NAME

ovs-test − Check Linux drivers for performance, vlan and L3 tunneling problems

SYNOPSIS

ovs−test −s *port*

ovs−test −c *server1 server2* [**−b** *targetbandwidth*] [**−i** *testinterval*] [**−d**]
[**−l** *vlantag*] [**−t** *tunnelmodes*]

DESCRIPTION

The **ovs−test** program may be used to check for problems sending 802.1Q or GRE traffic that Open vSwitch may uncover. These problems, for example, can occur when Open vSwitch is used to send 802.1Q traffic through physical interfaces running certain drivers of certain Linux kernel versions. To run a test, configure IP addresses on *server1* and *server2* for interfaces you intended to test. These interfaces could also be already configured OVS bridges that have a physical interface attached to them. Then, on one of the nodes, run **ovs−test** in server mode and on the other node run it in client mode. The client will connect to **ovs−test** server and schedule tests between both of them. The **ovs−test** client will perform UDP and TCP tests.

UDP tests can report packet loss and achieved bandwidth for various datagram sizes. By default target bandwidth for UDP tests is 1Mbit/s.

TCP tests report only achieved bandwidth, because kernel TCP stack takes care of flow control and packet loss. TCP tests are essential to detect potential TSO related issues.

To determine whether Open vSwitch is encountering any problems, the user must compare packet loss and achieved bandwidth in a setup where traffic is being directly sent and in one where it is not. If in the 802.1Q or L3 tunneled tests both **ovs−test** processes are unable to communicate or the achieved bandwidth is much lower compared to direct setup, then, most likely, Open vSwitch has encountered a pre−existing kernel or driver bug.

Some examples of the types of problems that may be encountered are:

- When NICs use VLAN stripping on receive they must pass a pointer to a *vlan_group* when reporting the stripped tag to the networking core. If no *vlan_group* is in use then some drivers just drop the extracted tag. Drivers are supposed to only enable stripping if a *vlan_group* is registered but not all of them do that.
- On receive, some drivers handle priority tagged packets specially and don't pass the tag onto the network stack at all, so Open vSwitch never has a chance to see it.
- Some drivers size their receive buffers based on whether a *vlan_group* is enabled, meaning that a maximum size packet with a VLAN tag will not fit if no *vlan_group* is configured.
- On transmit, some drivers expect that VLAN acceleration will be used if it is available, which can only be done if a *vlan_group* is configured. In these cases, the driver may fail to parse the packet and correctly setup checksum offloading or TSO.

Client Mode

An **ovs−test** client will connect to two **ovs−test** servers and will ask them to exchange test traffic. It is also possible to spawn an **ovs−test** server automatically from the client.

Server Mode

To conduct tests, two **ovs−test** servers must be running on two different hosts where the client can connect. The actual test traffic is exchanged only between both **ovs−test** servers. It is recommended that both servers have their IP addresses in the same subnet, otherwise one would have to make sure that routing is set up correctly.

OPTIONS

–s <port>, ––server <port>
Run in server mode and wait for the client to establish XML RPC Control Connection on this TCP port. It is recommended to have *ethtool(8)* installed on the server so that it could retrieve information about the NIC driver.

–c <server1> <server2>, ––client <server1> <server2>
Run in client mode and schedule tests between *server1* and *server2*, where each server must be given in the following format:

```
OuterIP[:OuterPort],InnerIP[/Mask][:InnerPort].
```

The *OuterIP* must be already assigned to the physical interface which is going to be tested. This is the IP address where client will try to establish XML RPC connection. If *OuterIP* is 127.0.0.1 then client will automatically spawn a local instance of **ovs–test** server. OuterPort is TCP port where server is listening for incoming XML/RPC control connections to schedule tests (by default it is 15531). The **ovs–test** will automatically assign *InnerIP[/Mask]* to the interfaces that will be created on the fly for testing purposes. It is important that *InnerIP[/Mask]* does not interfere with already existing IP addresses on both **ovs–test** servers and client. InnerPort is port which will be used by server to listen for test traffic that will be encapsulated (by default it is 15532).

–b <targetbandwidth>, ––bandwidth <targetbandwidth>
Target bandwidth for UDP tests. The targetbandwidth must be given in bits per second. It is possible to use postfix *M* or *K* to alter the target bandwidth magnitude.

–i <testinterval>, ––interval <testinterval>
How long each test should run. By default 5 seconds.

–h, ––help
Prints a brief help message to the console.

–V, ––version
Prints version information to the console.

The following test modes are supported by **ovs–test**. It is possible to combine multiple of them in a single **ovs–test** invocation.

–d, ––direct
Perform direct tests between both OuterIP addresses. These tests could be used as a reference to compare 802.1Q or L3 tunneling test results.

–l <vlantag>, ––vlan–tag <vlantag>
Perform 802.1Q tests between both servers. These tests will create a temporary OVS bridge, if necessary, and attach a VLAN tagged port to it for testing purposes.

–t <tunnelmodes>, ––tunnel–modes <tunnelmodes>
Perform L3 tunneling tests. The given argument is a comma sepa- rated string that specifies all the L3 tunnel modes that should be tested (e.g. gre). The L3 tunnels are terminated on interface that has the OuterIP address assigned.

EXAMPLES

On host 1.2.3.4 start **ovs–test** in server mode:

```
ovs-test -s 15531
```

On host 1.2.3.5 start **ovs–test** in client mode and do direct, VLAN and GRE tests between both nodes:

```
ovs-test -c 127.0.0.1,1.1.1.1/30 1.2.3.4,1.1.1.2/30 -d -l 123 -t
gre
```

SEE ALSO

ovs−vswitchd(8), *ovs−ofctl(8)*, *ovs−vsctl(8)*, **ovs−vlan−test**, *ethtool(8)*, *uname(1)*

AUTHOR

The Open vSwitch Development Community

COPYRIGHT