# VRRP Automation Test Script

This document describes the procedure of the automated script that tests the basic functionality of VRRP. The test set up is described in <https://confluence.sierrawireless.com/display/ALENGINEERING/VRRP+Testbed>. The automation script assumes that the Cisco router acting as the VRRP Master is configured correctly and the settings in the VRRP section of \config\common\_testbed\_conf.yml are correct. Figure 1 shows the settings in the configuration file.

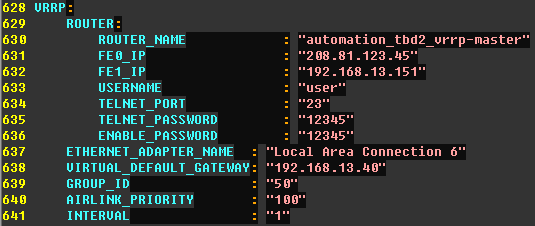


Figure : VRRP Settings in \config\common\_testbed\_conf.yml

# Integration with automation test bed

The test set up described on Confluence is used for manual testing. The automation scripts runs on an automation test bed with a controller PC separate from the host computer. The VRRP set up has been integrated with automation test bed 2 in the Richmond lab by adding a Cisco router to the Airlink LAN switch. This addition does not disturb other test scripts. The full test bed set up is shown in Figure 2.

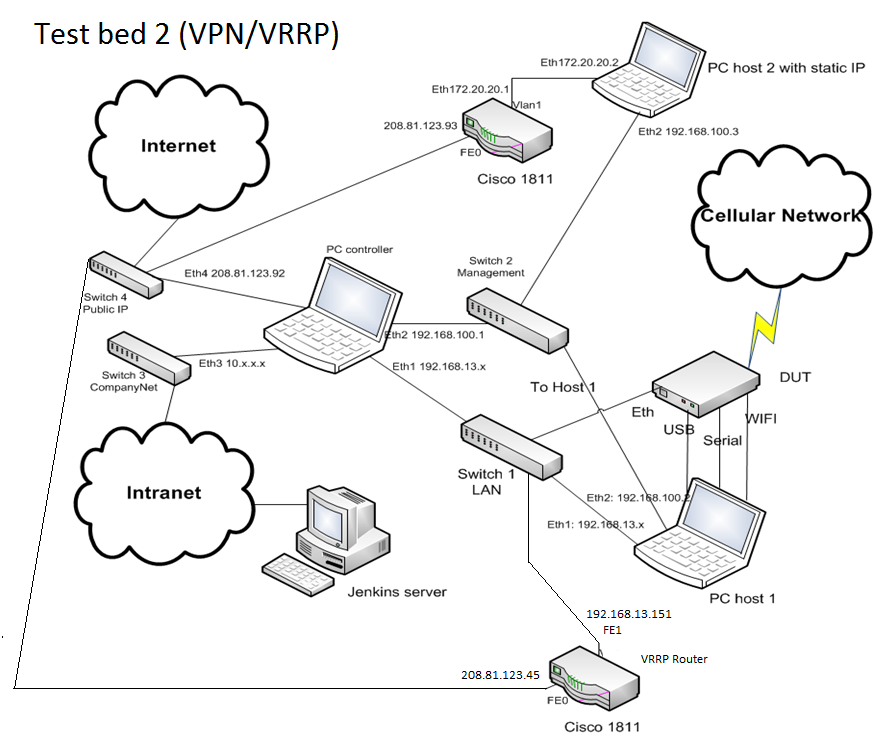


Figure 2: Automation Test Bed Setup

# Test Procedures

This section describes the test procedures programmed into the script.

## Change Airlink device’s default gateway

The Airlink device’s default gateway is changed from the usual 192.168.13.31 to the virtual IP address of the VRRP setup. This is direct all traffic destined for the internet to pass through the VRRP Master which currently is the Cisco router.

## Configure VRRP settings on the Airlink device

The Airlink device’s VRRP settings are configured using ACEmanager and the device is rebooted. The settings are verified to be correct after the reboot before continuing with the test.

## Simulating the VRRP Master malfunctioning

This is done by shutting down the inside interface (FE1) of the Cisco router. From the Airlink device’s perspective, the Cisco router is gone as it is removed from the network completely and no VRRP keepalive messages are received from the Cisco router. In the background, 50 ping packets are sent from the host computer to 8.8.8.8. When FE1 is shut down, the Airlink device is expected to become the VRRP Master and the outgoing traffic should pass through it though some of the ping packets are dropped during the transition but it should be minimal. The VRRP state of the Cisco router is checked to ensure that it is no longer the VRRP Master.

FE1 is then turned back on and the Cisco router preempts (after the preempt timeout has passed) the Airlink device and returns to being the VRRP Master. Ping packets are also expected to be dropped during this transition. The VRRP state of the Cisco router is checked that it has returned as Master.

The statistics of the 50 ping packets is collected. Some dropped packets are expected during the transition but it is unacceptable to have too many. The threshold is set to 10 packets being dropped and any more than that will result in a test failure.

## Simulating ISP failure

This is done by shutting down the outside interface (FE0) of the Cisco router. The track object on the Cisco router detects that the outside route is down and lowers its priority as VRRP Master. This causes the Airlink device to have a higher priority and becomes the new VRRP Master. The test procedure and pass conditions are identical to the previous part where FE1 is shut down.

## Test bed clean up

The test bed is returned to its original state so that other test cases may be run. The default gateway on the host computer is changed back to 192.168.13.31 and VRRP is turned off on the Airlink device. The Cisco router is left as is because no configuration changes were made in the script.

# Running the script

To run the script, \testsuite\Feature\LAN\testsuite\_lan\_launcher.py is the Python file that will start the process. The test case number if currently 26 as shown in Figure 3, so the argument is –n [26-26]. The –t and –v argument used by the automation framework can also be used, but be sure to update the entry in \testsuite\Feature\LAN\lan\_test\_conf.yml.

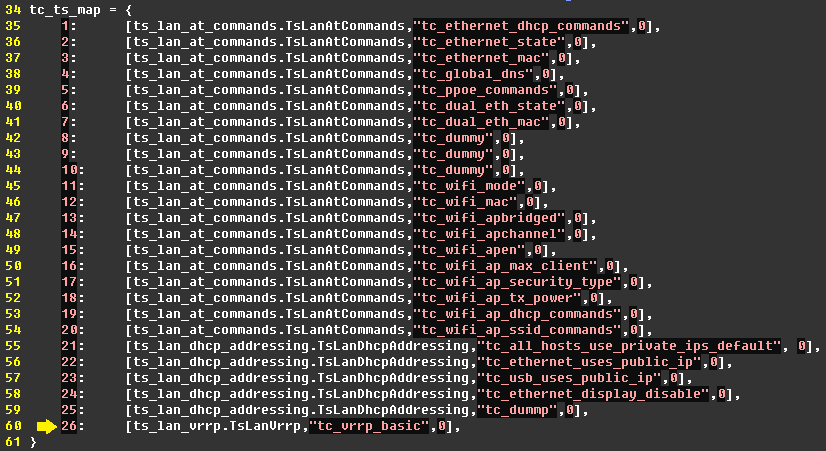


Figure : tc\_ts\_map in testsuite\_lan\_launcher.py