

**NAME**

`ovs-vlan-test` – Check Linux drivers for problems with vlan traffic

**SYNOPSIS**

`ovs-vlan-test` [`-s` | `--server`] *control\_ip* *vlan\_ip*

**DESCRIPTION**

The `ovs-vlan-test` utility has some limitations, for example, it does not use TCP in its tests. Also it does not take into account MTU to detect potential edge cases. To overcome those limitations a new tool was developed – `ovs-test`. `ovs-test` is currently supported only on Debian so, if possible, try to use that on instead of `ovs-vlan-test`.

The `ovs-vlan-test` program may be used to check for problems sending 802.1Q traffic which may occur when running Open vSwitch. These problems 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 Open vSwitch to tag traffic originating from *vlan\_ip* and forward it out the target interface. Then run the `ovs-vlan-test` in client mode connecting to an `ovs-vlan-test` server. `ovs-vlan-test` will display “OK” if it did not detect problems.

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-vlan-test` client may be run on a host to check for VLAN connectivity problems. The client must be able to establish HTTP connections with an `ovs-vlan-test` server located at the specified *control\_ip* address. UDP traffic sourced at *vlan\_ip* should be tagged and directed out the interface whose connectivity is being tested.

**Server Mode**

To conduct tests, an `ovs-vlan-test` server must be running on a host known not to have VLAN connectivity problems. The server must have a *control\_ip* on a non-VLAN network which clients can establish connectivity with. It must also have a *vlan\_ip* address on a VLAN network which clients will use to test their VLAN connectivity. Multiple clients may test against a single `ovs-vlan-test` server concurrently.

**OPTIONS**

`-s, --server`

Run in server mode.

`-h, --help`

Prints a brief help message to the console.

`-V, --version`

Prints version information to the console.

**EXAMPLES**

Display the Linux kernel version and driver of *eth1*:

```
uname -r
ethtool -i eth1
```

Set up a bridge which forwards traffic originating from *1.2.3.4* out *eth1* with VLAN tag 10:

```
ovs-vsctl -- add-br vlan-br \
-- add-port vlan-br eth1 \
-- add-port vlan-br vlan-br-tag tag=10 \
-- set Interface vlan-br-tag type=internal
ip addr add 1.2.3.4/8 dev vlan-br-tag
ip link set vlan-br-tag up
```

Run an **ovs-vlan-test** server listening for client control traffic on *172.16.0.142* port *8080* and VLAN traffic on the default port of *1.2.3.3*:

```
ovs-vlan-test -s 172.16.0.142:8080 1.2.3.3
```

Run an **ovs-vlan-test** client with a control server located at *172.16.0.142* port *8080* and a local VLAN IP of *1.2.3.4*:

```
ovs-vlan-test 172.16.0.142:8080 1.2.3.4
```

## SEE ALSO

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

## AUTHOR

The Open vSwitch Development Community

## COPYRIGHT

2016, The Open vSwitch Development Community