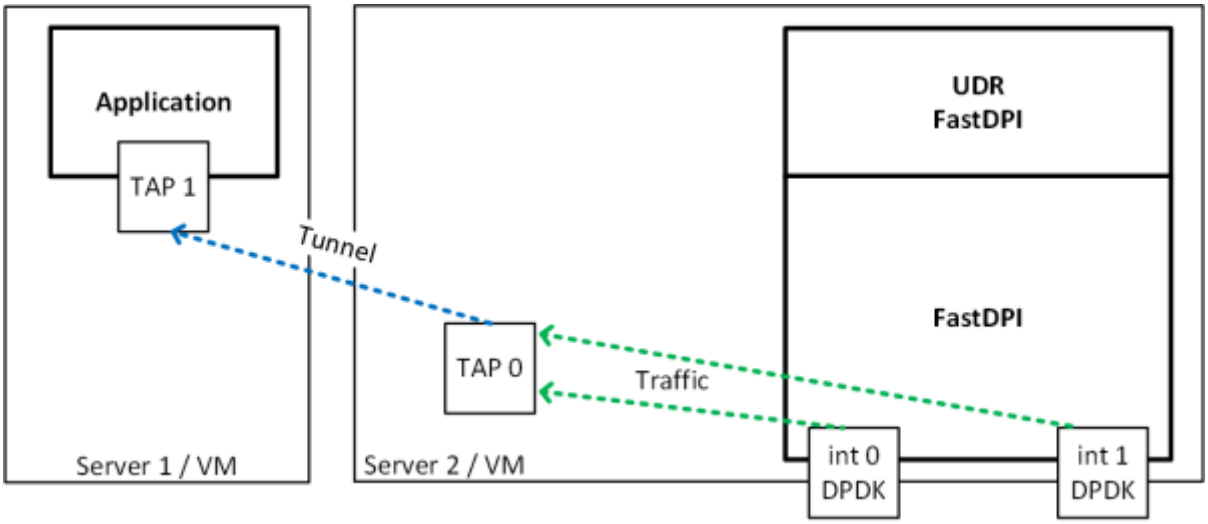
Table of Contents

Traffic Mirroring to External Platforms

A part of the traffic can be mirror from the common channel and transmitted for analysis and processing to external platforms.



TAP Interfaces Configuration

- TAP0 - is used for traffic diversion
- TAP1 - server side port that receives traffic
- Tunnel is created between TAP0 and TAP1 to transmit traffic to an external platform
- mac learning is disabled on the TAP0 interface

Run the following commands from the console:

```
ip tuntap add tap0 mode tap
ip tuntap add tap1 mode tap

ip link set dev tap0 up
ip link set dev tap1 up

ip link add br0 type tunnel

ip link set tap0 master br0
bridge link set dev tap0 learning off
ip link set tap1 master br0

ifconfig tap0 192.168.4.20 up
ifconfig tap1 192.168.4.21 up
ifconfig br0 up
```

Warning: TAP interfaces must be started after server restart!

Traffic Diversion Configuration

Set up traffic diversion (the variant with the named profile):

```
  fdpi_ctrl load profile --service 14 --profile.name radius_accounting  --
profile.json '{ "typedev" : "tap","dev" : "tap0","udp" : [ 1813 ] }' --
outformat=json
  fdpi_ctrl load --service 14 --profile.name radius_accounting --ip
10.16.252.11
```

here "typedev" : "tap" - type of device fox traffic diversion (TAP)

```
  "dev" : "tap0" - the name of the network interface to which traffic is
diverted
  "udp" : [ 1813 ] - protocol and port number of the diverted traffic
```

An example of use can be found in the section Radius Configuration Example #2
<html> </html>

Autorun Setup

To configure the traffic diversion interfaces at system startup, you have to:

1. Add to the file **/etc/rc.d/rc.local**:

```
 ip tuntap add tap0 mode tap
ip tuntap add tap1 mode tap

ip link set dev tap0 up
ip link set dev tap1 up

ip link add br0 type tunnel

ip link set tap0 master br0
bridge link set dev tap0 learning off
ip link set tap1 master br0

ifconfig tap0 192.168.4.20 up
ifconfig tap1 192.168.4.21 up
ifconfig br0 up
```

2. Add **rc.local** processing permission:

```
 chmod +x /etc/rc.d/rc.local
```

3. Change the file **/usr/lib/systemd/system/rc-local.service**:

```
 [Unit]
Description=/etc/rc.d/rc.local Compatibility
```

```
Documentation=man:systemd-rc-local-generator(8)
ConditionFileIsExecutable=/etc/rc.d/rc.local
After=network.target network-online.target

[Service]
Type=forking
ExecStart=/etc/rc.d/rc.local start
TimeoutSec=0
RemainAfterExit=yes
GuessMainPID=no

[Install]
WantedBy=multi-user.target
```

4. Enable **rc.local**, so that it starts every time after reboot:

```
systemctl enable rc-local
```