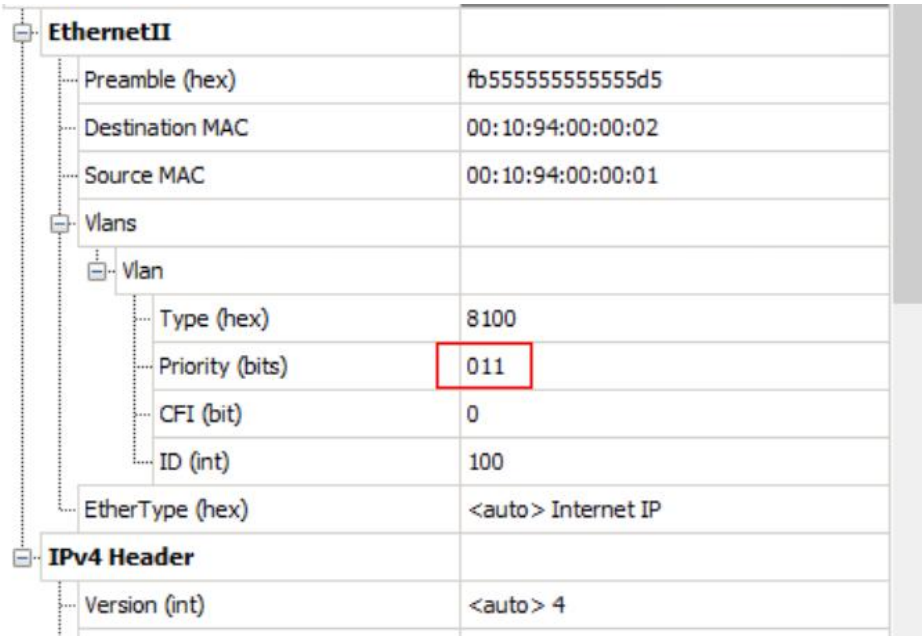
Frame	
EthernetII	
Preamble (hex)	fb555555555555d5
Destination MAC	00:10:94:00:00:02
Source MAC	00:10:94:00:00:03
Vlans	
Vlan	
Type (hex)	8100
Priority (bits)	000
CFI (bit)	0
ID (int)	100
EtherType (hex)	<auto> Internet IP
IPv4 Header	
Version (int)	<auto> 4
Header length (int)	<auto> 5



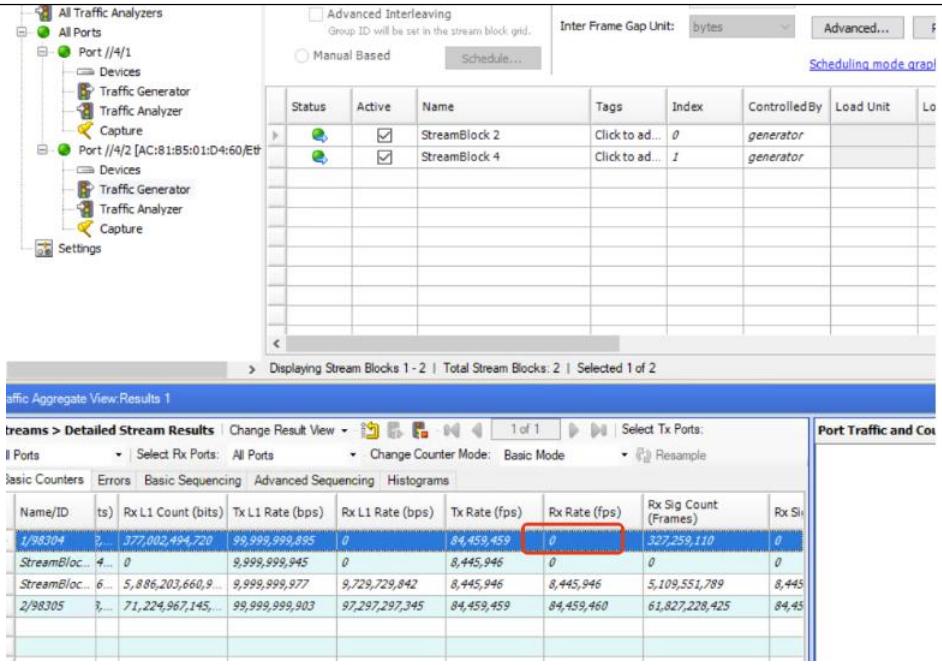
pfc-priority-queue
config qos pfc-priority-queue add pfc_queue_profile --pfc-priority 3 --queue 3
config interface qos pfc-priority-queue bind Ethernet8 pfc_queue_profile

Stop packet transmission on all ports.
TGA sends line-rate VLAN 100 packets with priority 3 to TGB.
TGB sends destination learning packets and PFC priority 3 packets.

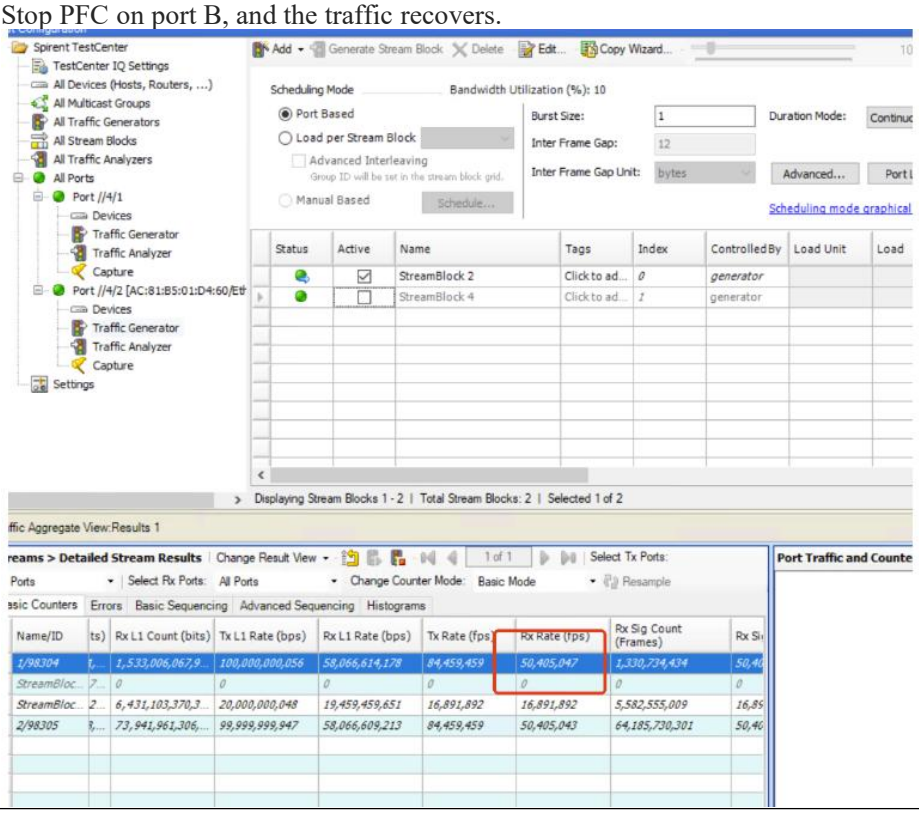
Port A



Port B



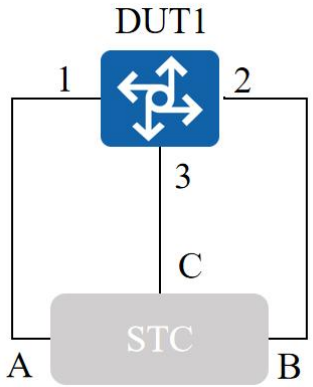
Stop PFC on port B, and the traffic recovers.



Test results	**PASS**
Remarks	

5.8 PFC Watchdog (WD)

Test Items	PFC watchdog Functional Test

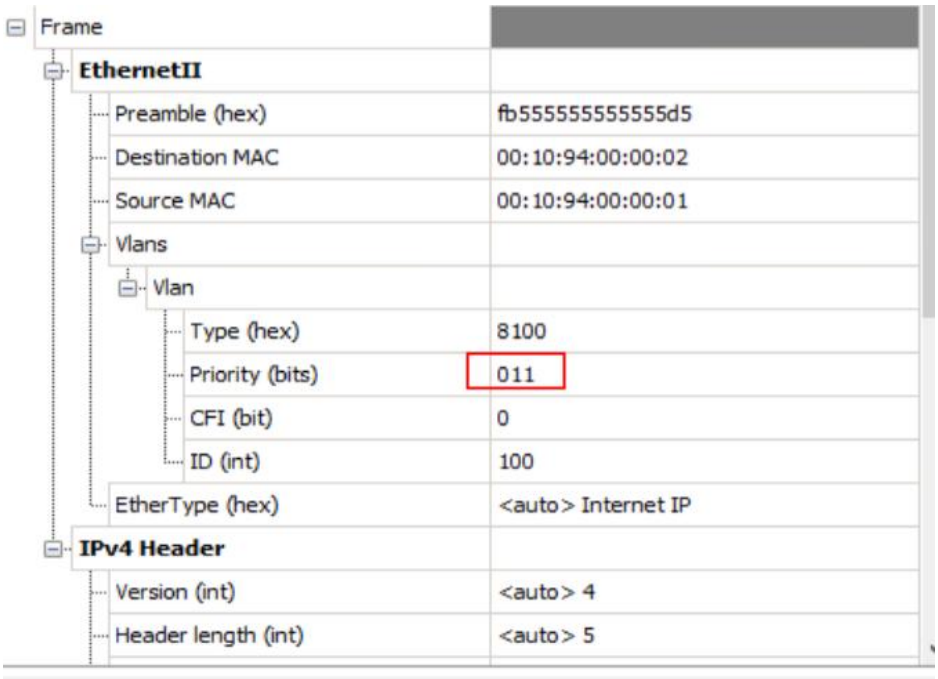
Test Content	The PFC watchdog of the testing equipment can work.
Test Topology	
Test Steps	1、Set up the test environment according to the diagram. 2、Map dot1p priority 3 to tc3 and pg3. 3、Bind the QoS mapping to the port. 4、Enable PFC priority 3 on DUTP1 and DUTP2. 5、Create VLAN100 and add DUT P1-P3 as tagged members. 6、Create pfc-priority-queue to map PFC priority 3 to queue3. 7、Bind the created pfc-priority-queue to TGB (check1). 8、TGA sends line-rate VLAN100 packets with priority 3 to TGB (check2). 9、TGB sends PFC priority 3 packets. 10、Enable PFC Watchdog (PFCWD) on the port.
Expected Results	Check1: TGB stops receiving data packets from TGA. Check2: TGB starts receiving partial data packets from TGA.
Actual Test Results	QOS config qos dot1p-tc add dot1p_to_tc_profile --dot1p 3 --tc 3 config interface qos dot1p-tc bind Ethernet0 dot1p_to_tc_profile config interface qos dot1p-tc bind Ethernet8 dot1p_to_tc_profile config interface qos dot1p-tc bind Ethernet16 dot1p_to_tc_profile config qos tc-pg add tc-pg-prof --tc 3 --pg 3 config interface qos tc-pg bind Ethernet0 tc-pg-prof config interface qos tc-pg bind Ethernet8 tc-pg-prof config interface qos tc-pg bind Ethernet16 tc-pg-prof config qos tc-queue add tc-queue-prof --tc 3 --queue 3 config interface qos tc-queue bind Ethernet0 tc-queue-prof config interface qos tc-queue bind Ethernet8 tc-queue-prof config interface qos tc-queue bind Ethernet16 tc-queue-prof config interface pfc priority Ethernet0 3 on config interface pfc priority Ethernet8 3 on

```
root@sonic:/home/admin# config interface pfc priority Ethernet0 3 on

Interface      Lossless priorities
-----------  ---------------------
Ethernet0                        3
root@sonic:/home/admin# config interface pfc priority Ethernet8 3 on

Interface      Lossless priorities
-----------  ---------------------
Ethernet8                        3
root@sonic:/home/admin#
```

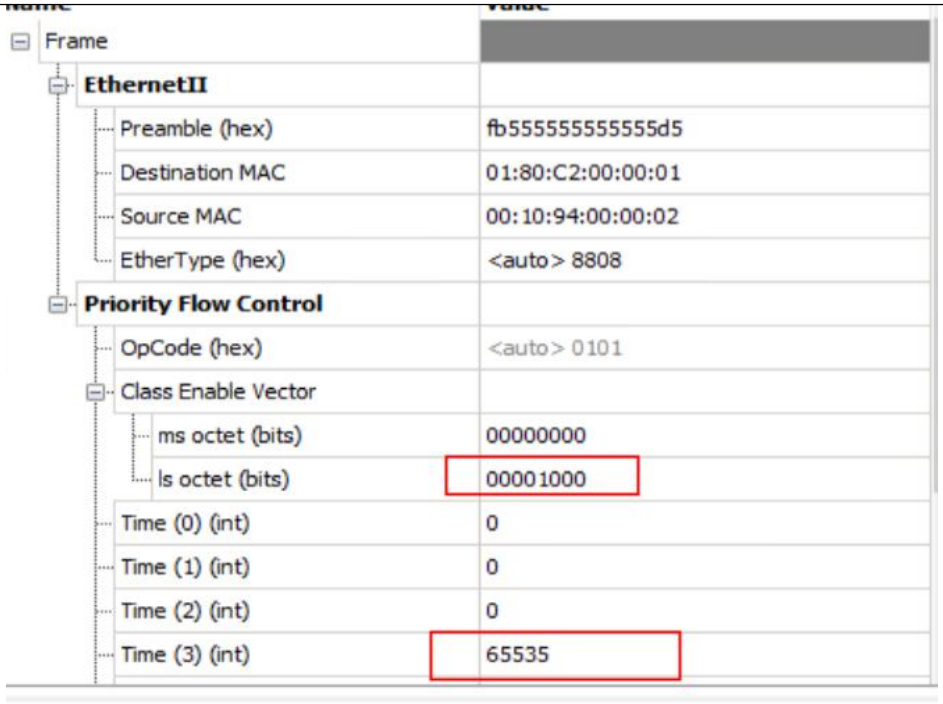
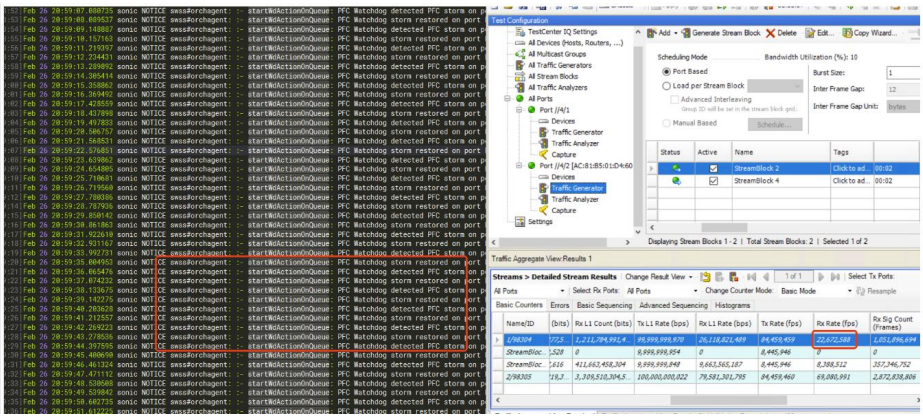
vlan
config vlan add 100
config vlan member add 100 Ethernet0
config vlan member add 100 Ethernet8
config vlan member add 100 Ethernet16

pfc-priority-queue
config qos pfc-priority-queue add pfc_queue_profile --pfc-priority 3 --queue 3
config interface qos pfc-priority-queue bind Ethernet8 pfc_queue_profile

TGA sends VLAN 100 data packets with priority 3 to TGB at line rate.
TGB sends destination learning packets and PFC priority 3 packets.
Port A

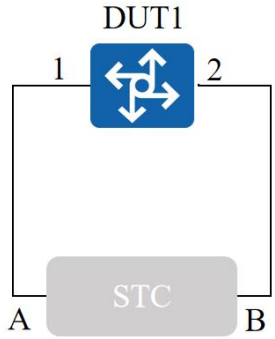


Port B

	 config pfcwd start --action forward --restoration-time 4000 ports Ethernet0 detection-time 4000 config pfcwd start --action forward --restoration-time 4000 ports Ethernet8 detection-time 4000 
Test results	**PASS**
Remarks	

5.9 PFC Watermark (WM)

Test Items	PFC Watetmark Functional Test
Test Content	The PFC watermark of the testing equipment can be displayed according to statistics.

Test Topology	
Test Steps	1、Set up the test environment according to the diagram. 2、Load the QoS configuration file. 3、Configure buffer parameters and bind them to the port. 4、Map DSCP 3 to tc3 and pg3; map DSCP 0 to tc0 and pg0. 5、Bind the QoS mapping to the port. 6、 Create a pfc-priority-queue to map PFC priority 3 to queue3 and PFC priority 0 to queue0. 7、Bind the created pfc-priority-queue to the port. 8、Enable PFC priorities 3 and 0 on the port. 9、Create VLAN 100 and add ports to VLAN 100 ， check1. 10、TGA sends line-rate DSCP 3 and DSCP 0 data packets of 1230 bytes to TGB. 11、Check the watermark statistics. 12、Rate limit the egress of TGB to 10G. 13、Check the watermark statistics again ， check2. 14、Delete the relevant configurations.
Expected Results	Check1: Configuration is successful. Check2: View the statistics.
Actual Test Results	QOS sudo config qos reload config interface buffer bind priority-group Ethernet0 0 ingress_lossless_profile config interface buffer bind priority-group Ethernet2 0 ingress_lossless_profile config interface buffer bind queue Ethernet0 0 egress_lossless_profile config interface buffer bind queue Ethernet2 0 egress_lossless_profile config interface buffer bind priority-group Ethernet0 3 ingress_lossless_profile config interface buffer bind priority-group Ethernet2 3 ingress_lossless_profile config interface buffer bind queue Ethernet0 3 egress_lossless_profile config interface buffer bind queue Ethernet2 3 egress_lossless_profile config qos dscp-tc add dscp_to_tc_profile --dscp 3 --tc 3 config qos dscp-tc update dscp_to_tc_profile --dscp 0 --tc 0 config interface qos dscp-tc bind Ethernet0 dscp_to_tc_profile config interface qos dscp-tc bind Ethernet2 dscp_to_tc_profile config qos tc-pg add tc-pg-prof --tc 3 --pg 3 config qos tc-pg update tc-pg-prof --tc 0 --pg 0 config interface qos tc-pg bind Ethernet0 tc-pg-prof config interface qos tc-pg bind Ethernet2 tc-pg-prof config qos tc-queue add tc-queue-prof --tc 3 --queue 3 config qos tc-queue update tc-queue-prof --tc 0 --queue 0 config interface qos tc-queue bind Ethernet0 tc-queue-prof

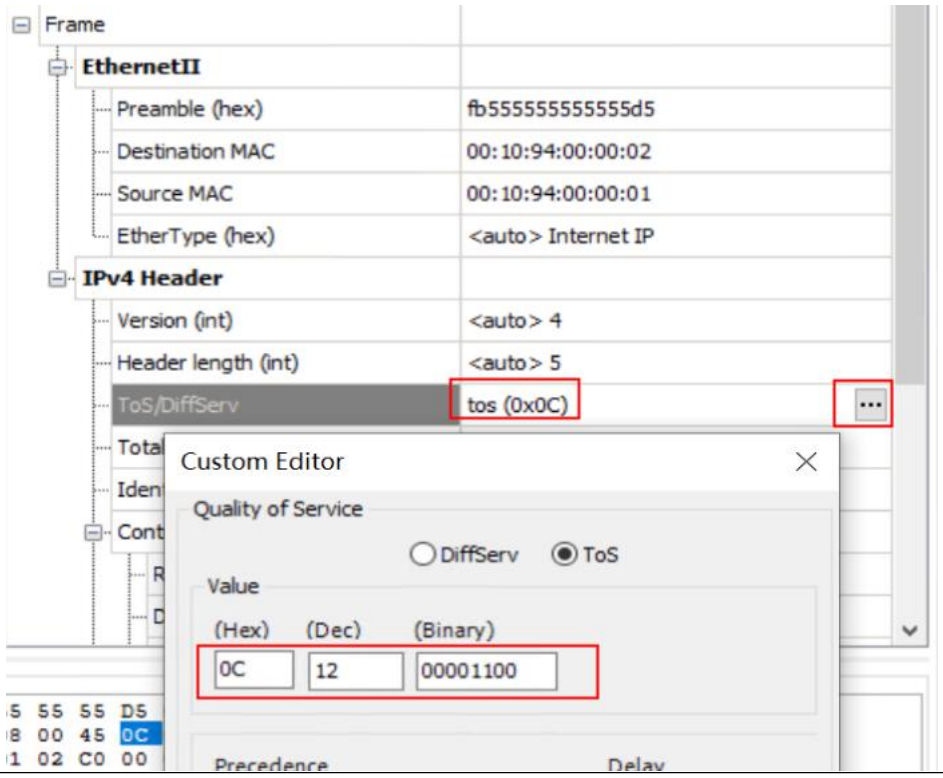
config interface qos tc-queue bind Ethernet2 tc-queue-prof

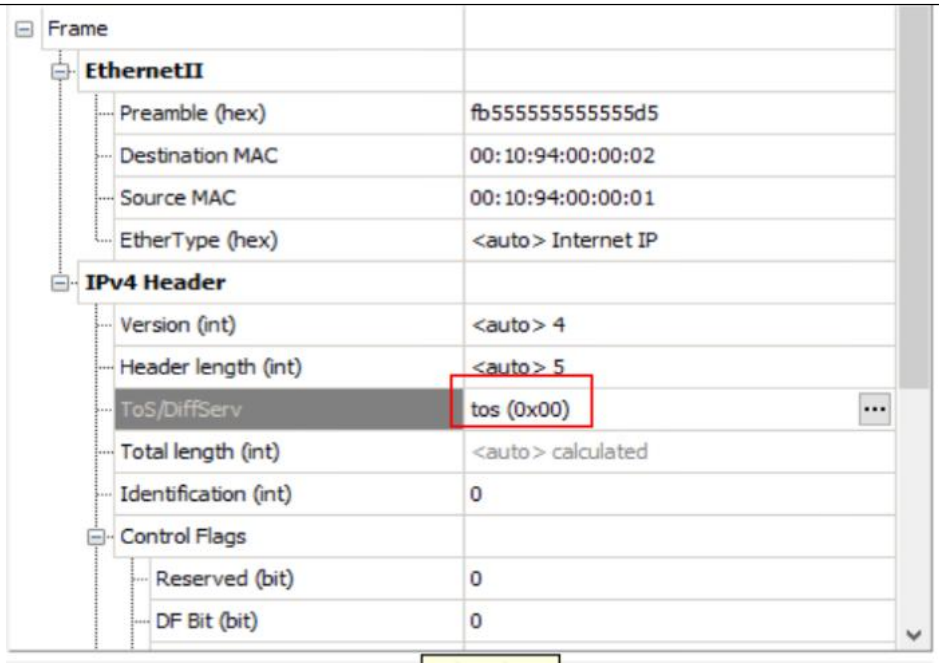
pfc-priority-queue
config qos pfc-priority-queue add pfc_queue_profile --pfc-priority 3 --queue 3
config qos pfc-priority-queue update pfc_queue_profile --pfc-priority 0 --queue 0
config interface qos pfc-priority-queue bind Ethernet0 pfc_queue_profile
config interface qos pfc-priority-queue bind Ethernet2 pfc_queue_profile

config interface pfc priority Ethernet0 3 on
config interface pfc priority Ethernet0 0 on
config interface pfc priority Ethernet2 3 on
config interface pfc priority Ethernet2 0 on

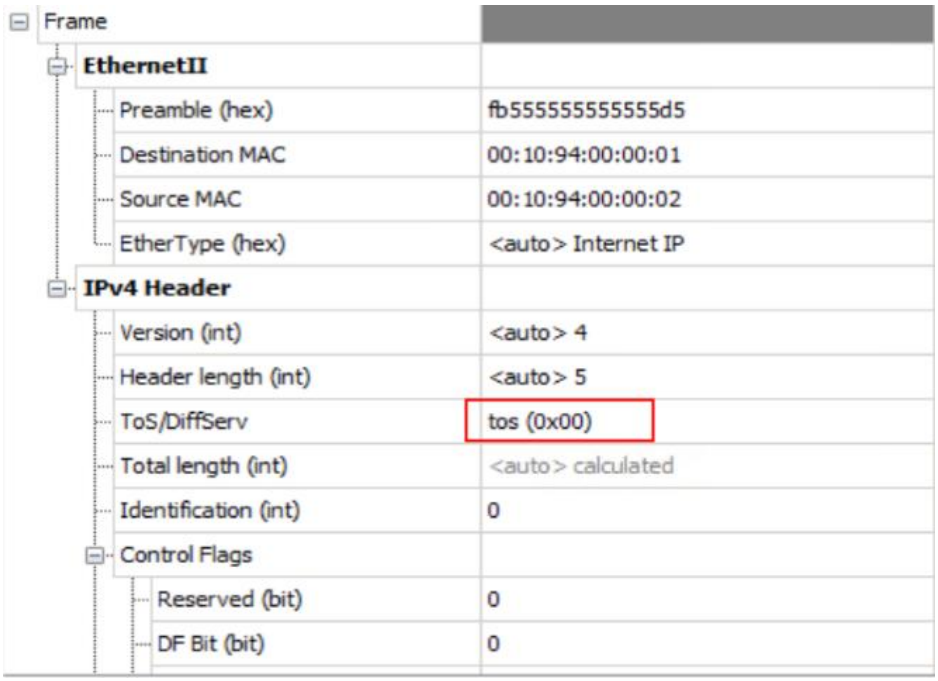
vlan
config vlan add 100
config vlan member add 100 Ethernet0 -u
config vlan member add 100 Ethernet2 -u

TGB sends MAC learning packets, and TGA sends line-rate packets with DSCP 3 and DSCP 0.
Port A



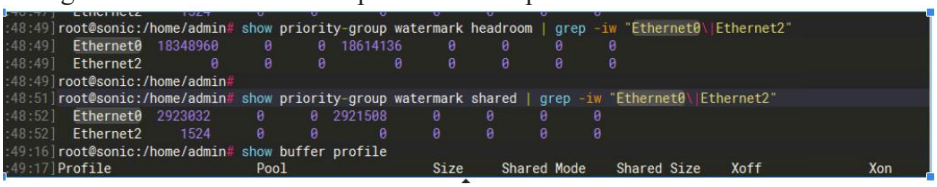


Port B



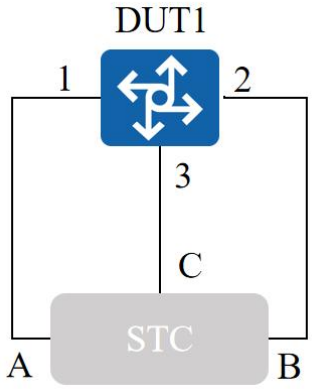
sonic-clear priority-group watermark headroom

config scheduler add profile-1 --shaper_type=bytes --bandwidth=10g
config interface scheduler bind port Ethernet2 profile-1



Test results	**PASS**
Remarks	

5.10 Scheduling

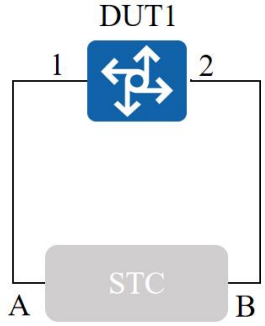
Test Items	Scheduling Functional Test
Test Content	The testing equipment forwards packets according to QoS Scheduling.
Test Topology	
Test Steps	1、Set up the test environment according to the diagram. 2、Map DSCP3 to tc3 and pg3, and map DSCP0 to tc0 and pg0. 3、Bind the QoS mapping to the port. 4、Configure schedulers wrr60 and wrr40, and bind them to queue0 and queue3 of DUTP3. 5、Create VLAN100, add DUT P1-P3 to VLAN100 as tagged members (check1). TGC sends learning packets. 6、TGA is set to send DSCP0 packets with a length of 1230 at line rate to TGC; 7、TGB is set to send DSCP3 packets with a length of 1230 at line rate to TGC. 8、TGA and TGB send packets simultaneously, check the ratio of packets received by TGC from TGA and TGB (check2). 9、Stop packet transmission on all ports.
Expected Results	Check1: Configuration is successful. Check2: The DSCP packets received by TGC from TGA and TGB conform to the WRR ratio.
Actual Test Results	qos config qos dscp-tc add dscp_to_tc_profile --dscp 3 --tc 3 config qos dscp-tc update dscp_to_tc_profile --dscp 0 --tc 0 config interface qos dscp-tc bind Ethernet0 dscp_to_tc_profile config interface qos dscp-tc bind Ethernet8 dscp_to_tc_profile config interface qos dscp-tc bind Ethernet16 dscp_to_tc_profile config qos tc-pg add tc-pg-prof --tc 3 --pg 3 config qos tc-pg update tc-pg-prof --tc 0 --pg 0 config interface qos tc-pg bind Ethernet0 tc-pg-prof config interface qos tc-pg bind Ethernet8 tc-pg-prof config interface qos tc-pg bind Ethernet16 tc-pg-prof config qos tc-queue add tc-queue-prof --tc 3 --queue 3

	config qos tc-queue update tc-queue-prof --tc 0 --queue 0 config interface qos tc-queue bind Ethernet0 tc-queue-prof config interface qos tc-queue bind Ethernet8 tc-queue-prof config interface qos tc-queue bind Ethernet16 tc-queue-prof scheduler config scheduler add sched-prof-wrr60 --sched_type WRR --weight 60 config scheduler add sched-prof-wrr40 --sched_type WRR --weight 40 config interface scheduler bind queue Ethernet16 0 sched-prof-wrr60 config interface scheduler bind queue Ethernet16 3 sched-prof-wrr40 vlan config vlan add 100 config vlan member add 100 Ethernet0 -u config vlan member add 100 Ethernet8 -u config vlan member add 100 Ethernet16 -u TGC sends learning packets. The packet reception rates at TGC for DSCP 0 (from TGA) and DSCP 3 (from TGB) are measured when TGA and TGB send simultaneously. The port receives DSCP 0 and DSCP 3 packets at a rate ratio of 3:2.

Name/ID	t	Late Count (Frames)	Dropped Rate (fps)	Dropped Frame Perce...	In-order Rate (fps)	R
StreamBlock...		0	0	0	0	0
StreamBlock...		0	0	0	31,581,791	0
StreamBlock...		0	0	0	0	0
StreamBlock...		0	0	0	20,091,821	0
StreamBlock...		0	28,441,934	67.35	13,787,794	0
StreamBlock...		0	23,231,268	55.011	18,998,461	0

Test results	**PASS**
Remarks	

5.11 Egress Shaping (Port, Queue)

Test Items	Egress Shaping Functional Test
Test Content	The testing equipment forwards packets according to the set Egress shaping value.
Test Topology	
Test Steps	1. Set up the test environment according to the diagram. 2. Map DSCP3 to tc3 and pg3, and map DSCP0 to tc0 and pg0. 3. Bind the QoS mapping to the port. 4. Set the port bandwidth of DUTP2 to 10G and the bandwidth of queue0 to 1G. 5. Create VLAN2, add DUT P1-P2 to VLAN2 as tagged members (check1).

	6. TGB sends learning packets. 7. TGA is set to send DSCP0 packets with a length of 1230 at line rate to TGB, and check the packet reception rate (check2). 8. TGA is set to send DSCP3 packets with a length of 1230 at line rate to TGB, and check the packet reception rate (check3). 9. Stop packet transmission on all ports.
Expected Results	Check1: Configuration is successful. Check2: The rate of DSCP0 packets received by TGB is 1G. Check3: The rate of DSCP3 packets received by TGB is 10G.
Actual Test Results	qos config qos dscp-tc add dscp_to_tc_profile --dscp 3 --tc 3 config qos dscp-tc update dscp_to_tc_profile --dscp 0 --tc 0 config interface qos dscp-tc bind Ethernet0 dscp_to_tc_profile config interface qos dscp-tc bind Ethernet8 dscp_to_tc_profile config qos tc-pg add tc-pg-prof --tc 3 --pg 3 config qos tc-pg update tc-pg-prof --tc 0 --pg 0 config interface qos tc-pg bind Ethernet0 tc-pg-prof config interface qos tc-pg bind Ethernet8 tc-pg-prof config qos tc-queue add tc-queue-prof --tc 3 --queue 3 config qos tc-queue update tc-queue-prof --tc 0 --queue 0 config interface qos tc-queue bind Ethernet0 tc-queue-prof config interface qos tc-queue bind Ethernet8 tc-queue-prof config scheduler add profile-1 --shaper_type=bytes --bandwidth=10g config scheduler add profile-2 --shaper_type=bytes --bandwidth=1g config interface scheduler bind port Ethernet8 profile-1 config interface scheduler bind queue Ethernet8 0 profile-2 vlan config vlan add 100 config vlan member add 100 Ethernet0 -u config vlan member add 100 Ethernet8 -u TGB sends learning packets. TGA is set to send DSCP0 packets with a length of 1230 at line rate to TGB, and check the packet reception rate.

```
root@sonic:~# show qos dscp-tc
root@sonic:~# show qos tc-pg
root@sonic:~# show qos tc-queue
root@sonic:~#
root@sonic:~# config qos dscp-tc add dscp_to_tc_profile --dscp 3 --tc 3
root@sonic:~# config qos dscp-tc update dscp_to_tc_profile --dscp 0 --tc 0
root@sonic:~#
root@sonic:~# config interface qos dscp-tc bind Ethernet16 dscp_to_tc_profile
root@sonic:~# config interface qos dscp-tc bind Ethernet18 dscp_to_tc_profile
root@sonic:~#
root@sonic:~# show qos dscp-tc
scp-tc policy: dscp_to_tc_profile
 DSCP    TC
-----  ----
    0     0
    3     3

root@sonic:~#
root@sonic:~# config qos tc-pg add tc-pg-prof --tc 3 --pg 3
root@sonic:~# config qos tc-pg update tc-pg-prof --tc 0 --pg 0
root@sonic:~#
root@sonic:~# config interface qos tc-pg bind Ethernet0 tc-pg-prof
root@sonic:~# config interface qos tc-pg unbind Ethernet0 tc-pg-prof
root@sonic:~# config interface qos tc-pg unbind Ethernet16 tc-pg-prof
sage: config interface qos tc-pg [OPTIONS] {bind|unbind}
                                 {<interface_name>|all} <profile>
ry "config interface qos tc-pg -h" for help.

rror: 'tc-pg-prof' profile is not configured on interface Ethernet16.
root@sonic:~# config interface qos tc-pg bind Ethernet16 tc-pg-prof
root@sonic:~#
root@sonic:~# config interface qos tc-pg bind Ethernet18 tc-pg-prof
root@sonic:~#
root@sonic:~# show qos tc-pg
c-pg policy: tc-pg-prof
 TC    PG
---  ----
  0     0
  3     3

root@sonic:~#
root@sonic:~# config qos tc-queue add tc-queue-prof --tc 3 --queue 3
root@sonic:~# config qos tc-queue update tc-queue-prof --tc 0 --queue 0
root@sonic:~#
root@sonic:~# config interface qos tc-queue bind Ethernet16 tc-queue-prof
root@sonic:~# config interface qos tc-queue bind Ethernet18 tc-queue-prof
root@sonic:~#
root@sonic:~# show qos tc-queue
c-queue policy: tc-queue-prof
 TC    Queue
---  -------
  0        0
  3        3

root@sonic:~#
```

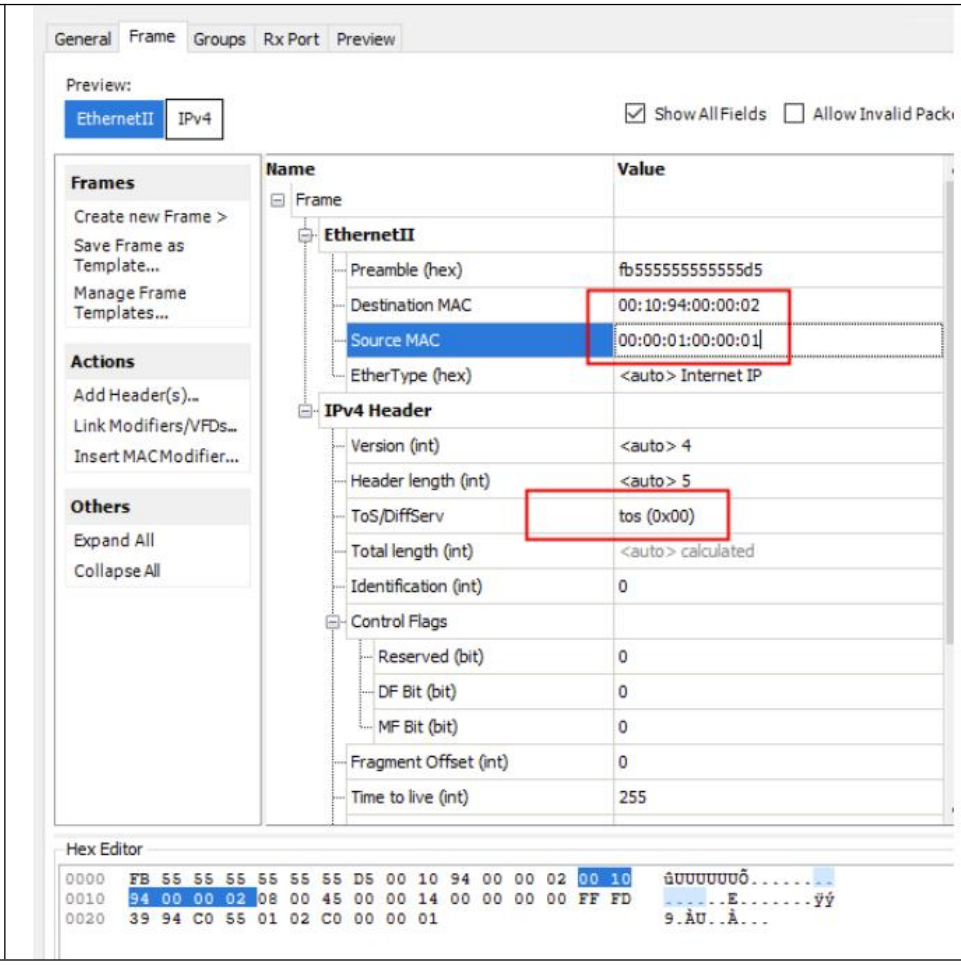
Set the port bandwidth of DUTP2 to 10G and the bandwidth of queue0 to 1G.

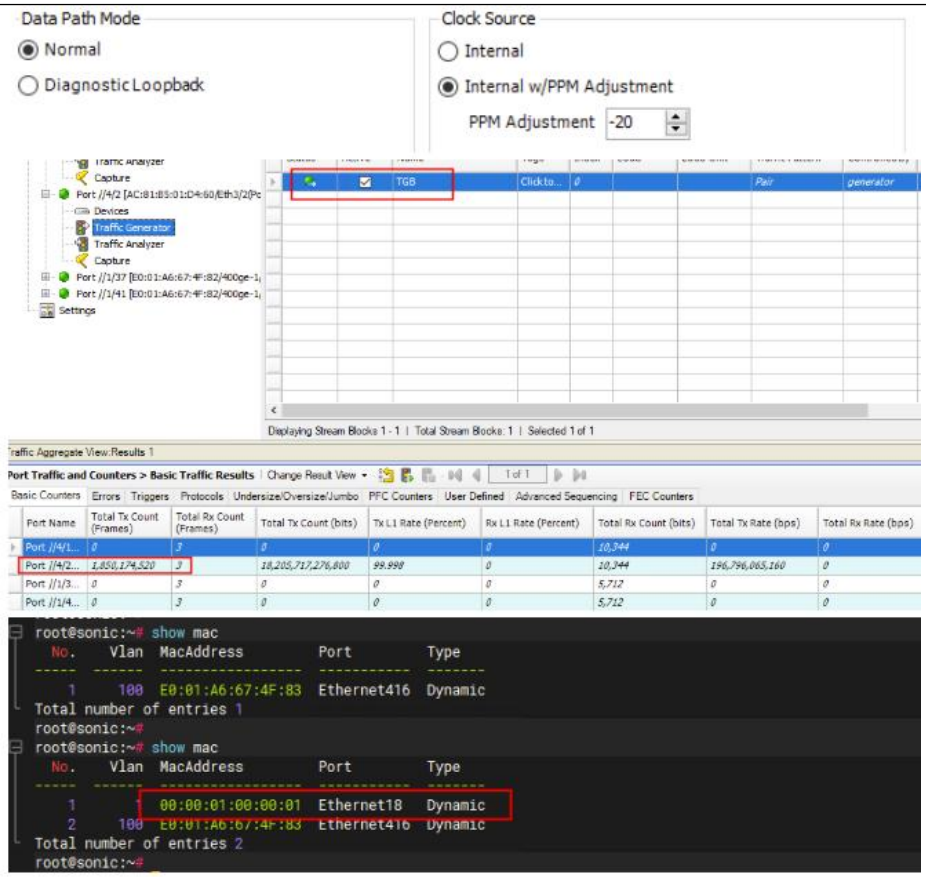
```
root@sonic:~# config scheduler add profile-1 --shaper_type=bytes --bandwidth=10g
root@sonic:~# config scheduler add profile-2 --shaper_type=bytes --bandwidth=1g
root@sonic:~#
root@sonic:~# config interface scheduler bind port Ethernet18 profile-1
root@sonic:~# config interface scheduler bind queue Ethernet18 0 profile-2
root@sonic:~#
root@sonic:~# show interfaces scheduler
Ethernet18
Type    Queue    Profile    Scheduling Mode    Weight    Shaper Type    Bandwidth
------  -------  ---------  -----------------  --------  -------------  -----------
Port    N/A      profile-1  N/A                N/A       bytes          10 Gbps
Queue   0        profile-2  N/A                N/A       bytes          1 Gbps

root@sonic:~#
```

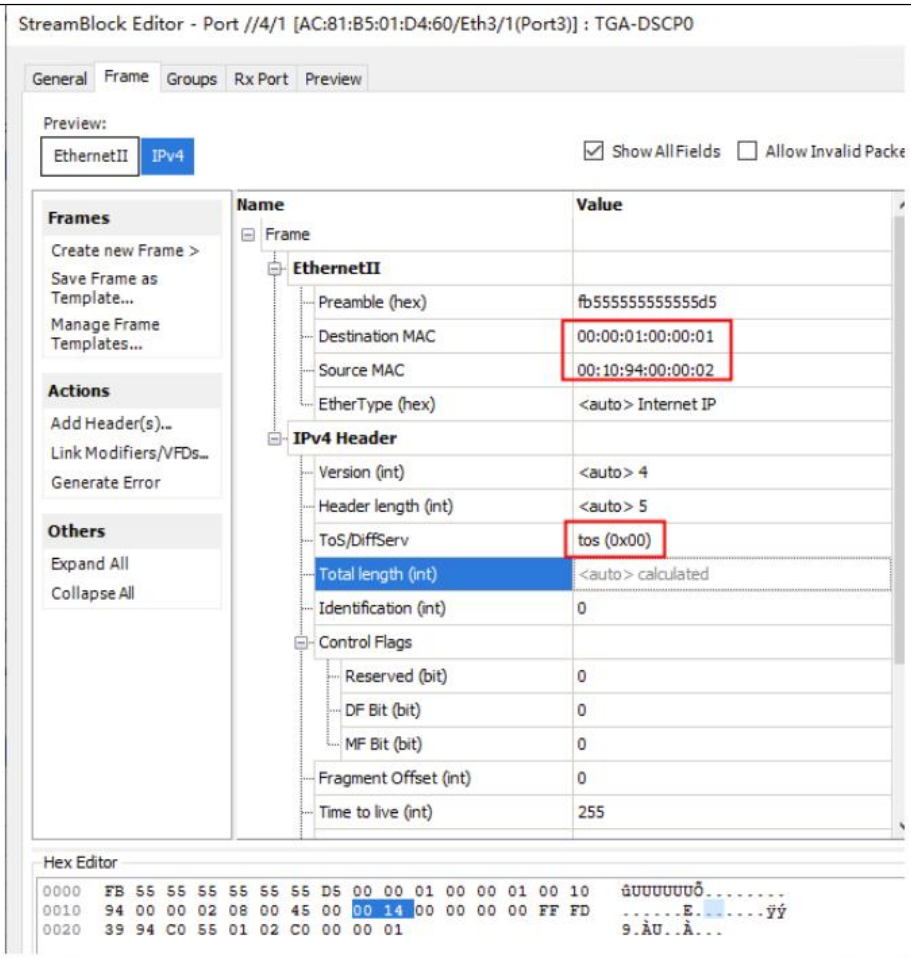
Create VLAN2 and add DUT Port1-Port2 to VLAN2 as tagged members.

```
root@sonic:~# config vlan add 2
root@sonic:~#
root@sonic:~# config vlan member add 2 Ethernet16
root@sonic:~# config vlan member add 2 Ethernet18
root@sonic:~#
root@sonic:~# show vlan brief
+-----------+--------------+-------------+-----------+-----------+---------------+-----------------------------+
| VLAN ID   | IP Address   | Ports       | Port      | Proxy     | DHCP Helper   | DHCP Relay Configuration    |
|           |              |             | Tagging   | ARP       | Address       |                             |
+===========+==============+=============+===========+===========+===============+=============================+
|         2 |              | Ethernet16  | tagged    | disabled  |               | Source Interface:           |
|           |              | Ethernet18  | tagged    |           |               | Link Selection:             |
|           |              |             |           |           |               | Server Vrf:                 |
|           |              |             |           |           |               | Server ID Override:         |
+-----------+--------------+-------------+-----------+-----------+---------------+-----------------------------+
|       100 |              | Ethernet416 | untagged  | disabled  |               | Source Interface:           |
|           |              |             |           |           |               | Link Selection:             |
|           |              |             |           |           |               | Server Vrf:                 |
|           |              |             |           |           |               | Server ID Override:         |
+-----------+--------------+-------------+-----------+-----------+---------------+-----------------------------+
root@sonic:~# show vlan config
Name       VID  Member       Mode
-------  -----  -----------  --------
Vlan2        2  Ethernet16   tagged
Vlan2        2  Ethernet18   tagged
Vlan100    100  Ethernet416  untagged
root@sonic:~#
```

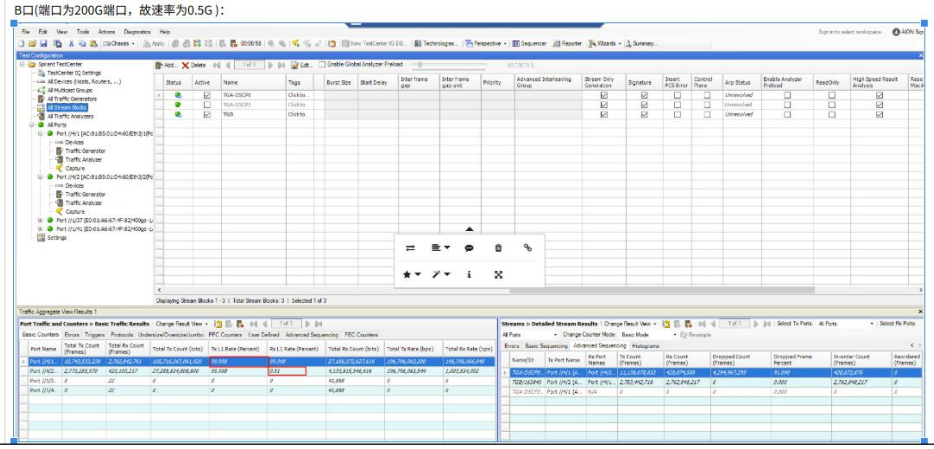
TGB sends learning packets.

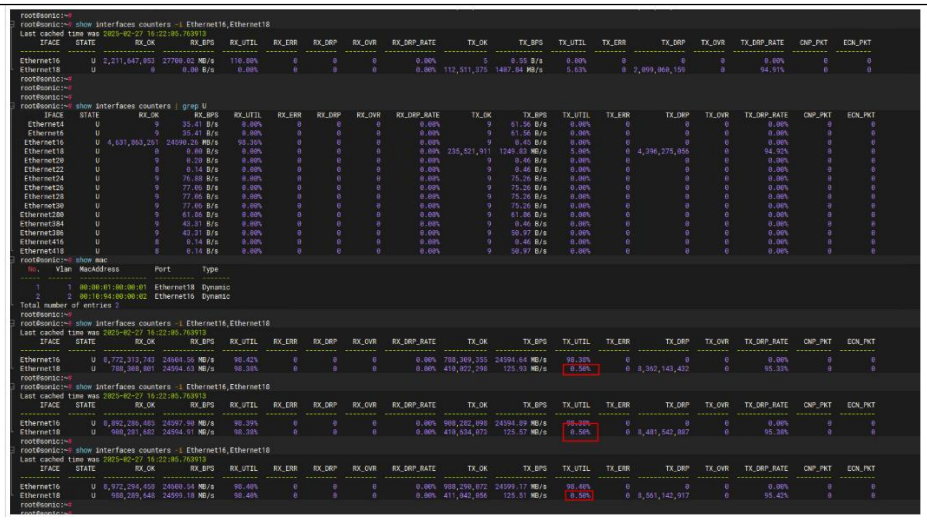

General
Frame
Groups
Rx Port
Preview
Preview:
EthernetII
IPv4
Show All Fields
Allow Invalid Pack
Frames
Create new Frame >
Save Frame as Template...
Manage Frame Templates...
Actions
Add Header(s)...
Link Modifiers/VFDs...
Insert MAC Modifier...
Others
Expand All
Collapse All
Name
Value
Frame
EthernetII
Preamble (hex)
fb555555555555d5
Destination MAC
00:10:94:00:00:02
Source MAC
00:00:01:00:00:01
EtherType (hex)
<auto> Internet IP
IPv4 Header
Version (int)
<auto> 4
Header length (int)
<auto> 5
ToS/DiffServ
tos (0x00)
Total length (int)
<auto> calculated
Identification (int)
0
Control Flags
Reserved (bit)
0
DF Bit (bit)
0
MF Bit (bit)
0
Fragment Offset (int)
0
Time to live (int)
255
Hex Editor
0000 FB 55 55 55 55 55 55 D5 00 10 94 00 00 02 00 10
0010 94 00 00 02 08 00 45 00 00 14 00 00 00 00 FF FD
0020 39 94 C0 55 01 02 C0 00 00 01


Data Path Mode
Normal
Diagnostic Loopback
Clock Source
Internal
Internal w/PPM Adjustment
PPM Adjustment -20
Capture
Port //4/2 [AC:81:85:01:D4:60/Eth3/2/Pc
Devices
Traffic Generator
Traffic Analyzer
Capture
Port //1/37 [E0:01:A6:67:4F:82/400ge-1
Port //1/41 [E0:01:A6:67:4F:82/400ge-1
Settings
TG8
Clickto...
Pair
generator
Displaying Stream Blocks 1 - 1 | Total Stream Blocks: 1 | Selected 1 of 1
Traffic Aggregate View:Results 1
Port Traffic and Counters > Basic Traffic Results
Change Result View
Basic Counters
Errors
Triggers
Protocols
Undersize/Oversize/Jumbo
PFC Counters
User Defined
Advanced Sequencing
FEC Counters
Port Name
Total Tx Count (Frames)
Total Rx Count (Frames)
Total Tx Count (bits)
Tx L1 Rate (Percent)
Rx L1 Rate (Percent)
Total Rx Count (bits)
Total Tx Rate (bps)
Total Rx Rate (bps)
Port //4/1...
Port //4/2...
1,850,174,520
18,205,717,276,800
99.999
10,344
196,796,065,160
Port //1/3...
5,712
Port //1/4...
root@sonic:~# show mac
No. Vlan MacAddress Port Type
1 100 E0:01:A6:67:4F:83 Ethernet416 Dynamic
Total number of entries 1
root@sonic:~#
root@sonic:~# show mac
No. Vlan MacAddress Port Type
1 1 00:00:01:00:00:01 Ethernet18 Dynamic
2 100 E0:01:A6:67:4F:83 Ethernet416 Dynamic
Total number of entries 2
root@sonic:~#

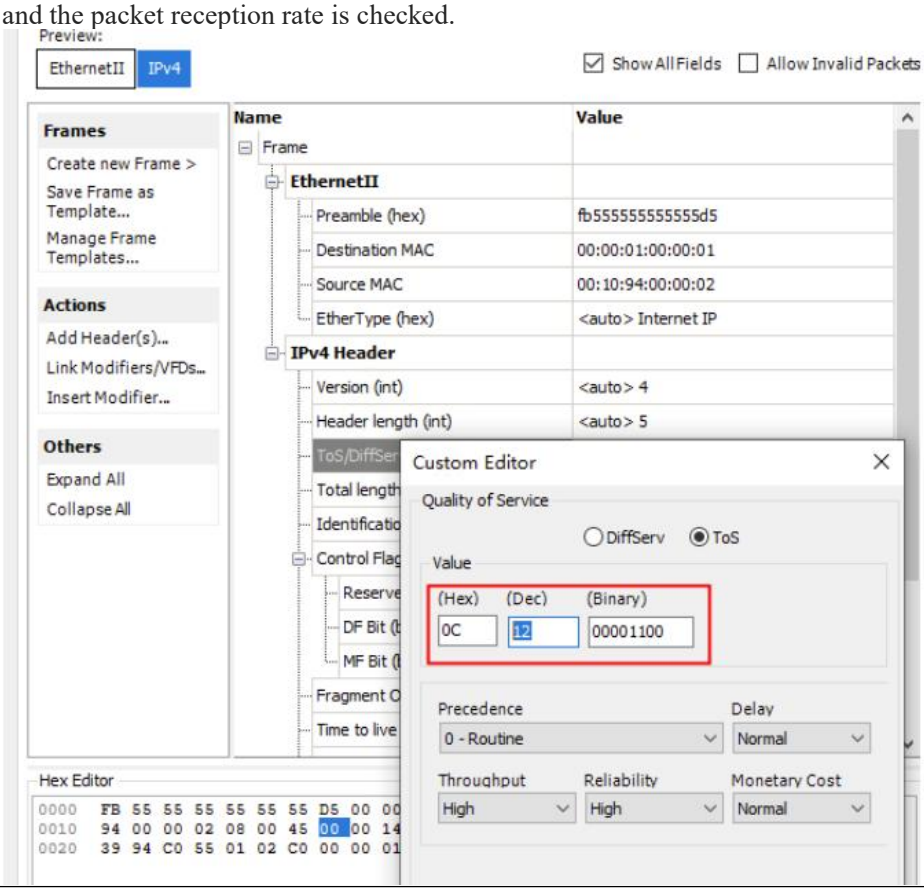


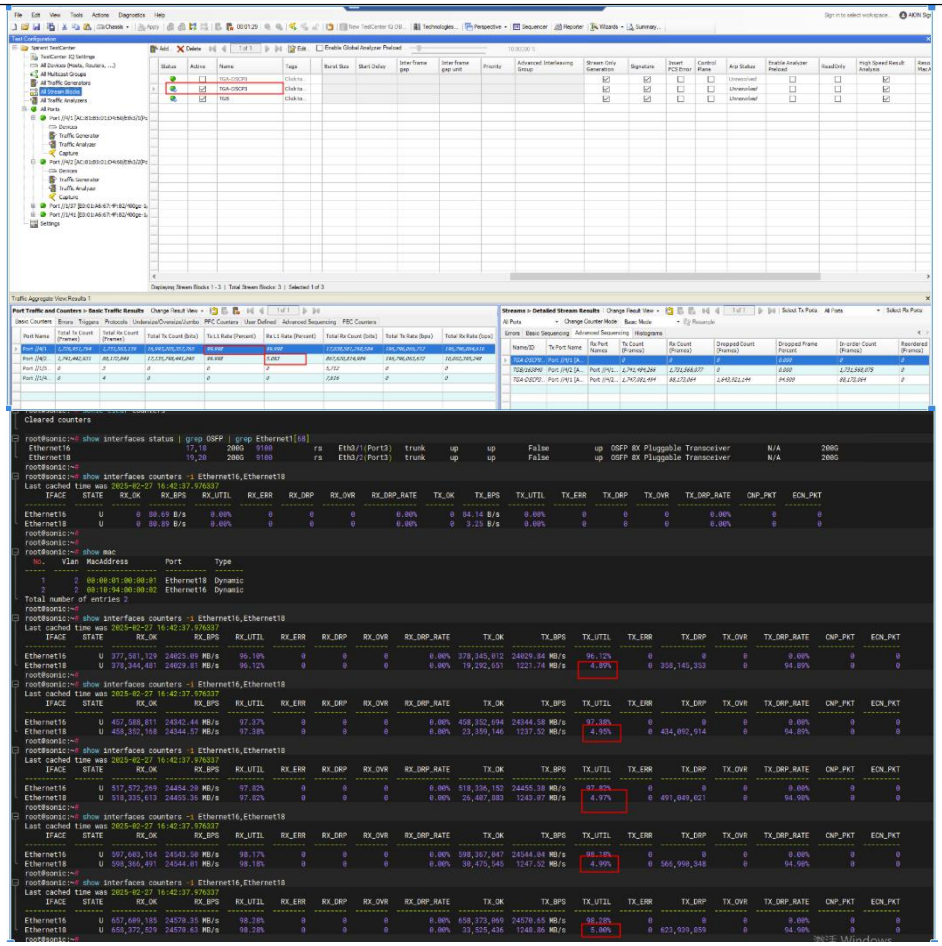
B口(端口为200G端口，故速率为0.5G)：



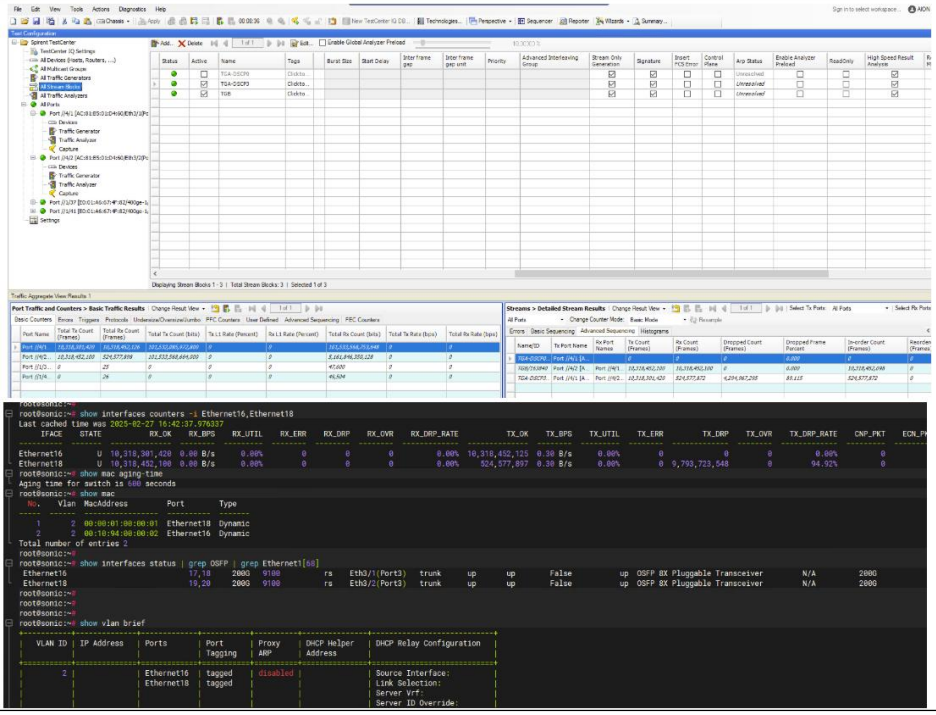


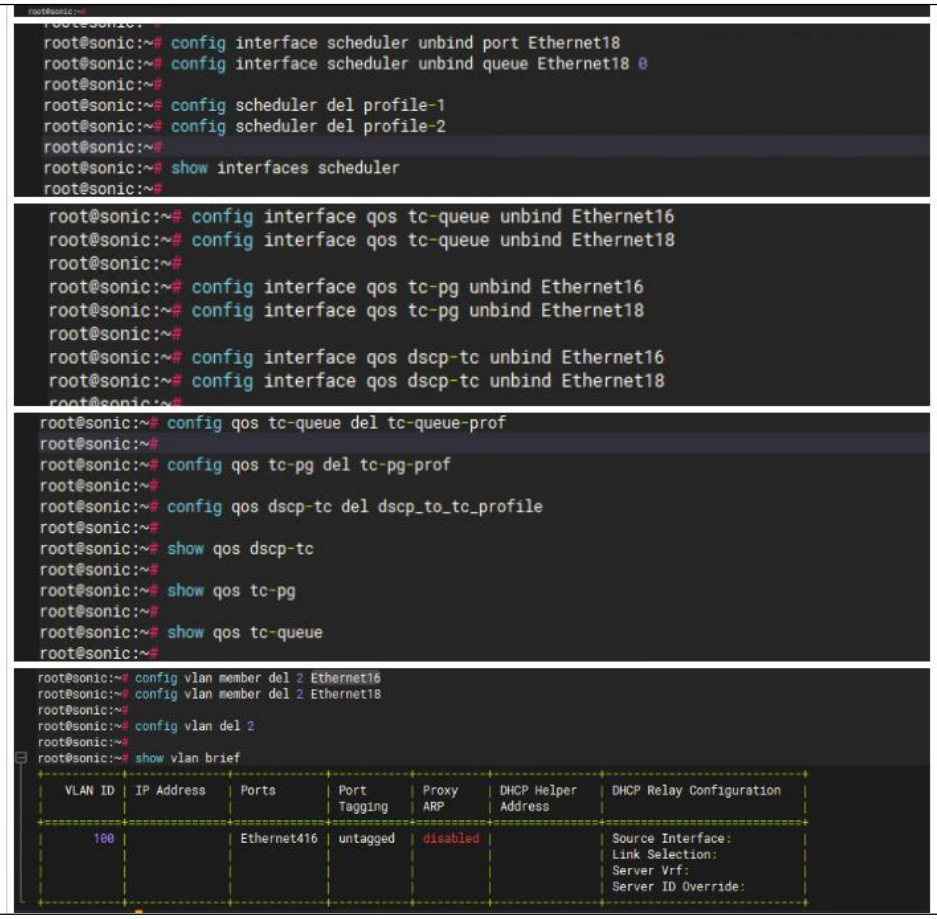
TGA is configured to send DSCP3 packets with a length of 1230 at line rate to TGB, and the packet reception rate is checked.



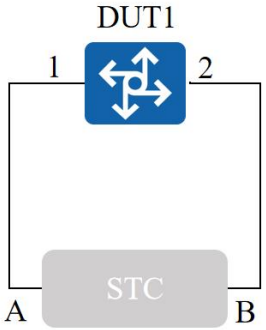


Stop packet transmission on all ports.



	
Test results	**PASS**
Remarks	

5.12 WRED

Test Items	WRED Functional Test
Test Content	The testing equipment can forward packets according to the WRED (Weighted Random Early Detection) settings.
Test Topology	

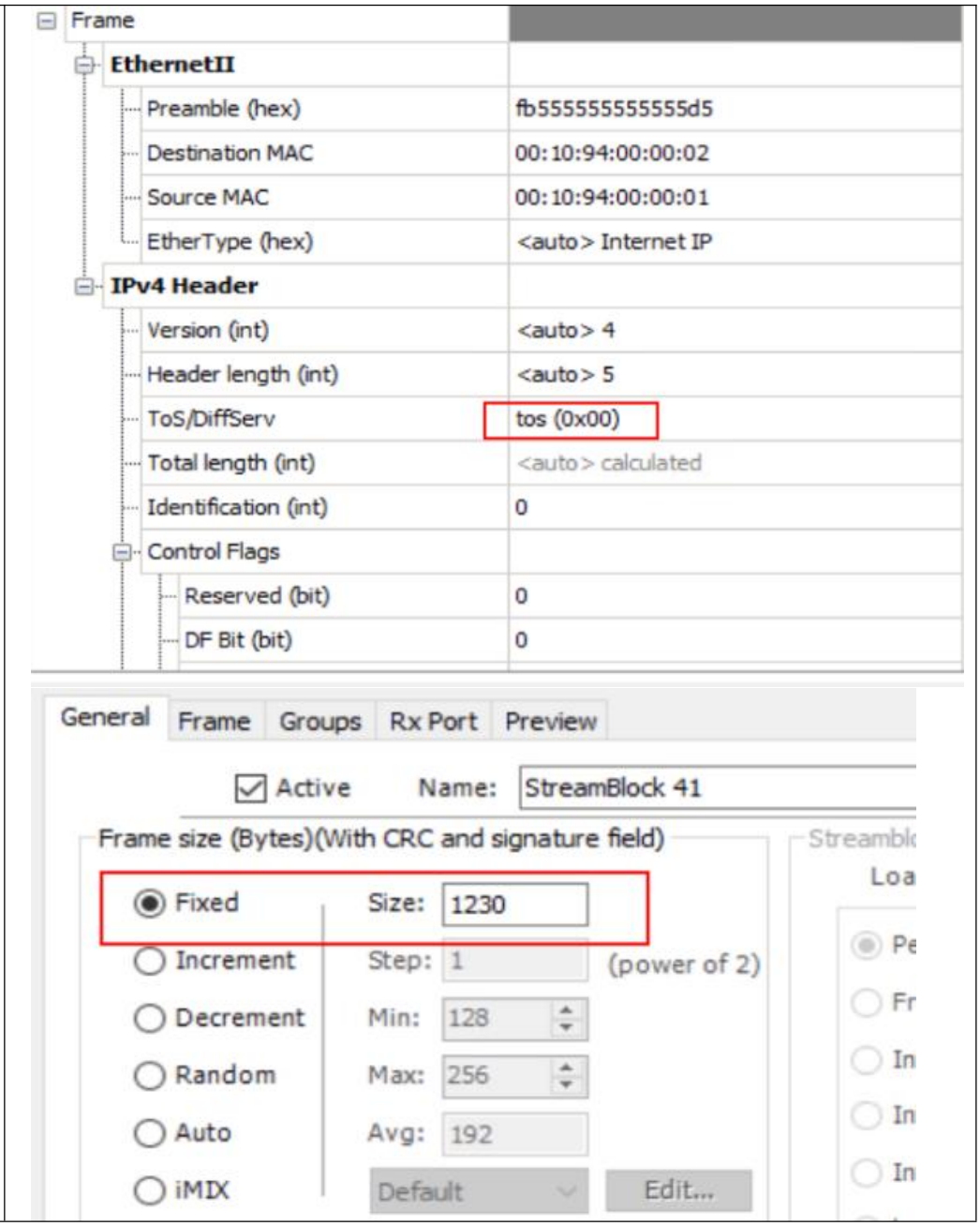
Test Steps	1、Set up the test environment according to the diagram. 2、Configure two WRED profiles. 3、Bind the WRED profiles to egress port Queue 0 and Queue 3 respectively. 4、Configure a QOS profile, bind it to the port, and map DSCP 0 to Queue 0 and DSCP 3 to Queue 3. 5、Configure egress rate limiting to 10G. 6、Create VLAN 100 and add DUT P1 and DUT P2 to VLAN 100 as untagged members. 7、TGB continuously sends MAC learning packets. 8、TGA sends line-rate packets to TGB for 10 seconds, and check the packet loss ratio received by TGB (check1). 9、delete the WRED configuration, repeat step 8, and check the packets received by TGB (check2).
Expected Results	Check1: WRED works as configured. Check2: No WRED behavior.
Actual Test Results	QOS sudo config qos reload config interface buffer bind priority-group Ethernet0 0 ingress_lossless_profile config interface buffer bind priority-group Ethernet8 0 ingress_lossless_profile config interface buffer bind queue Ethernet0 0 egress_lossless_profile config interface buffer bind queue Ethernet8 0 egress_lossless_profile config interface buffer bind priority-group Ethernet0 3 ingress_lossless_profile config interface buffer bind priority-group Ethernet8 3 ingress_lossless_profile config interface buffer bind queue Ethernet0 3 egress_lossless_profile config interface buffer bind queue Ethernet8 3 egress_lossless_profile config qos dscp-tc add dscp_to_tc_profile --dscp 0 --tc 0 config qos dscp-tc update dscp_to_tc_profile --dscp 3 --tc 3 config interface qos dscp-tc bind Ethernet0 dscp_to_tc_profile config interface qos dscp-tc bind Ethernet8 dscp_to_tc_profile config qos tc-pg add tc_to_pg_profile --tc 0 --pg 0 config qos tc-pg update tc_to_pg_profile --tc 3 --pg 3 config interface qos tc-pg bind Ethernet0 tc_to_pg_profile config interface qos tc-pg bind Ethernet8 tc_to_pg_profile config qos tc-queue add tc_to_queue_profile --tc 0 --queue 0 config qos tc-queue update tc_to_queue_profile --tc 3 --queue 3 config interface qos tc-queue bind Ethernet0 tc_to_queue_profile config interface qos tc-queue bind Ethernet8 tc_to_queue_profile config wred add wred-prof-high --mode wred --gmin 0 --gmax 133168898 --gdrop 100 config wred add wred-prof --mode wred --gmin 0 --gmax 133168898 --gdrop 10 config interface wred bind queue Ethernet8 0 wred-prof-high config interface wred bind queue Ethernet8 3 wred-prof show wred show interfaces wred

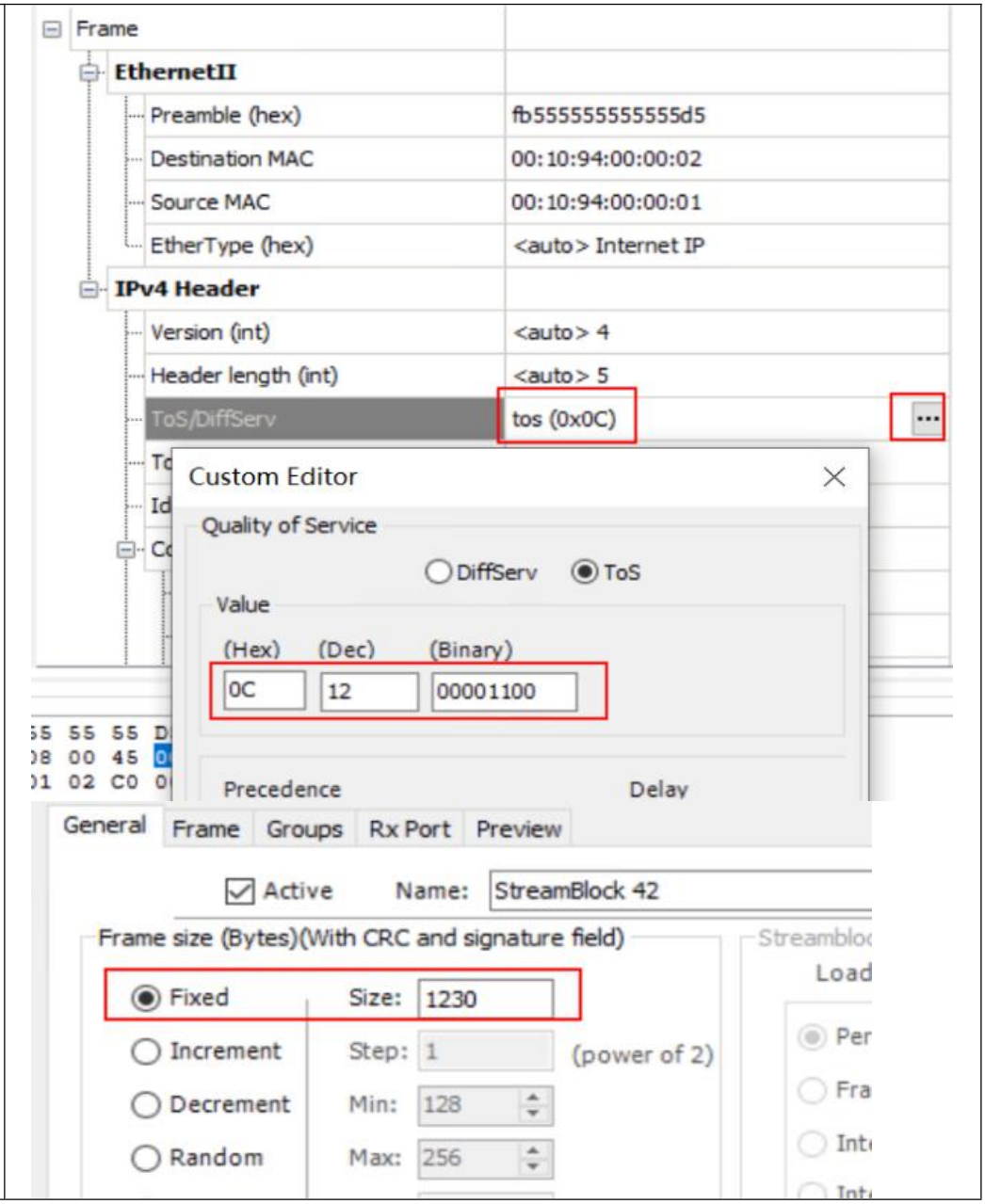
```
root@sonic:/home/admin# show wred
Profile: wred-prof
Color     Mode      Min Threshold    Max Threshold    Drop Probability
--------  ------  ---------------  ---------------  ------------------
Green     WRED                 0        133168898                 100
Yellow
Red

Profile: wred-prof-high
Color     Mode      Min Threshold    Max Threshold    Drop Probability
--------  ------  ---------------  ---------------  ------------------
Green     WRED                 0        133168898                  10
Yellow
Red

root@sonic:/home/admin#
root@sonic:/home/admin# show interfaces wred
Ethernet8
Queue: 0
ECN/WRED: wred-prof
Color     Mode      Min Threshold    Max Threshold    Drop Probability
--------  ------  ---------------  ---------------  ------------------
Green     WRED                 0        133168898                 100
Yellow
Red

Ethernet8
Queue: 3
ECN/WRED: wred-prof-high
Color     Mode      Min Threshold    Max Threshold    Drop Probability
--------  ------  ---------------  ---------------  ------------------
Green     WRED                 0        133168898                  10
Yellow
Red

root@sonic:/home/admin#
```

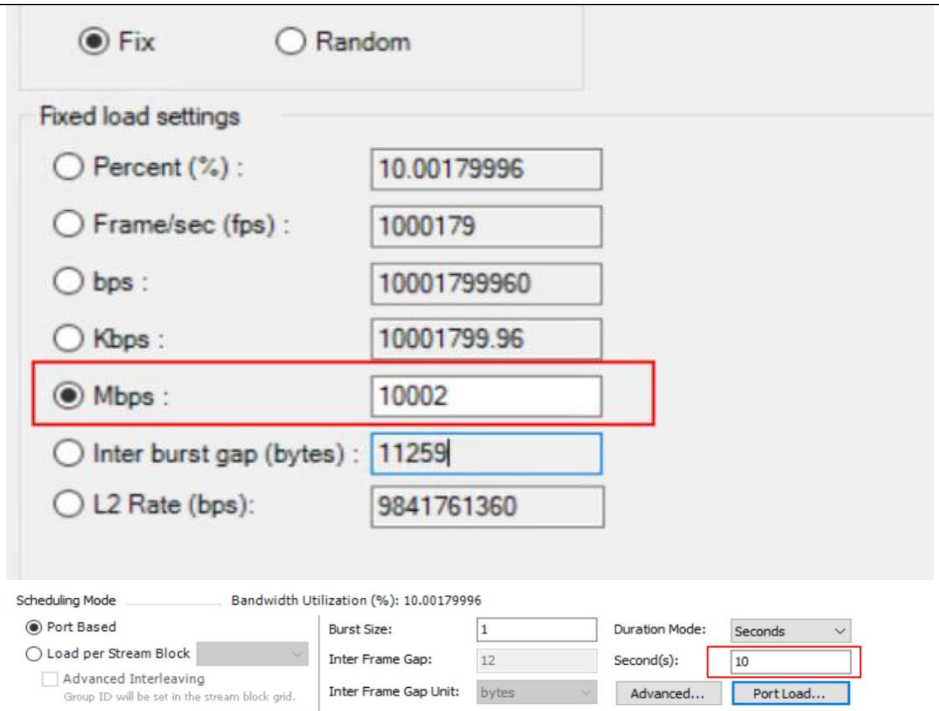
Configure egress rate limiting to 10G.
config scheduler add profile-1 --shaper_type=bytes --bandwidth=10g
config interface scheduler bind port Ethernet8 profile-1

Configure vlan
config vlan add 100
config vlan member add 100 Ethernet0 -u
config vlan member add 100 Ethernet8 -u

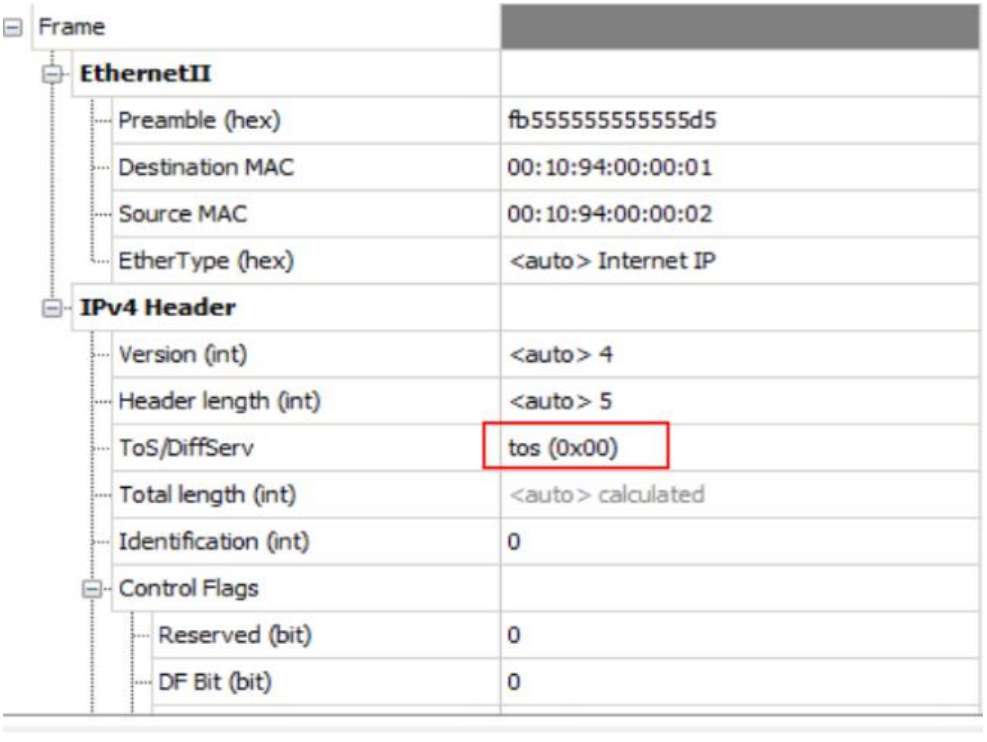
TGB continuously sends MAC learning packets, and TGA sends packets to TGB at a rate of 10.002 Gb.
port A


Frame
EthernetII
Preamble (hex)
fb555555555555d5
Destination MAC
00:10:94:00:00:02
Source MAC
00:10:94:00:00:01
EtherType (hex)
<auto> Internet IP
IPv4 Header
Version (int)
<auto> 4
Header length (int)
<auto> 5
ToS/DiffServ
tos (0x00)
Total length (int)
<auto> calculated
Identification (int)
0
Control Flags
Reserved (bit)
0
DF Bit (bit)
0
General
Frame
Groups
Rx Port
Preview
Active
Name:
StreamBlock 41
Frame size (Bytes)(With CRC and signature field)
Fixed
Size:
1230
Increment
Step:
1
(power of 2)
Decrement
Min:
128
Random
Max:
256
Auto
Avg:
192
iMIX
Default
Edit...

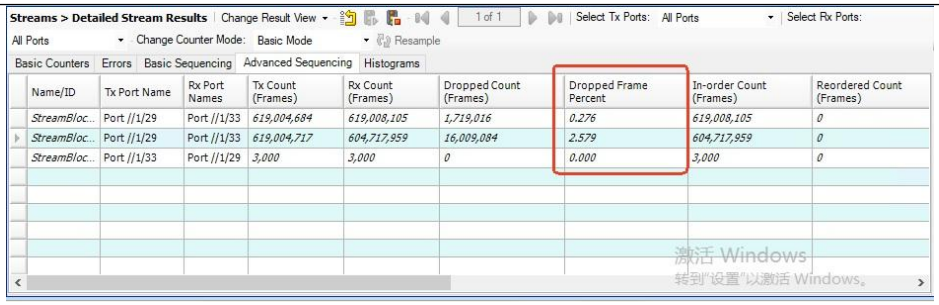

Frame
EthernetII
Preamble (hex)
fb5555555555d5
Destination MAC
00:10:94:00:00:02
Source MAC
00:10:94:00:00:01
EtherType (hex)
<auto> Internet IP
IPv4 Header
Version (int)
<auto> 4
Header length (int)
<auto> 5
ToS/DiffServ
tos (0x0C)
Custom Editor
Quality of Service
DiffServ
ToS
Value
(Hex)
(Dec)
(Binary)
0C
12
00001100
Precedence
Delay
General
Frame
Groups
Rx Port
Preview
Active
Name:
StreamBlock 42
Frame size (Bytes)(With CRC and signature field)
Fixed
Size:
1230
Increment
Step:
1
(power of 2)
Decrement
Min:
128
Random
Max:
256



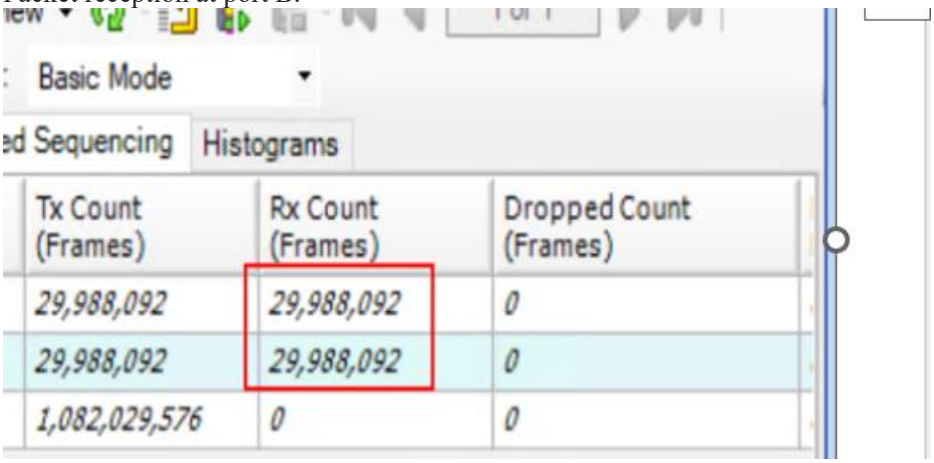
Port B



The packet reception at port B (the packet loss ratio is close to 1:10).

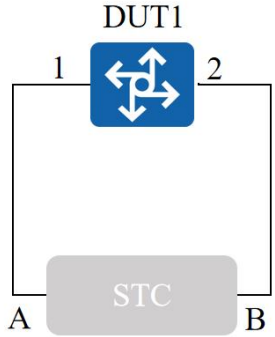


Remove WRED binding.
The instrument configuration remains unchanged.
Packet reception at port B.

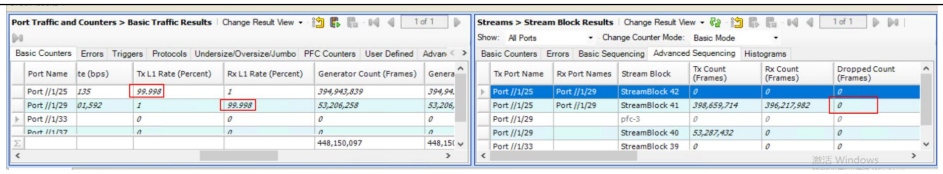


Test results	**PASS**
Remarks	

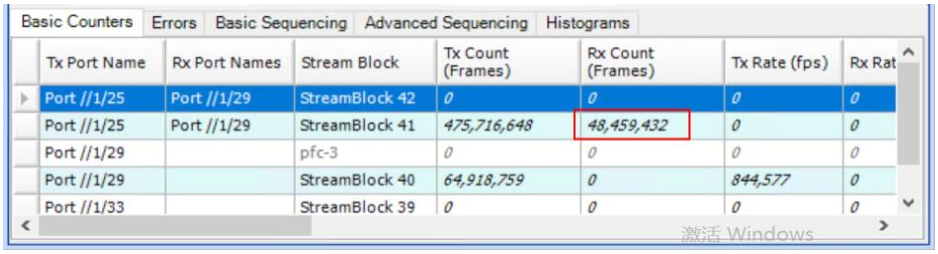
5.13 ECN

Test Items	ECN Functional Test
Test Content	The testing equipment can forward packets according to the ECN (Explicit Congestion Notification) settings.
Test Topology	
Test Steps	1、Set up the test environment according to the diagram. 2、Create VLAN 100 and add DUT P1 and DUT P2 to VLAN 100 as untagged members. 3、Configure a WRED profile and bind it to the port ， check1.

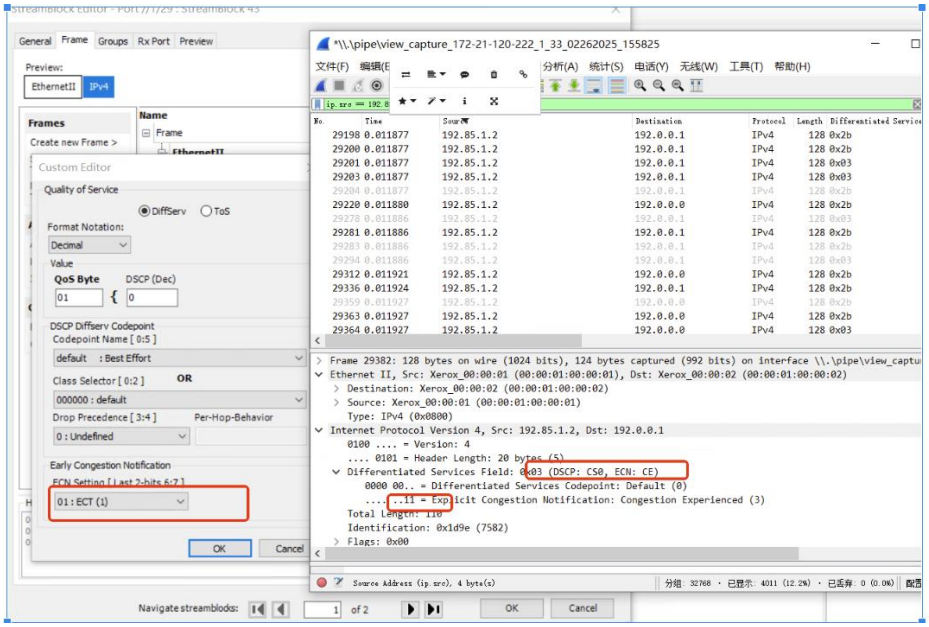
	4、Check the ECN configuration. 5、Configure a QoS profile, bind it to the port, and map DSCP 0 to Queue 0 ， check2. 6、Set the egress rate limit to 10G ,check3. 7、TGB continuously sends MAC learning packets. 8、TGA sends IPv4 packets with ECN 0x01 at line rate. 9、Capture packets at TGB for inspection ,check4. 10、TGA sends IPv4 packets with ECN 0x02 at line rate. 11、Capture packets at TGB for inspection ,check5. 12、TGA sends IPv6 packets with ECN 0x01 at line rate. 13、Capture packets at TGB for inspection. 14、TGA sends IPv6 packets with ECN 0x02 at line rate. 15、Capture packets at TGB for inspection.
Expected Results	Check1: Configuration is successful. Check2: No packet loss in packet forwarding. Check3: Packet loss occurs according to WRED settings. Check4: ECN captured by TGB is 3. Check5: ECN captured by TGB is 3.
Actual Test Results	Configure VLAN config vlan add 100 config vlan member add 100 Ethernet0 -u config vlan member add 100 Ethernet8 -u config wred add wred-prof --mode ecn --gmin 100000 --gmax 800000 --gdrop 100 config interface wred bind queue Ethernet0 0 wred-prof config interface wred bind queue Ethernet8 0 wred-prof show ecn Configure a QoS profile, bind it to the port, and map DSCP 0 to Queue 0. config qos dscp-tc add dscp_to_tc_profile --dscp 0 --tc 0 config interface qos dscp-tc bind Ethernet0 dscp_to_tc_profile config interface qos dscp-tc bind Ethernet8 dscp_to_tc_profile config qos tc-pg add tc_to_pg_profile --tc 0 --pg 0 config interface qos tc-pg bind Ethernet0 tc_to_pg_profile config interface qos tc-pg bind Ethernet8 tc_to_pg_profile config qos tc-queue add tc_to_queue_profile --tc 0 --queue 0 config interface qos tc-queue bind Ethernet0 tc_to_queue_profile config interface qos tc-queue bind Ethernet8 tc_to_queue_profile



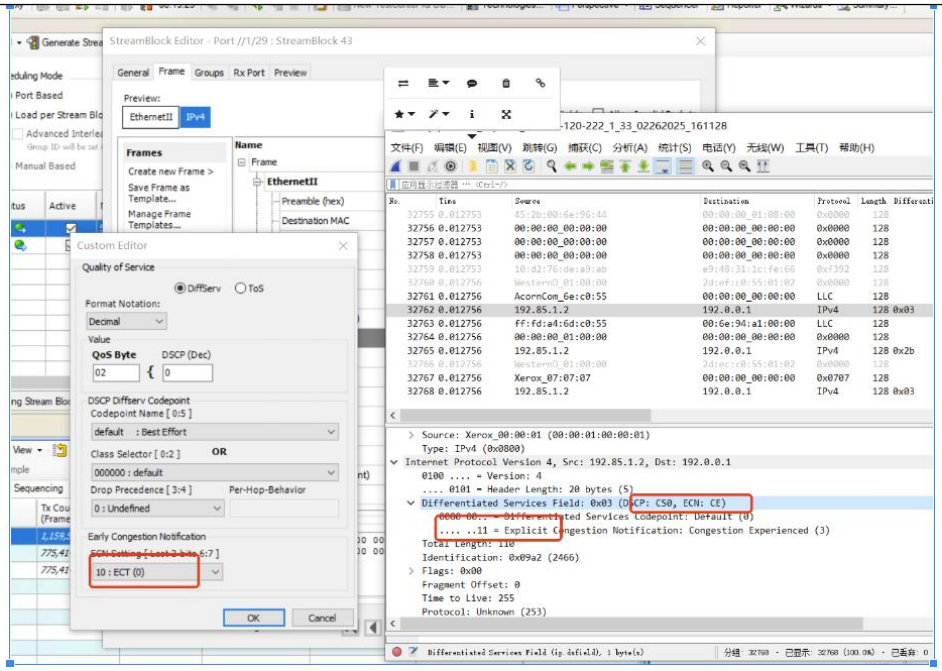
config scheduler add profile-1 --shaper_type=bytes --bandwidth=10g
config interface scheduler bind port Ethernet8 profile-1



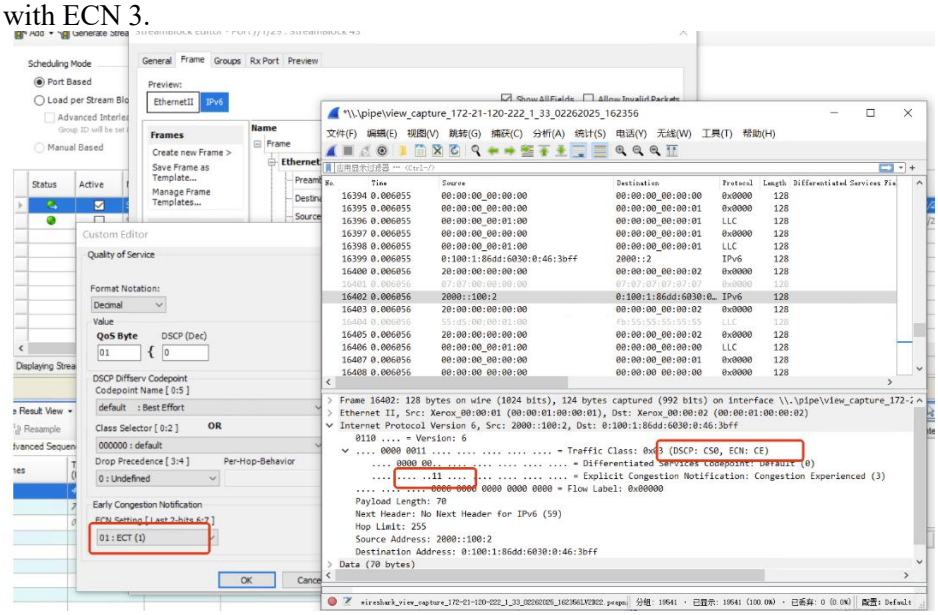
TGA sends DSCP0 packets with ECN 1 at line rate, and TGB captures packets with ECN



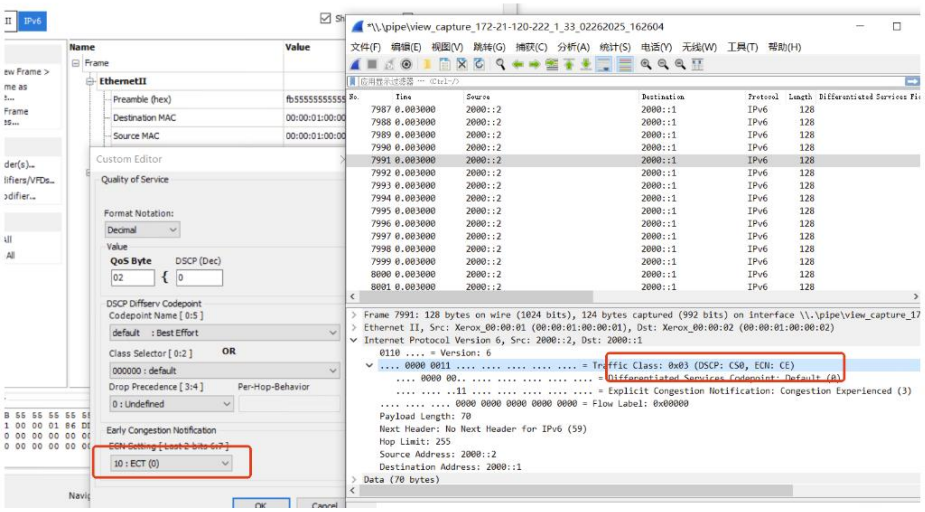
TGA sends DSCP0 packets with ECN 2 at line rate, and TGB captures packets with ECN 3.



TGA sends DSCP0 IPv6 packets with ECN 1 at line rate, and TGB captures packets with ECN 3.

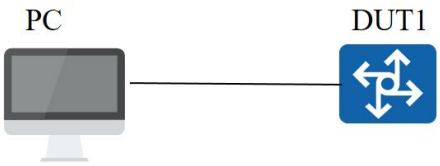


TGA sends DSCP0 IPv6 packets with ECN 2 at line rate, and TGB captures packets with ECN 3.

	
Test results	**PASS**
Remarks	

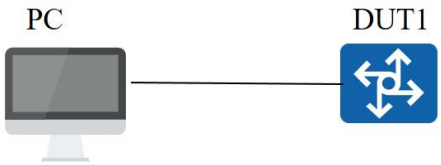
六、System/ Management

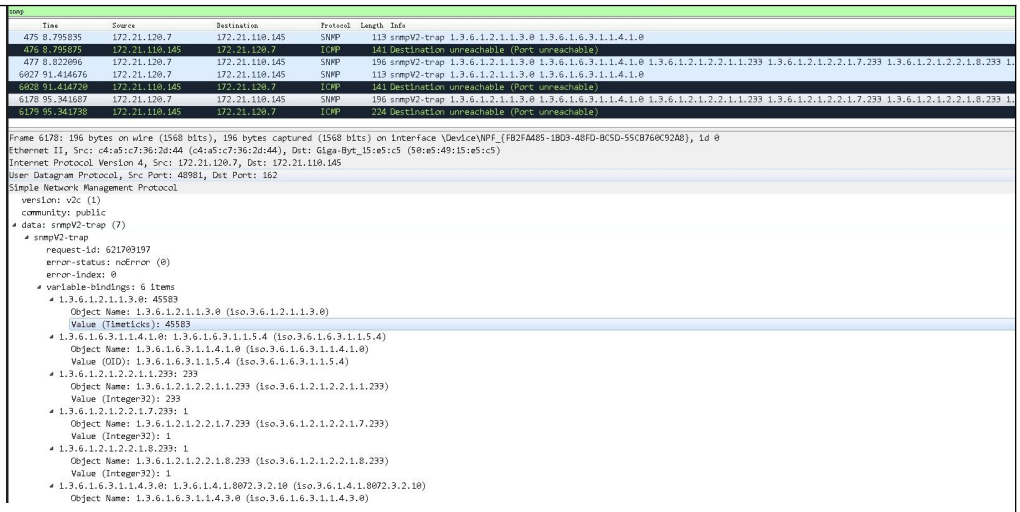
6.1 SNMP

Test Items	SNMP Support Test
Test Content	Verify that the switch supports the SNMP.
Test Topology	
Test Steps	1、Connect the host with MIB software installed to the switch. 2、Configure the SNMP community string and agent settings on the switch. 3、Use the snmpwalk tool to query MIB information，check1.
Expected Results	Check1：MIB information can be successfully retrieved.,
Actual Test Results	Step 2： config snmp community add testcomm rw config snmpagentaddress add x.x.x.x Step 3： snmpwalk -c testcomm -v 2c x.x.x.x F:\xijun>snmpget.exe -v 2c -c comtest 172.21.120.7 1.3.6.1.2.1.1.5.0 SNMPv2-MIB::sysName.0 = STRING: sonic

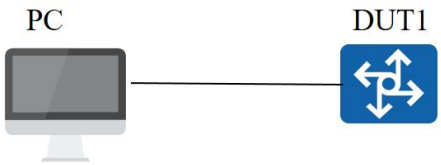
	F:\xijun>snmpwalk.exe -v 2c -c comtest 172.21.120.7 1.3.6.1.2.1.1.5.0 SNMPv2-MIB::sysName.0 = STRING: sonic
Test results	**PASS**
Remarks	

6.2 SNMP Trap

Test Items	SNMP trap
Test Content	Verify that the device sends SNMP traps as per the configured trap settings.
Test Topology	
Test Steps	1、Set up the test environment as illustrated. 2、Configure the device to send SNMP traps to the PC and start the SNMP trap server on the PC. 3、View relevant logs on the PC.
Expected Results	1、The configuration is successful. 2、The SNMP Trap server receives relevant trap messages from the device.
Actual Test Results	Configure snmp trap server config snmptrap modify 2 -c public x.x.x.x root@sonic:/home/admin# config interface shutdown Ethernet232 root@sonic:/home/admin# config interface startup Ethernet232 Show snmp trap Configure show snmptrap root@sonic:/home/admin# config snmptrap modify 2 -c public 172.21.110.145 root@sonic:/home/admin# show snmptrap Version TrapReceiverIP Port VRF Community --------- ---------- ------ ----- ----------- 2 172.21.110.145 162 None public root@sonic:/home/admin#

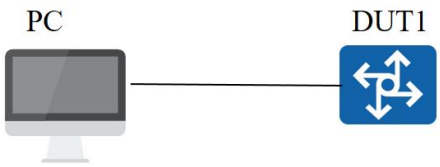
	
Test results	**PASS**
Remarks	

6.3 Rsyslog

Test Items	Rsyslog
Test Content	Verify that the device sends system log messages as per the rsyslog configuration.
Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2、Configure the device to send syslog to PC1 and start the log server on PC1. 3、View relevant logs on PC1.
Expected Results	2、The configuration is successful. 3、The log server receives relevant logs from the device.
Actual Test Results	Configure a remote syslog server. config syslog add x.x.x.x root@sonic:/home/admin# config syslog add 172.21.110.145 Show syslog 172.21.120.7 03/03 14:31:52.101 <86>Mar 3 14:34:11.071570 sonic INFO sudo[225503]: pam_unix(sudo:session): session opened for user root(uid=0) by admin(uid=0) 172.21.120.7 03/03 14:31:52.101 <35>Mar 3 14:34:11.131290 sonic ERR PAM-tacplus[225503]: ACC: TACACS+ service type not configured

	172.21.120.7 03/03 14:31:52.160 <86>Mar 3 14:34:11.131402 sonic INFO sudo[225503]: pam_unix(sudo:session): session closed for user root 172.21.120.7 03/03 14:31:52.161 <14>Mar 3 14:34:54.990603 sonic INFO syncd#syncd: [none] _brcm_sai_link_event_cb:870 Send port 165 up notification done. 172.21.120.7 03/03 14:32:36.020 <13>Mar 3 14:34:54.991438 sonic NOTICE swss#orchagent: :- doTask: Get port state change notification id:1000000000020 status:1 172.21.120.7 03/03 14:32:36.021 <13>Mar 3 14:34:54.991438 sonic NOTICE swss#orchagent: :- updatePortOperStatus: Port Ethernet232 oper state set from down to up 172.21.120.7 03/03 14:32:36.022 <13>Mar 3 14:34:54.993034 sonic NOTICE iccpd#iccpd: [iccp_event_handler_obj_input_newlink.NOTICE] Update local port Ethernet232 state up 172.21.120.7 03/03 14:32:36.022 <13>Mar 3 14:34:54.993127 sonic NOTICE iccpd#iccpd: [iccp_event_handler_obj_input_newaddr.NOTICE] ifname Ethernet232 index 60 address fe80::8e5d:b2ff:feb6:0 l3_proto 0, prefix_len 64 172.21.120.7 03/03 14:32:36.023 <13>Mar 3 14:34:54.993225 sonic NOTICE swss#portsyncd: :- onMsg: nlmsg type:16 key:Ethernet232 admin:1 oper:1 addr:8c:5d:b2:b6:00:00 ifindex:60 master:0 172.21.120.7 03/03 14:32:36.024 <13>Mar 3 14:34:54.993424 sonic NOTICE swss#orchagent: :- setHostIntfsOperStatus: Set operation status UP to host interface Ethernet232 172.21.120.7 03/03 14:32:36.024 <13>Mar 3 14:34:54.993424 sonic NOTICE swss#orchagent: :- updateMonitorLinkGrpDownlinkOperStatus: update port Ethernet232, oper status 1 172.21.120.7 03/03 14:32:36.025 <6>Mar 3 14:34:54.994522 sonic INFO kernel: [3699.737388] IPv6: ADDRCONF(NETDEV_CHANGE): Ethernet232: link becomes ready 172.21.120.7 03/03 14:32:36.026 <13>Mar 3 14:34:54.994632 sonic NOTICE swss#portsyncd: :- onMsg: Publish Ethernet232(ok:up) to state db
Test results	**PASS**
Remarks	

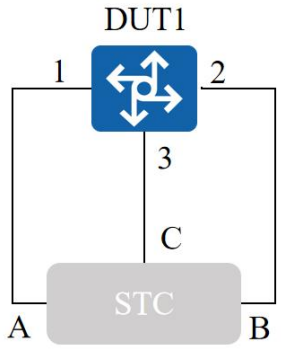
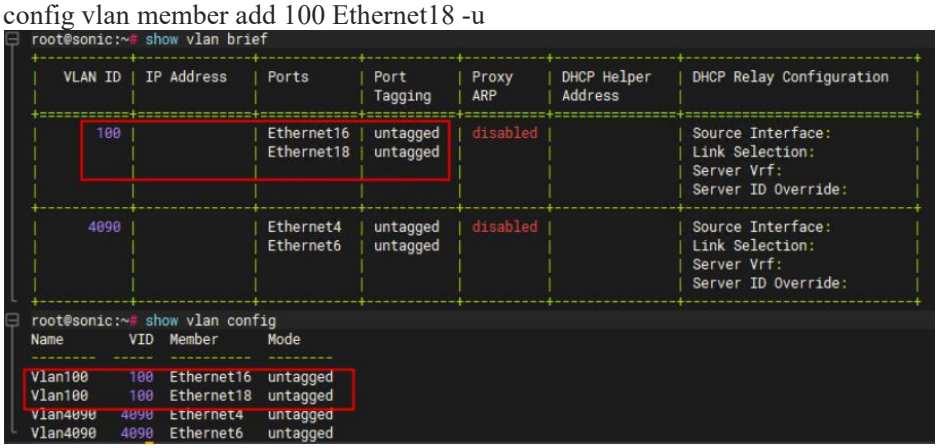
6.4 Open SSH/SCP/SFTP

Test Items	Open SSH/SCP/SFTP
Test Content	Test that the switch supports Open SSH/SCP/SFTP.
Test Topology	
Test Steps	1、Use the command line on PC1 to remotely log in via SSH: ssh admin@DUT_IP (follow prompts).

	2、After successful login, use config and show commands for configuration. 3、Use SCP on PC1 to transfer a file to the switch: scp testfile admin@DUT_IP:testfile. 4、Use SCP on PC1 to retrieve a file from the switch: scp admin@DUT_IP:testfile ./testfile. 5、Use SFTP on PC1 to remotely log in: sftp admin@DUT_IP (follow prompts). 6、Within the SFTP interactive shell, upload a file: put testfile. 7、Within the SFTP interactive shell, download a file: get testfile.
Expected Results	1、Login is successful. 2、Commands take effect. 3-4. File transfer is successful. 5、SFTP login is successful. 6-7. File transfer via SFTP is successful.
Actual Test Results	The SSH and SFTP services are enabled by default. ssh sonic login: admin Password: Linux sonic 5.10.0-8-2-amd64 #1 SMP Debian 5.10.46-4 (2021-08-03) x86_64 You are on -- Software for Open Networking in the Cloud -- Unauthorized access and/or use are prohibited. All access and/or use are subject to monitoring. Help: http://azure.github.io/SONiC/ Last login: Fri Feb 28 17:23:03 CST 2025 on ttyS0 scp E:\Version\CN9500-64OC>scp admin@172.21.120.7:/tmp/gu/test_scp.txt E:\Version\CN9500-64OC admin@172.21.120.7's password: E:\Version\CN9500-64OC> sftp sftp> put E:\Version\CN9500-64OC\test_sftp.txt Uploading E:/Version/CN9500-64OC/test_sftp.txt to /tmp/gu/test_sftp.txt test_sftp.txt 100% 0 0.0KB/s 00:00 sftp> sftp> get test_sftp.txt E:\Version\CN9500-64OC Fetching /tmp/gu/test_sftp.txt to E:/Version/CN9500-64OC/test_sftp.txt sftp>
Test results	**PASS**

Remarks	

6.5 Mirroring

Test Items	Mirroring
Test Content	MirroringVerify whether the DUT (Device Under Test) supports Mirroring.
Test Topology	
Test Steps	1、Connect the devices as per the test environment. 2、Configure port mirroring on the DUT, with Interface 1 as the source port and Interface 3 as the destination port. 3、Send data packets from Port A to Port B of the test instrument. 4、Reconfigure port mirroring on the DUT, setting Interface 1 and Interface 2 as source ports and Interface 3 as the destination port, then send data packets between Port A and Port B of the test instrument.
Expected Results	3、The test traffic sent from Port A can also be observed on DUT's Interface 3. 4、The test traffic sent from both Port A and Port B can be observed on DUT's Interface 3.
Actual Test Results	Vlan config vlan add 100 config vlan member add 100 Ethernet16 -u config vlan member add 100 Ethernet18 -u  Step 3：

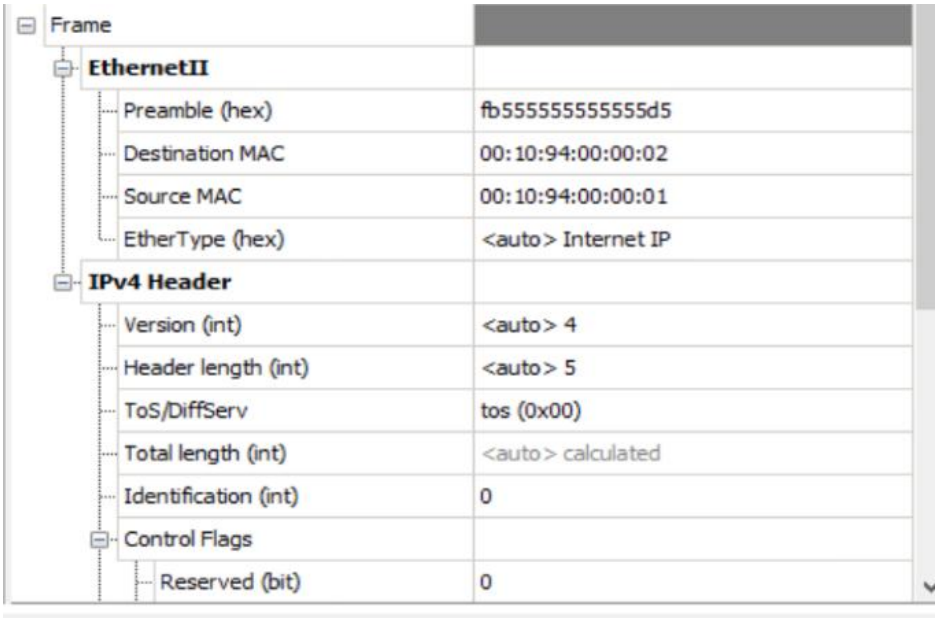
```
root@sonic:~# show vlan brief
+-----------+--------------+-----------+------------+---------+---------------+--------------------------+
| VLAN ID   | IP Address   | Ports     | Port       | Proxy   | DHCP Helper   | DHCP Relay Configuration |
|           |              |           | Tagging    | ARP     | Address       |                          |
+===========+==============+===========+============+=========+===============+==========================+
|       100 |              | Ethernet16| untagged   | disabled|               | Source Interface:        |
|           |              | Ethernet18| untagged   |         |               | Link Selection:          |
|           |              |           |            |         |               | Server Vrf:              |
|           |              |           |            |         |               | Server ID Override:      |
+-----------+--------------+-----------+------------+---------+---------------+--------------------------+
|      4090 |              | Ethernet4 | untagged   | disabled|               | Source Interface:        |
|           |              | Ethernet6 | untagged   |         |               | Link Selection:          |
|           |              |           |            |         |               | Server Vrf:              |
|           |              |           |            |         |               | Server ID Override:      |
+-----------+--------------+-----------+------------+---------+---------------+--------------------------+
root@sonic:~# show vlan config
Name        VID  Member      Mode
--------  -----  ----------  --------
Vlan100     100  Ethernet16  untagged
Vlan100     100  Ethernet18  untagged
Vlan4090   4090  Ethernet4   untagged
Vlan4090   4090  Ethernet6   untagged
```

config mirror_session span add port0 Ethernet16 Ethernet0

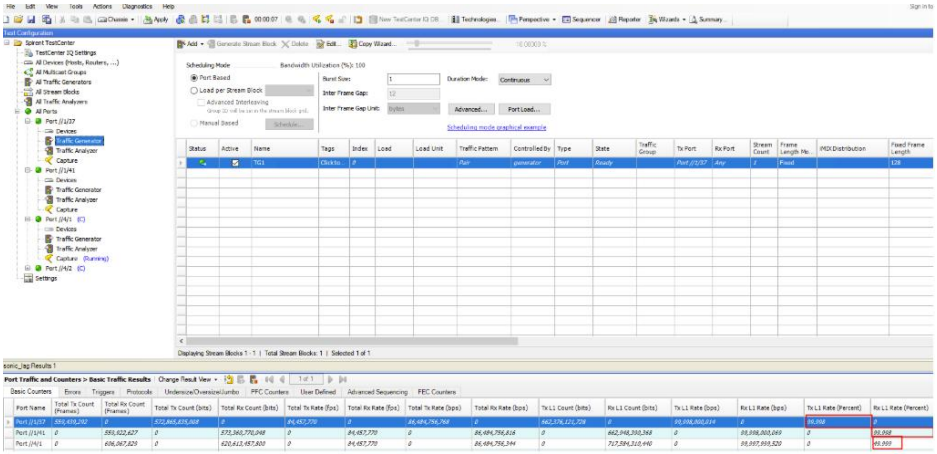
```
root@sonic:~# show mirror_session
ERSPAN Sessions
Name    Status    SRC IP    DST IP    GRE    DSCP    TTL    Queue    Policer    Monitor Port    SRC Port    Direction
------  --------  --------  --------  -----  ------  -----  -------  ---------  --------------  ----------  -----------

SPAN Sessions
Name    Status    DST Port    SRC Port    Direction    Queue    Policer
------  --------  ----------  ----------  -----------  -------  ---------

root@sonic:~#
root@sonic:~# config mirror_session span add port0 Ethernet20 Ethernet16
root@sonic:~#
root@sonic:~# show mirror_session
ERSPAN Sessions
Name    Status    SRC IP    DST IP    GRE    DSCP    TTL    Queue    Policer    Monitor Port    SRC Port    Direction
------  --------  --------  --------  -----  ------  -----  -------  ---------  --------------  ----------  -----------

SPAN Sessions
Name    Status    DST Port    SRC Port    Direction    Queue    Policer
------  --------  ----------  ----------  -----------  -------  ---------

port0   active    Ethernet20  Ethernet16  both         MC0
```

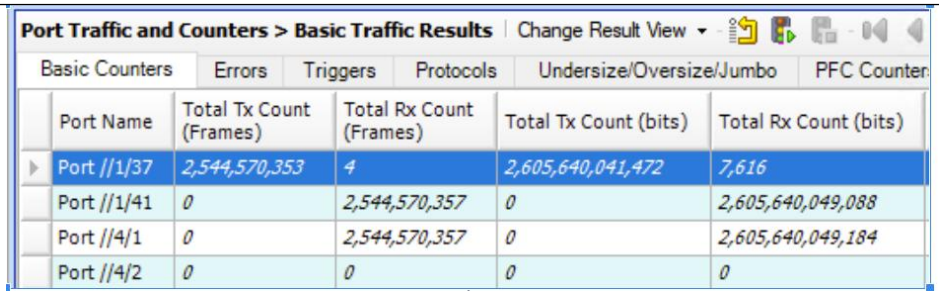
TGA sends packets to TGB.

Port A



Port C





Port Traffic and Counters > Basic Traffic Results | Change Result View

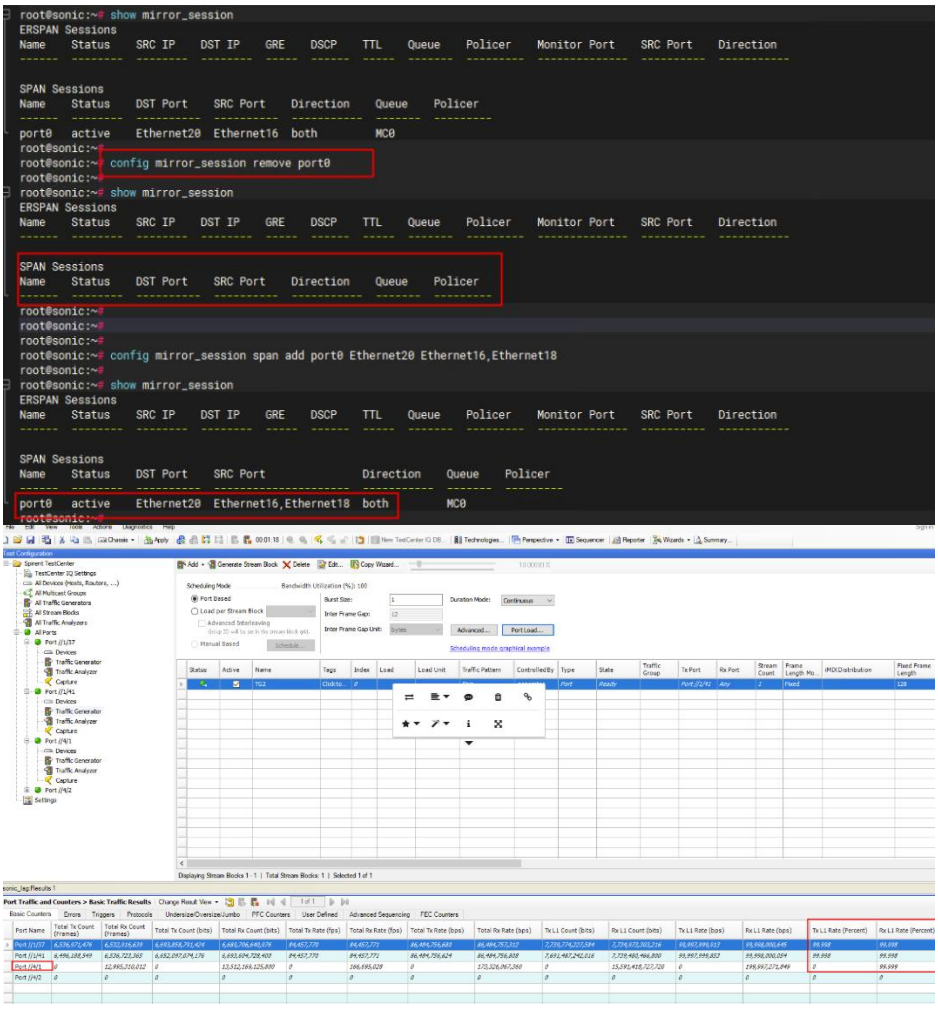
Basic Counters | Errors | Triggers | Protocols | Undersize/Oversize/Jumbo | PFC Counter

Port Name	Total Tx Count (Frames)	Total Rx Count (Frames)	Total Tx Count (bits)	Total Rx Count (bits)
Port //1/37	2,544,570,353	4	2,605,640,041,472	7,616
Port //1/41	0	2,544,570,357	0	2,605,640,049,088
Port //4/1	0	2,544,570,357	0	2,605,640,049,184
Port //4/2	0	0	0	0

Step 4：

config mirror_session remove port0
config mirror_session span add port0 Ethernet16 Ethernet20,Ethernet18

Port C

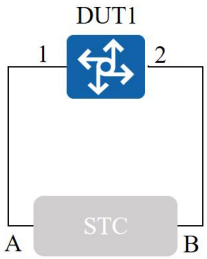
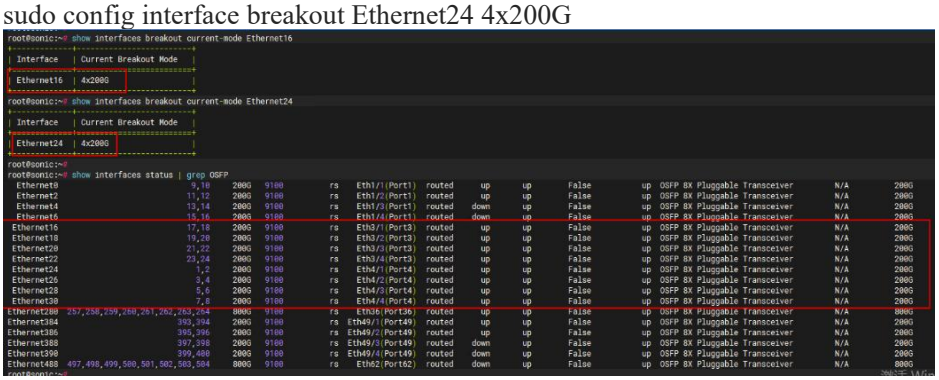


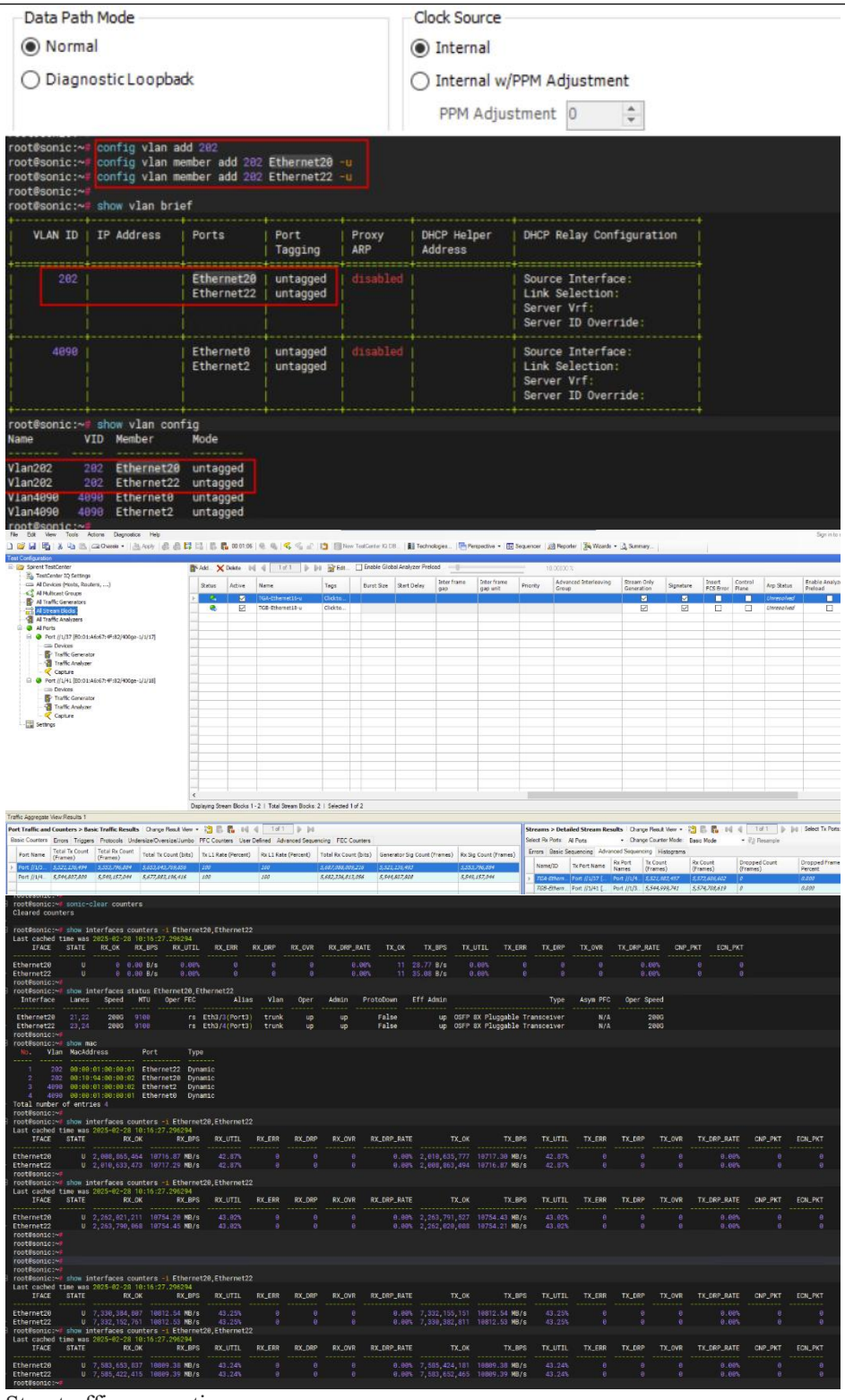
```
root@sonic:~# show mirror_session
ERSPAN Sessions
Name    Status    SRC IP    DST IP    GRE    DSCP    TTL    Queue    Policer    Monitor Port    SRC Port    Direction
------  --------  --------  --------  -----  ------  -----  -------  ---------  --------------  ----------  -----------

SPAN Sessions
Name    Status    DST Port    SRC Port    Direction    Queue    Policer
------  --------  ----------  ----------  -----------  -------  ---------
port0   active    Ethernet20  Ethernet16  both         MC0
root@sonic:~#
root@sonic:~# config mirror_session remove port0
root@sonic:~#
root@sonic:~# show mirror_session
ERSPAN Sessions
Name    Status    SRC IP    DST IP    GRE    DSCP    TTL    Queue    Policer    Monitor Port    SRC Port    Direction
------  --------  --------  --------  -----  ------  -----  -------  ---------  --------------  ----------  -----------

SPAN Sessions
Name    Status    DST Port    SRC Port    Direction    Queue    Policer
------  --------  ----------  ----------  -----------  -------  ---------
root@sonic:~#
root@sonic:~#
root@sonic:~#
root@sonic:~# config mirror_session span add port0 Ethernet20 Ethernet16,Ethernet18
root@sonic:~#
root@sonic:~# show mirror_session
ERSPAN Sessions
Name    Status    SRC IP    DST IP    GRE    DSCP    TTL    Queue    Policer    Monitor Port    SRC Port    Direction
------  --------  --------  --------  -----  ------  -----  -------  ---------  --------------  ----------  -----------

SPAN Sessions
Name    Status    DST Port    SRC Port              Direction    Queue    Policer
------  --------  ----------  --------------------  -----------  -------  ---------
port0   active    Ethernet20  Ethernet16,Ethernet18  both         MC0
root@sonic:~#
```

Test results	**PASS**
Remarks	

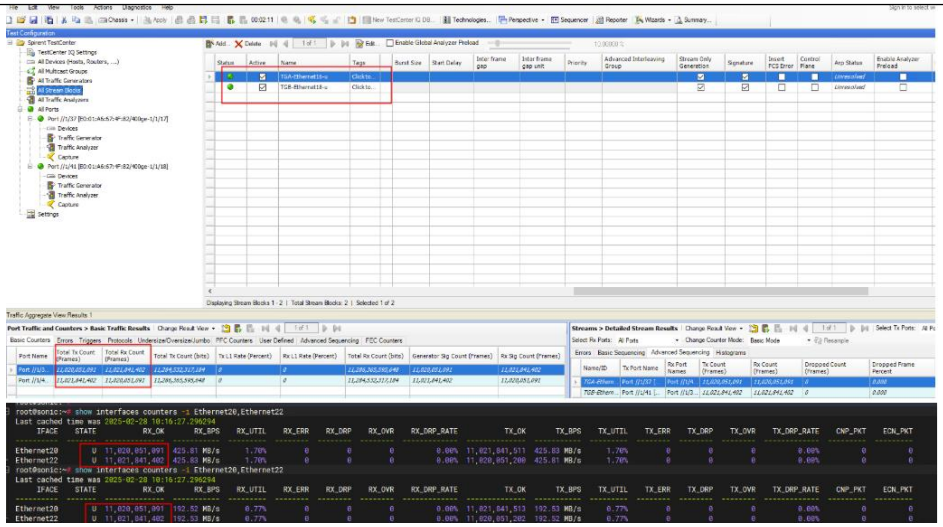
6.6 DPB (Dynamic Port Breakout)

Test Items	Dynamic Port Breakout
Test Content	Test that the switch supports Dynamic Port Breakout.

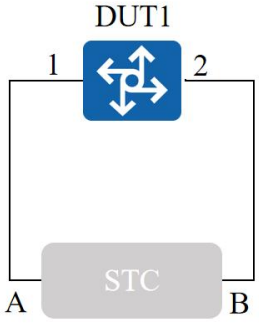
Test Topology	
Test Steps	Modify DUT ports 472, 480, and 488 to 2x200G mode.
Expected Results	The ports can be split correctly, and the docking of the remaining ports can be up.
Actual Test Results	The optical module is in place to modify the 2x200G mode. config interface breakout Ethernet472 2x200G config interface breakout Ethernet480 2x200G config interface breakout Ethernet488 2x200G config interface startup Ethernet472 config interface startup Ethernet476 config interface startup Ethernet480 config interface startup Ethernet488 The ports are 800G ports by default. Split the ports into 200G ports using the CLI commands: sudo config interface breakout Ethernet16 4x200G sudo config interface breakout Ethernet24 4x200G  Connect DUT's Ethernet20/22 to the instrument ports for data forwarding according to the topology diagram. The results are as follows:



Stop traffic generation.

	
Test results	**Pass**
Remarks	

6.8 COPP

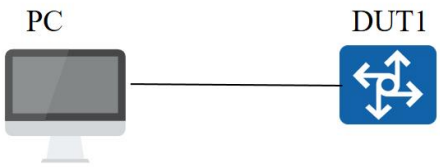
Test Items	COPP
Test Content	Test that the device can resist attacks after enabling COPP.
Test Topology	
Test Steps	1、Set up the test environment as shown in the diagram. 2、Configure DUT P1 IP as 100.100.100.100/24 and P2 IP as 200.200.200.200/24. 3、Check the device CPU load. 4、Send packets to the device CPU at line rate via TGA, and check the device CPU load. 5、Test ARP miss, wait for 5 minutes, and check the device CPU load.
Expected Results	4.5. The CPU utilization of the device is within the normal range.

Actual Test Results	Port 1 IP:100.100.100.100/24, Port 2 IP:200.200.200.200/24 config interface ip add Ethernet0 100.100.100.100/24 config interface ip add Ethernet8 200.200.200.200/24 TGA sends packets at line rate with the destination MAC as the device CPU MAC and the destination IP as 100.100.100.100. Observe for 5 minutes and capture the average value. root@sonic:/home/admin# top top - 10:00:52 up 7:33, 1 user, load average: 0.65, 0.83, 0.88 Tasks: 406 total, 1 running, 402 sleeping, 0 stopped, 3 zombie %Cpu(s): 2.5 us, 2.2 sy, 0.0 ni, 95.1 id, 0.0 wa, 0.0 hi, 0.2 si, 0.0 st MiB Mem : 15160.7 total, 9805.4 free, 3332.4 used, 2022.9 buff/cache MiB Swap: 0.0 total, 0.0 free, 0.0 used. 11386.4 avail Mem PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND 11627 root 20 0 3829364 850656 228728 S 15.9 5.5 40:55.94 syncd 8740 root 20 0 263944 32932 12744 S 8.0 0.2 28:23.35 python3 5615 root 20 0 111012 58524 7400 S 5.0 0.4 16:27.16 redis-server 5746 root 20 0 42136 26396 13224 S 4.7 0.2 9:06.98 procdockerstats 5739 root 20 0 104884 21296 10720 S 2.7 0.1 9:42.95 main.py 5616 root 20 0 85412 38916 7360 S 1.3 0.3 6:44.89 redis-server 25898 root 20 0 199104 38888 13664 S 1.3 0.3 6:20.35 python3 8108 root 20 0 17404 10944 6236 S 1.0 0.1 3:17.75 python 13 root 20 0 0 0 0 I 0.3 0.0 0:11.02 rcu_sched 38 root 20 0 0 0 0 S 0.3 0.0 0:02.53 ksoftirqd/5 849 root 20 0 2985472 67904 35268 S 0.3 0.4 3:08.59 containerd 8112 root 20 0 17804 11164 6312 S 0.3 0.1 0:39.65 python 8741 root 20 0 37148 25340 12252 S 0.3 0.2 2:19.75 python3 13469 admin 20 0 11004 4564 3980 S 0.3 0.0 0:11.56 vty_log.sh 25894 root 20 0 2426288 186632 20156 S 0.3 1.2 1:19.96 telemetry 26060 root 20 0 2496040 186292 18228 S 0.3 1.2 1:21.52 dialout_client_ 1679555 root 20 0 11108 4304 3284 R 0.3 0.0 0:00.03 top 1 root 20 0 100464 11812 8044 S 0.0 0.1 0:08.93 systemd 2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd 3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp 4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp 6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-kblockd 7 root 20 0 0 0 0 I 0.0 0.0 0:00.11 kworker/0:1-cgroup_destroy 9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq 10 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu_tasks_rude_ 11 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu_tasks_trace 12 root 20 0 0 0 0 S 0.0 0.0 0:02.88 ksoftirqd/0 14 root rt 0 0 0 0 S 0.0 0.0 0:00.05 migration/0 15 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0 16 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/1 17 root rt 0 0 0 0 S 0.0 0.0 0:00.22 migration/1 18 root 20 0 0 0 0 S 0.0 0.0 0:02.76 ksoftirqd/1 20 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/1:0H-events_highpri 21 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/2 22 root rt 0 0 0 0 S 0.0 0.0 0:00.23 migration/2 23 root 20 0 0 0 0 S 0.0 0.0 0:02.67 ksoftirqd/2 25 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/2:0H-events_highpri

	26 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/3 27 root rt 0 0 0 0 S 0.0 0.0 0:00.23 migration/3 28 root 20 0 0 0 0 S 0.0 0.0 0:02.47 ksoftirqd/3 30 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/3:0H-events_highpri 31 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/4 32 root rt 0 0 0 0 S 0.0 0.0 0:00.23 migration/4 33 root 20 0 0 0 0 S 0.0 0.0 0:02.53 ksoftirqd/4 35 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/4:0H-events_highpri 36 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/5 37 root rt 0 0 0 0 S 0.0 0.0 0:00.24 migration/5 40 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/5:0H-kblockd 41 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/6 root@sonic:/home/admin# TGA sends packets at line rate with the destination MAC as the device CPU MAC and the destination IP as 200.200.200.200. After waiting for 5 minutes. check the device CPU load. top root@sonic:/home/admin# top top - 10:01:42 up 7:33, 1 user, load average: 0.43, 0.74, 0.85 Tasks: 410 total, 2 running, 405 sleeping, 0 stopped, 3 zombie %Cpu(s): 2.3 us, 2.1 sy, 0.0 ni, 95.6 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st MiB Mem : 15160.7 total, 9775.3 free, 3362.3 used, 2023.0 buff/cache MiB Swap: 0.0 total, 0.0 free, 0.0 used. 11357.2 avail Mem PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND 11627 root 20 0 3829364 850656 228728 S 19.9 5.5 41:00.65 syncd 8740 root 20 0 263944 32932 12744 S 7.3 0.2 28:26.36 python3 5615 root 20 0 111012 58352 7400 S 4.0 0.4 16:28.98 redis-server 5616 root 20 0 85412 39116 7360 S 3.3 0.3 6:45.67 redis-server 5739 root 20 0 104884 21296 10720 S 3.0 0.1 9:44.04 main.py 25898 root 20 0 199104 38888 13664 S 1.7 0.3 6:21.08 python3 1683142 root 20 0 1790756 64456 29572 S 1.7 0.4 0:00.05 docker 8108 root 20 0 17404 10944 6236 S 1.0 0.1 3:18.11 python 1302 root 20 0 3384108 130064 50576 S 0.7 0.8 4:08.10 dockerd 8741 root 20 0 37148 25340 12252 S 0.7 0.2 2:20.02 python3 11717 root 20 0 406040 24904 13564 S 0.7 0.2 0:25.88 orchagent 145 root 20 0 0 0 0 I 0.3 0.0 0:00.69 kworker/9:1-events 8107 root 20 0 19080 12524 6280 S 0.3 0.1 0:53.01 python 9756 root 20 0 29808 24896 8928 S 0.3 0.2 0:04.21 supervisord 14376 300 20 0 1568560 133248 11712 S 0.3 0.9 0:01.88 zebra 25894 root 20 0 2426288 186632 20156 S 0.3 1.2 1:20.05 telemetry 26060 root 20 0 2496040 186292 18228 S 0.3 1.2 1:21.60 dialout_client_ 1682959 root 20 0 11104 4228 3212 R 0.3 0.0 0:00.02 top 1683213 root 20 0 6212 1984 1732 R 0.3 0.0 0:00.01 du 1 root 20 0 100464 11812 8044 S 0.0 0.1 0:08.93 systemd 2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd 3 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_gp 4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 rcu_par_gp 6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-kblockd 7 root 20 0 0 0 0 I 0.0 0.0 0:00.11 kworker/0:1-cgroup_destroy 9 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 mm_percpu_wq 10 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu_tasks_rude_ 11 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcu_tasks_trace 12 root 20 0 0 0 0 S 0.0 0.0 0:02.88 ksoftirqd/0 13 root 20 0 0 0 0 I 0.0 0.0 0:11.04 rcu_sched

	14 root rt 0 0 0 0 S 0.0 0.0 0:00.05 migration/0 15 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0 16 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/1 17 root rt 0 0 0 0 S 0.0 0.0 0:00.22 migration/1 18 root 20 0 0 0 0 S 0.0 0.0 0:02.76 ksoftirqd/1 20 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/1:0H-events_highpri 21 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/2 22 root rt 0 0 0 0 S 0.0 0.0 0:00.23 migration/2 23 root 20 0 0 0 0 S 0.0 0.0 0:02.67 ksoftirqd/2 25 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/2:0H-events_highpri 26 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/3 27 root rt 0 0 0 0 S 0.0 0.0 0:00.23 migration/3 28 root 20 0 0 0 0 S 0.0 0.0 0:02.47 ksoftirqd/3 30 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/3:0H-events_highpri 31 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/4 32 root rt 0 0 0 0 S 0.0 0.0 0:00.23 migration/4 33 root 20 0 0 0 0 S 0.0 0.0 0:02.53 ksoftirqd/4 35 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/4:0H-events_highpri 36 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/5
Test results	**PASS**
Remarks	

6.9 ZTP

Test Items	ZTP
Test Content	Test that the switch supports ZTP (Zero Touch Provisioning).
Test Topology	
Test Steps	1、Export the current configdb file (skip if already exported), and delete the current configdb file. 2、On PC1, enable the DHCP server to configure option 67, enable the TFTP server containing the ZTP configuration file, and enable the HTTP server containing the configdb. 3、Reboot the DUT and check whether it obtains the ZTP configuration and configdb.
Expected Results	2.The ZTP configuration and configdb file can be obtained.
Actual Test Results	Default configuration: Enabled state. root@sonic:/home/admin# cat /host/ztp/ztp_cfg.json {

	"admin-mode": true, "ztp-json-local": "/mnt/usb/ztp.json" } Delete configdbfile rm /etc/sonic/config_db.json Reboot Check the downloaded configdb file. Mar 4 14:58:07.994214 sonic INFO sonic-ztp[1167229]: Downloading provisioning data from tftp://192.168.1.150/ztp.json to /var/run/ztp/ztp_data_opt67.json Mar 4 14:58:08.353372 sonic INFO sonic-ztp[1167229]: Starting ZTP using JSON file /var/run/ztp/ztp_data_opt67.json at 2025-03-04 06:58:08 UTC. Mar 4 14:58:08.354179 sonic INFO sonic-ztp[1167229]: Processing configuration section 01-configdb-json at 2025-03-04 06:58:08 UTC. Mar 4 14:58:09.569112 sonic INFO sonic-ztp[1178666]: configdb-json: Downloading config_db.json file from 'tftp://192.168.1.150/N9500_config_db.json'. Mar 4 14:58:09.650892 sonic INFO sonic-ztp[1178666]: configdb-json: Configuration change detected. Removing ZTP configuation from Config DB. Mar 4 14:58:09.651051 sonic INFO sonic-ztp[1178666]: configdb-json: Stopping ZTP discovery on interfaces. Mar 4 14:58:28.316779 sonic INFO sonic-ztp[1178666]: configdb-json: Reloading config_db.json to Config DB. Mar 4 14:58:28.624094 sonic INFO sonic-ztp[1182422]: Running command: rm -rf /tmp/dropstat-* Mar 4 14:58:28.647855 sonic INFO sonic-ztp[1182422]: Disabling container monitoring ... Mar 4 14:58:28.666403 sonic INFO sonic-ztp[1182422]: Stopping SONiC target ... Mar 4 14:58:48.979418 sonic INFO sonic-ztp[1182422]: Running command: /usr/local/bin/sonic-cfggen -j /etc/sonic/init_cfg.json -j /tmp/config_dl.json --write-to-db Mar 4 14:58:49.246952 sonic INFO sonic-ztp[1182422]: Running command: /usr/local/bin/db_migrator.py -o migrate Mar 4 14:58:49.431069 sonic INFO sonic-ztp[1182422]: Running command: /usr/local/bin/sonic-cfggen -d -y /etc/sonic/sonic_version.yml -t /usr/share/sonic/templates/sonic-environment.j2,/etc/sonic/sonic-environment Mar 4 14:58:50.558117 sonic INFO sonic-ztp[1182422]: Restarting SONiC target ... Mar 4 14:58:50.642379 sonic INFO sonic-ztp[1182422]: Enabling container monitoring ... Mar 4 14:58:50.667017 sonic INFO sonic-ztp[1182422]: Updating hostname ... Mar 4 14:58:51.749871 sonic INFO sonic-ztp[1182422]: Reloading Monit configuration ... Mar 4 14:58:51.770692 sonic INFO sonic-ztp[1182422]: Reinitializing monit daemon Mar 4 14:58:52.036079 sonic INFO sonic-ztp[1185892]: user info loaded. Mar 4 14:58:52.157316 sonic INFO sonic-ztp[1167229]: Processed Configuration section 01-configdb-json with result SUCCESS, exit code (0) at 2025-03-04 06:58:08 UTC. Mar 4 14:58:52.158124 sonic INFO sonic-ztp[1167229]: Checking configuration section 01-configdb-json result: SUCCESS, ignore-result: False. Mar 4 14:58:52.169687 sonic INFO sonic-ztp[1167229]: ZTP successfully completed at 2025-03-04 06:58:52 UTC. root@sonic:/etc/sonic# show ztp ZTP Admin Mode : True ZTP Service : Inactive ZTP Status : SUCCESS ZTP Source : dhcp-opt67 (eth0) Runtime : 01m 42s

	Timestamp : 2025-03-04 06:58:52 UTC ZTP Service is not running 01-configdb-json: SUCCESS
Test results	**PASS**
Remarks	