FS Network TAP Series

Configuration Guide

Table of Contents

List of Figures

1 Brief Introduction

This document describes the basic conceptions, applications and usages (include network topology, configuration examples and limitations) of TAP series devices.

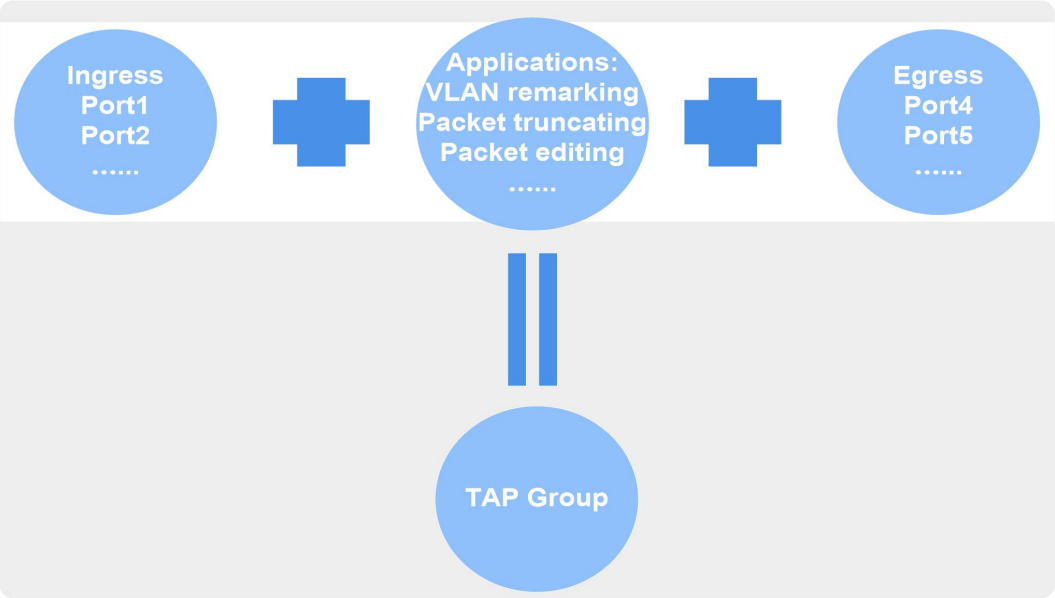
The following descriptions are based on T5850_TAP (T5850-48S2Q4C / T5850-48S6Q / T5850-32S2Q), T8050_TAP (T8050-20Q4C) . Differences between T5850 (T8050) and T5800 (T5800-8TF12S) are pointed out.

1.1 TAP Group introduction

A TAP Group has at least one ingress port and one egress port. The ingress and egress ports should be link aggregation or physical ports.

TAP series devices support 2 modes: PORT and PORT WITH FLOW.

Figure 1-1 Composition of TAP group



1.1.2 Port mode

Applications are taking effect on all packets which pass through the port.

One ingress port can only belong to one TAP group. One Egress port can belong to several TAP groups.

All packets enter the ingress port should be forward to the egress port.

1.1.3 Port with flow mode

Applications are taking effect on packets which pass through the port and match the flow rule.

One ingress port with different flow rules can join different TAP Groups. One Egress port can join several TAP groups.

Packets enter the ingress port should compare with the flow rule, only the packets matching the flow rule can be forward to the egress port.

E.g.: eth-0-1 with Flow A is the ingress member of TAP group 1; eth-0-1 with Flow B is the ingress member of TAP group 2. When the packets enter the port eth0-01, packets which match Flow A should forward to TAP group1's egress port; packets which match Flow B should forward to TAP group2's egress port.

1.2 FLOW types

TAP series devices support 2 types of the flow: default (UDF) Flow ; decap (inner-match) Flow.

Default Flow is used for matching normal packets.

Decap Flow is used for matching the inner header of the packet which is encapsulated with GRE/NVGRE/VXLAN, etc.

1.3 Precondition

The following actions are supported for both PORT and PORT WITH FLOW mode:

VLAN remarking

VLAN heading stripping

Packet editing

Packet truncating

Time stamp

The following actions are only supported on PORT WITH FLOW mode:

GRE/NVGRE/VXLAN header stripping

Inner header field matching

Table 1-1 Supported actions for different mode:

Action\Mode	**PORT**	**PORT with FLOW**
VLAN remarking	√	√
VLAN heading stripping	√	√
Packet truncating	√	√
Packet editing	√	√
Inner header field matching	×	√
Packet header stripping	×	√
Inner VXLAN header stripping	×	√
Time STAMP	√ (Apply to the egress port of TAP Group)	

1.4 Limitaions

Table 1-2 Mutual exclusion table

	VLAN header stripping	**VLAN remarking**	**Packet truncating**	**Packet editing**	**Packet head stripping**	**Time stamp**	**Inner VXLAN header stripping**
VLAN header stripping	N/A	×	×	√	×	√	
VLAN remarking	×	N/A	×	√	√	√	
Packet truncating	×	×	N/A	×	×	×	
Packet editing	√	√	×	N/A	√	×	
Packet head stripping	×	√	×	√	N/A	√	
Time stamp	√	√	×	×	√	N/A	
Inner VXLAN header stripping	×	√	×	√	√	√	N/A

√ : These 2 actions can be configured together.

× : These 2 actions are mutual exclusive and cannot be configured together.

2 Device Management Configuration

TAP series devices have 2 types of management ports: Ethernet port and console port.

User can choose any of these management ports to manage the device.

2.1 Configuring console port for management

2.1.1 Configuration

Before you can assign switch information, make sure you have connected a PC or terminal to the console port, and configured the PC or terminal software parameters to match the default console port parameters.

The follow list describes the default value of console parameters for TAP series switches:

- Baud rate default is 115200.
- Data bits default is 8.
- Stop bits default is 1.
- Parity settings default is none.

2.1.2 Validation

The following example shows how to display the configuration of the console port:

```
TAP# show console
Current console configuration:
-------------------------------------------------------------
line console 0
 speed 115200
 parity none
 databits 8
 stopbits 1
 exec-timeout 10 0
 privilege level 1
 no line-password
 no login
```

2.2 Configuring out band Ethernet port for management

User should set the management ip address by console port before managing the device by out band ethernet port.

2.2.1 Configuration

Set the management ip address as 10.10.10.11/23:

```
TAP# configure terminal
TAP(config)# management ip address 10.10.10.11/23
```

(optional) Set the management gateway address:

```
TAP# configure terminal
TAP(config)# management route gateway 10.10.10.1
```

2.2.2 Validation

The following example shows how to display the configuration:

```
TAP# show management ip address
Management IP address:    10.10.10.11/23
Gateway:    10.10.10.1
```

2.3 Configuring Temperature

TAP series switches support temperature alarm management.

User can configure three temperature thresholds: low, high and critical. When the temperature of the device is lower than low threshold or higher than higher threshold, the device will give an alarm. If the temperature of the device is higher than critical threshold, the device will cut off its power automatically.

NOTE The critical threshold is not recommended to set too low, otherwise it may lead the device reboot unnecessary

2.3.1 Configuration

The following example shows how to set the low threshold of the device as 10° C; high threshold of the device as 70° C; critical threshold of the device as 85° C:

```
TAP# configure terminal
TAP(config)# temperature 10 70 85
```

NOTE User can set the temperature of the board. The temperature of the chip cannot be changed.

2.3.2 Validation

The following example shows how to display the configuration of the temperature:

```
TAP# show environment
Fan tray status:
Index          Status          SpeedRate       Mode
----------+------------+------------+------------
1-1            OK              40%             AUTO
1-2            OK              40%             AUTO
1-3            OK              40%             AUTO
1-4            OK              40%             AUTO

Power status:
Index          Status          Power           Type            Alert
----------+------------+------------+------------+------------
1              PRESENT         OK              AC              NO
2              PRESENT         FAIL            -               ALERT

Sensor status (Degree Centigrade):
Index          Temperature     Lower_alarm     Upper_alarm     Critical        Position
----------+------------+------------+------------+------------+------------
1              41              10              70              85              BEFORE_CHIP
2              43              10              70              85              BEHIND_CHIP
3              34              10              70              85              AROUND_FAN
4              41              10              70              85              AROUND_CPU
5              65              -10             100             110             SWITCH_CHIP0
```

2.4 Configuring Fan

TAP series switches support to manage fan automatically according to the temperature of the board and chip.

Table 2-1 Correspondence of the chip temperature and the fan speed:

Chip temperature (ºC)	Work mode of the FAN	Speed rate of the FAN
⩾100	Full	100%
90⩽ Temperature < 100	High	80%

80≤ Temperature < 90	Low	60%
≤80	Bottom	40%

Table 2-2 Correspondence of the board temperature and the fan speed

Board temperature (ºC)	**Work mode of the FAN**	**Speed rate of the FAN**
≥80	Full	100%
65 ≤ Temperature < 80	High	80%
50 ≤ Temperature < 65	Low	60%
≤50	Bottom	40%

NOTE e.g. When the chip and the board are both 65 ºC, according to Table 2-1 the FAN speed should be 40%, according to Table 2-2 the FAN speed should be 80%. The real speed should be according the higher one (80%).

2.4.1 Configuration

This application does not have any command line.

2.4.2 Validation

The following example shows how to display the fan information:

```
TAP# show environment
Fan tray status:
Index          Status          SpeedRate       Mode
----------+------------+------------+------------
1-1            OK               40%              AUTO
1-2            OK               40%              AUTO
1-3            OK               40%              AUTO
1-4            OK               40%              AUTO

Power status:
Index          Status          Power           Type             Alert
----------+------------+------------+------------+------------
1               PRESENT         OK               AC               NO
```

```
2               PRESENT         FAIL            -               ALERT

Sensor status (Degree Centigrade):
Index           Temperature     Lower_alarm     Upper_alarm     Critical        Position
----------+------------+------------+------------+------------+------------
1               41              10              70              85              BEFORE_CHIP
2               43              10              70              85              BEHIND_CHIP
3               34              10              70              85              AROUND_FAN
4               41              10              70              85              AROUND_CPU
5               65              -10             100             110             SWITCH_CHIP0
```

2.5 Configuring Power

TAP series switches support to manage power status automatically. When the power is failed or the fan is failed because of the power issue, the device should give an alarm.

If power is removed or inserted, the switch should give an alarm too.

2.5.1 Configuration

This application does not have any command line.

2.5.2 Validation

The following example shows how to display the power information:

```
TAP# show environment
Fan tray status:
Index           Status          SpeedRate       Mode
----------+------------+------------+------------
1-1             OK              40%             AUTO
1-2             OK              40%             AUTO
1-3             OK              40%             AUTO
1-4             OK              40%             AUTO

Power status:
Index           Status          Power           Type            Alert
----------+------------+------------+------------+------------
1               PRESENT         OK              AC              NO
2               PRESENT         FAIL            -               ALERT

Sensor status (Degree Centigrade):
Index           Temperature     Lower_alarm     Upper_alarm     Critical        Position
----------+------------+------------+------------+------------+------------
1               41              10              70              85              BEFORE_CHIP
2               43              10              70              85              BEHIND_CHIP
3               34              10              70              85              AROUND_FAN
4               41              10              70              85              AROUND_CPU
5               65              -10             100             110             SWITCH_CHIP0
```

2.6 Configuring Transceiver

TAP series switches support to check up the information of the transceiver.

The transceiver information includes basic information and diagnostic information. The basic information includes transceiver type, vendor name, PN, S/N, wavelength and link length for supported type.

The diagnostic information includes real-time temperature, voltage, current, optical transmit power, optical receive power and the threshold about these parameters.

when the transceiver is inserted or removed or the real-time parameter is out of threshold, the switch should notice the users.

2.7 Configuration

This application does not have any command line.

2.7.1 Validation

The following example shows how to display the transceiver information:

```
TAP# show transceiver

Port eth-0-1 transceiver info:
Transceiver Type: 1000BASE-SX
Transceiver Vendor Name : FINISAR CORP.
Transceiver PN                : FTLF8519P3BNL
Transceiver S/N               : PL36KUC
Transceiver Output Wavelength: 850 nm
Supported Link Type and Length:
        Link Length for 50/125um multi-mode fiber: 300 m
        Link Length for 62.5/125um multi-mode fiber: 150 m
```

The following example shows how to display the detailed transceiver information:

```
TAP# show transceiver detail eth-0-1

Port eth-0-1 transceiver info:
Transceiver Type: 1000BASE-SX
Transceiver Vendor Name : FINISAR CORP.
Transceiver PN                : FTLF8519P3BNL
Transceiver S/N               : PL36KUC
Transceiver Output Wavelength: 850 nm
Supported Link Type and Length:
```

```
        Link Length for 50/125um multi-mode fiber: 300 m
        Link Length for 62.5/125um multi-mode fiber: 150 m
--------------------------------------------------------------------------------
Transceiver is internally calibrated.
mA: milliamperes, dBm: decibels (milliwatts), NA or N/A: not applicable.
++ : high alarm, +  : high warning, -  : low warning, -- : low alarm.
The threshold values are calibrated.
--------------------------------------------------------------------------------
                                    High Alarm  High Warn   Low Warn    Low Alarm
                    Temperature     Threshold   Threshold   Threshold   Threshold
Port                (Celsius)       (Celsius)   (Celsius)   (Celsius)   (Celsius)
--------------+---------------+-----------+-----------+-----------+----------
eth-0-1         39.10           110.00      93.00       -30.00      -40.00
--------------------------------------------------------------------------------
                                    High Alarm  High Warn   Low Warn    Low Alarm
                    Voltage         Threshold   Threshold   Threshold   Threshold
Port                (Volts)         (Volts)     (Volts)     (Volts)     (Volts)
--------------+---------------+-----------+-----------+-----------+----------
eth-0-1         3.32            3.60        3.50        3.10        3.00
--------------------------------------------------------------------------------
                                    High Alarm  High Warn   Low Warn    Low Alarm
                    Current         Threshold   Threshold   Threshold   Threshold
Port                (milliamperes)  (mA)        (mA)        (mA)        (mA)
--------------+---------------+-----------+-----------+-----------+----------
eth-0-1         6.56            13.00       12.50       2.00        1.00
--------------------------------------------------------------------------------
                    Optical         High Alarm  High Warn   Low Warn    Low Alarm
                    Transmit Power  Threshold   Threshold   Threshold   Threshold
Port                (dBm)           (dBm)       (dBm)       (dBm)       (dBm)
--------------+---------------+-----------+-----------+-----------+----------
eth-0-1         -5.11           0.00        -3.00       -9.50       -13.50
--------------------------------------------------------------------------------
                    Optical         High Alarm  High Warn   Low Warn    Low Alarm
                    Receive Power   Threshold   Threshold   Threshold   Threshold
Port                (dBm)           (dBm)       (dBm)       (dBm)       (dBm)
--------------+---------------+-----------+-----------+-----------+----------
eth-0-1         -6.15           0.50        -1.00       -16.99      -21.02
```

3 Interface configuration

3.1 Configuring Interface Split

3.1.1 Configuration

The following example shows how to split a 40G port into four 10G ports:

```
TAP# configure terminal
TAP(config)# split interface eth-0-1 10giga
```

NOTE User must reboot the switch to take effect.

3.1.2 Validation

The following example shows how to display the splitting information:

```
TAP# show interface status
Name          Status        Duplex    Speed     Mode      Type            Description
----------+-----------+--------+--------+-------+------------+----------------
eth-0-1/1   down          auto      auto      trunk     UNKNOWN
eth-0-1/2   down          auto      auto      trunk     UNKNOWN
eth-0-1/3   down          auto      auto      trunk     UNKNOWN
eth-0-1/4   down          auto      auto      trunk     UNKNOWN
```

3.2 Configuring Interface State

3.2.1 Configuration

The following example shows how to turn up eth-0-1 and turn down eth-0-2:

```
TAP# configure terminal
TAP(config)# interface eth-0-1
TAP(config-if)# no shutdown
TAP(config-if)# exit
TAP(config)# interface eth-0-2
TAP(config-if)# shutdown
```

3.2.2 Validation

The following example shows how to display the interface information:

```
TAP# show interface status
Name          Status        Duplex    Speed      Mode      Type            Description
----------+-----------+--------+--------+-------+------------+----------------
eth-0-1       up            a-full    a-1000     trunk     1000BASE_SX
eth-0-2       admin down    auto      a-1000     trunk     1000BASE_SX
```

3.3 Configuring Interface Duplex

3.3.1 Configuration

The following example shows how to set duplex of eth-0-1 to full and duplex of eth-0-2 to auto:

```
TAP# configure terminal
TAP(config)# interface eth-0-1
TAP(config-if)# duplex full
TAP(config-if)# exit
TAP(config)# interface eth-0-2
TAP(config-if)# duplex auto
```

3.3.2 Validation

The following example shows how to display the duplex information:

```
TAP# show interface status
Name          Status        Duplex    Speed      Mode      Type            Description
----------+-----------+--------+--------+-------+------------+----------------
eth-0-1       up            full      a-1000     trunk     1000BASE_SX
eth-0-2       up            a-full    a-1000     trunk     1000BASE_SX
```

3.4 Configuring Interface Speed

3.4.1 Configuration

The following example shows how to set speed of eth-0-1 to 1000M:

```
TAP# configure terminal
TAP(config)# interface eth-0-1
TAP(config-if)# speed 1000
```

3.4.2 Validation

The following example shows how to display the speed information:

```
TAP# show interface status
Name          Status          Duplex    Speed      Mode      Type             Description
----------+-----------+--------+--------+-------+------------+----------------
eth-0-1       up              full      1000       trunk     1000BASE_SX
```

3.5 Configuring Unidirectional

3.5.1 Configuration

The following example shows how to set unidirectional of eth-0-1:

```
TAP# configure terminal
TAP(config)# interface eth-0-1
TAP(config-if)# unidirectional enable
TAP(config-if)# speed 1000
TAP(config-if)# duplex full
TAP(config-if)# end
```

3.5.2 Validation

The following example shows how to display the unidirectional information:

```
TAP# show interface status
Name          Status          Duplex    Speed      Mode      Type             Description
----------+-----------+--------+--------+-------+------------+----------------
eth-0-1       up              full      1000       trunk     1000BASE_SX
```

NOTE Interface state is always up when unidirectional is enabled. Duplex auto and speed auto are not supported when unidirectional is enabled, user should set proper duplex and speed value.

4 SSH configuration

The Secure Shell (SSH) is a protocol that provides a secure, remote connection to a device. SSH provides more security for remote connections than Telnet does by providing strong encryption when a device is authenticated. SSH supports the Data Encryption Standard (DES) encryption algorithm, the Triple DES (3DES) encryption algorithm, and password-based user authentication. The SSH feature has an SSH server and an SSH integrated client, which are applications that run on the switch. You can use an SSH client to connect to a switch running the SSH server. The SSH server works with the SSH client supported in this release and with SSH clients. The SSH client also works with the SSH server supported in this release and with SSH servers.

4.1 Configuration

The following example shows how to create a key which is named by “a”:

```
TAP# configure terminal
TAP(config)# rsa key a generate
```

The following example shows how to generate private key “a.pri” and public key “a.pub”, then put them on the FTP server:

```
TAP(config)# rsa key a export mgmt-if url ftp://username:password@host:port/a.pri private ssh2
TAP(config)# rsa key a export mgmt-if url ftp://username:password@host:port/a.pub public ssh2
```

The following example shows how to download the public key from the FTP server and configure the user name of the device which need to login with SSH:

```
TAP(config)# rsa key a.pub import mgmt-if url ftp:// username:password@host:port/a.pub public ssh2
TAP(config)# username aaa privilege 4 password 123
TAP(config)# username aaa assign rsa key a.pub
```

4.2 Validation

The following example shows how to download the private key on the client and login with SSH:

```
[TAP@localhost]$ ssh -i a.pri aaa@10.10.33.122
```

5 Syslog configuration

System information can be saved in log file or be sent to other servers on the network.

By default, The TAP series devices logs normal but significant system messages to its internal buffer and sends these messages to the system console.

User can check out the messages on the system console or the specified log server.

The messages are time-stamped to enhance real-time debugging and management.

Table 5-1 System message types

Name	**definition**
kern	Kernel message
user	Random user level message
mail	Mail system message
daemon	System daemon message
auth	Security/certification message
syslog	Inner message generated by daemon “syslogd”
lpr	Line printer message
news	Network news message
uucp	UUCP message
cron	Clock daemon message
authpriv	Privacy security certification message
ftp	FTP message

5.2 Configuring log server

5.2.1 Configuration

The following shows how to enable the log server, how to set the IP address of the server and how to set the log level:

```
TAP# configure terminal
TAP(config)# logging server enable
TAP(config)# logging server address mgmt-if 10.10.22.204
TAP(config)# logging server severity debug
```

NOTE

Table 5-2 Log level definition

Severity Level	Definition
emergency	system is unusable(0)
alert	action must be taken immediately(1)
critical	critical conditions(2)
error	error conditions(3)
warning	warning conditions(4)
notice	normal but significant condition(5)
information	Informational(6)
debug	debug-level messages(7)

5.2.2 Validation

The following example shows how to display the system log configuration information:

```
TAP# show logging
Current logging configuration:
------------------------------------------------------------
logging buffer 500
logging timestamp bsd
logging file enable
logging level file warning
```

```
logging level module debug
logging server enable
logging server severity debug
logging server facility local4
logging server address 10.10.22.204
logging merge enable
logging merge fifo-size 1024
logging merge timeout 10
```

5.3 Configuring Logging Buffer Size

5.3.1 Configuration

The following example shows how to set the logging buffer size to 700 messages:

```
TAP# configure terminal
TAP(config)# logging buffer 700
```

5.3.2 Validation

The following example shows how to display the system log configuration information:

```
TAP# show logging
Current logging configuration:
------------------------------------------------------------
logging buffer 700
logging timestamp bsd
logging file enable
logging level file warning
logging level module debug
logging server enable
logging server severity debug
logging server facility local4
logging server address 10.10.22.204
logging merge enable
logging merge fifo-size 1024
logging merge timeout 10
```

6 Time configuration

6.1 Configuration

The following example shows how to set system time:

```
TAP# configure terminal
TAP(config)# clock set datetime 10:10:12 3 7 2017
```

6.2 Validation

The following example shows how to display the system time:

```
TAP# show clock
10:10:16 Beijing Tue Mar 07 2017
Time Zone(Beijing) : UTC+08:00:00
```

7 User Management configuration

User management can improve the security level of the system.

Only the authorized users can login to the system.

Table 7-1 Login modes for TAP series devices

mode	definition
Login local	Login with the username and password configured in the system.
Login	Login with the password configured in the ”line vty” mode.
No login	Login with out password

7.2 Configuring the user management in login local mode

7.2.1 Configuration

The following example shows how to use the “login local” mode.

Set username to “test”, set password to “123”, and choose “login local” mode:

```
TAP# configure terminal
TAP(config)# line vty 0 7
TAP(config-line)# login local
TAP(config-line)# exit
TAP(config)# username test privilege 4 password 123
```

Validation

The following example shows how to login the device via Telnet:

```
Username: test
Password:
TAP#
```

7.3 Configuring the user management in login mode

7.3.1 Configuration

The following example shows how to use the “login” mode.

Set password to “123”, and choose “login” mode:

```
TAP# configure terminal
TAP(config)# line vty 0 7
TAP(config-line)# login
TAP(config-line)# line-password 123
TAP(config-line)# privilege level 4
```

7.3.2 Validation

The following example shows how to login the device via Telnet:

```
Password:
TAP#
```

NOTE The examples above show how to configure on Ethernet management port. The configuration of the console management port is similar as Ethernet port. Use “line console 0” to enter the console configuration mode.

7.4 Password recovery

7.4.1 Configuration

If the password is forgotten unfortunately, it can be recovered by following steps.

Connect the device by console port.

Reset the system by plug out and plug in the power.

The follow information will be printed on Console:

```
NAND read: device 0 offset 0x200000, size 0x400000
 4194304 bytes read: OK
Press ctrl+b to stop autoboot:    5
```

Choose “no pass” mode in bootrom:

```
Bootrom# boot_flash_nopass
Bootrom# Do you want to revert to the default config file ? [Y|N|E]: Y
```

NOTE After recover the password the configuration on the device may be lost. Please remember the password to avoid the service interruption.

7.4.2 Validation

Then system will reboot without loading startup-configuration. No password will be required.

8 SNMP configuration

SNMP is a communication protocol to connect a network management systems (NMS) and agents. It defines the standardized management frame work, common communication language, security and access control mechanism for monitoring and managing the devices in the network environment.

Via SNMP, the administrator can connect to the device to query the information, modify the configuration, monitor the state, get the failures and generate a report automatically

NOTE TAP series devices support SNMP V1/V2, Only part of the OID and trap are supported.

8.1 Configuring SNMP GET

8.1.1 Configuration

The following example shows how to set the SNMP community word:

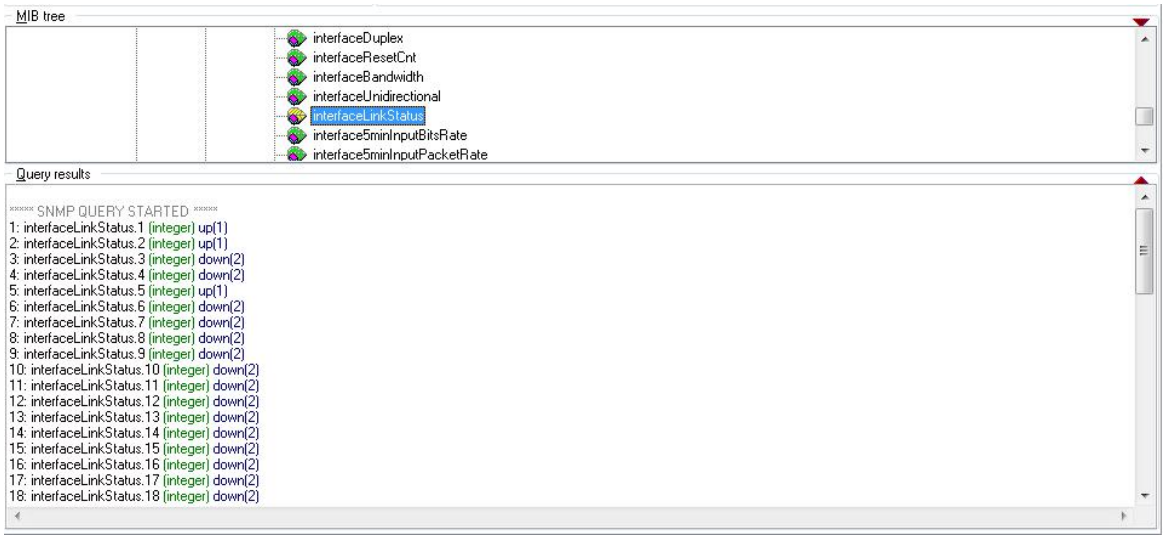
```
TAP(config)# snmp-server community test read-only
```

The following example shows how to enable SNMP service:

```
TAP(config)# snmp-server enable
```

8.1.2 Validation

Display the OID interfaceLinkStatus by applications:



8.2 Configuring SNMP TRAP

8.2.1 Configuration

The following example shows how to set the SNMP TRAP server IP and the SNMP community word:

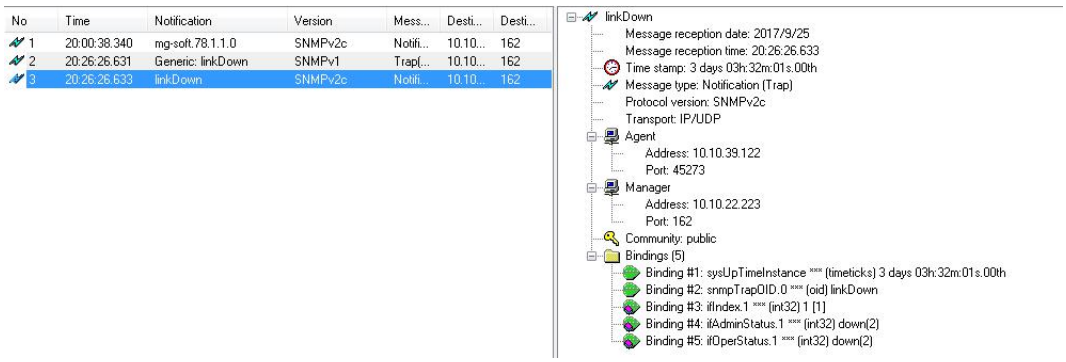
```
TAP(config)# snmp-server trap target-address mgmt-if 10.10.22.215 community public
```

The following example shows how to enable SNMP TRAP service:

```
TAP(config)# snmp-server trap enable all
```

8.2.2 Validation

Display the Trap information of linkDown by applications:



9 File Copy Configuration

9.1 Copy the file form the flash of device

The following example shows how to copy the file named “diagnostic-information.txt”.

9.1.1 Copy to TFTP server

```
TAP# copy flash:/ diagnostic-information.txt mgmt-if tftp://10.10.38.160
TFTP server [10.10.38.160]
Name of the TFTP file to access []diagnostic-information.txt
```

9.1.2 Copy to FTP server

```
TAP# copy flash:/ diagnostic-information.txt    mgmt-if ftp://10.10.25.33
FTP server [10.10.25.33]
User name [] test
Password []
Name of the FTP file to access []diagnostic-information.txt
```

9.1.3 Copy to USB disk

```
TAP# copy flash:/diagnostic-information.txt    udisk:
```

9.2 Copy the file to the flash of device

9.2.1 Copy from TFTP server

```
TAP# copy mgmt-if tftp://10.10.38.160/diagnostic-information.txt flash:
```

9.2.2 Copy from FTP server

```
TAP# copy mgmt-if ftp://10.10.25.33/diagnostic-information.txt flash:/
FTP server [] 10.10.25.33
User name [] test
Password []
Name of the FTP file to access []diagnostic-information.txt
```

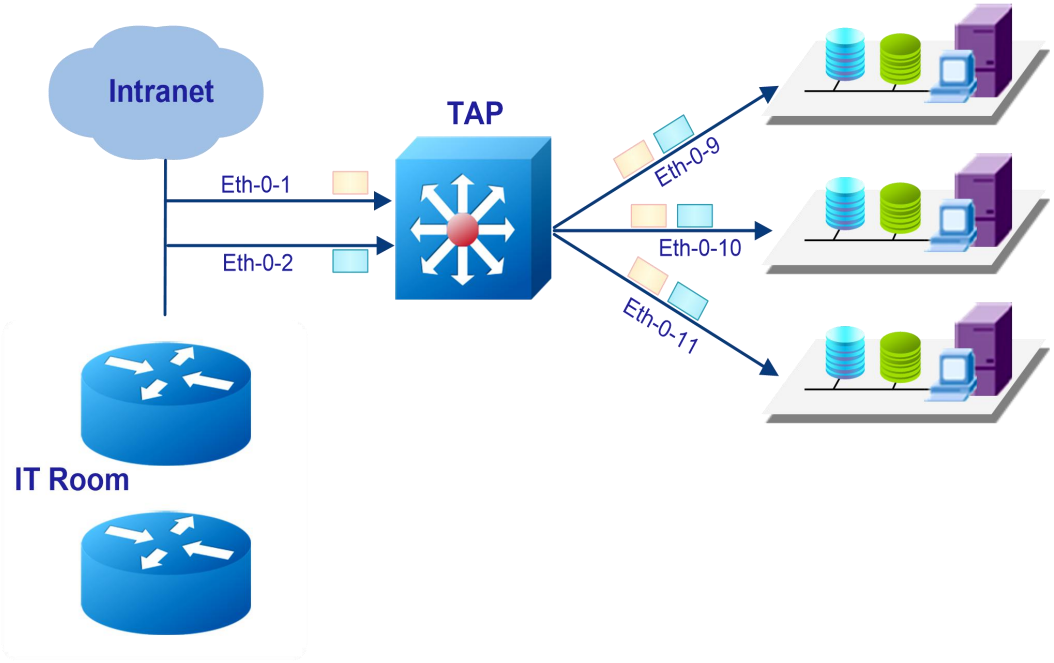
9.2.3 Copy from USB disk

```
TAP# copy udisk:/diagnostic-information.txt    flash:
```

10 M:N configuration

10.1 Networking requirements

Figure 10-1 Topology of M:N networking:



10.2 Configuration Ideas

In some cases, packets enter the device from different port need to be sent to different monitors. Therefore TAP M:N mode is required. The packets enter the ingress ports will send copies to all egress ports.

Reference to Figure 10-1: Packets enter eth-0-1 will send copies to eth-0-9/eth-0-10/eth-0-11. Packets enter eth-0-1 will also send copies to eth-0-9/eth-0-10/eth-0-11.

10.3 Configuration

The following example shows to create a TAP group with ingress port eth-0-1/eth-0-2, with egress port eth-0-9/eth-0-10/eth-0-11:

```
TAP# configure terminal
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1
TAP(config-tap-tap1)# ingress eth-0-2
TAP(config-tap-tap1)# egress eth-0-9
TAP(config-tap-tap1)# egress eth-0-10
TAP(config-tap-tap1)# egress eth-0-11
```

10.4 Validation

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
   ID: 1
   Ingress:
            eth-0-1
            eth-0-2
   egress:
            eth-0-9
            eth-0-10
            eth-0-11
```

10.5 Configuration file

User can display the configuration files as below:

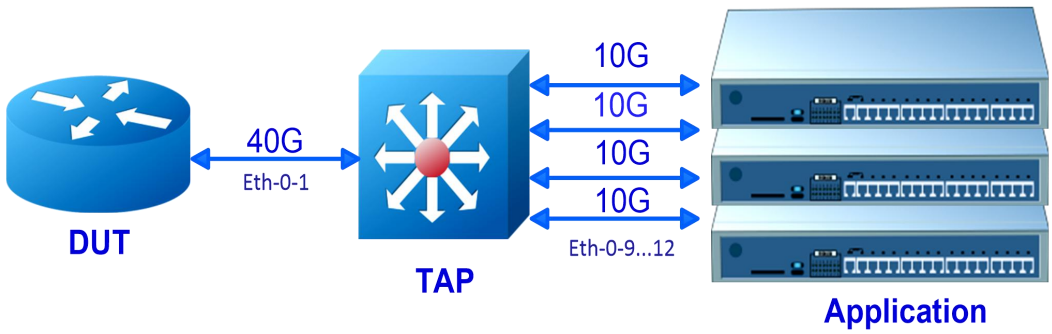
```
TAP# show running-config

tap-group tap1 1
  ingress eth-0-1
  ingress eth-0-2
  egress eth-0-9
  egress eth-0-10
  egress eth-0-11
```

11 load balance configuration(HASH)

11.1 Networking requirements

Figure 11-1 Topology of load balance:



11.2 Configuration Ideas

In some cases, the capability of the port is 40G/s, but the capability of the server or analyzer is 10G/s. Therefore, load balance is required to resolve this problem.

Reference to Figure 11-1, eth-0-1 is a 40G port, Agg1 is a link aggregation port with four 10G members (eth-0-9/eth-0-10/eth-0-11/eth-0-12).

Packets enter eth-0-1 should choose an outgoing port among eth-0-9/eth-0-10/eth-0-11/eth-0-12, according the load balance rule.

11.3 Configuration

The following example shows how to add eth-0-9/eth-0-10/eth-0-11/eth-0-12 into the link aggregation port Agg1:

```
TAP# configure terminal
TAP(config)# interface eth-0-9
TAP(config-if-eth-0-9)# static-channel-group 1
TAP(config)# interface eth-0-10
```

```
TAP(config-if0)# static-channel-group 1
TAP(config)# interface eth-0-11
TAP(config-if1)# static-channel-group 1
TAP(config)# interface eth-0-12
TAP(config-if2)# static-channel-group 1
```

The flowing example shows how to create a TAP group with ingress port eth-0-1, egress port Agg1:

```
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1
TAP(config-tap-tap1)# egress agg1
```

The flowing example shows how to set the load balance rule to hash by source mac address:

(The default rule is hash by source IP , destination IP , source port, destination port)

```
TAP(config)# port-channel load-balance set src-mac
TAP(config)# end
```

NOTE T5800_TAP use the following command to set load balance rule,

E.g. use source IP & destination IP:

```
TAP (config)# port-channel load-balance src-dst-ip
```

(Optional) T5850& T8050 TAP support detailed harsh rule, e.g. inner IP/ inner Mac, .etc.

```
TAP(config)# port-channel load-balance set inner-dst-ip
TAP(config)# end
```

NOTE The follow command is necessary if user enable to load balance by inner fields:

```
TAP (config)# port-channel load-balance tunnel-hash-mode both
```

11.4 Validation

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
ID: 1
```

```
  Ingress:
          eth-0-1
  egress:
          agg1
```

The following example shows how to display the load balance rule:

```
TAP# show port-channel load-balance
Port-channel load-balance hash fields:
-------------------------------------
src-mac
src-ip
dst-ip
src-port-l4
dst-port-l4
```

11.5 Configuration file

User can display the configuration files as below:

```
TAP# show running-config
!
port-channel load-balance set src-mac
!
interface eth-0-9
 static-channel-group 1
!
interface eth-0-10
 static-channel-group 1
!
interface eth-0-11
 static-channel-group 1
!
interface eth-0-12
 static-channel-group 1
!
tap-group tap1 1
   ingress eth-0-1
   egress agg1
```

Table 11-1 T5850 & T8050_TAP load balance fields

Load balance field	Description
src-mac	Load balance by source MAC address

dst-mac	Load balance by destination MAC address
src-ip	Load balance by source IP address
dst-ip	Load balance by destination IP address
ip-protocol	Load balance by ip-protocol
src-port-l4	Load balance by source port
dst-port-l4	Load balance by destination port
vxlan-vni	Vni of vxlan
inner-dst-mac	Inner Source MAC address based load balancing
inner-src-mac	Inner Destination MAC address based load balancing
inner-src-ip	Inner Source IP address based load balancing
inner-dst-ip	Inner Destination IP address based load balancing
gre-key	Key of GRE
nvgre-vsid	Vsid of nvgre
nvgre-flow-id	Flow ID of GRE

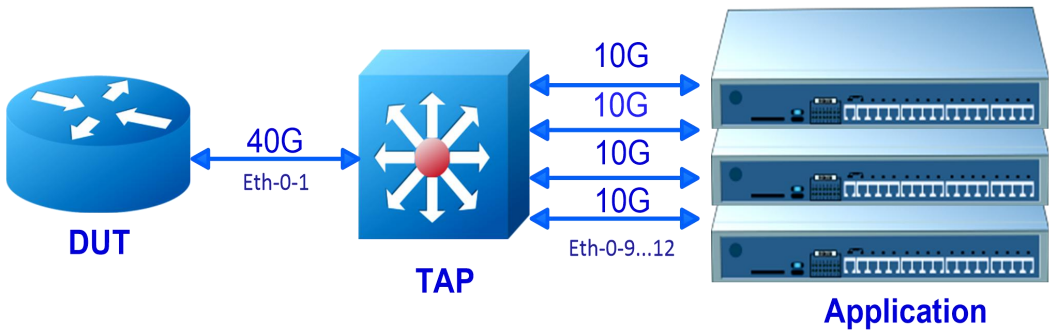
Table 11-2 T5800_TAP load balance fields

Load balance field	**Description**
src-mac	Load balance by source MAC address
dst-mac	Load balance by destination MAC address
src-ip	Load balance by source IP address
dst-ip	Load balance by destination IP address
src-dst-ip-src-dst-port	Load balance by IP address port
src-port	Load balance by source port
dst-port	Load balance by destination port
src-dst-mac	Load balance by MAC address.
src-dst-ip	Load balance by IP address
src-dst-port	Load balance by port

12 load balance configuration(RR)

12.1 Networking requirements

Figure 12-1 Topology of load balance:



NOTE Round-Robin mode is only supported by T5850 & T8050_TAP.

12.2 Configuration Ideas

In some cases, the capability of the port is 40G/s, but the capability of the server or analyzer is 10G/s. Therefore, load balance is required to resolve this problem.

Reference to Figure 11-1, eth-0-1 is a 40G port, Agg1 is a link aggregation port with four 10G members (eth-0-9/eth-0-10/eth-0-11/eth-0-12).

Packets enter eth-0-1 should choose an outgoing port among eth-0-9/eth-0-10/eth-0-11/eth-0-12, according the round-robin rule.

12.3 Configuration

The flowing example shows how to set the load balance mode to round-robin:

```
TAP# configure terminal
TAP(config)# port-channel 1 load-balance-mode round-robin
```

NOTE TAP series device supports at most 16 link aggregation ports to use round-robin mode. Round-robin mode must configure before ink aggregation port is created.

The following example shows how to add eth-0-9/eth-0-10/eth-0-11/eth-0-12 into the link aggregation port Agg1:

```
TAP# configure terminal
TAP(config)# interface eth-0-9
TAP(config-if-eth-0-9)# static-channel-group 1
TAP(config)# interface eth-0-10
TAP(config-if0)# static-channel-group 1
TAP(config)# interface eth-0-11
TAP(config-if1)# static-channel-group 1
TAP(config)# interface eth-0-12
TAP(config-if2)# static-channel-group 1
```

The flowing example shows how to create a TAP group with ingress port eth-0-1, egress port Agg1:

```
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1
TAP(config-tap-tap1)# egress agg1
```

12.4 Validation

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
ID: 1
  Ingress:
        eth-0-1
  egress:
        agg1
```

12.5 Configuration file

User can display the configuration files as below:

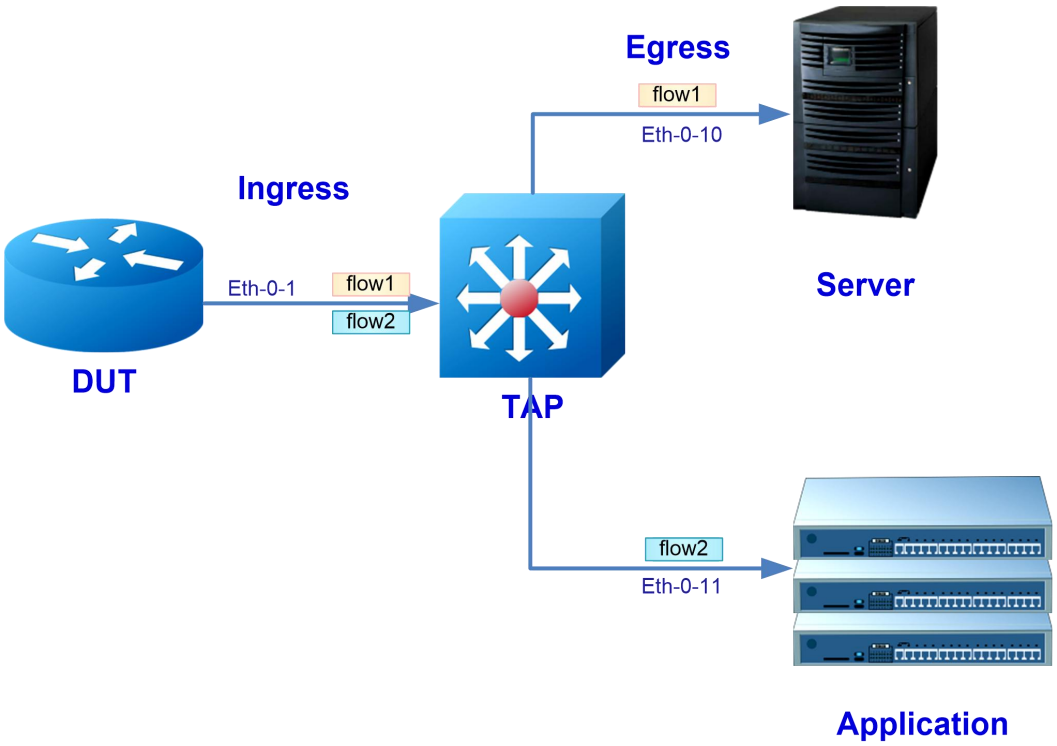
```
TAP# show running-config
!
port-channel 1 load-balance-mode round-robin
!
interface eth-0-9
 static-channel-group 1
!
interface eth-0-10
 static-channel-group 1
!
interface eth-0-11
 static-channel-group 1
!
interface eth-0-12
 static-channel-group 1
!
tap-group tap1 1
   ingress eth-0-1
   egress agg1
```

13 Ingress PORT with FLOW configuration

13.1 Configuring basic Flow

13.1.1 Networking requirements

Figure 13-1 Topology of PORT with FLOW



13.1.2 Configuration Ideas

In some cases, packets from one interface need to copy to different outgoing ports.

Use the PORT with FLOW TAP groups can redirect the packets to different ports.

Reference to Figure 13-1 packets with source IP address 1.1.1.0/24 or 2.2.2.0/24 should copy to eth-0-10. Packets with source IP address 10.1.1.0/24 or 20.1.1.0/24 should copy to eth-0-11 Packets with other source IP address should be discard.

13.1.3 Configuration

The follow example shows how to create a Flow rule:

```
TAP# configure terminal
TAP(config)# flow flow1
TAP(config-flow-flow1)# permit any src-ip 1.1.1.0 0.0.0.255 dst-ip any
TAP(config-flow- flow1)# permit any src-ip 2.2.2.0 0.0.0.255 dst-ip any
TAP(config-flow- flow1)# exit
TAP(config)# flow flow2
TAP(config-flow- flow2)# permit any src-ip 10.1.1.0 0.0.0.255 dst-ip any
TAP(config-flow- flow2)# permit any src-ip 20.1.1.0 0.0.0.255 dst-ip any
```

NOTE The packets not matched by the flow rule should be discarded by default.

The following example shows how to create a TAP group with flow1 and flow2:

```
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 flow flow1
TAP(config-tap-tap1)# egress eth-0-10
TAP(config-tap-tap1)# exit
TAP(config)# tap-group tap2
TAP(config-tap-tap2)# ingress eth-0-1 flow flow2
TAP(config-tap-tap2)# egress eth-0-11
```

13.1.4 Validation

The following example shows how to display the flow rule information:

```
TAP# show flow1
flow flow1
 sequence-num 10 permit any src-ip 1.1.1.0 0.0.0.255 dst-ip any
 sequence-num 20 permit any src-ip 2.2.2.0 0.0.0.255 dst-ip any
flow flow2
 sequence-num 10 permit any src-ip 10.1.1.0 0.0.0.255 dst-ip any
 sequence-num 20 permit any src-ip 20.1.1.0 0.0.0.255 dst-ip any
```

The following example shows how to display the TAP group information:

```
TAP# show tap-group

TAP-group tap1
ID: 1
   Ingress:
            eth-0-1                flow flow1
   egress:
            eth-0-10
TAP-group tap2
ID: 2
   Ingress:
```

```
        eth-0-1             flow flow2
egress:
        eth-0-11
```

13.1.5 Configuration file

User can display the configuration files as below:

```
TAP# show running-config
!
flow flow1
 sequence-num 10 permit any src-ip 1.1.1.0 0.0.0.255 dst-ip any
 sequence-num 20 permit any src-ip 2.2.2.0 0.0.0.255 dst-ip any
!
flow flow2
 sequence-num 10 permit any src-ip 10.1.1.0 0.0.0.255 dst-ip any
 sequence-num 20 permit any src-ip 20.1.1.0 0.0.0.255 dst-ip any
!
tap-group tap1 1
ingress eth-0-1 flow flow1
egress eth-0-10
!
tap-group tap2 2
ingress eth-0-1 flow flow2
egress eth-0-11
```

Table 13-1 T5850 & T8050_TAP Flow rule fields

Field	Description
IP protocol[number| any| icmp| igmp|gre| nvgre| tcp |udp|]	Specify the IP protocol number of the flow rule. Well known IP protocols can also be specified by name. e.g. IP protocol 1 = icmp, 2 = igmp, 6 = tcp, 17 = udp, 47 = gre/nvgre (gre protocol 0x0800 = gre, 0x6558 = nvgre). Parameter “any” indicates packets with any IP protocol can match this rule.
src-ip/src-ipv6	Source IPv4/IPv6 address
dst-ip/dst-ipv6	Destination IPv4/IPv6 address
flow-label	Flow label of IPv6
Inner-match	Specify the inner match profile of the flow rule. The inner-match profile is created by

	“inner-match” command in global configuration mode.
ip-precedence	IP precedence
src-port	Source layer 4 port
dst-port	Destination layer 4 port
first-fragment	Match packets with first fragment
non-first-fragment	Match packets with non first fragment
non-fragment	Match packets with non fragment
non-or-first-fragment	Match packets with non first fragment
small-fragment	Match packets with small fragment
any-fragment	Match packets with any fragment
options	Match packets with IP options
dscp	DSCP in IPv4 packets value
vxlan-vni	VNI of VXLAN
vlan	Vlan ID
inner-vlan	Inner vlan ID
cos	CoS value in vlan header
inner-cos	CoS value in inner vlan header
ether-type	Ether type
src-mac	Source mac address
dst-mac	Destination mac address
Ipv4-head	IPv4 packet header
l4-head	Layer 4 header

Table 13-2 T5850 & T8050_TAP Flow rule actions

Action	**Description**
un-tag/un-tag-outer-vlan/un-tag-inner-vlan	Remove vlan tags of the packets.
mark-source	Specify additional outer vlan id of the outgoing packets.
edit-macda	Edit the destination mac address of the outgoing packet.
edit-macsa	Edit the source mac address of the outgoing packet.
edit-ipda/edit-ipv6da	Edit the destination IPv4/IPv6 address of the outgoing packet.

edit-ipsa/edit-ipv6sa	Edit the source IPv4/IPv6 address of the outgoing packet.
edit-vlan	Edit the vlan tag of the outgoing packet
strip-header	Strip the gre/nvgre/vxlan header
truncation	Truncate the packet

Table 13-3 T5800_TAP Flow rule fields

Field	**Description**
IP protocol[number| any| icmp| igmp| tcp | udp|]	Specify the IP protocol number of the flow rule. The valid range for IP protocol number is 0-255. Well known IP protocols can also be specified by name. e.g. IP protocol 1 = icmp, 2 = igmp, 6 = tcp, 17 = udp, 47 = gre/nvgre (gre protocol 0x0800 = gre, 0x6558 = nvgre). Parameter “any” indicates packets with any IP protocol can match this rule.
src-ip/src-ipv6	Source IPv4/IPv6 address
dst-ip/dst-ipv6	Destination IPv4/IPv6 address
flow-label	Flow label of IPv6
ip-precedence	IP precedence
src-port	Source layer 4 port
dst-port	Destination layer 4 port
first-fragment	Match packets with first fragment
non-first-fragment	Match packets with non first fragment
non-fragment	Match packets with non fragment
non-or-first-fragment	Match packets with non first fragment
small-fragment	Match packets with small fragment
any-fragment	Match packets with any fragment
options	Match packets with IP options
dscp	DSCP in IPv4 packets value of the flow rule.
vlan	Vlan ID
inner-vlan	Inner vlan ID

cos	CoS value in vlan header
inner-cos	CoS value in inner vlan header
ether-type	Ether type
src-mac	Source mac address
dst-mac	Destination mac address

Table 13-4 T5800_TAP Flow rule actions

Action	**Description**
un-tag/un-tag-outer-vlan/un-tag-inner-vlan	Remove vlan tags of the packets.
mark-source	Specify additional outer vlan id of the outgoing packets.
edit-macda	Edit the destination mac address of the outgoing packet.
edit-macsa	Edit the source mac address of the outgoing packet.
edit-ipda/edit-ipv6da	Edit the destination IPv4/IPv6 address of the outgoing packet.
edit-vlan	Edit the vlan tag of the outgoing packet

13.2 Configuring UDF Flow

13.2.1 Networking requirements

Figure 13-2 Topology of UDF FLOW

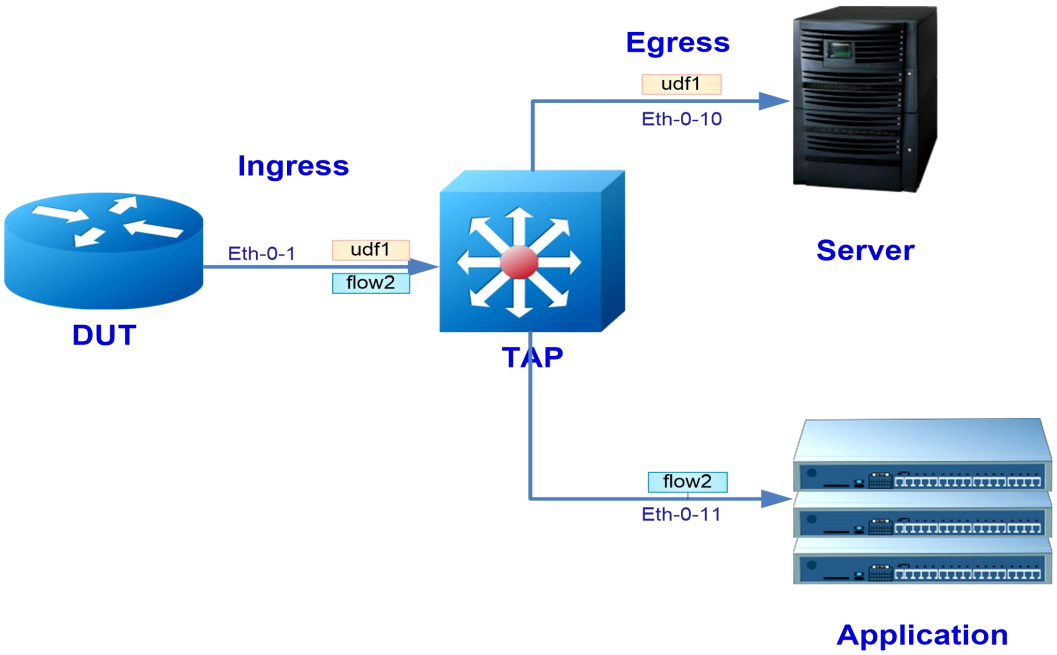


Figure 13-3 Packet structure for match the UDF flow rule

```
/*
 * L3/L4 UDF offset
 * +-------+-------+-------+------------+--------+-------+--------------------+
 * | MacDa | MacSa |  Vlan |  EtherType | Layer3 |Layer4 |      PayLoad       |
 * +-------+-------+-------+------------+--------+-------+--------------------+
 *                                      ^                                     |
 *                                      |          L3 UDF offset              |
 *                                      |<----------------------------------->|
 *
 * +-------+-------+-------+------------+----------+-----------------+--------+
 * |  MacDa|  MacSa|   Vlan|   EtherType| IP Header|Layer4(TCP or UDP)|PayLoad |
 * +-------+-------+-------+------------+----------+-----------------+--------+
 *                                                 ^                         |
 *                                                 |      L4 UDF offset      |
 *                                                 |<----------------------->|
```

NOTE UDF flow is only supported on T5850 & T8050_TAP

13.2.2 Configuration Ideas

In some cases, user needs more detailed rules to filter the packets. The TAP UDF (User defined format) can accurately match the specified field

UDF use the specified value and the reversed wildcard bits to match the field which is concerned.

An offset is needed to point out the position in the packet to match the UDF field.

13.2.3 Configuration

The following example shows how to create an IPv4 UDF rule, match field " 00 ab cd ef", the offset is 30 bytes starting at the layer 3 header.

```
TAP# configure terminal
TAP(config)# flow udf1
TAP(config-flow-udf1)# permit any src-ip any dst-ip any ipv4-head 0xabcdef 0x0 30
```

NOTE These examples are based on IPv4 rules.

The following example shows how to create basic flow rule:

```
TAP(config)# flow flow2
TAP(config-flow-map2)# permit any src-ip 10.1.1.0 0.0.0.255 dst-ip any
TAP(config-flow-map2)# permit any src-ip 20.1.1.0 0.0.0.255 dst-ip any
```

The following example shows how to create a TAP group with udf1 and flow2:

```
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 flow udf1
TAP(config-tap-tap1)# egress eth-0-10
TAP(config-tap-tap1)# exit
TAP(config)# tap-group tap2
TAP(config-tap-tap2)# ingress eth-0-1 flow flow2
TAP(config-tap-tap2)# egress eth-0-11
```

NOTE The TAP series device supports one profile for IPv4 UDF rule, or three profiles for layer 4 UDF rule. The IPv4 UDF rule and layer 4 UDF rule are mutual exclusive.

Layer 4 UDF rule has 2 key words: protocol & offset. If any of the key words is different, they belong to different profiles.

IPv4 UDF rue has 1 key word: offset. (One profile for IPv4 UDF rule means all IPv4 UDF rule should use same offset)

13.2.4 Validation

The following example shows how to display the IPv4 UDF rules:

```
TAP# show flow
flow udf1
 sequence-num 10 permit any src-ip any dst-ip any ipv4-head 0x00abcdef 0x00000000 30
flow flow2
 sequence-num 10 permit any src-ip 10.1.1.0 0.0.0.255 dst-ip any
 sequence-num 20 permit any src-ip 20.1.1.0 0.0.0.255 dst-ip any
```

The following example shows how to display the TAP group:

```
TAP# show tap-group

TAP-group tap1
ID: 1
   Ingress:
            eth-0-1              flow udf1
   egress:
            eth-0-10

TAP-group tap2
ID: 2
   Ingress:
            eth-0-1              flow flow2
   egress:
            eth-0-11
```

13.2.5 Configuration file

User can display the configuration files as below:

```
TAP# show running-config
!
flow udf1
 sequence-num 10 permit any src-ip any dst-ip any ipv4-head 0x00abcdef 0x00000000 30
!
flow flow2
 sequence-num 10 permit any src-ip 10.1.1.0 0.0.0.255 dst-ip any
 sequence-num 20 permit any src-ip 20.1.1.0 0.0.0.255 dst-ip any
!
tap-group tap1 1
ingress eth-0-1 flow udf1
egress eth-0-10
!
tap-group tap2 2
ingress eth-0-1 flow flow2
egress eth-0-11
```

13.3 Configuring Inner-match

13.3.1 Networking requirements

Figure 13-4 Topology of Inner match

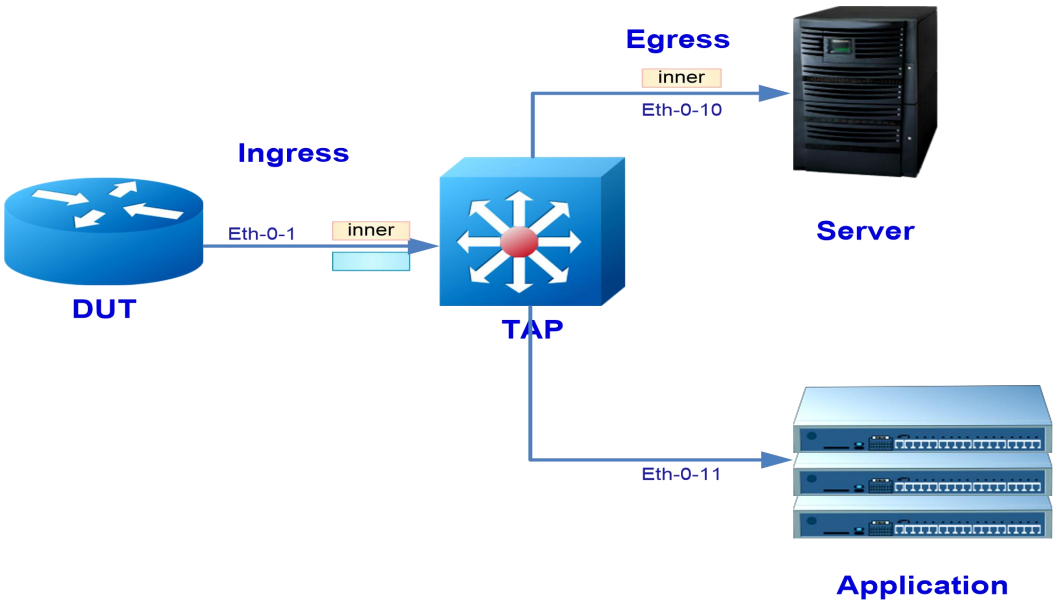


Figure 13-5 Packet for inner-match

GRE/ NVGRE/ VXLAN header	Original Inner Packet

NOTE Inner match is only supported by T5850 & T8050_TAP

13.3.2 Configuration Ideas

In some cases, user needs to match the inner field of GRE/NVGRE/VXLAN packets. To meet the requirement, use the inner-match configuration.

13.3.3 Configuration

The following example shows how to create a inner-match profile, matching the destination IP address 1.1.1.1 or 1.1.1.2:

```
TAP(config)# inner-match imf
TAP(config-inner-match-imf)# match any src-ip any dst-ip 1.1.1.1 0.0.0.0
TAP(config-inner-match-imf)# match any src-ip any dst-ip 1.1.1.2 0.0.0.0
TAP(config-inner-match-imf)# exit
```

The following example shows how to create a Flow with decap enabled, matching the GRE packets with destination IP address 11.1.1.1, NVGRE packets with the destination IP address 12.1.1.1, VXLAN packets with the destination IP address 13.1.1.1, and apply the inner-match imf to this flow:

```
TAP(config)# flow inner type decap
TAP(config-flow-inner)# permit gre src-ip any dst-ip 11.1.1.1 0.0.0.0 inner-match imf
TAP(config-flow-inner)# permit nvgre src-ip any dst-ip 12.1.1.1 0.0.0.0 inner-match imf
TAP(config-flow-inner)# permit udp dst-port eq 4789 src-ip any dst-ip 13.1.1.1 0.0.0.0 inner-match
imf
```

NOTE To match the VXLAN packets, set the type to udp and set the destination port to 4789.

13.3.4 Validation

The following example shows how to display the inner-match rule and the flow rule:

```
TAP# show inner-match
inner-match imf
 sequence-num 1 match any src-ip any dst-ip host 1.1.1.1
 sequence-num 2 match any src-ip any dst-ip host 1.1.1.2

TAP# show flow
flow inner type decap
 sequence-num 10 permit gre src-ip any dst-ip host 11.1.1.1 inner-match imf
 sequence-num 20 permit nvgre src-ip any dst-ip host 12.1.1.1 inner-match imf
 sequence-num 30 permit udp dst-port eq 4789 src-ip any dst-ip host 13.1.1.1 inner-match imf
```

NOTE Flows with decap enabled and disabled cannot bind to the same interface. E.g. eth-0-1 with decap flow inner is the ingress of TAP Group tap1, so eth-0-1 cannot bind with other flows without decap in any other TAP groups.

The following example shows the error notification when configure different types of flow:

```
DUT1(config)# flow flow1 type decap
DUT1(config-flow-flow1)# exit
DUT1(config)# flow flow2
DUT1(config-flow-flow2)# exit
DUT1(config)# tap-group tap1
DUT1(config-tap-tap1)# ingress eth-0-1 flow flow1
DUT1(config-tap-tap1)# exit
DUT1(config)# tap-group tap2
DUT1(config-tap-tap2)# ingress eth-0-1 flow flow2
% Interface mode conflict
```

Reference to Figure 13-4 the packets remark with blue rectangle is not matched by any flow rule so they should be discard.

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
ID: 1
   Ingress:
            eth-0-1                flow inner
   egress:
            eth-0-10
```

13.3.5 Configuration file

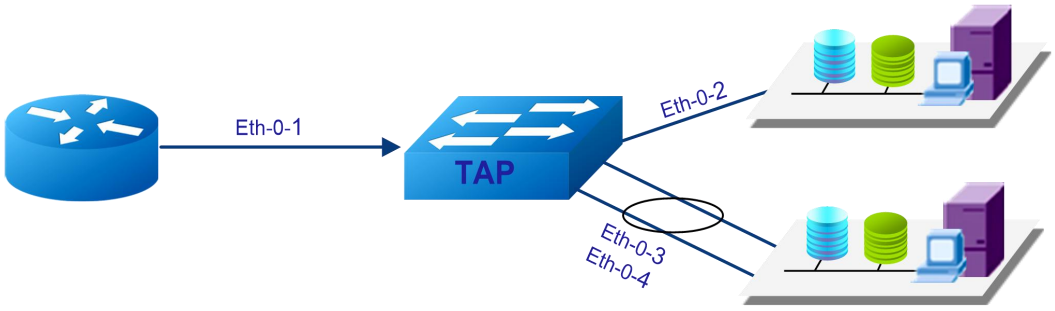
User can display the configuration files as below:

```
TAP# show running-config
!
inner-match imf
 sequence-num 1 match any src-ip any dst-ip host 1.1.1.1
 sequence-num 2 match any src-ip any dst-ip host 1.1.1.2
!
flow inner type decap
 sequence-num 10 permit gre src-ip any dst-ip host 11.1.1.1 inner-match imf
 sequence-num 20 permit nvgre src-ip any dst-ip host 12.1.1.1 inner-match imf
 sequence-num 30 permit udp dst-port eq 4789 src-ip any dst-ip host 13.1.1.1 inner-match imf
!
tap-group tap1 1
ingress eth-0-1 flow inner
egress eth-0-10
```

14 Port Filter configuration

14.1 Networking Requirements

Figure 14-1 Topology of port filter usage



14.2 Configuration Ideas

In some cases, after packets forward to the destination port, a filter is required to discard some unneeded packets.

Reference to Figure 14-1, packets with source IP address 1.0.0.0/24 from eth-0-1 should forward to eth-0-2 and Agg1(with two members eth-0-3/eth-0-4).

Eth-0-3 need to monitor the web packets, Agg1 need to monitor all packets.

14.3 Configuration

The following example shows how to add eth-0-3/eth-0-4 into the link aggregation port Agg1:

```
TAP# configure terminal
TAP(config)# interface eth-0-3
TAP(config-if-eth-0-3)# static-channel-group 1
TAP(config)# interface eth-0-4
TAP(config-if-eth-0-4)# static-channel-group 1
```

The following example shows how to create the filter:

```
TAP# configure terminal
TAP(config)# ip access-list filter1
TAP(config-acl-filter1)# permit tcp dst-port eq 80 src-ip any dst-ip any
TAP(config-acl-filter1)# exit
TAP(config)# ip access-list filter2
TAP(config-acl-filter2)# deny tcp dst-port eq 80 src-ip any dst-ip any
TAP(config-acl-filter2)# permit any src-ip any dst-ip any
TAP(config-acl-filter2)# end
```

NOTE After apply the filter to the egress port, Packets which not matched by any filter rule should be discard by default.

The following example shows how to apply the filter:

```
TAP# configure terminal
TAP(config)# interface eth-0-2
TAP(config-if- eth-0-2)# egress filter1
TAP(config-if- eth-0-2)# exit
TAP(config)# interface agg1
TAP(config-if-agg1)# egress filter2
```

The following example shows to create a TAP group with ingress port eth-0-1, with egress port eth-0-2/Agg1:

```
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1
TAP(config-tap-tap1)# egress agg1
TAP(config-tap-tap1)# egress eth-0-2
```

14.4 Validation

The following example shows how to display the filter rules:

```
TAP# show ip access-list
ip access-list filter1
 sequence-num 10 permit tcp dst-port eq 80 src-ip any dst-ip any
ip access-list filter2
 sequence-num 10 deny tcp dst-port eq 80 src-ip any dst-ip any
 sequence-num 20 permit any src-ip any dst-ip any
```

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
ID: 1
   Ingress:
            eth-0-1
   egress:
            eth-0-2
            agg1
```

14.5 Configuration file

User can display the configuration files as below:

```
TAP# show running-config
!
ip access-list filter1
 sequence-num 10 permit tcp dst-port eq 80 src-ip any dst-ip any
!
ip access-list filter2
 sequence-num 10 deny tcp dst-port eq 80 src-ip any dst-ip any
 sequence-num 20 permit any src-ip any dst-ip any
!
interface eth-0-2
 egress filter1
!
interface eth-0-3
 static-channel-group 1
!
interface eth-0-4
 static-channel-group 1
!
interface agg1
 egress filter2
!
tap-group tap1 1
 ingress eth-0-1
 egress eth-0-2
egress agg1
```

Table 14-1 T5850 & T8050_TAP Filter fields

Field	Description
IP protocol[number| any| icmp| igmp|gre| nvgre| tcp |udp|]	Specify the IP protocol number of the flow rule. Well known IP protocols can also be

	specified by name. e.g. IP protocol 1 = icmp, 2 = igmp, 6 = tcp, 17 = udp, 47 = gre/nvgre (gre protocol 0x0800 = gre, 0x6558 = nvgre). Parameter “any” indicates packets with any IP protocol can match this rule.
src-ip/src-ipv6	Source IPv4/IPv6 address
dst-ip/dst-ipv6	Destination IPv4/IPv6 address
flow-label	Flow label of IPv6
ip-precedence	IP precedence
first-fragment	Match packets with first fragment
non-first-fragment	Match packets with non first fragment
non-fragment	Match packets with non fragment
non-or-first-fragment	Match packets with non first fragment
small-fragment	Match packets with small fragment
any-fragment	Match packets with any fragment
options	Match packets with IP options
dscp	DSCP in IPv4 packets value
vlan	Vlan ID
inner-vlan	Inner vlan ID
cos	CoS value in vlan header
inner-cos	CoS value in inner vlan header
ether-type	Ether type
src-mac	Source mac address
dst-mac	Destination mac address
Ipv4-head	IPv4 packet header
l4-head	Layer 4 header

Table 14-2 T5800_TAP Filter rule fields

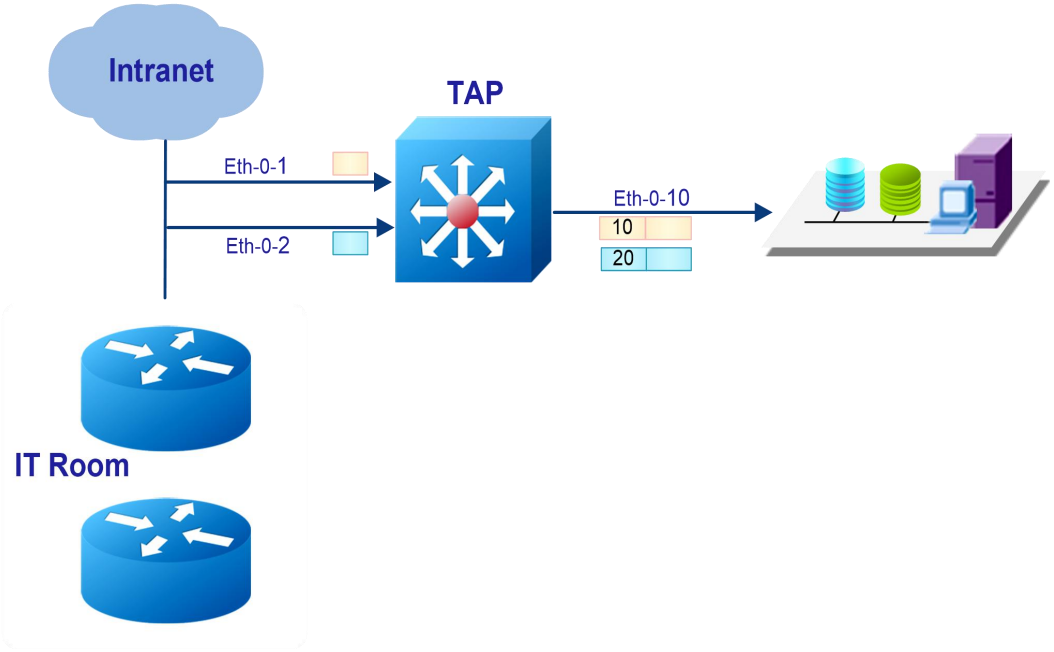
Field	**Description**
IP protocol[number| any| icmp| igmp| tcp | udp|]	Specify the IP protocol number of the flow rule. The valid range for IP protocol number is 0-255. Well known IP protocols can also be specified by name. e.g. IP protocol 1 = icmp, 2 = igmp, 6 =

	tcp, 17 = udp, 47 = gre/nvgre (gre protocol 0x0800 = gre, 0x6558 = nvgre). Parameter “any” indicates packets with any IP protocol can match this rule.
src-ip/src-ipv6	Source IPv4/IPv6 address
dst-ip/dst-ipv6	Destination IPv4/IPv6 address
flow-label	Flow label of IPv6
ip-precedence	IP precedence
first-fragment	Match packets with first fragment
non-first-fragment	Match packets with non first fragment
non-fragment	Match packets with non fragment
non-or-first-fragment	Match packets with non first fragment
small-fragment	Match packets with small fragment
any-fragment	Match packets with any fragment
options	Match packets with IP options
dscp	DSCP in IPv4 packets value of the flow rule.
vlan	Vlan ID
inner-vlan	Inner vlan ID
cos	CoS value in vlan header
inner-cos	CoS value in inner vlan header
ether-type	Ether type
src-mac	Source mac address
dst-mac	Destination mac address

15 VLAN Remarking Configuration

15.1 Networking Requirements

Figure 15-1 Topology of VLAN Remarking



15.2 Configuration Ideas

In some cases, the server and analyzer need to separate different packets. The VLAN Remarking application can meet the requirement.

Reference to Figure 15-1 Packets from eth-0-1 should add vlan tag 10. Packets from eth-0-2 should add vlan tag 20

15.3 Configuration

PORT mode and PORT WITH FLOW mode both support vlan remarking.

15.3.1 VLAN Remarking for PORT mode

The following example shows how to create TAP group, and remark the vlan tag to 10 for the packets form eth-0-1, remark the vlan tag to 20 for the packets form eth-0-2:

```
TAP# configure terminal
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 mark-source 10
TAP(config-tap-tap1)# ingress eth-0-2 mark-source 20
TAP(config-tap-tap1)# egress eth-0-10
```

15.3.2 VLAN Remarking for PORT WITH FLOW mode

The following example shows how to create TAP group, and remark the vlan tag to 10 for the packets with destination IP 1.1.1.1 form eth-0-1, remark the vlan tag to 20 for the packets with destination IP 1.1.1.2 form eth-0-2:

```
TAP(config)# flow flow1
TAP(config-flow-map1)# permit any src-ip any dst-ip 1.1.1.1 0.0.0.0 mark-source 10
TAP(config)# flow flow2
TAP(config-flow-map1)# permit any src-ip any dst-ip 1.1.1.2 0.0.0.0 mark-source 20
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 flow flow1
TAP(config-tap-tap1)# ingress eth-0-2 flow flow2
TAP(config-tap-tap1)# egress eth-0-10
```

15.4 Validation

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
  ID: 1
  Ingress:
        eth-0-1              mark-src 10
        eth-0-2              mark-src 20
  egress:
        eth-0-10
```

NOTE The result above shows the TAP group for PORT mode.

15.5 Configuration file

User can display the configuration files as below:

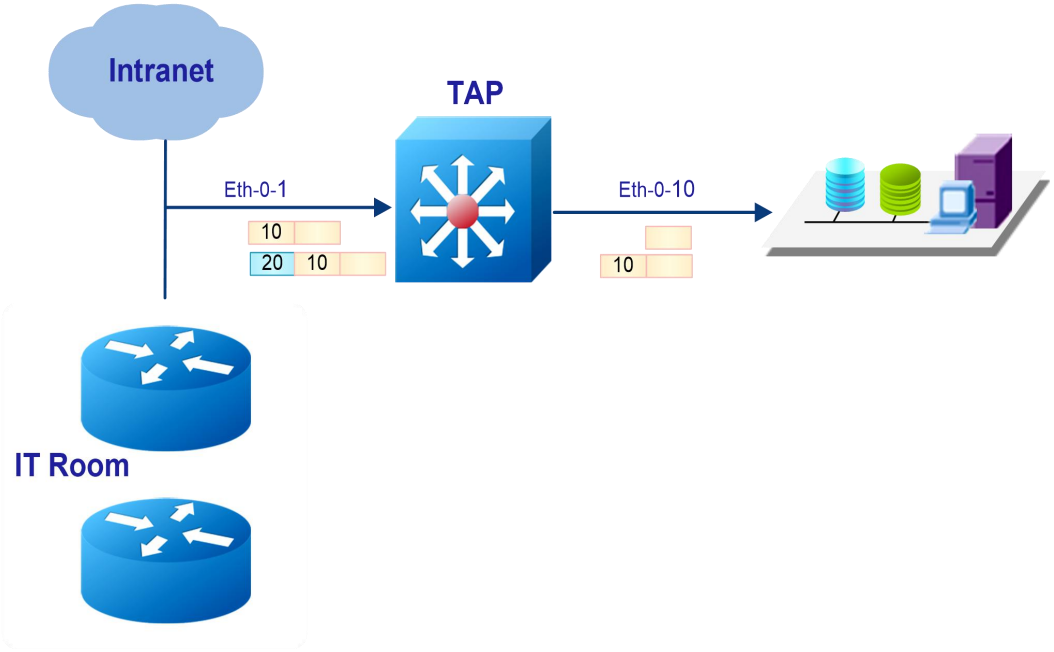
```
TAP# show running-config
!
tap-group tap1 1
 ingress eth-0-1 mark-source 10
 ingress eth-0-2 mark-source 20
 egress eth-0-10
```

NOTE The result above shows the configuration for PORT mode.

16 VLAN Stripping Configuration

16.1 Networking Requirements

Figure 16-1 Topology for VLAN stripping



16.2 Configuration Ideas

In some cases server or analyzer cannot deal with the packets with vlan tag or double vlan tags. The VLAN stripping application can resolve the problem.

Reference to Figure 16-1, Packets from eth-0-1 with VLAN 10 should be stripped the vlan tag, Packets from eth-0-1 with S-VLAN 20 C-VLAN 10 should be stripped the outer vlan tag S-VLAN 20.

VLAN stripping application should do nothing to untagged packets.

16.3 Configuration

PORT mode and PORT WITH FLOW mode both support vlan stripping.

16.3.1 VLAN Stripping for PORT mode

The following example shows how to create TAP group, strip the vlan for the packets form eth-0-1, and send a copy to eth-0-10:

```
TAP# configure terminal
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 un-tag-outer-vlan
TAP(config-tap-tap1)# egress eth-0-10
```

16.3.2 VLAN Stripping for PORT WITH FLOW mode

The following example shows how to create TAP group, strip the vlan for the packets with destination IP address 1.1.1.1form eth-0-1, and send a copy to eth-0-2:

```
TAP(config)# flow flow1
TAP(config-flow-map1)# permit any src-ip any dst-ip 1.1.1.1 0.0.0.0 un-tag-outer-vlan
TAP(config-flow-map1)# permit any src-ip any dst-ip any
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 flow flow1
TAP(config-tap-tap1)# egress eth-0-2
```

16.4 Validation

The following example shows how to display the information of the TAP group:

```
TAP# show    tap-group

TAP-group tap1
   ID: 1
   Ingress:
            eth-0-1                  un-tag-outer-vlan
   egress:
            eth-0-10
```


NOTE The result above shows the TAP group for PORT mode.

16.5 Configuration file

User can display the configuration files as below:

```
TAP# show running-config
```

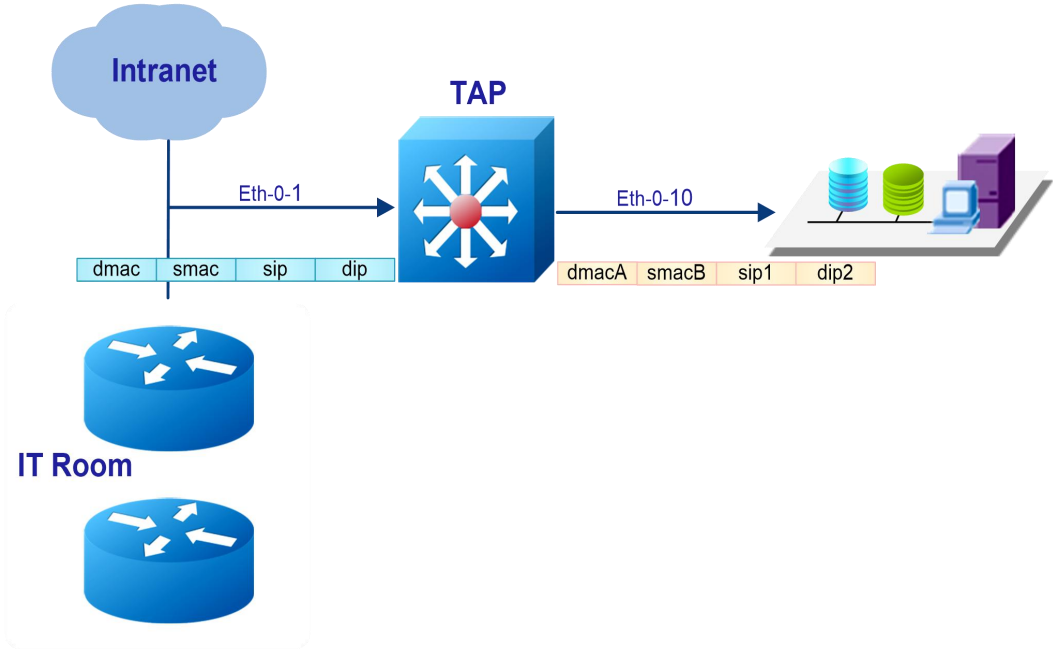
```
!
tap-group tap1 1
 ingress eth-0-1 un-tag-outer-vlan
egress eth-0-10
```

NOTE The result above shows the configuration for PORT mode.

17 Packet Editing Configuration

17.1 Networking Requirements

Figure 17-1 Topology for packet editing



NOTE Editing Source IP is not supported by T5800_TAP

17.2 Configuration Ideas

In some cases, the server or analyzer can only receive the packets with the destination address equal to its own address The packet editing application can meet the requirement. Source and destination MAC address, Source and destination IP address of the packets can be modified when enter the ingress port.

Reference to Figure 17-1, the device should modify the source and destination MAC address, Source and destination IP address of the packets from eth-0-1 and send a copy to eth-0-10.

17.3 Configuration

PORT mode and PORT WITH FLOW mode both support packet editing.

17.3.1 Packet editing for PORT mode

The following example shows how to create TAP group, edit the source and destination IP/MAC address of the packets form eth-0-1, and send a copy to eth-0-10:

```
TAP# configure terminal
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 edit-macsa a.a.a edit-macda b.b.b edit-ipda 1.1.1.1 edit-ipsa 2.2.2.2
TAP(config-tap-tap1)# egress eth-0-10
```

17.3.2 Packet editing for PORT WITH FLOW mode

The following example shows how to create TAP group with flow rule, and edit the destination IP address to 100.100.100.1 for the packets with destination IP address 1.1.1.1, edit the destination IP address to 100.100.100.2 for the packets with destination IP address 1.1.1.2:

```
TAP(config)# flow flow1
TAP(config-flow-map1)# permit any src-ip any dst-ip 1.1.1.1 0.0.0.0 edit-ipda 100.100.100.1
TAP(config-flow-map1)# permit any src-ip any dst-ip 1.1.1.2 0.0.0.0 edit-ipda 100.100.100.2
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 flow flow1
TAP(config-tap-tap1)# egress eth-0-10
```

17.4 Validation

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
  ID: 1
  Ingress:
        eth-0-1         edit-macda 000B.000B.000B
                        edit-macsa 000A.000A.000A
                        edit-ipda 1.1.1.1
                        edit-ipsa 2.2.2.2
  egress:
        eth-0-10
```

NOTE The result above shows the TAP group for PORT mode.

17.5 Configuration file

User can display the configuration files as below:

```
TAP# show running-config
tap-group tap1 1
 ingress eth-0-1 edit-macda 000B.000B.000B edit-macsa 000A.000A.000A edit-ipda 1.1.1.1
edit-ipsa 2.2.2.2
 egress eth-0-10
```

NOTE The result above shows the TAP group for PORT mode.

Editing Source IP is not supported by T5800_TAP

18 Time Stamp Configuration

18.1 Overview

To monitor the outgoing traffic of the data center is a common application scenario of TAP. With the increasement of data center scale and the improvement of the performance requirements, user need to monitor the inner traffic of the data center and get more detailed information.

TAP series device provides flexible packet remarking applications, which can insert an additional header before the original packet header. The additional header use a ether-type defined by private protocol, which can carry 16 bytes private data.

Figure 18-1 Packet structure

Flowid: ID of the flow. Default value is 0x1000, cannot be modified.

Srcport: source port ID of the packet. (The port ID is assigned by chip which is not same as the ID on the device panel)

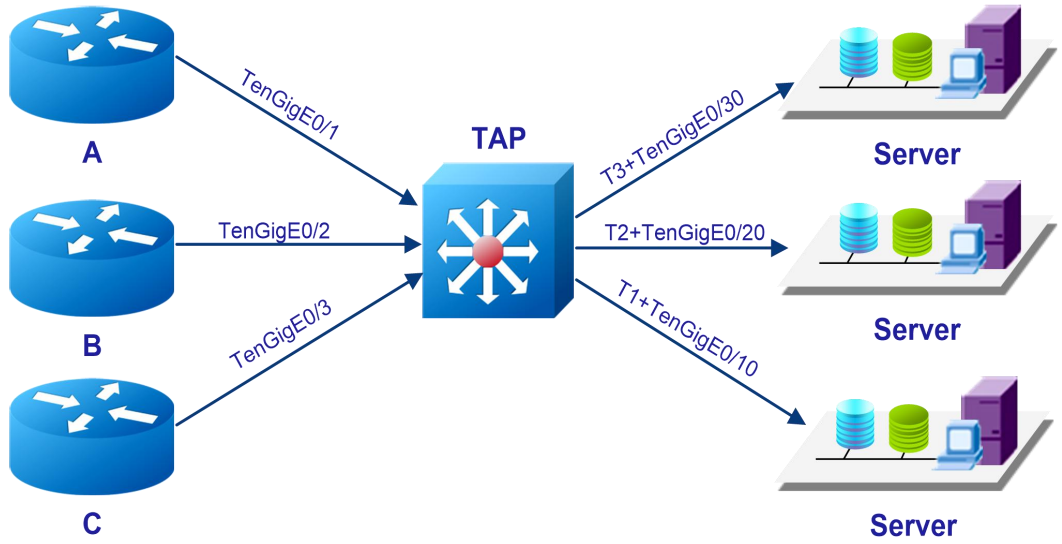
Time stamp: add the time stamp before the chip processing the packet. The duration of packets stay in the chip should not record by the timestamp.

Timestamp use standard Time of Day format. The high 32 bits record seconds (since 1970-01-01), the low 32 bits record nanosecond.

The analyzer can recognize the time stamp packets by ether header, and analyze the TCP traffic by the information carried in the packets.

18.2 Networking Requirements

Figure 18-2 Topology of Time stamp



NOTE Time stamp is not supported by T5800_TAP

18.3 Configuration Ideas

Reference to Figure 18-2, the cluster of the server can get the accurate duration the packet spent on each node of the data center by the source port and timestamp information. Use the source port to identify different devices, use the information in timestamp to get the latency.

18.4 Configuration

The following example shows how to set private ether-type to 0xFF12, and set the destination MAC address to 1.1.1, set the source MAC address to 2.2.2:

```
TAP# configure terminal
TAP(config)# timestamp-over-ether 1.1.1 2.2.2 0xff12
```

The following example shows how to create 3 TAP groups, with 3 source ports eth-0-1/eth-0-2/eth-0-3, and with 3 destination ports eth-0-10/eth-0-20/eth-0-30 which enabled time stamp:

```
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1
TAP(config-tap-tap1)# egress eth-0-10 timestamp
```

```
TAP(config-tap-tap1)# exit
TAP(config)# tap-group tap2
TAP(config-tap-tap2)# ingress eth-0-2
TAP(config-tap-tap2)# egress eth-0-20 timestamp
TAP(config-tap-tap2)# exit
TAP(config)# tap-group tap3
TAP(config-tap-tap3)# ingress eth-0-3
TAP(config-tap-tap3)# egress eth-0-30 timestamp
TAP(config-tap-tap3)# exit
```

18.5 Validation

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
ID:
   Ingress:
            eth-0-1
   egress:
            eth-0-10            time-stamp
TAP-group tap2
ID: 2
   Ingress:
            eth-0-2
   egress:
            eth-0-20            time-stamp
TAP-group tap3
ID: 3
   Ingress:
            eth-0-3
   egress:
            eth-0-30            time-stamp
```

18.6 Configuration file

User can display the configuration files as below:

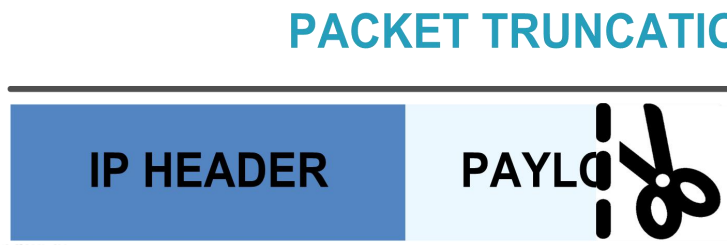
```
TAP# show running-config
!
timestamp-over-ether 0001.0001.0001 0002.0002.0002 0xff12
!
tap-group tap1 1
  ingress eth-0-1
egress eth-0-10 timestamp
!
tap-group tap2 2
  ingress eth-0-2
  egress eth-0-20 timestamp
```

```
!
tap-group tap3 3
 ingress eth-0-3
 egress eth-0-30 timestamp
```

19 Packet truncation Configuration

19.1 Networking requirements

Figure 19-1 sketch map of packet truncation



NOTE Packet truncation is not supported by T5800_TAP

19.2 Configuration Ideas

In some cases, packets need to be truncated in order to reduce the pressure of the server or in order to protect privacy. The packet truncation application can meet the requirement.

E.g. the size of packet enters the TAP device from eth-0-1 is 1518 bytes. The size of packet leaves destination port eth-0-10 is 64 byte.

19.3 Configuration

PORT mode and PORT WITH FLOW mode both support packet truncation.

19.3.1 Packet Truncation for PORT mode

The following example shows how to set the packet length after truncated to 64 byte:

```
TAP# configure terminal
TAP(config)# truncation 64
```

The follow example shows how to create TAP group with ingress port eth-0-1 and enable packet truncation:

```
TAP# configure terminal
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 truncation
TAP(config-tap-tap1)# egress eth-0-10
```

19.3.2 Packet Truncation for PORT WITH FLOW mode

The following example shows how to set a flow rule to match the packets with destination IP address 1.1.1.0/24 and enable truncation. Packets with other destination IP address should not be truncated:

```
TAP(config)# flow flow1
TAP(config-flow-flow1)# permit any src-ip any dst-ip 1.1.2.0 0.0.0.255 truncation
TAP(config-flow-flow1)# permit any src-ip any dst-ip any
TAP(config-flow-flow1)# exit
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 flow flow1
TAP(config-tap-tap1)# egress eth-0-10
TAP(config-tap-tap1)# end
```

19.4 Validation

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
ID: 1
  Ingress:
        eth-0-1              truncation
  egress:
        eth-0-10
```

NOTE The result above shows the TAP group for PORT mode.

19.5 Configuration file

User can display the configuration files as below:

```
TAP# show running-config
!
truncation 64
!
```

```
tap-group tap1 1
 ingress eth-0-1 truncation
egress eth-0-10
```

NOTE Packet truncation is mutual exclusive to other actions. E.g. Only Packet truncation is effective and all other configuration(egress-filter/time stamp etc.) is invalid in the following configuration:

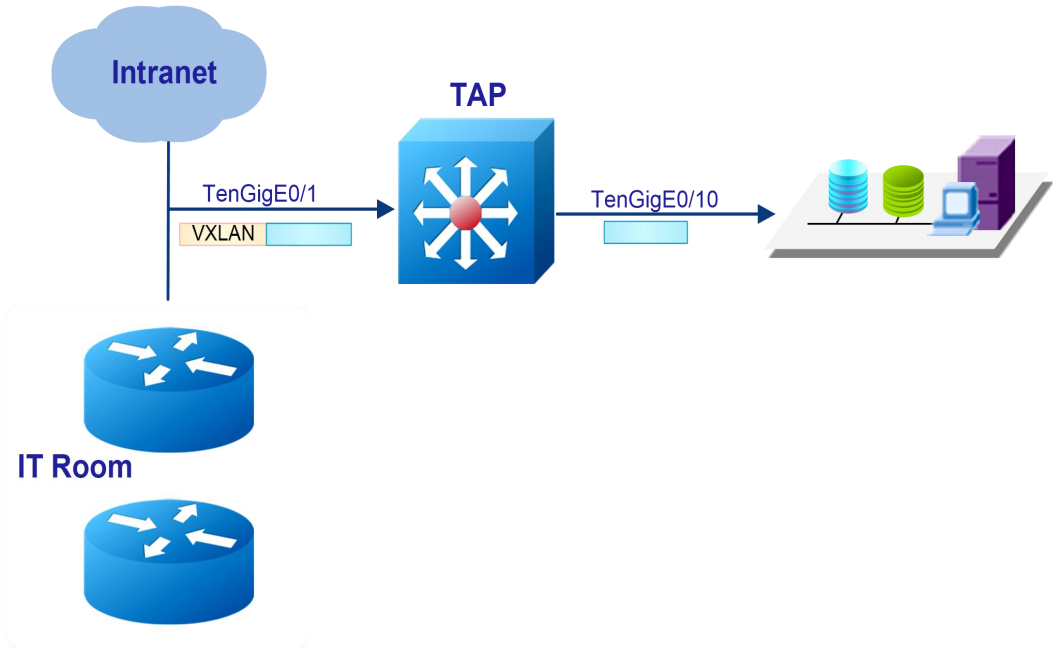
```
ip access-list filter1
sequence-num 10 deny any src-ip any dst-ip any
!
interface eth-0-2
   egress filter1
!
timestamp-over-ether 000A.000A.000A 000B.000B.000B 0xff12
!
tap-group tap1
 ingress eth-0-1 truncation
 egress eth-0-2 timestamp
```

20 Packet header stripping Configuration

20.1 Configuring strip the VXLAN header

20.1.1 Networking Requirements

Figure 20-1 Topology of stripping VXLAN header



NOTE Packet header stripping is not supported by T5800_TAP

20.1.2 Configuration Ideas

In some cases, server or analyzer cannot parse the packet with VXLAN/NVGRE/GRE header.

The packet header stripping application can resolve the problem.

Reference to Figure 20-1 the packet enter eth-0-1, the VLAN header should be stripped

20.1.3 Configuration

The following example shows how to create a flow rule the match the VXLAN packets and strip the header:

```
TAP(config)# flow flow1
TAP(config-flow-flow1)# permit udp dst-port eq 4789 src-ip any dst-ip any strip-header
TAP(config-flow-flow1)# exit
```

The following example shows how to create a TAP group with ingress port eth-0-1 and flow1:

```
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 flow flow1
TAP(config-tap-tap1)# egress eth-0-10
TAP(config-tap-tap1)# end
```

20.1.4 Validation

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
ID: 1
   Ingress:
            eth-0-1                 flow flow1
   egress:
            eth-0-10
```

20.1.5 Configuration file

User can display the configuration files as below:

```
TAP# show running-config
!
flow flow1
 sequence-num 10 permit udp dst-port eq 4789 src-ip any dst-ip any strip-header
!
tap-group tap1 1
 ingress eth-0-1 flow flow1
egress eth-0-10
```

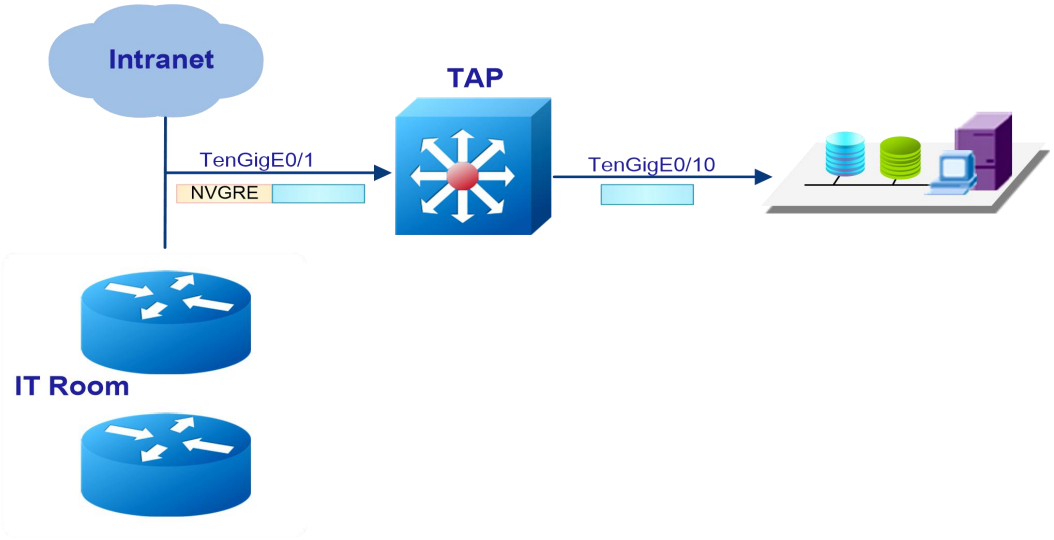
NOTE TAP series devices support to match the specified VNI. E.g. match VNI 1000 and strip the VXLAN header:

```
TAP(config)# flow flow1
TAP(config-flow-map1)# permit udp dst-port eq 4789 vxlan-vni 1000 0x0 src-ip any dst-ip any
strip-header
TAP(config-tap-tap1)# end
```

20.2 Configuring strip the NVGRE header

20.2.1 Networking Requirements

Figure 20-2 Topology of stripping NVGRE header



20.2.2 Configuration Ideas

In some cases, server or analyzer cannot parse the packet with VXLAN/NVGRE/GRE header.

The packet header stripping application can resolve the problem.

Reference to Figure 20-2 the packet enter eth-0-1, the NVGRE header should be stripped

20.2.3 Configuration

The following example shows how to create a flow rule the match the NVGRE packets and strip the header:

```
TAP(config)# flow flow1
TAP(config-flow-flow1)# permit nvgre src-ip any dst-ip any strip-header
TAP(config-flow-flow1)# exit
```

The following example shows how to create a TAP group with ingress port eth-0-1 and flow1:

```
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 flow flow1
TAP(config-tap-tap1)# egress eth-0-10
TAP(config-tap-tap1)# end
```

20.2.4 Validation

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
ID: 1
   Ingress:
            eth-0-1                 flow flow1
   egress:
            eth-0-10
```

20.2.5 Configuration file

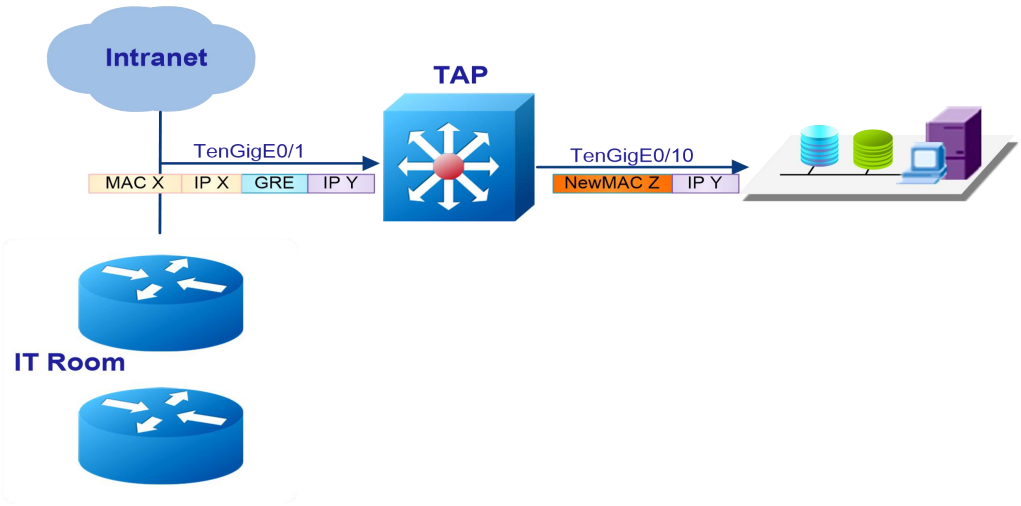
User can display the configuration files as below:

```
TAP# show running-config
!
flow flow1
sequence-num 10 permit nvgre src-ip any dst-ip any strip-header
!
tap-group tap1 1
  ingress eth-0-1 flow flow1
egress eth-0-10
```

20.3 Configuring strip the GRE header

20.3.1 Networking Requirements

Figure 20-3 Topology of stripping GRE header



20.3.2 Configuration Ideas

In some cases, server or analyzer cannot parse the packet with VXLAN/NVGRE/GRE header.

The packet stripping application for GRE packet should strip the outer IP address, MAC address and GRE header, only inner IP address and payload are left. Packet editing application should be configured together with packet header stripping, in order to add outer MAC address.

Reference to Figure 20-3 the packet enter eth-0-1, the GRE header should be stripped and a new MAC address should be added.

20.3.3 Configuration

The following example shows how to create a flow rule the match the GRE packets and strip the header:

```
TAP(config)# flow flow1
TAP(config-flow-flow1)# permit gre src-ip any dst-ip any strip-header edit-macsa a.a.a edit-macda b.b.b
TAP(config-flow-flow1)# exit
```

The following example shows how to create a TAP group with ingress port eth-0-1 and flow1:

```
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 flow flow1
TAP(config-tap-tap1)# egress eth-0-10
TAP(config-tap-tap1)# end
```

20.3.4 Validation

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
ID: 1
  Ingress:
        eth-0-1              flow flow1
  egress:
        eth-0-10
```

NOTE If the original GRE packets have GRE-KEY field, the key word "gre-key" should be configured when create the flow rule:

```
TAP(config)# flow flow1
TAP(config-flow-flow1)# permit gre gre-key any src-ip any dst-ip any strip-header edit-macsa a.a.a edit-macda b.b.b
TAP(config-flow-flow1)# exit
```

20.3.5 Configuration file

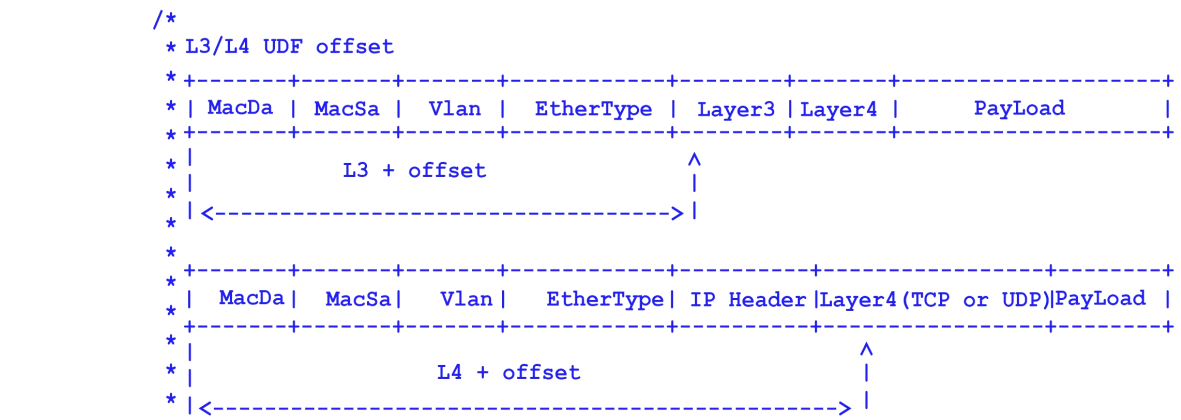
User can display the configuration files as below:

```
TAP# show running-config
!
flow flow1
sequence-num 10 permit gre src-ip any dst-ip any strip-header edit-macda 000B.000B.000B
edit-macsa 000A.000A.000A
!
tap-group tap1 1
 ingress eth-0-1 flow flow1
egress eth-0-10
```

20.4 Configuring strip the User Defined header

20.4.1 Networking Requirements

Figure 20-4 Packet structure



20.4.2 Configuration Ideas

Normal packet header stripping application can strip the standard VXLAN/GRE/NVGRE header, which cannot match all cases. e.g. GRE header may have variable length because GRE-KEY/Checksum/Sequence Num inserted..

By default, packet header stripping can strip GRE header and one option field of 4 bytes. When the GRE packet has more than one option fields, the packet header stripping cannot strip them correctly.

The user defined header stripping application can resolve the problem.

A starting position (L2, L3 or L4)and offset (up to 30 bytes) should be specified before using user defined header stripping.

The following example shows how to strip the GRE packets with GRE-KEY/Checksum/Sequence Number

20.4.3 Configuration

Create a flow rule to match GRE packets and enable user defined stripping:

```
TAP(config)# flow flow1
TAP(config-flow-flow1)# permit gre src-ip any dst-ip any strip-header strip-position l4 strip-offset 16 edit-macsa a.a.a edit-macda b.b.b
TAP(config-flow-flow1)# exit
```

NOTE Strip-position is L4 and offset is 16 means remove 16 bytes after L4 header and remove all fields before L4 header.

Create a TAP group with ingress port eth-0-1 and flow1:

```
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 flow flow1
TAP(config-tap-tap1)# egress eth-0-10
TAP(config-tap-tap1)# end
```

20.4.4 Validation

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
ID: 1
  Ingress:
        eth-0-1              flow flow1
  egress:
        eth-0-10
```

20.4.5 Configuration file

User can display the configuration files as below:

```
TAP# show running-config
!
flow flow1
sequence-num 10 permit gre src-ip any dst-ip any strip-header strip-position l4 strip-offset 16 edit-macda 000B.000B.000B edit-macsa 000A.000A.000A
!
tap-group tap1 1
 ingress eth-0-1 flow flow1
```

```
egress eth-0-10
```

20.5 Configuring strip the ERSPAN header

20.5.1 Networking Requirements

Figure 20-5 Topology of stripping ERSPAN header

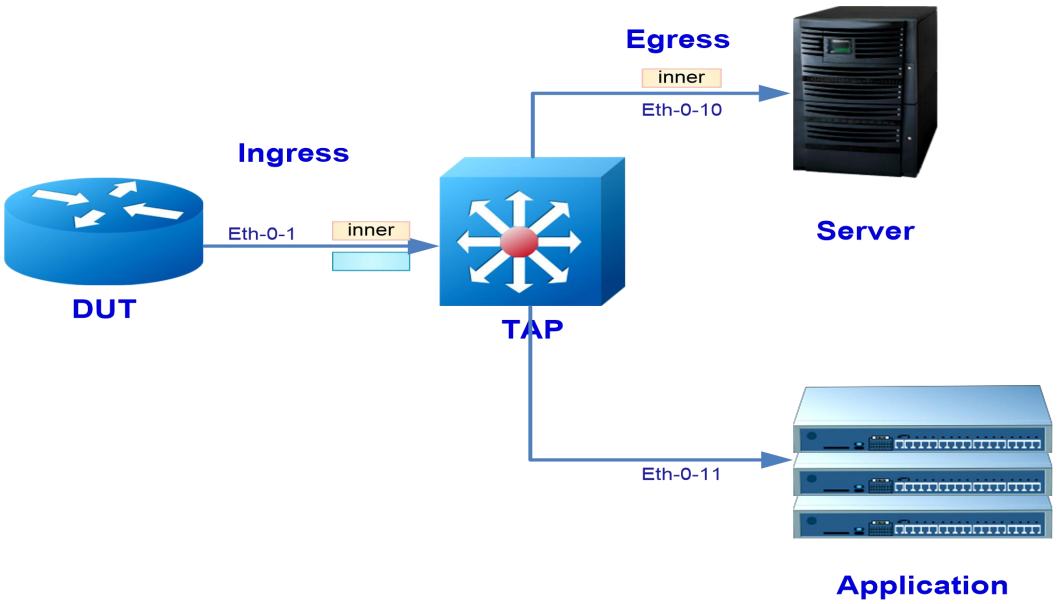


Figure 20-6 Packet structure

NOTE Flow rule which can match ERSPAN is only supported by T5850 & T8050_TAP.

20.5.2 Configuration Ideas

In some cases, user needs to match the packets with ERSPAN ID among those packets with GRE+ERSPAN+VXLAN header, and need to strip the ERSPAN header. VXLAN header may also need to be stripped.

Matching ERSPAN ID can meet the requirement.

20.5.3 Configuration

The following example shows how to create a flow rule to match the ERSPAN packets with span id 1, and strip the ERSPAN header and edit vlan 1:

```
TAP(config)# flow erspan
TAP(config-flow- erspan)# permit gre erspan 1 0x0 src-ip any dst-ip any strip-header edit-vlan 1
```

The following example shows how to create a TAP group with ingress port eth-0-1 and flow erspan:

```
TAP(config)# tap-group tap1
TAP(config-tap-tap1)# ingress eth-0-1 flow erspan
TAP(config-tap-tap1)# egress eth-0-10
```

20.5.4 Validation

The following example shows how to display the information of the flow rule:

```
TAP# show flow
flow erspan
 sequence-num 10 permit gre erspan 1 0x0 src-ip any dst-ip any strip-header edit
-vlan 1
```

NOTE strip-header and edit-vlan must be configured to strip the erspan header. "edit-vlan" should add a vlan tag when packet is untagged and should edit the vlan when the packet is tagged.

The following example shows how to display the information of the TAP group:

```
TAP# show tap-group

TAP-group tap1
ID: 1
  Ingress:
        eth-0-1              flow erspan
  egress:

        eth-0-10
```

20.5.5 Configuration file

User can display the configuration files as below:

```
TAP# show running-config
!
flow erspan
```

```
 sequence-num 10 permit gre erspan 1 0x0 src-ip any dst-ip any strip-header edit
-vlan 1
!
tap-group tap1 1
ingress eth-0-1 flow erspan
egress eth-0-10
```

21 AAA Configuration

AAA（Authentication/Authorization/Accounting）is an security mechanism

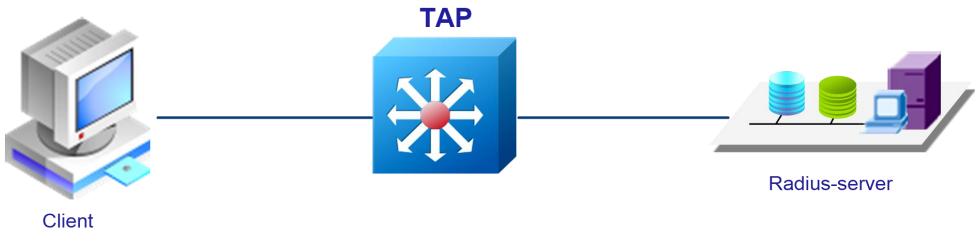
for network management, which support 3 applications: Authentication, Authorization and Accounting

The TAP series devices support to certify the users access the network.

21.1 Configuring Radius Authentication

21.1.1 Networking requirements

Figure 21-1 Topology of Radius Authentication



21.1.2 Configuration Ideas

Radius is a distributed server/client system to prevent unauthorized access and to guarantee the security of the network.

Radius server keeps all information of users' authentication and network service accessing. Radius server should do Authentication/Authorization/Accounting according the user information in local database, after it received request from a client.

21.1.3 Configuration

The following example shows how to enable AAA and set the mode of Authentication/Authorization/Accounting:

```
TAP(config)# aaa new-model
TAP(config)# aaa authentication login radius-authen radius
TAP(config)# aaa authorization exec radius-author radius
TAP(config)# aaa accounting exec radius-acct start-stop radius
```

The following example shows how to set the parameter of the radius server:

```
TAP(config)# radius-server host mgmt-if 10.10.1.1 key test auth-port 1819
```

The following example shows how to set the login mode to radius:

```
TAP(config)# line vty 0 7
TAP(config-line)# login authentication radius-authen
TAP(config-line)# privilege level 4
TAP(config-line)# no line-password
```

21.1.4 Validation

Use the username and password on radius server to login the device.

21.1.5 Configuration file

User can display the configuration files as below:

```
TAP# show running-config
aaa new-model
!
aaa authentication login radius-authen radius
!
aaa authorization exec radius-author radius
!
aaa accounting exec radius-acct start-stop radius
!
line vty 0 7
 exec-timeout 35791 0
 privilege level 4
 no line-password
 login authentication radius-authen
```

22 Sflow Configuration

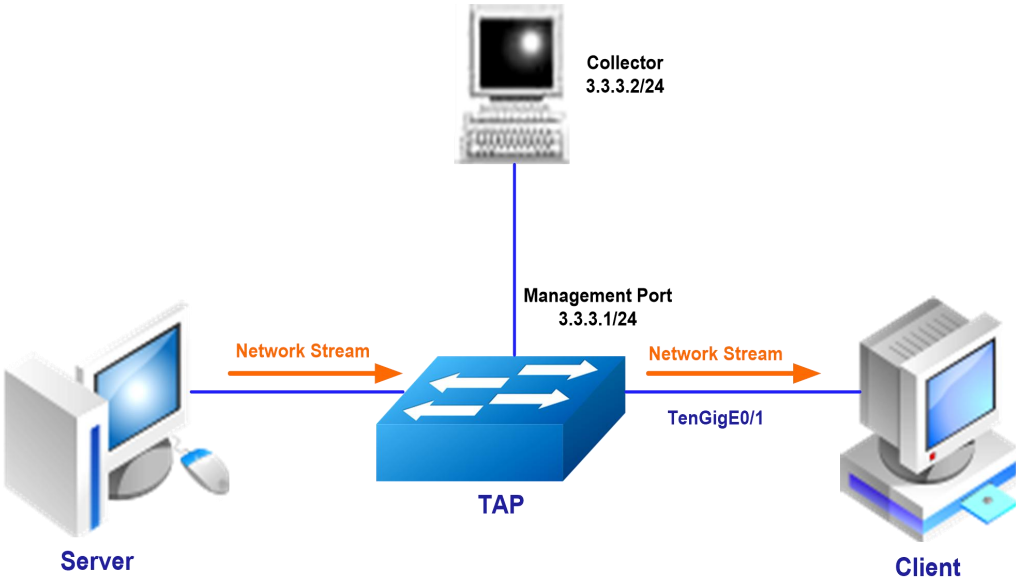
Sflow (Sampled Flow) is a traffic monitoring technology based on packet sampling.

Sflow is used to analyze the network traffic.

Sflow has 2 types of message: statistics information for ports and sampled packets information.

22.1 Networking requirements

Figure 22-1 Topology of sflow



22.2 Configuration Ideas

Traffic monitoring is a basic requirement of network management.

User need to find the source abnormal traffic and attacking traffic in time. Sflow, which is a traffic monitoring technology based on packet sampling can meet the requirement.

22.3 Configuration

The following example shows how to enable sflow and set the sampling interval, IP address of the agent and IP address of the collector:

```
TAP(config)# sflow enable
TAP(config)# sflow counter interval 20
TAP(config)# sflow agent ip 3.3.3.1
TAP(config)# sflow collector mgmt-if 3.3.3.2
```

The follow example shows how to enable sflow on a port and set the sampling rate:

```
TAP(config)# interface eth-0-1
TAP(config-if)# sflow flow-sampling rate 32768
TAP(config-if)# sflow flow-sampling enable input
TAP(config-if)# sflow counter-sampling enable
```

22.4 Validation

The following example shows how to display the information of sflow:

```
TAP# show sflow
sFlow Version: 4
sFlow Global Information:
  Agent IPv4 address                 : 3.3.3.1
  Counter Sampling Interval          : 20 seconds
  Collector 1:
    IPv4 Address: 3.3.3.2
    Port: 6343

sFlow Port Information:
                                          Flow-Sample     Flow-Sample
  Port            Counter     Flow        Direction       Rate
  ------------------------------------------------------------
  XGe0-1          enable      enable      Input           32768
```

22.5 Configuration file

User can display the configuration files as below:

```
TAP# show running-config
!
sflow enable
sflow agent ip 3.3.3.1
sflow counter interval 20
!
sflow collector mgmt-if 3.3.3.2
```

```
!
interface eth-0-1
 speed 1000
 duplex full
 sflow counter-sampling enable
 sflow flow-sampling enable input
!
```

23 Tips

1. To full fill the keyword of any command line in any command mode, use TAB on the keyboard. It is unnecessary to type every letter of the keywords.
2. To get the help information of the command line, use the “?” symbol.
3. To quit to the up level of the command mode, use “quit” or “exit”. To return to Privileged EXEC mode, use “end”.
4. To save the current configuration, use “write memory”. User should use the “write memory” command on time in order to prevent loss the configuration after device reboot.
5. To get more description of the command line, please reference to the CLI guide.
6. To get detailed information about the feature, please reference to the User guide.
7. Only one user can login to the configure mode/interface mode/application configuration mode at same time.
8. The “no” form of the command line is usually used to delete the configuration or restore the default value. E.g.: configuration “speed 1000”should be removed by “no speed”.