Aerohive Networks Inc.

WAN Interface Addressing Test Plan

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 0.1 | 06/06/2011 | Kevin Lin (klin) | Initial version |
| 0.2 | 06/16/2011 | Kevin Lin (klin) | Add Lei’s comments on:  Change descriptions copied over from unit test plan.  Change description on multi VLAN scenario for clients behind BR eth1.  Add test cases when NAT is disabled.  Add checks to prevent eth1 to be configured as backhaul mode  Add mesh links to wifi0/wifi1 interface of BR. Also check backhaul mode versus access mode. |
| 0.3 | 06/21/2011 | Kevin Lin (klin) | Internal meeting review. Attendees: Lei Wang, Rick Strong, Linda Knudstrup, Jing Han  Should add some test cases specific to BR100 (not applicable/testable with AP330). Such as traffic between eth1, eth2, eth3 and different interface names between BR100/BR200 and AP330.  Add test case to make sure management packets do not pass beyond WAN port, and what to do if/when BR receives such packet on the WAN interface.  Add application specific traffic to NAT testing. Specifically FTP, HTTP, SSH.  Be clear (how, when, what) with how routes get updated in the route table.  Check for DHCP client failure test cases. Also check NTP.  When AP comes up initially, how does it connect to HM? (HM test case to be added). |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

Contents

[Table of Contents 3](#_Toc296513600)

[1. Introduction 5](#_Toc296513601)

[2. Test Objectives 5](#_Toc296513602)

[3. Test Acceptance Criterion from Development 7](#_Toc296513603)

[4. Product Pass Criterion 7](#_Toc296513604)

[5. Test Bed Topology 8](#_Toc296513605)

[6. TestCase 9](#_Toc296513606)

[6.1. Key Scenarios 28](#_Toc296513607)

[6.2. Function Test Case <maybe has many sub-sections, up to you> 28](#_Toc296513608)

[6.3. Stress Test Case 28](#_Toc296513609)

[6.4. Duration Test Case 28](#_Toc296513610)

[6.5. Performance Test Case 28](#_Toc296513611)

[6.6. Scalability Test Case 28](#_Toc296513612)

[6.7. Compatibility Test Case 28](#_Toc296513613)

[6.8. CLI Management (Automation Status: Yes/No) 29](#_Toc296513614)

[6.9. GUI Management-HiveManager 29](#_Toc296513615)

[6.10. GUI Management-HiveUI 29](#_Toc296513616)

Glossary and Abbreviations

AP – Access Point

BR – Branch Router

# Introduction

This test plan addresses the test coverage for the WAN interface address feature during HiveOS Congo release. The goal of the testing is to cover IP forwarding and network interface configurations for the branch router products, defined by the Congo functional specification and MRD.

HiveOS devices are currently deployed in private LAN environments as Layer-2 network devices functioning as bridges for wireless and wired LANs. For the upcoming branch router product lines, HiveOS devices will be deployed in branch offices as gateways to the internet functioning as Layer-3 network devices, forwarding IP traffic between the private network (LAN) and the internet (WAN). The HiveOS network stack needs to be enhanced to support the IP forwarding requirement and configurations of network interfaces to function properly in a network topology connected to both a WAN and LAN environment.

Congo is the initial release which constitutes a telecommuter-optimized low-end AP, Cloud VPN Gateway, 3G Support, Routing, and a couple new HiveOS and HiveManager features. This document addresses the DCD and interface configuration for the WAN data-plane, DHCP client and static IP, NAT and AMRP support for WAN interface

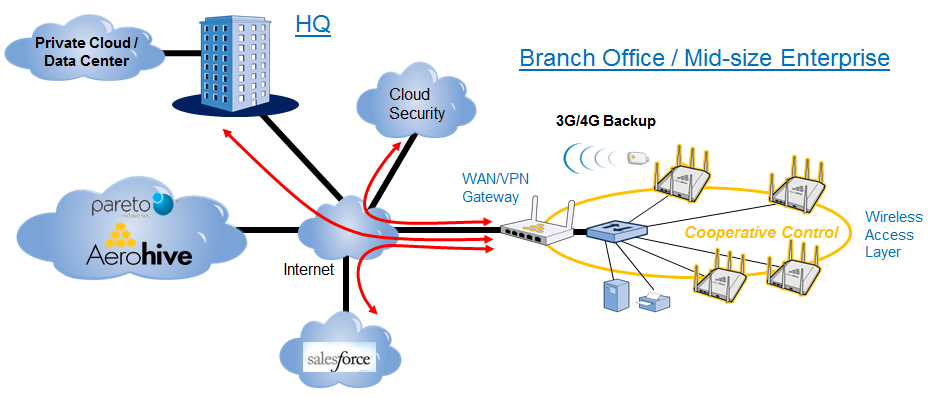


Figure – Overview of HiveOS L3 support

Feature MRD can be found on Jive <link?>

Feature FS can be found on Knowledge Tree: <http://kt.aerohive.com/view.php?fDocumentId=3245>

# Test Objectives

This feature is planned to be delivered during the milestone-A drop of Congo release.

Note that AP330 will be used initially for Congo software testing before the actual BR100 hardware is available.

BR100 will have some unique differences not testable by AP330. These will be addressed at a later time.

Currently, there is no scalability, performance or longevity requirement for this feature. The goal is to validate feature functionality on both HiveOS and HiveManager (configuration). Testing shall also provide customer and negative scenarios.

|  |  |  |
| --- | --- | --- |
| Feature name | Description | CLI/Info |
| WAN: data-plane support | When an interface is configured as a WAN interface, it must not be part of the Aerohive mesh bridge group, and must be unbind from the hive associated with mgt0.  Note that with multiple Ethernet interfaces or a PPP interface configured as WAN mode with IP address assigned through DHCP, a priority needs to be set for the WAN interfaces for default IP route installation and WAN failovers. | A single Ethernet WAN interface is supported in the Congo release. |
| WAN: DCD and CLI support | CLI commands must support a new “WAN” interface mode. | From CLI-Final: |
| WAN: DHCP client and static IP support | Both static and dynamic IP address assignment must be supported on a WAN Ethernet interface via CLI and DHCP respectively. DHCP client on HiveOS shall install a default IP route using the default IP gateway address obtained from its DHCP server. | From CLI-Final: |
| WAN: NAT support |  | Use iptable from Linux shell |
| WAN: AMRP support | AMRP module shall be notified for the interface mode change similar to current Ethernet interface mode change processing. | “show amrp interface” should not display the WAN interface |

Multi-WAN failover between Ethernet and 3G modem will be covered separately in another test plan.

# Test Acceptance Criterion from Development

* Approved – Functional Specifications
  + <http://kt.aerohive.com/view.php?fDocumentId=3245>
* Approved – Unit Test Plans
  + <http://kt.aerohive.com/view.php?fDocumentId=3283>

# Product Pass Criterion

Feature testing is considered pass when test result meets or exceeds the requirement defined in the expected result field. The expected result field is defined by the requirements stated in the functional specification and/or MRD; whichever is stricter; plus additional quality and usability expectations set by the test engineer.

# Test Bed Topology

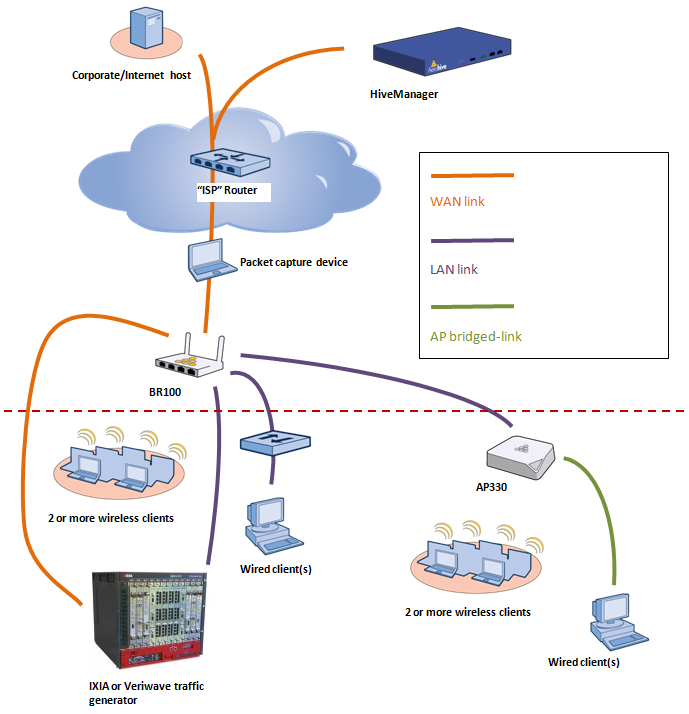


Figure – Test bed topology

# TestCase

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_1 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0101:  In initial deployment, customer power on AP, AP's mgt0 gets IP address by DHCP, then connect to NMS to download initial configuration, which sets eth0 interface to wan mode, (AP will enable eth0's DHCP client automatically), disable mgt0's DHCP client. After reboot, AP will act as a L3 device, eth0 is wan interface with DHCP client enabled, mgt0 acts as default gateway for downlink stations.  Wired clients connected to eth1 should be able to access Internet via AP. | | |
| Pre-condition | No precondition. Start from factory reset. | | |
| Test procedure | 1. Upgrade AP's image, **reset config**, then reboot.(AP330/350 support as BR)  2. After reboot, set eth0 to wan mode; disable mgt0's DHCP client, config mgt0 IP, enable mgt0 DHCP server, set default gateway to mgt0 IP; set eth1 mode to access, then save to reboot.  CLI commands:  no interface mgt0 dhcp client  interface eth0 mode wan  interface mgt0 vlan xx  user-profile vlan\_xx  user-profile vlan\_xx vlan-id xx attribute xx  interface mgt0 ip <MGT0\_IP/MASK>  interface mgt0 dhcp-server enable  interface mgt0 dhcp-server ip-pool <START\_IP> <END\_IP>  interface mgt0 dhcp-server options default-gateway <MGT0\_IP>  interface mgt0 dhcp-server options dns1 <MGT0\_IP>  interface mgt0 dns-server enable  interface eth1 mode bridge-access user-profile-attribute xx  interface eth1 mac-learning enable  show running-config  save config  Connect PC to eth1, check PC should be able to access Internet  3. Reboot and show running-config to check, configuration remains to be same as set, no redundance CLI be generated | | |
| Expect result | 1. Login into AP, check eth0 with backhaul mode, it can ping [www.baidu.com](http://www.baidu.com)  CLI: show interface  2. Login AP, check eth0 with wan mode, and install default route on eth0; also it obtained IP and with default-gateway,dns.  CLI commands:  show interface eth0  show interface eth0 dhcp client  Check PC can obtain ip from mgt0 dhcp-server.  CLI commands:  show interface m0 dhcp-server detail  PC ping www.baidu.com , check the ping packet will be first deliverd to gateway mgt0, and then do L3 routing by kernel and forward it out via eth0.  3. Show running-config should not generate CLI if an option is configured as its default value | | |
| Comments | 1. show interface  AH-9582c0#show interface  State=Operational state; Chan=Channel;  Radio=Radio profile; U=up; D=down;  Name MAC addr Mode State Chan VLAN Radio Hive SSID  ----------- -------------- -------- ----- ---- ---- ---------- ---------- ---------  Mgt0 0019:7795:82c0 - U - 1 - hive0 -  Agg0 0019:7795:82c3 backhaul D - 1 - hive0 -  **Eth0** 0019:7795:82c0 **backhaul** U - 1 - hive0 -  Eth1 0019:7795:82c1 backhaul D - 1 - hive0 -  Red0 0019:7795:82c2 backhaul D - 1 - hive0 -  Wifi0 0019:7795:82d0 access U 1 - radio\_ng0 - -  Wifi0.1 0019:7795:82d0 access D 1 - radio\_ng0 hive0 -  Wifi1 0019:7795:82e0 dual U 161 - radio\_na0 - -  Wifi1.1 0019:7795:82e0 backhaul U 161 1 radio\_na0 hive0 -  2. AH-9582c0#show interface eth0  Parent interface=none;  Mode=**wan**; Mac learning= disabled; Admin state=enabled;  WEB server=disable; NAT support=enable; DHCP client=enable; DHCP server=disable; DNS server=disable;  IP addr=**10.2.2.31**; Netmask=255.255.255.0;  Internal station traffic state=enabled;  Operational state=up; Duplex=full-duplex; Speed=100Mbps;  LLDP state=enabled; CDP state=enabled;  Native-vlan=1;  MAC addr=0019:7795:82c0; MTU=1500;  Rx packets=81301; errors=0; dropped=0;  Tx packets=23372; errors=0; dropped=0;  Rx bytes=11641555 (11.102 MB); Tx bytes=2030035 (1.936 MB);  show interface eth0 dhcp client  AH-9582c0#show interface eth0 dhcp client  HM=HiveManager; SLS=system log server;  DHCP client: Enabled  Timeout for applying static IP or default IP: 20 secs  State:Network configuration received; **Get IP address** **10.2.2.31** from the server 10.2.2.252; Relay agent 0.0.0.0  Lease time: 86400 seconds; Duration time: 748 seconds  Get options from server:  Netmask (option number 1): 255.255.255.0  **Router** (option number 3): **10.2.2.1**  **DNS server** (option number 6): **10.155.3.250**  Log server (option number 7):  DNS domain (option number 15):aerohive-qanet.cn  NTP server (option number 42):  AH-9582c0#show ip route  Ref=references; Iface=interface;  U=route is up;H=target is a host; G=use gateway;  Destination Gateway Netmask Flags Metric Ref Use Iface  --------------- --------------- --------------- ----- ------ ------ --- -----  10.2.2.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0  11.11.11.0 0.0.0.0 255.255.255.0 U 0 0 0 mgt0  127.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 lo  0.0.0.0 **10.2.2.1** 0.0.0.0 **UG** 0 0 0 **eth0**  show interface m0 dhcp-server detail, with client PC’s mac-address in dhcp-client list  AH-9582c0#show interface m0 dhcp-server detail  LPR: Lease Period Remaining  No. MAC address IP address LPR  ---- ---------------- ---------------- ------------------  1 **0023:14d9:cfcc** **11.11.11.20**  23 hr 59 min 48 sec  \_ff dst-ip <client\_pc\_ip> protocol 1 bidirectional  \_kdebug fe basic  \_kdebug fe detail  **Wired Ping logs when AP with wan mode:**  AH-9582c0#2012-09-17 07:51:58 debug kernel: L\*: **(i) eth1** **11.11.11.21->220.181.111.148**(2377) ttl(128) **icmp-echo-req**(1/7) 60 bytes  2012-09-17 07:51:58 debug kernel: [fe]: MAC session (id [5]) found  2012-09-17 07:51:58 debug kernel: [fe]: fflow b8ac:6fcb:f9c3->0019:7795:82c0 flag 0x41002, rflow 0019:7795:82c0->b8ac:6fcb:f9c3 flag 0xc1210  2012-09-17 07:51:58 debug kernel: [fe]: fflow acl 0x0/0x0, rflow acl 0x0/0x0  2012-09-17 07:51:58 debug kernel: [fe]: QoS: ingress pkt fwd(eth1) profile=11 qos=2  2012-09-17 07:51:58 debug kernel: [fe]: set pkt to self  2012-09-17 07:51:58 debug kernel: [fe]: swap incoming dev eth1 -> mgt0  2012-09-17 07:51:58 debug kernel: [fe]: allow to self-pak, dst-ip 220.181.111.148,allow-arp 0 allow-dhcp 0 allow-forwarding 1  2012-09-17 07:51:58 debug kernel: [fe]: **ip forward: ip pkt with dst-ip 220.181.111.148 received on eth1**  2012-09-17 07:51:58 debug kernel: [fe]: **deliver pak to self on mgt0** with fw mark 1  2012-09-17 07:51:58 debug kernel: L\*: **(u) mgt0 11.11.11.21->220.181.111.148**(2377) ttl(128) **icmp-echo-req**(1/7) 60 bytes  2012-09-17 07:51:58 debug kernel: [kernel]: iptables PRE\_ROUTING pkt 11.11.11.21(mgt0)->220.181.111.148(<NULL>) proto(1) 60 bytes: ACCEPT  2012-09-17 07:51:58 debug kernel: [kernel]: iptables FORWARD pkt 11.11.11.21(mgt0)->220.181.111.148(eth0) proto(1) 60 bytes: ACCEPT  2012-09-17 07:51:58 debug kernel: [kernel]: iptables POST\_ROUTING pkt 10.2.2.31(<NULL>)->220.181.111.148(eth0) proto(1) 60 bytes: ACCEPT  2012-09-17 07:51:58 debug kernel: L\*: **(o) eth0 10.2.2.31->220.181.111.148**(2377) ttl(127) mtu(1500) **icmp-echo-req**(1/7) 74 bytes  2012-09-17 07:51:58 debug kernel: [fe]: bypass fe egress procesing and deliver packet to stack on eth0  2012-09-17 07:51:58 debug kernel: L\*: **(i) eth0 220.181.111.148->10.2.2.31**(2377) ttl(51) **icmp-echo-reply**(1/7) 60 bytes  2012-09-17 07:51:58 debug kernel: [fe]: bypass fe ingress procesing and deliver packet to stack on eth0  2012-09-17 07:51:58 debug kernel: [kernel]: iptables PRE\_ROUTING pkt 220.181.111.148(eth0)->11.11.11.21(<NULL>) proto(1) 60 bytes: ACCEPT  2012-09-17 07:51:58 debug kernel: [kernel]: iptables FORWARD pkt 220.181.111.148(eth0)->11.11.11.21(mgt0) proto(1) 60 bytes: ACCEPT  2012-09-17 07:51:58 debug kernel: [kernel]: iptables POST\_ROUTING pkt 220.181.111.148(<NULL>)->11.11.11.21(mgt0) proto(1) 60 bytes: ACCEPT  2012-09-17 07:51:58 debug kernel: [fe]: bypass fe egress procesing and deliver packet to stack on mgt0  2012-09-17 07:51:58 debug kernel: [fe]: mark pkt as from self  2012-09-17 07:51:58 debug kernel: [fe]: inject pkt back into flow from mgt0 xmit  2012-09-17 07:51:58 debug kernel: L\*: **(i) mgt0 220.181.111.148->11.11.11.21**(2377) ttl(50) **icmp-echo-reply**(1/7) 60 bytes  2012-09-17 07:51:58 debug kernel: [fe]: MAC session (id [5]) found  2012-09-17 07:51:58 debug kernel: [fe]: fflow 0019:7795:82c0->b8ac:6fcb:f9c3 flag 0xc1210, rflow b8ac:6fcb:f9c3->0019:7795:82c0 flag 0x41002  2012-09-17 07:51:58 debug kernel: [fe]: fflow acl 0x0/0x0, rflow acl 0x0/0x0  2012-09-17 07:51:58 debug kernel: [fe]: QoS: host pkt fwd(mgt0) qos=2 profile=0  2012-09-17 07:51:58 debug kernel: [fe]: eth1 Tx 0019:7795:82c0 -> b8ac:6fcb:f9c3 type 0x0800 74 bytes  2012-09-17 07:51:58 debug kernel: L\*: **(o) eth1 220.181.111.148->11.11.11.21**(2377) ttl(50) **icmp-echo-reply**(1/7) 74 bytes  2012-09-17 07:51:58 debug kernel: [fe]: QoS: pkt forwarded (eth1) qos=2 profile=0  (Wan mode, nat is enabled by default, so icmp packet’s src-ip or dst-ip will be updated.) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_2 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0102:  Wireless clients connected to wifi0/wifi1 should be able to access Internet via AP. | | |
| Pre-condition |  | | |
| Test procedure | 1. AP eth0 be set to wan mode, and is reachable to Internet  2. config Wifi SSID on AP(wan mode).  CLI commands:  ssid xx  interface wifi1 ssid xx  user-profile vlan\_xx  user-profile vlan\_xx vlan-id xx attribute xx  security-object xx  security-object test default-user-profile-attr xx  save config  3. Connect PC to ssid, check it can obtaion ip and ping [www.baidu.com](http://www.baidu.com) to check it can access Internet | | |
| Expect result | 1. Check AP eth0 obtained IP and reachable to Internet  CLI: show interface eth0 dhcp client  2. CLI set successful  3. Connect PC to AP's ssid test, it can obtain ip from mgt0 dhcp-server;  CLI: show interface m0 dhcp-server detail  PC ping [www.baidu.com](http://www.baidu.com) , check the ping packet will be first deliverd to gateway mgt0, and then do L3 routing by kernel and forward it out via eth0. | | |
| Comments | 3. show interface m0 dhcp-server detail, with client PC’s mac-address in dhcp-client list  AH-9582c0#show interface m0 dhcp-server detail  LPR: Lease Period Remaining  No. MAC address IP address LPR  ---- ---------------- ---------------- ------------------  1 **0023:14d9:cfcc** **11.11.11.20** 23 hr 59 min 48 sec  \_ff dst-ip <client\_pc\_ip> protocol 1 bidirectional  \_kdebug fe basic  \_kdebug fe detail  **Wireless Ping logs when AP with wan mode:**  AH-9582c0#2012-09-17 06:58:33 debug kernel: L\*: **(i) wifi0.1** **11.11.11.20->220.181.112.143**(19073) ttl(128) **icmp-echo-req**(1/80) 60 bytes  2012-09-17 06:58:33 debug kernel: [fe]: MAC session (id [3]) found  2012-09-17 06:58:33 debug kernel: [fe]: fflow 0023:14d9:cfcc->0019:7795:82c0 flag 0x41002, rflow 0019:7795:82c0->0023:14d9:cfcc flag 0xe1210  2012-09-17 06:58:33 debug kernel: [fe]: fflow acl 0x0/0x0, rflow acl 0x0/0x0  2012-09-17 06:58:33 debug kernel: [fe]: QoS: ingress pkt fwd(wifi0.1) profile=11 qos=2  2012-09-17 06:58:33 debug kernel: [fe]: set pkt to self  2012-09-17 06:58:33 debug kernel: [fe]: swap incoming dev wifi0.1 -> mgt0  2012-09-17 06:58:33 debug kernel: [fe]: allow to self-pak, dst-ip 220.181.112.143,allow-arp 0 allow-dhcp 0 allow-forwarding 1  2012-09-17 06:58:33 debug kernel: [fe]: **ip forward: ip pkt with dst-ip 220.181.112.143 received on wifi0.1**  2012-09-17 06:58:33 debug kernel: [fe]: **deliver pak to self on mgt0** with fw mark 1  2012-09-17 06:58:33 debug kernel: L\*: **(u) mgt0 11.11.11.20->220.181.112.143**(19073) ttl(128) **icmp-echo-req**(1/80) 60 bytes  2012-09-17 06:58:33 debug kernel: [kernel]: iptables PRE\_ROUTING pkt 11.11.11.20(mgt0)->220.181.112.143(<NULL>) proto(1) 60 bytes: ACCEPT  2012-09-17 06:58:33 debug kernel: [kernel]: iptables FORWARD pkt 11.11.11.20(mgt0)->220.181.112.143(eth0) proto(1) 60 bytes: ACCEPT  2012-09-17 06:58:33 debug kernel: [kernel]: iptables POST\_ROUTING pkt 10.2.2.31(<NULL>)->220.181.112.143(eth0) proto(1) 60 bytes: ACCEPT  2012-09-17 06:58:33 debug kernel: L\*: **(o) eth0 10.2.2.31->220.181.112.143**(19073) ttl(127) mtu(1500) **icmp-echo-req**(1/80) 74 bytes  2012-09-17 06:58:33 debug kernel: [fe]: bypass fe egress procesing and deliver packet to stack on eth0  2012-09-17 06:58:33 debug kernel: L\*: **(i) eth0 220.181.112.143->10.2.2.31**(19073) ttl(51) **icmp-echo-reply**(1/80) 60 bytes  2012-09-17 06:58:33 debug kernel: [fe]: bypass fe ingress procesing and deliver packet to stack on eth0  2012-09-17 06:58:33 debug kernel: [kernel]: iptables PRE\_ROUTING pkt 220.181.112.143(eth0)->11.11.11.20(<NULL>) proto(1) 60 bytes: ACCEPT  2012-09-17 06:58:33 debug kernel: [kernel]: iptables FORWARD pkt 220.181.112.143(eth0)->11.11.11.20(mgt0) proto(1) 60 bytes: ACCEPT  2012-09-17 06:58:33 debug kernel: [kernel]: iptables POST\_ROUTING pkt 220.181.112.143(<NULL>)->11.11.11.20(mgt0) proto(1) 60 bytes: ACCEPT  2012-09-17 06:58:33 debug kernel: [fe]: bypass fe egress procesing and deliver packet to stack on mgt0  2012-09-17 06:58:33 debug kernel: [fe]: mark pkt as from self  2012-09-17 06:58:33 debug kernel: [fe]: inject pkt back into flow from mgt0 xmit  2012-09-17 06:58:33 debug kernel: L\*: **(i) mgt0 220.181.112.143->11.11.11.20**(19073) ttl(50) **icmp-echo-reply**(1/80) 60 bytes  2012-09-17 06:58:33 debug kernel: [fe]: MAC session (id [3]) found  2012-09-17 06:58:33 debug kernel: [fe]: fflow 0019:7795:82c0->0023:14d9:cfcc flag 0xe1210, rflow 0023:14d9:cfcc->0019:7795:82c0 flag 0x41002  2012-09-17 06:58:33 debug kernel: [fe]: fflow acl 0x0/0x0, rflow acl 0x0/0x0  2012-09-17 06:58:33 debug kernel: [fe]: QoS: host pkt fwd(mgt0) qos=2 profile=0  2012-09-17 06:58:33 debug kernel: [fe]: wifi0.1 Tx 0019:7795:82c0 -> 0023:14d9:cfcc type 0x0800 74 bytes  2012-09-17 06:58:33 debug kernel: L\*: **(o) wifi0.1 220.181.112.143->11.11.11.20**(19073) ttl(50) **icmp-echo-reply**(1/80) 74 bytes  2012-09-17 06:58:33 debug kernel: [fe]: QoS: pkt queued (wifi0.1) qos=2 profile=0  2012-09-17 06:58:33 debug kernel: [fe]: Tx:wifi0.1:0> 0019:7795:82c0->0023:14d9:cfcc profile idx=0 pkt\_len=74 q\_len=0 QoS buf=0  (Wan mode, nat is enabled by default, so icmp packet’s src-ip or dst-ip will be updated.) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_3 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0103:  Configuration save, upload config and download config should work for WAN interface | | |
| Pre-condition |  | | |
| Test procedure | save config to remote scp server, reset config, reboot, then upload the saved config back to AP, then reboot.  1. AP eth0 set to wan mode and save config to remote scp server  CLI commands:  no interface mgt0 dhcp client  interface eth0 mode wan  interface mgt0 vlan xx  interface mgt0 ip <MGT0\_IP/MASK>  interface mgt0 dhcp-server enable  interface mgt0 dhcp-server ip-pool <START\_IP> <END\_IP>  interface mgt0 dhcp-server options default-gateway <MGT0\_IP>  interface mgt0 dhcp-server options dns1 218.108.248.200  interface eth1 mode bridge-access user-profile-attribute xx  interface eth1 mac-learning enable  ssid xx  interface wifi1 ssid xx  user-profile vlan\_xx  user-profile vlan\_xx vlan-id xx attribute xx  security-object xx  security-object test default-user-profile-attr xx  show running-config  save config  save config current scp://user@server\_ip:/filename  2. Reset config on AP  CLI commands:  reset config  3. Upload the saved config back to AP and reboot to check  CLI commands:  save config scp://user@server\_ip:/filename current  reboot | | |
| Expect result | 1. Config should be copied to flash after save, and the same config should be copied over to remote server  2. Reset config successful, check it with default configuration, eht0 is backhaul mode  3. When retrieving config from server, it should be exactly the same as the one saved before.  Verify WAN feature continue to work by connecting clients (wired or wireless) and access internet. | | |
| Comments | 3. wired or wireless clients can ping [www.baidu.com](http://www.baidu.com). | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_4 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0104:  Manual insertion of static routes should work, and BR should route packets correctly, provided the static route is accurate and reachable | | |
| Pre-condition |  | | |
| Test procedure | CLI commands:  One more AP (AP2) is needed for this case (we just use AP to simulate a L2 VLAN switch for convenience). Orginal testing AP is denoted by AP1, the new extra AP is AP2.  After WANAddressing\_UnitTest\_ConfigCLI\_3, change AP1 eth1 mode to bridge-802.1q, connect AP2 eth0 to AP1 eth1, connect PC to AP2 eth1, set AP2 eth1 in bridge-access mode and associate VLAN 2.  CLI commands:  (on AP1)  interface eth1 mode bridge-802.1q  interface eth1 allowed-vlan all  interface mgt0.2 vlan 2  interface mgt0.2 ip <MGT2\_IP/MASK>  interface mgt0.2 dhcp-server enable  interface mgt0.2 dhcp-server ip-pool <MGT2\_START\_IP> <MGT2\_END\_IP>  interface mgt0.2 dhcp-server options default-gateway <MGT2\_IP>  interface mgt0.2 dhcp-server options dns1 218.108.248.200  save config | | |
| Expect result | 1. Verify the new static route would work by passing traffic through the new route. Make sure the packet is forwarded to the correct next hop by using sniffer. 2. Ensure the default route is not affected by the new static route, and packets not routed to this new static route should not be effected. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_5 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0105:  Creating new user-profile and putting clients on new VLAN should work | | |
| Pre-condition |  | | |
| Test procedure | create a new user-profile with VLAN 2, create a security-object test-vlan with attribute 2, create a new ssid test-vlan, set ssid test-vlan to reference the security-object  CLI commands:  user-profile test-vlan vlan-id 2 attribute 2  security-object test-vlan  security-object test-vlan default-user-profile-attr 2  ssid test\_vlan  ssid test\_vlan security-object test-vlan  interface wifi0 ssid test-vlan  save config | | |
| Expect result | (1) Connect PC to AP's ssid test-vlan, it should be able to access Internet  (2) Check PC get correct IP in the ip pool | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_6 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0106:  Verify eth1 works with bridge-access mode | | |
| Pre-condition |  | | |
| Test procedure | change eth1 mode to bridge-access, set it to reference security-object test-vlan  CLI commands:  interface eth1 mode bridge-access  interface eth1 security-object test-vlan  save config | | |
| Expect result | (1) Connect PC to eth1, it should be able to access Internet.  (2) Check PC get correct IP in the ip pool | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_7 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0201:  Repeatedly change ethx interface to/from wan mode, check its behaviors, like DHCP client, routing, NAT and connectivity. | | |
| Pre-condition |  | | |
| Test procedure | Change eth0 interface mode between wan and backhaul 20 times. Eventually end configuration after switching to WAN mode. | | |
| Expect result | (1) Login into AP, it can ping www.baidu.com  (2) Connect PC to eth1, it should be able to access Internet.  (3) Connect PC to AP's ssid <SSID\_NAME>, it should be able to access Internet  (4) show running-config indicates eth0 is wan mode  (5) show interface eth0 indicates it's in wan mode with correct IP  (6) Ensure there are no memory leaks | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_8 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0202:  Change ethx interface between WAN and backhaul mode, check behavior and functionality on each mode. | | |
| Pre-condition |  | | |
| Test procedure | Change ethx mode from/to wan mode, check DHCP client status, NAT status  CLI commands:  show interface eth0 dhcp client | | |
| Expect result | (1) When change to wan mode, DHCP client should be default to enable; NAT should be enabled by default  (2) When change from wan mode, DHCP client should be disabled. NAT should be disabled | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_9 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0301:  When DHCP client is enabled on ethx interface, it should send DHCP discovery packet, and go through DHCP state transition. If DHCP is configured in the network, AP should eventually receive an IP,and configure the ethx interface correctly.  Only mgt0 and ethx in wan mode can be configued as DHCP client. | | |
| Pre-condition |  | | |
| Test procedure | Change ethx mode to wan, enable DHCP client on it.  Change ethx mode to backhaul, try to enable DHCP client on it.  CLI commands:  interface eth0 dhcp client  show interface eth0 dhcp client  show running-config  save config | | |
| Expect result | (1) For wan mode ethx, succed to enable DHCP client, and it will receive and configure correct IP and route.  Login into AP, can ping www.baidu.com  (2) For non-wan mode ethx, it will fail with error message when trying to enable DHCP client.  (3) show running-config should only display the options with non-default values. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_10 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0302:  When DHCP client is disabled on ethx interface, it should not send discovery packet via the interface.  Only mgt0 and ethx in wan mode can be configued as DHCP client. | | |
| Pre-condition |  | | |
| Test procedure | Disable DHCP client on ethx in wan mode  CLI commands:  no interface eth0 dhcp client  show interface eth0 dhcp client  show running-config  save config | | |
| Expect result | (1) Previous obtained IP and route information should be cleared. Login into AP, it can't ping www.baidu.com  (2) Check system log by "show logging buffered", there should be no log like "DHCPC: send discovery on eth0"  (3) show running-config should only display the options with non-default values. | | |

.

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_11 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0401:  An ethx interface in wan mode can be assigned a static IP when DHCP client is disabled. | | |
| Pre-condition |  | | |
| Test procedure | Try to configure static IP on ethx in backhaul/bridge-access mode without security object.  CLI commands:  show interface eth0  interface eth0 ip <ETH0\_IP/MASK> | | |
| Expect result | (1) show commands should indicate eth0 is in backhaul/bridge-access mode, if not, first change it.  (2) CLI will fail with error message | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_12 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0402:  When DHCP client is enabled, ethx can't be assigned a static IP.  Bachhaul mode ethx can't be assigned an IP. | | |
| Pre-condition |  | | |
| Test procedure | change eth0 mode to wan and enable dhcp client, then try to configure static IP on eth0  CLI commands:  interface eth0 mode wan  interface eth0 dhcp client  show interface eth0  interface eth0 ip <ETH0\_IP/MASK> | | |
| Expect result | CLI will fail with error message | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_13 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0403:  When changing WAN interface IP from DHCP to static IP, the old DHCP address obtained should be flushed and replaced by the static IP address | | |
| Pre-condition |  | | |
| Test procedure | disable eth0 dhcp client, try to configure static IP on eth0. Check eth0 interface info, route info, and running-config  CLI commands:  show interface eth0  show ip route  show running-config  save config | | |
| Expect result | (1) Previous IP abtained by DHCP should be cleared;  (2) New static IP is configured, network route is installed. Can ping host in same subnet as eth0.  (3) show running-config shoud include static IP CLI. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_14 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0404:  When changing WAN interface from static IP address to DHCP, the old static IP address should be flushed and replaced by the new DHCP address obtained from DHCP server | | |
| Pre-condition |  | | |
| Test procedure | enable DHCP client on ethx, then check running-config and interface IP.  CLI commands:  show running-config | | |
| Expect result | (1) The static IP should be cleared  (2) show running-config does not include a line with "interface eth0 ip <ETH0\_IP/MASK>"  (3) show interface eth0 should indicate that eth0 gets IP from DHCP server | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_15 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0501:  In previous releases, "ip route" commands are used to configure static routes. The routes are always installed on mgt0 interface.  When installing/removing a route, AP will search ethx interfaces for one with same subnet as the route gateway. If one interface is found, then install route on it. Otherwise, CLI will fail with error message. | | |
| Pre-condition |  | | |
| Test procedure | Set eth0 to wan mode, disable dhcp client, configure static IP, install a default route with gateway not in eth0's subnet.  CLI commands:  interface eth0 mode wan  no interface eth0 dhcp client  interface eth0 ip <ETH0\_IP/MASK>  ip route default gateway <GATEWAY\_IP>  show ip route  show running-config  save config | | |
| Expect result | (1) CLI will fail with error message  (2) show ip route should not include the default gateway | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_16 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0502:  In previous releases, "ip route" commands are used to configure static routes. The routes are always installed on mgt0 interface.  When installing/removing a route, AP will first search mgt0 interfaces for one with same subnet as the route gateway. If one interface is found, then install route on it. | | |
| Pre-condition |  | | |
| Test procedure | install a default route with gateway in eth0's subnet  CLI commands:  ip route default gateway <GATEWAY\_IP>  show ip route  show running-config  save config | | |
| Expect result | (1) show ip route should indicate that route is installed on eth0  (2) show running-config should include default gateway CLI  (3) AP can ping IP of www.baidu.com via eth0 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_17 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0503:  Verify that when changing eth0 to a different subnet, the original default route should automatically be removed and BR should thus no longer be able to route packets from clients. | | |
| Pre-condition |  | | |
| Test procedure | change eth0 IP to another subnet  CLI commands:  interface eth0 ip <ETH0\_IP/MASK>  show ip route  show running-config  show logging buffered  save config | | |
| Expect result | (1) show ip route should indicate that default route on eth0 has been removed.  (2) show running-config should not include default gateway CLI.  (3) CLI should print message that the default route on eth0 has been removed  (4) show logging buffered should include a log that default route on eth0 has been removed  (5) AP can't ping IP of www.baidu.com | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_18 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0504:  Verify that after changing eth0 to a different subnet, updating the default route with a new and valid IP address will allow BR and clients to reach hosts outside WAN interface. | | |
| Pre-condition |  | | |
| Test procedure | install a default route with gateway in eth0's new subnet  CLI commands:  ip route default gateway <GATEWAY\_IP>  show ip route  show running-config  save config | | |
| Expect result | (1) show ip route should indicate that route is installed on eth0  (2) show running-config should include default gateway CLI  (3) AP can ping IP of www.baidu.com via eth0  (5) show interface eth0 indicates it's in wan mode with correct IP | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_19 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0505:  Verify default route should be removed when eth0 is changed from WAN to backhaul | | |
| Pre-condition |  | | |
| Test procedure | change eth0 mode to backhaul  CLI commands:  interface eth0 mode backhaul  show ip route  show running-config  save config | | |
| Expect result | (1) show ip route should indicate that default route on eth0 has been removed.  (2) show running-config should not include default gateway CLI.  (3) CLI should print message that the default route on eth0 has been removed  (4) show logging buffered should include a log that default route on eth0 has been removed  (5) AP can't ping IP of www.baidu.com | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_20 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0506:  Verify that default route could be removed, if user chooses to, and that BR and clients will no longer to reach hosts beyond WAN interface. | | |
| Pre-condition |  | | |
| Test procedure | remove the default route on eth0.  CLI commands:  no ip route default gateway <GATEWAY\_IP>  show ip route  show running-config  save config | | |
| Expect result | (1) show ip route should indicate that default route on eth0 has been removed.  (2) show running-config should not include default gateway CLI.  (3) AP can't ping IP of www.baidu.com | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_21 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0601:  Verify that NAT could be disabled, and clients with non-routable IP address behind BR would not reach the internet without NAT. | | |
| Pre-condition |  | | |
| Test procedure | disable NAT on eth0  CLI commands:  no interface eth0 wan nat enable  show running-config  save config | | |
| Expect result | (1) Login into AP, it can ping www.baidu.com  (2) Connect PC to eth1, it should NOT be able to access Internet.  (3) Connect PC to AP's ssid <SSID\_NAME>, it should NOT be able to access Internet  (4) show running-config include a CLI to disable NAT on eth0  (5) show interface eth0 indicates NAT is disabled  (6) Login into busybox by \_shell, command "iptables -t nat -nvL" should NOT include a NAT rule for eth0 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_22 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0602:  Verify that when NAT is enabled, clients with non-routable IP address could reach the internet with NAT functionality. | | |
| Pre-condition |  | | |
| Test procedure | enable NAT on eth0  CLI commands:  interface eth0 wan nat enable  show running-config  save config | | |
| Expect result | (1) Login into AP, it can ping www.baidu.com  (2) Connect PC to eth1, it should be able to access Internet.  (3) Connect PC to AP's ssid <SSID\_NAME>, it should be able to access Internet  (4) show running-config should NOT include a CLI to enable NAT on eth0  (5) show interface eth0 indicates NAT is enabled  (6) Login into busybox by \_shell, command "iptables -t nat -nvL" should include a NAT rule for eth0 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_UnitTest\_ConfigCLI\_23 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Unit Test 0603:  Verify that NAT works with static IP and/or static route assignment | | |
| Pre-condition |  | | |
| Test procedure | connect AP1' eth0 to another AP2's eth1, config static ip on AP1 eth0, set default gateway to AP2 mgt0 ip.  Config static ip route on AP2 such that traffic to PC's subnet can route to AP1's eth0  CLI commands:  (on AP1)  no interface eth0 dhcp client  interface eth0 ip <AP1\_ETH0\_IP/MASK>  dns server-ip 218.108.200.248  ip route default gateway <AP2\_MGT0\_IP>  show running-config  save config  (on AP2)  interface mgt0 ip <AP2\_MGT0\_IP/MASK>  interface eth0 mode wan  interface eth1 mode bridge-access  interface eth1 mac-learning enable  ip route net <AP1\_MGT0\_SUBNET> <AP1\_MGT0\_MASK> gateway <AP1\_ETH0\_IP>  show running-config  save config | | |
| Expect result | (1) Login into AP1, it can ping www.baidu.com  (2) Connect PC to AP1 eth1, it should be able to access Internet.  (3) Connect PC to AP's ssid <SSID\_NAME>, it should be able to access Internet | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_ConfigCLI\_24 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that eth1 should not be allowed to be configured as backhaul mode when eth0 is in WAN mode | | |
| Pre-condition | Configure WAN mode on eth0 and WAN functions properly | | |
| Test procedure | HiveManager should prevent such configuration.  From CLI, configure eth1 to backhaul mode  Connect clients to both eth1 and wifi0/wifi1 and pass traffic between each other and towards internet  Observe NAT and routing table | | |
| Expect result | (1) CLI should throw out error/warning message when configuring eth1 to backhaul mode while eth0 is in WAN mode  (2) Both local traffic and ingress/egress traffic through WAN interface should not be impacted | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_ConfigCLI\_25 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that wifi0/wifi1 should not be allowed to be configured as backhaul mode when eth0 is in WAN mode | | |
| Pre-condition | Configure WAN mode on eth0 and WAN functions properly | | |
| Test procedure | HiveManager should prevent such configuration.  From CLI, configure wifi0/wifi1 to backhaul mode  Connect clients to both eth1 and wifi0/wifi1 and pass traffic between each other and towards internet  Observe NAT and routing table | | |
| Expect result | (1) CLI should throw out error/warning message when configuring wifi0/1 to backhaul mode while eth0 is in WAN mode  (2) Both local traffic and ingress/egress traffic through WAN interface should not be impacted | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_ConfigCLI\_26 | | |
| Priority | Accept | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that wifi0/wifi1 should not be allowed to be configured as backhaul mode when eth0 is in WAN mode, and connect to mesh link | | |
| Pre-condition | Configure WAN mode on eth0 and WAN functions properly | | |
| Test procedure | HiveManager should prevent such configuration.  From CLI, configure wifi0/wifi1 to backhaul mode, and connect to mesh network wifi0/wifi1  Connect clients to both eth1, wifi0/wifi1, behind mesh AP, and pass traffic between each other and towards internet  Observe NAT and routing table | | |
| Expect result | (1) CLI should throw out error/warning message when configuring wifi0/1 to backhaul mode while eth0 is in WAN mode  (2) Both local traffic and ingress/egress traffic through WAN interface should not be impacted | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_NAT\_31 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that clients with routable IP address, attached to wifi0/wifi1 or eth1 of BR can still reach the internet even with NAT disabled | | |
| Pre-condition | Basic NAT features are validated | | |
| Test procedure | Connect clients to both eth1 and wifi0/wifi1 interface of BR, configure the clients with the routable IP addresses via the WAN interface.  Disable NAT on BR. | | |
| Expect result | (1) For all clients, traffic can be passed and received from either directions  (2) Verify NAT is indeed disabled, and route table is accurate. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_NAT\_32 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that outbound FTP, HTTP and SSH connections can be established with NAT enabled (inbound requires port forwarding feature). | | |
| Pre-condition | Basic NAT features are validated | | |
| Test procedure | Connect clients to both eth1 and wifi0/wifi1 interface of BR. Establish FTP, HTTP and SSH connections to remote host. | | |
| Expect result | (1) For all clients, traffic can be passed and received from either directions  (2) Verify FTP file transfer works, HTTP works, and SSH connections can be established. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_MultipleClients\_41 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that multiple clients on same VLAN behind eth1 interface can pass traffic via WAN port properly | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | Attach a L2 switch to eth1, and 2 (wired) clients behind the switch. Have the clients on same VLAN, and pass traffic in both ingress and egress directions through the WAN interface.  All clients should be sending traffic to:   * Same host * Different host   All clients should be receiving traffic from:   * Same host * Different host | | |
| Expect result | (1) For all clients, traffic can be passed and received from either directions  (2) Verify route and NAT table to be correct | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_MultipleClients\_42 | | |
| Priority | Middle | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that multiple clients on different VLANs behind eth1 interface can pass traffic via WAN port properly | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | Attach L2 switch (with VLAN support) to eth1, and 2 (wired) clients behind the switch. Have the clients on different VLAN, and pass traffic in both ingress and egress directions through the WAN interface.  All clients should be sending traffic to:   * Same host * Different host   All clients should be receiving traffic from:   * Same host * Different host | | |
| Expect result | (1) For all clients, traffic can be passed and received from either directions  (2) Verify route and NAT table to be correct | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_MultipleClients\_43 | | |
| Priority | Middle | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that multiple clients on same VLAN behind wifi0/wifi1 interface can pass traffic via WAN port properly | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | Attach a L2 switch (with VLAN support) to eth1, and 2 (wireless) clients behind the switch. Have the clients on same VLAN, and pass traffic in both ingress and egress directions through the WAN interface.  All clients should be sending traffic to:   * Same host * Different host   All clients should be receiving traffic from:   * Same host * Different host | | |
| Expect result | (1) For all clients, traffic can be passed and received from either directions  (2) Verify route and NAT table to be correct | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_MultipleClients\_44 | | |
| Priority | Middle | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that multiple clients on different VLANs behind wifi0/wifi1 interface can pass traffic via WAN port properly | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | Attach a L2 switch (with VLAN support) to eth1, and 2 (wireless) clients behind the switch. Have the clients on different VLAN, and pass traffic in both ingress and egress directions through the WAN interface.  All clients should be sending traffic to:   * Same host * Different host   All clients should be receiving traffic from:   * Same host * Different host | | |
| Expect result | (1) For all clients, traffic can be passed and received from either directions  (2) Verify route and NAT table to be correct | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_MultipleClients\_45 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that multiple clients on different VLANs behind eth1 and wifi0/wifi1 interface can pass traffic via WAN port properly | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | Attach a L2 switch (with VLAN support) to eth1, and 4 (2 wireless, 2 wired) clients behind the switch. Have the clients on different VLAN, and pass traffic in both ingress and egress directions through the WAN interface.  All clients should be sending traffic to:   * Same host * Different host   All clients should be receiving traffic from:   * Same host * Different host | | |
| Expect result | (1) For all clients, traffic can be passed and received from either directions  (2) Verify route and NAT table to be correct | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_APBehindBR\_51 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that Aerohive AP can be connected to eth1 of BR, and establish CAPWAP to HiveManager on WAN | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | Attach a Aerohive AP330 to eth1, with:   * AP330 mgt0 using DHCP * AP330 mgt0 using static IP   Configure CAPWAP server on AP330 to point it to the HiveManager | | |
| Expect result | (1) AP330 behind eth1 should obtain DHCP address from BR, or local DHCP server is DHCP proxy is used  (2) Verify AP330 is able to establish CAPWAP tunnel to HiveManager with either DHCP or static IP as mgt0 IP  (3) Verify route and NAT information is correct | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_APBehindBR\_52 | | |
| Priority | Middle | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that Aerohive AP can be connected to eth1 of BR, and establish CAPWAP to HiveManager on WAN. Then wired clients can be connected off the eth1 interface of this AP. | | |
| Pre-condition |  | | |
| Test procedure | Attach a wired client to eth1 interface of AP330, and send data in both ingress and egress direction on the WAN interface | | |
| Expect result | (1) Client behind eth1 of BR should obtain DHCP address from BR, or local DHCP server is DHCP proxy is used  (2) Verify client is able to send and receive traffic between itself and the host beyond WAN interface  (3) Verify route and NAT information is correct | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_APBehindBR\_53 | | |
| Priority | Middle | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that Aerohive AP can be connected to eth1 of BR, and establish CAPWAP to HiveManager on WAN. Then wireless clients can be connected off the eth1 interface of this AP. | | |
| Pre-condition |  | | |
| Test procedure | Attach a wireless client to wifi0/wifi1 interface of AP330, and send data in both ingress and egress direction on the WAN interface | | |
| Expect result | (1) Client behind eth1 of BR should obtain DHCP address from BR, or local DHCP server is DHCP proxy is used  (2) Verify client is able to send and receive traffic between itself and the host beyond WAN interface  (3) Verify route and NAT information is correct | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_APBehindBR\_54 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that Aerohive AP can be connected to eth1 of BR, and establish CAPWAP to HiveManager on WAN. Then multiple wired and wireless clients can be connected off the eth1 interface of this AP. | | |
| Pre-condition |  | | |
| Test procedure | Attach 2 wired clients to eth1 interface of AP330 (using a switch), and 2 wireless clients to wifi0/wifi1 interface of AP330, and send data in both ingress and egress direction on the WAN interface | | |
| Expect result | (1) Client behind eth1 of BR should obtain DHCP address from BR, or local DHCP server is DHCP proxy is used  (2) Verify clients are able to send and receive traffic between itself and the host beyond WAN interface  (3) Verify route and NAT information is correct | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_PacketSize\_61 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Verify that WAN feature – routing and NAT works for different packet sizes, both small and large, including jumbo frames that require fragmentation | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | Connect traffic generator (IXIA or Veriwave) on both the LAN and WAN interface of the DUT. Ensure proper routes between the 2 links and pass traffic between the 2 points. Use the following traffic sizes:  64, 65, 88, 256, 512, 513, 1024, 1025, 1518, 1519, 2048, 3000, 5000, 5001  Send out these packets at a relatively low rate to ensure packet drops are not due to other performance reasons. Eg. 10 packets/sec  Repeat test for LAN 🡪 WAN and WAN 🡪 LAN | | |
| Expect result | (1) Traffic can be passed and received from either directions  (2) There should not be any packet loss at such low rate  (3) Fragmentation and reassembly is taken care of properly | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_AMRP\_65 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | AMRP traffic should not be sent out via the WAN interface, but whatever AMRP packets received via WAN interface should be accepted and processed. | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | Set up WAN interface, place it in a hive, and have other APs in the same hive both inside the LAN and also out in the WAN. Do “show amrp interface” on BR to see which interfaces are participating in the AMRP group. | | |
| Expect result | (1) “show amrp interface” should not contain the interface currently configured as WAN port  (2) “sh int” should not have the WAN interface listed under a hive  (3) Verify no AMRP packets are going out of the WAN interface  (4) Any AMRP packets coming into the WAN interface should still be accepted and not dropped | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_Negative\_71 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Set static IP (un-routable) on AP WAN port, wired client, wireless client | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | Set a non-valid IP address to the WAN port, and/or combinations of invalid static IP addresses on wired and wireless clients (including APs and clients attached to eth1 of BR).  Attempt to pass outbound traffic and traffic within the remote location (both L2 and L3 traffic). | | |
| Expect result | (1) Traffic not be passed nor received from either directions  (2) Check for abnormities in route and NAT tables | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_Negative\_72 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Disconnected/non-existent DHCP server | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | Configure BR to use external DHCP server.  Disconnect the DHCP server link, or assign bogus IP address | | |
| Expect result | (1) Clients should not get IP addresses via DHCP discovery  (2) BR should not flood WAN port with DHCP request | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_Negative\_73 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Disconnecting/disabling WAN next hop from BR | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | After WAN mode is operational, disconnect or disable the next hop from the BR and observe behavior | | |
| Expect result | (1) Clients should not be able to communicate with hosts beyond the WAN interface  (2) Local clients should still be able to communicate between each other | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_Negative\_74 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Out of sequence fragmented packets through WAN interface | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | After WAN mode is operational, use traffic generator to send out of sequence packets in both ingress and egress directions of the WAN interface | | |
| Expect result | (1) BR should be able to handle out of sequence frames gracefully and not impact connectivity | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_Negative\_75 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Dropped fragments through WAN interface | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | After WAN mode is operational, use traffic generator to send fragmented packets (randomly dropped) in both ingress and egress directions of the WAN interface | | |
| Expect result | (1) BR should be able to handle dropped frames gracefully and not impact connectivity | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_Negative\_76 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Handling high latency links between WAN port and HiveManager | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | After WAN mode is operational, increase latency (gradually, up to 5 seconds, and back down to 10ms) between BR and HiveManager | | |
| Expect result | (1) HiveManager should be able to see the increase in latency between itself and BR  (2) BR should be able to handle frequent CAPWAP disconnect and re-connect without affecting client traffic | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_Negative\_77 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | WAN interface DHCP client should keep retrying after DHCP request timeout | | |
| Pre-condition | Basic WAN addressing test case | | |
| Test procedure | Configure interface for WAN mode operation and set interface to use DHCP. Disconnect/disable DHCP server on the network to force DHCP time out. Reconnect/enable DHCP server after 10 minutes. | | |
| Expect result | (1) WAN interface should not be able to obtain an IP address in the beginning, but should eventually get one after the DHCP server is back online.  (2) Also validate other functions such as NTP is obtained after WAN interface, provided that DHCP server is also NTP server. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_Multicast\_81 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Handling multicast traffic through WAN interface | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | After WAN mode is operational, use traffic generator to send multicast traffic in both ingress and egress directions of the WAN interface | | |
| Expect result | (1) Clients should be able to join IGMP group  (2) Clients should be able to send and receive multicast streams | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_Multicast\_82 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | Handling fragmented multicast traffic through WAN interface | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure | After WAN mode is operational, use traffic generator to send fragmented multicast traffic in both ingress and egress directions of the WAN interface | | |
| Expect result | (1) Clients should be able to join IGMP group  (2) Clients should be able to send and receive multicast streams | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | WANAddressing\_FeatureTest\_HiveManager\_101 | | |
| Priority | High | Automation Flag | Yes/No/NA |
| Topology to use | Figure 2 | | |
| Description | HM-TBD | | |
| Pre-condition | Basic WAN routing and NAT features are validated | | |
| Test procedure |  | | |
| Expect result | (1) | | |

## Key Scenarios

## Function Test Case <maybe has many sub-sections, up to you>

<Do not forget negative/boundary case>

## Stress Test Case

## Duration Test Case

## Performance Test Case

## Scalability Test Case

## Compatibility Test Case

## CLI Management (Automation Status: Yes/No)

<Just list all cli that this feature has one by one>

< memory leak case for these CLI.Normally, the leak is happened when we do some commands repeatedly.  Like create an object, then delete that object, it should release all the memory it allocated. But this is not true for all the cases. If you create/delete an object several times(but how many times?) and the memory just going down and never recovered, it maybe a memory leak(again, how can we decide it is really a memory leak?).>

## GUI Management-HiveManager

<List HM test case or test log>

## GUI Management-HiveUI

<List HiveUI test case or test log>