Aerohive Networks Inc.

L7 Engine and Application Reporting Test Case

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 0.1 | 2012-09-11 | Cao Dihua | Initial version |
| 0.2 | 2012-10-15 | Lei Xu | The 2nd version, modify test case according to FS enhancement. |
| 0.3 | 2012-10-29 | Lei Xu | Enahnce test case as per internal review. |
| 0.4 | 2013-07-08 | Lei Xu | Add test case on HiveOS application discovery. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

Glossary and Abbreviations

# Introduction

HiveOS Layer 7 app classification and reporting are enabled by an advanced network technology named Deep Packet Inspection (DPI). These features are intended to allow IT administrators to effectively identify the content and origin of each packet of data that passes through Aerohive Access Point (AP) and Branch Router (BR) product lines, and thus help Aerohive customers gain more insight and visibility into their wireless or wired network in addition to the traditionally available network information including IP addresses and port numbers, etc.

Furthermore, knowledge of Layer 7 applications and their associated users information and content types collected on HiveOS based device, work in conjunction with Hive Manager to provide a powerful, graphically rich, highly interactive, and fully customizable application reporting solution allowing Aerohive customers to keep their fingers on the pulse of what is really going on on their networks.

# Test Objectives

Verify HiveOS can classify application, which goes through AP and BR, and report it to HM correctly.

Note: Aerohive required L7 engine, which supports more than 700 L7 applications, from other vendor. It will cost much time and effort to verify all of L7 applications. But if QA does NOT verify L7 application, we can’t guarantee the quality of Aerohive product.

As per QA and HiveOS Dev discussion, we apply following policy to verify L7 engine:

1. Distinguish priority of L7 application to different level: High, Middle and Low.

Make different effort for different priority applications:

|  |  |  |
| --- | --- | --- |
| App Priority | Verify App Classification | Verify App Reporting |
| High | Permit | Permit |
| Middle | Permit | BE |
| Low | BE | BE |

1. L7 engine vendor provided L7 application sample trace, IXIA replay sample trace to verify application as the 1st choice.
2. Verify critical application with live traffic.
3. Require L7 engine vendor’s test plan document to make sure all applications have been covered by vendor.

The HiveOS supporting L7 applications:

<https://wiki.aerohive.com/wiki/download/attachments/5832918/apps.txt?version=1&modificationDate=1351225273223>

# Test Acceptance Criterion from Development

* Approved – MRD

<https://aerohive.jiveon.com/docs/DOC-4070>

* Approved – Functional Specifications

<http://saturn.aerohive.com/view.php?fDocumentId=3972>

* Approved – Unit Test Plans

<http://saturn.aerohive.com/view.php?fDocumentId=4071>

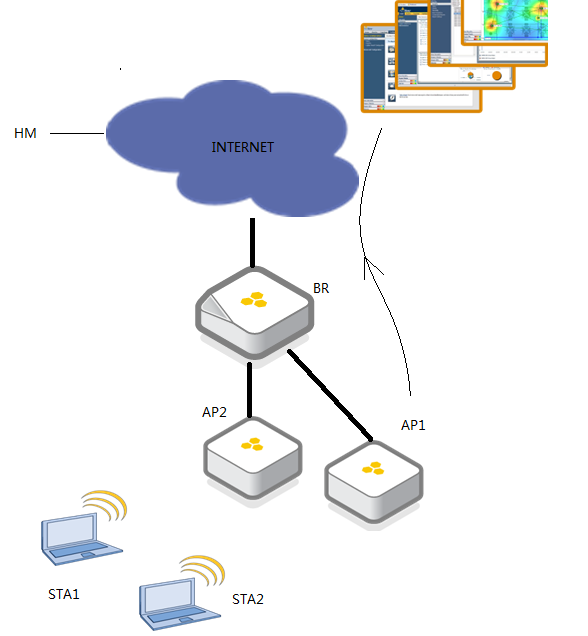
# Product Pass Criterion

Meet all objects in marketing requirement or function spec which may include key function objectives, capacity objectives, and performance objectives and so on.

# Test Bed/Topo Design

# Test Case

## Solution



### ApplicationReporting\_Solution\_01

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Solution\_01 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  | |  | |  Laptop2 Internet | | |
| Description | HiveOS device (AP and BR) can classify application launched at laptops and report it to HM correctly. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 and laptop2 connect with SSID. | | |
| Test procedure | 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR. 2. ~~Keep default settings of application reporting for every individual interface at AP and BR~~ 3. Enable application HTTP and BT for reporting. 4. Launch HTTP web browse at Laptop1. 5. Check if AP can classify HTTP session and report it to HM correctly. 6. Check if BR can classify HTTP session and report it to HM correctly. 7. Launch BT at Laptop2. 8. Check if AP can classify BT session and report it to HM correctly. 9. Check if BR can classify BT session and report it to HM correctly. | | |
| Expect result | Step 5. AP can classify HTTP session and report it to HM correctly.  Step 6. BR can classify HTTP session and report it to HM correctly.  Setp 8. AP can classify BT session and report it to HM correctly.  Step 9. BR can classify BT session and report it to HM correctly. | | |
| Test Result |  | | |
| Comment | Test ICMP as application traffic  This is the application list from PLM, QA should verify HiveOS application identification for all of them. Some are hard to set up. Please pick out those you can cover, and Sunnyvale will cover those China blocked:  File Sharing:  Bittorrent – encrypted, icloud, dropbox,  Email and Chat:  Skype – audio, video, chat, exchange, gmail, yahoo mail/msgr, hotmail  Social and Meetings:  Facebook, Facetime, WebEx,  GoToMeeting, Google+,twitter,  Video / Audio:  Hulu, Youtube, iTunes, QuickTime, HTTPVideo, HTTP Audio, Netflix, Pandora, Google video  Business and productivity:  Salesforce, CITRIX ICA, CITRIX CGP, Microsoft AD, Exchange, sharepoint, McAfee, windows updates, CITRIX streaming.  We should take some of them as application under test during function test. | | |

## Function Test Case

### L7 Engine Application Classification Test

#### L7\_Engine\_Function\_01

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Function\_01 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet or  application server | | |
| Description | Verify HiveOS device (AP and BR) can classify specific application. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR.     2. Keep default settings of application reporting for every individual interface at AP and BR     3. Launch application session at Laptop1.     4. Check if AP can classify application.     5. Check if BR can classify application.   Note: Record applications which are verified. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. The default mode of application reporting at interfaces is auto.  Step 5. AP classifies application.  Step 6. BR classifies application. | | |
| Test Result | AP330 Amazon App ID 33:  Ap330 Daily Build Nov 25  Id:3; Ageout:297575; Flags:0x20200; QOS:2; L7 ID:33; Up: 0 min 5 sec; InPol:L7Verify/2;  10.155.33.6/49613 -> 72.21.215.232/80; Proto 6; Flg:0x112; Pkts:5 Bytes:1739 Parent-MAC-Sess: 25  72.21.215.232/80 -> 10.155.33.6/49613; Proto 6; Flg:0x110; Pkts:46 Bytes:50240  Id:8; Ageout:298018; Flags:0x20400; QOS:2; L7 ID:33; Up: 0 min 10 sec; InPol:L7Verify/2;  10.155.33.6/49612 -> 176.32.98.166/80; Proto 6; Flg:0x102; Pkts:3 Bytes:1779 Parent-MAC-Sess: 25  176.32.98.166/80 -> 10.155.33.6/49612; Proto 6; Flg:0x100; Pkts:3 Bytes:1684  AP330 Apple App ID 41:  Id:5; Ageout:295566; Flags:0x20400; QOS:2; L7 ID:41; Up: 0 min 15 sec; InPol:L7Verify/2;  10.155.33.6/49669 -> 23.15.237.15/80; Proto 6; Flg:0x112; Pkts:9 Bytes:3928 Parent-MAC-Sess: 25  23.15.237.15/80 -> 10.155.33.6/49669; Proto 6; Flg:0x110; Pkts:10 Bytes:11268  AP330 CBS App ID 95:  Id:7; Ageout:295615; Flags:0x20400; QOS:2; L7 ID:95; Up: 0 min 36 sec; InPol:L7Verify/2;  10.155.33.6/49686 -> 64.30.230.36/80; Proto 6; Flg:0x112; Pkts:11 Bytes:4500 Parent-MAC-Sess: 25  64.30.230.36/80 -> 10.155.33.6/49686; Proto 6; Flg:0x110; Pkts:11 Bytes:11992  AP330 CBSInteractive.com App ID 96:  Id:7; Ageout:289375; Flags:0x20400; QOS:2; L7 ID:96; Up: 0 min 44 sec; InPol:L7Verify/2;  10.155.33.6/49716 -> 216.239.120.112/80; Proto 6; Flg:0x112; Pkts:10 Bytes:4736 Parent-MAC-Sess: 25  216.239.120.112/80 -> 10.155.33.6/49716; Proto 6; Flg:0x110; Pkts:17 Bytes:22960  AP330 CNN App ID 112  Id:6; Ageout:293765; Flags:0x20400; QOS:2; L7 ID:112; Up: 0 min 16 sec; InPol:L7Verify/2;  10.155.33.6/49720 -> 157.166.240.13/80; Proto 6; Flg:0x112; Pkts:9 Bytes:3914 Parent-MAC-Sess: 25  157.166.240.13/80 -> 10.155.33.6/49720; Proto 6; Flg:0x110; Pkts:6 Bytes:1542  AP330 eBay App ID 174:  Id:6; Ageout:294551; Flags:0x20400; QOS:2; L7 ID:174; Up: 0 min 23 sec; InPol:L7Verify/2;  10.155.33.6/49728 -> 66.135.200.161/80; Proto 6; Flg:0x112; Pkts:10 Bytes:4468 Parent-MAC-Sess: 25  66.135.200.161/80 -> 10.155.33.6/49728; Proto 6; Flg:0x110; Pkts:12 Bytes:10632  AP330 FTP App ID 217/218/219 bug 21015:  Id:5; Ageout:697; Flags:0x24251; QOS:2; L7 ID:618; Up: 0 min 5 sec; OutPol:L7Verify/6;  10.155.33.189/20 -> 10.155.33.6/50779; Proto 6; Flg:0x112; Pkts:10657 Bytes:15607286 Parent-MAC-Sess: 10  10.155.33.6/50779 -> 10.155.33.189/20; Proto 6; Flg:0x110; Pkts:5346 Bytes:214088  Id:4; Ageout:1797955; Flags:0x624400; QOS:2; L7 ID:618; Up: 0 min 17 sec; InPol:L7Verify/6;  10.155.33.6/50776 -> 10.155.33.189/21; Proto 6; Flg:0x112; Pkts:20 Bytes:921 Parent-MAC-Sess: 10  10.155.33.189/21 -> 10.155.33.6/50776; Proto 6; Flg:0x110; Pkts:14 Bytes:1016  AP330 Gmail APP ID 229:  Id:20; Ageout:1794130; Flags:0x20400; QOS:2; L7 ID:229; Up: 0 min 27 sec; InPol:L7Verify/1;  10.155.33.6/49912 -> 74.125.128.18/443; Proto 6; Flg:0x112; Pkts:271 Bytes:23003 Parent-MAC-Sess: 25  74.125.128.18/443 -> 10.155.33.6/49912; Proto 6; Flg:0x110; Pkts:357 Bytes:504808  Id:22; Ageout:1789953; Flags:0x20400; QOS:2; L7 ID:229; Up: 0 min 10 sec; InPol:L7Verify/1;  10.155.33.6/49915 -> 74.125.128.18/443; Proto 6; Flg:0x112; Pkts:7 Bytes:2255 Parent-MAC-Sess: 25  74.125.128.18/443 -> 10.155.33.6/49915; Proto 6; Flg:0x110; Pkts:5 Bytes:863  Id:8; Ageout:297122; Flags:0x20400; QOS:2; L7 ID:229; Up: 0 min 42 sec; InPol:L7Verify/2;  10.155.33.6/49960 -> 74.125.128.19/80; Proto 6; Flg:0x112; Pkts:11 Bytes:4532 Parent-MAC-Sess: 25  74.125.128.19/80 -> 10.155.33.6/49960; Proto 6; Flg:0x110; Pkts:10 Bytes:5114  Id:22; Ageout:291674; Flags:0x20400; QOS:2; L7 ID:229; Up: 0 min 18 sec; InPol:L7Verify/2;  10.155.33.6/49968 -> 74.125.128.19/443; Proto 6; Flg:0x112; Pkts:10 Bytes:1339 Parent-MAC-Sess: 25  74.125.128.19/443 -> 10.155.33.6/49968; Proto 6; Flg:0x110; Pkts:9 Bytes:8045  AP330 HTTP App ID 275:  Id:2; Ageout:779; Flags:0x24211; QOS:2; L7 ID:275; Up: 0 min 1 sec; InPol:L7Verify/1;  10.155.33.6/51497 -> 10.155.3.249/80; Proto 6; Flg:0x112; Pkts:15 Bytes:1788 Parent-MAC-Sess: 25  10.155.3.249/80 -> 10.155.33.6/51497; Proto 6; Flg:0x110; Pkts:22 Bytes:28299  AP330 MSN App ID 393:  Id:7; Ageout:297633; Flags:0x24400; QOS:2; L7 ID:393; Up: 0 min 36 sec; InPol:L7Verify/2;  10.155.33.6/51735 -> 131.253.13.140/80; Proto 6; Flg:0x112; Pkts:11 Bytes:4500 Parent-MAC-Sess: 27  131.253.13.140/80 -> 10.155.33.6/51735; Proto 6; Flg:0x110; Pkts:24 Bytes:31922  Id:87; Ageout:298387; Flags:0x24400; QOS:2; L7 ID:393; Up: 0 min 6 sec; InPol:L7Verify/2;  10.155.33.6/51795 -> 65.54.82.179/80; Proto 6; Flg:0x11a; Pkts:10 Bytes:805 Parent-MAC-Sess: 27  65.54.82.179/80 -> 10.155.33.6/51795; Proto 6; Flg:0x114; Pkts:15 Bytes:16169  Id:25; Ageout:286265; Flags:0x24400; QOS:2; L7 ID:393; Up: 0 min 17 sec; InPol:L7Verify/2;  10.155.33.6/51755 -> 207.46.68.17/80; Proto 6; Flg:0x112; Pkts:6 Bytes:2430 Parent-MAC-Sess: 27  207.46.68.17/80 -> 10.155.33.6/51755; Proto 6; Flg:0x110; Pkts:3 Bytes:866  Id:50; Ageout:285500; Flags:0x24400; QOS:2; L7 ID:393; Up: 0 min 15 sec; InPol:L7Verify/2;  10.155.33.6/51770 -> 64.4.21.39/80; Proto 6; Flg:0x112; Pkts:7 Bytes:1924 Parent-MAC-Sess: 27  64.4.21.39/80 -> 10.155.33.6/51770; Proto 6; Flg:0x110; Pkts:3 Bytes:1394  Id:8; Ageout:285933; Flags:0x24400; QOS:2; L7 ID:393; Up: 0 min 18 sec; InPol:L7Verify/2;  10.155.33.6/51747 -> 207.46.49.133/80; Proto 6; Flg:0x112; Pkts:9 Bytes:3204 Parent-MAC-Sess: 27  207.46.49.133/80 -> 10.155.33.6/51747; Proto 6; Flg:0x110; Pkts:7 Bytes:5884  AP330 Google App ID 239:  Id:5; Ageout:296348; Flags:0x24400; QOS:2; L7 ID:239; Up: 0 min 29 sec; InPol:L7Verify/2;  10.155.33.6/51864 -> 74.125.224.81/80; Proto 6; Flg:0x112; Pkts:10 Bytes:4484 Parent-MAC-Sess: 27  74.125.224.81/80 -> 10.155.33.6/51864; Proto 6; Flg:0x110; Pkts:8 Bytes:8012  AP330 SIP App ID 559:  Id:2; Ageout:456260; Flags:0x24400; QOS:2; L7 ID:559; Up: 2 min 24 sec; InPol:L7Verify/1;  10.155.33.6/40708 -> 10.155.30.124/5060; Proto 17; Flg:0x102; Pkts:3 Bytes:1936 Parent-MAC-Sess: 27  10.155.30.124/5060 -> 10.155.33.6/40708; Proto 17; Flg:0x100; Pkts:2 Bytes:928  AP330 RTP App ID 534:  Id:3; Ageout:89537; Flags:0x24400; QOS:2; L7 ID:534; Up: 0 min 18 sec; InPol:L7Verify/1;  10.155.33.6/21414 -> 10.155.30.28/10642; Proto 17; Flg:0x102; Pkts:358 Bytes:71288 Parent-MAC-Sess: 27  10.155.30.28/10642 -> 10.155.33.6/21414; Proto 17; Flg:0x100; Pkts:382 Bytes:76204  AP330 Hotmail App ID 271:  Id:6; Ageout:296910; Flags:0x24400; QOS:2; L7 ID:271; Up: 0 min 24 sec; InPol:L7Verify/2;  10.155.33.6/52101 -> 64.4.2.107/80; Proto 6; Flg:0x112; Pkts:10 Bytes:4492 Parent-MAC-Sess: 27  64.4.2.107/80 -> 10.155.33.6/52101; Proto 6; Flg:0x110; Pkts:6 Bytes:6282  Id:11; Ageout:282616; Flags:0x24400; QOS:2; L7 ID:271; Up: 0 min 17 sec; InPol:L7Verify/2;  10.155.33.6/52217 -> 64.4.56.183/80; Proto 6; Flg:0x112; Pkts:5 Bytes:2611 Parent-MAC-Sess: 27  64.4.56.183/80 -> 10.155.33.6/52217; Proto 6; Flg:0x110; Pkts:3 Bytes:1058 | | |
| Comment | Aerohive required L7 engine, which supports more than 700 L7 applications, from other vendor. It will cost much time and effort to verify all of L7 applications. But if QA does NOT verify L7 application, we can’t guarantee the quality of Aerohive product.  As per QA and HiveOS Dev discussion, we apply following policy to verify L7 engine:   1. L7 engine vendor provided L7 application sample trace, IXIA replay sample trace to verify L7 application. 2. Verified critical or common applications by live traffic. 3. Require L7 engine vendor’s test plan document to make sure all applications have been covered by vendor. | | |

#### L7\_Engine\_Function\_02

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Function\_02 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | For AP only with 1 eth interface:  (eth0)MP(wifi1.1)-----AP\_\_\_\_SW\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|  For AP with 2 eth interfaces:  (eth1)AP(eth0)\_\_\_\_SW\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_|  For BR:  (eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify HiveOS device (AP and BR) can classify specific application. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR.     2. Keep default settings of application reporting for every individual interface at AP and BR     3. IXIA replay application sample trace which is provided by L7 engine vendor.     4. Check if AP can classify application.     5. Check if BR can classify application.   Note: Record applications which are verified. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. The default mode of application reporting at interfaces is auto.  Step 4. AP classifies application and reports it to HM correctly.  Step 5. BR classifies application and reports it to HM correctly. | | |
| Test Result |  | | |
| Comment | Aerohive required L7 engine, which supports more than 700 L7 applications, from other vendor. It will cost much time and effort to verify all of L7 applications. But if QA does NOT verify L7 application, we can’t guarantee the quality of Aerohive product.  As per QA and HiveOS Dev discussion, we apply following policy to verify L7 engine:   1. L7 engine vendor provided L7 application sample trace, IXIA replay sample trace to verify L7 application. 2. Verified critical or common applications by live traffic. 3. Require L7 engine vendor’s test plan document to make sure all applications have been covered by vendor. | | |

#### L7\_Engine\_Function\_03

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Function\_03 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Verify HiveOS device (AP and BR) can classify game which works as application widgets on social networking websites such as Facebook and MySpace. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR.     2. ~~Keep default settings of application reporting for every individual interface at AP and BR~~     3. Launch web game application at Laptop1. For example, launch Zynga game integrated with Facebook.     4. Check if AP can classify web game application. For example, classify Zynga and Facebook respectively.     5. Check if BR can classify web game application. Forexample, classify Zynga and Facebook respectively.   Note: Record applications which are verified. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. The default mode of application reporting at interfaces is auto.  Step 4. AP classifies applications.  Step 5. BR classifies applications. | | |
| Test Result |  | | |
| Comment |  | | |

### L7 Engine Classification State Verification

#### L7\_Engine\_Function\_04

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Function\_04 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | L7 engine classification state verification: “Monitored” | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR.     2. Keep default settings of application reporting for every individual interface at AP and BR     3. Laptop1 connects to a FTP server and launch file download from server.     4. Check if AP and BR can classify FTP. | | |
| Expect result | Step 4. AP and BR can classify sessions as FTP, which includs IP sessions for both FTP control and file download. | | |
| Test Result |  | | |
| Comment | For a few applications like FTP, SIP VoIP call and FaceTime, there is a middle state as “Monitored” before L7 engine classify them.  L7 DPI engine is required to continue monitoring traffic for further analysis, and packets need to be sent to L7 daemon for processing before reaching the maximum number of “monitored” packets allowed. | | |

#### ~~L7\_Engine\_Function\_05~~

|  |  |  |  |
| --- | --- | --- | --- |
| ~~Case ID~~ | ~~L7\_Engine\_Function\_05~~ | | |
| ~~Priority~~ | ~~Low~~ | ~~Automation Flag~~ | ~~No~~ |
| ~~Topology to use~~ | ~~For BR:~~  ~~(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM~~  ~~| |~~  ~~IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|~~  ~~For AP:~~  ~~(eth0)MP(wifi1.1)-----AP\_\_\_\_SW\_\_\_\_HM~~  ~~| |~~  ~~IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|~~ | | |
| ~~Description~~ | ~~L7 engine classification state verification: “Unknown”~~ | | |
| ~~PlatformDependence~~ | ~~AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,~~  ~~BR: BR200,BR200-WP,BRAP330,BRAP350,~~ | | |
| ~~Pre-condition~~ | ~~For BR：~~  ~~BR is managed by HM.~~  ~~Set BR eth1 mode as bridge-access, and eth0 as wan.~~  ~~For AP:~~  ~~Set MP eth0 as bridge-access mode, and wifi1.1 as backhaul mode.~~ | | |
| ~~Test procedure~~ | * + 1. ~~Turn on L7 engine and set application reporting mode as “auto” globally at MP or BR.~~     2. ~~Keep default settings of application reporting for every individual interface at MP or BR~~     3. ~~IXIA launch abnormal BT session with modified payload option to make sure MP and BR can’t classify BT session.~~     4. ~~Check if MP or BR can classify BT session. And check MP or BR CPU utilization rate and used memory.~~ | | |
| ~~Expect result~~ | ~~Step 4. MP and BR can NOT classify session as BT. Set App ID of associated IP session as “0”.~~  ~~The MP’s CPU utilization rate and used memory will return to regular level after the maximum number of packets allowed for classification has been reached.~~  ~~The BR’s CPU utilization rate and used memory will return to regular level after the maximum number of packets allowed for classification has been reached.~~ | | |
| ~~Test Result~~ |  | | |
| ~~Comment~~ | ~~UNKNOWN~~  ~~This state means that DPI engine has attempted and failed to classify the traffic, or, the maximum number of packets allowed for classification has been reached without receiving result, and the session has thus been marked as unknown, and no more packets are needed by L7 daemon.~~  ~~Ignore this case.~~  ~~L7 engine can set application traffic as “TCP/UDP/IP”. And it can be covered in stress test.~~ | | |

### L7 Engine Roaming Scenario Test

#### L7\_Engine\_Function\_06

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Function\_06 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP1(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  -----(wifi0/wifi1)AP2(eth0)\_\_\_\_\_\_\_|  |  FTP server | | |
| Description | Verify “Classified” ip session with app id will be synchronized during station roaming. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP1 and AP2 are managed by HM.  Create a SSID and bind it with APs sub-interface, which is set as access mode.  Laptop1 associates with AP1 by SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP1 and AP2.     2. ~~Keep default settings of application reporting for APs’ interfaces which application traffic will go through~~     3. Enable application FTP, TCP and IP for reporting.     4. Laptop1 accesses FTP server and download file.     5. Check ip-sessions for FTP at AP1.     6. Laptop1 roams from AP1 to AP2.     7. Check ip-session for FTP at AP2. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  ~~Step 2. The default mode of application reporting at interfaces is auto.~~  Step 3. Enable application for reporting successfully.  Step 5. The ip-sessions for ftp control and file download are classified as FTP session at AP1.  Step 7. The ip-sessions for ftp control and file download are synchronized to AP2. These ip-sessions are marked as FTP session. | | |
| Test Result |  | | |
| Comment |  | | |

#### L7\_Engine\_Function\_07

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Function\_07 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP1(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  -----(wifi0/wifi1)AP2(eth0)\_\_\_\_\_\_\_|  |  FTP server | | |
| Description | Verify “monitored” ip session with app id will be synchronized during station roaming. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP1 and AP2 are managed by HM.  Create a SSID and bind it with APs sub-interface, which is set as access mode.  Laptop1 associates with AP1 by SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP1 and AP2.     2. ~~Keep default settings of application reporting for APs’ interfaces which application traffic will go through~~     3. Enable application FTP, TCP and IP for reporting.     4. Laptop1 accesses FTP server but does NOT download file.     5. Check ip-sessions for FTP at AP1.     6. Laptop1 roams from AP1 to AP2.     7. Check ip-session for FTP at AP2.     8. Download file from FTP server     9. Check ip-session for FTP at AP2 again. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. The default mode of application reporting at interfaces is auto.  Step 3. Enable application for reporting successfully.  Step 5. AP1 classify ip-session for ftp control.  Step 7. The ip-session for ftp control is synchronized to AP2.  Step 9. AP2 classifies session for FTP file download. | | |
| Test Result |  | | |
| Comment |  | | |

#### L7\_Engine\_Function\_08

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Function\_08 | | |
| Priority | Middle | Automation Flag | No |
| Topology | Laptop1-----(wifi0/wifi1)AP1(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  -----(wifi0/wifi1)AP2(eth0)\_\_\_\_\_\_\_|  |  Internet | | |
| Description | Verify “un-classified” application session will be reported as “IP” after roaming. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP1 and AP2 are managed by HM.  Create a SSID and bind it with APs sub-interface, which is set as access mode.  Laptop1 associates with AP1 by SSID. | | |
| TestProcedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP1 and AP2.     2. Keep default settings of application reporting for APs’ interfaces which application traffic will go through     3. Launch application at Laptop1, and make sure application has NOT been classified by AP1.     4. Laptop1 roams from AP1 to AP2.     5. Continue L7 application at Laptop1.     6. IP session of “un-classified” application and application reporting file at AP2. | | |
| ExpectResult | Step 3. AP1 does NOT classify L7 application.  Step 6. AP2 does NOT classify this application traffic, but report it as “IP” with TLV record correctly. | | |
| TestResult | At dst AP:  2013-03-06 15:40:36 debug l7d: [dpi\_basic]: wk 0 conn 401 ip\_sess 2 : Notified appid 0 apps /0 (10.155.33.225/80 -> 10.155.33.30/63614 6)  2013-03-06 15:41:02 info l7d: A new report file created: "08EA440C5EC00049.hpr"  Check reporting file, it only contain one record for App “IP” | | |
| Comment | Need automation script to control Laptop1 roaming during test.  And need to select a L7 application for test. This application cannot be classified by L7 engine with few packets.  When clients roam, active IP sessions will be synced over to new AP along with their L7 status. There will be cases when a "synced over" IP session being in an "un-classified" state meaning L7 classification has not completed on these. Due to DPI engine's limitation we cannot resume classification on new AP. These "un-classified" sessions are reported as L7 application "IP" and are permitted by new AP.  During test, need to make sure application reporting accuracy of station roaming. | | |

#### L7\_Engine\_Function\_09

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Function\_09 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP1(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  -----(wifi0/wifi1)AP2(eth0)\_\_\_\_\_\_\_|  |  FTP server | | |
| Description | Verify scenario that application has been classified at SRC AP, after roaming initiate a new ip-session, it should be classified correctly. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP1 and AP2 are managed by HM.  Create a SSID and bind it with APs sub-interface, which is set as access mode.  Laptop1 associates with AP1 by SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP1 and AP2.     2. Keep default settings of application reporting for APs’ interfaces which application traffic will go through     3. Enable application BT, TCP, UDP and IP for reporting.     4. Launch BT at laptop1 for file download.     5. Check ip-sessions for BT at AP1.     6. Laptop1 roams from AP1 to AP2.     7. Check ip-session for BT at AP2.     8. Laptop1 finds a new peer, and generates connection with it.     9. Check this new ip-session can be classified as BT session. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. The default mode of application reporting at interfaces is auto.  Step 3. Enable application for reporting successfully.  Step 5. The ip-sessions are classified as BT at AP1.  Step 7. The ip-sessions belonging to BT are synchronized to AP2 with BT app ID successfully.  Step 9. New ip-session is classified as BT. | | |
| Test Result |  | | |
| Comment |  | | |

#### L7\_Engine\_Function\_10

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Function\_10 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP1(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  -----(wifi0/wifi1)AP2(eth0)\_\_\_\_\_\_\_|  |  Internet | | |
| Description | Verify roaming between APs which one enables L7 engine and another one disable it. During roaming, 2 APs will sync ip-session which can associates with App ID. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP1 and AP2 are managed by HM.  Create a SSID and bind it with APs sub-interface, which is set as access mode.  Laptop1 associates with AP1 by SSID. | | |
| Test procedure | * + 1. Turn on L7 engine at AP1.   CLI: no application identification shutdown   * + 1. Turn off L7 engine at AP2.   CLI: application identification shutdown   * + 1. Keep default setting of application reporting at AP1 and AP2.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet via HTTP session.     4. Check ip-sessions for HTTP at AP1.     5. Laptop1 roams from AP1 to AP2.     6. Check ip-session for HTTP at AP2.     7. Clear HTTP session at AP2, and launch it again at Laptop1.     8. Repeat step 6 to 8 again, but at this time roams from AP2 to AP1. | | |
| Expect result | Step 1. Turn on L7 engine at AP1 successfully.  Step 2. Turn off L7 engine at AP2 successfully.  Step 4. Enable application for reporting successfully.  Step 6. The HTTP session is classified at AP1.  Step 7. Laptop1 roams from AP1 to AP2. No AP crash.  Step 8. The HTTP session is synchronized to AP2. But no application reporting at AP2.  AP1 syslog  Show log buffer  2012-12-11 14:04:35 debug kernel: [fe]: send ip-session (id 4) 10.155.33.24/62794 <-> 119.147.241.43/80, proto 6 qos 2, mac-sess dir (reverse) app 1 flag 0x0  2012-12-11 14:04:35 debug kernel: [fe]: send ip session from sync 10.155.33.24/62794 <-> 119.147.241.43/80, 6, mac sess dir (reverse)  2012-12-11 14:04:35 debug kernel: [fe]: send ip-session (id 5) 10.155.33.24/62796 <-> 119.147.241.43/80, proto 6 qos 2, mac-sess dir (reverse) app 1 flag 0x0  2012-12-11 14:04:35 debug kernel: [fe]: send ip session from sync 10.155.33.24/62796 <-> 119.147.241.43/80, 6, mac sess dir (reverse)  2012-12-11 14:04:35 debug kernel: [fe]: send ip-session (id 6) 10.155.33.24/62800 <-> 119.147.241.43/80, proto 6 qos 2, mac-sess dir (reverse) app 1 flag 0x0  2012-12-11 14:04:35 debug kernel: [fe]: send ip session from sync 10.155.33.24/62800 <-> 119.147.241.43/80, 6, mac sess dir (reverse)  2012-12-11 14:04:35 debug kernel: [fe]: send ip-session (id 7) 10.155.33.24/62801 <-> 119.147.241.43/80, proto 6 qos 2, mac-sess dir (reverse) app 1 flag 0x0  2012-12-11 14:04:35 debug kernel: [fe]: send ip session from sync 10.155.33.24/62801 <-> 119.147.241.43/80, 6, mac sess dir (reverse)  2012-12-11 14:04:35 debug kernel: [fe]: send ip-session (id 9) 10.155.33.24/62804 <-> 119.147.241.43/80, proto 6 qos 2, mac-sess dir (reverse) app 1 flag 0x0  2012-12-11 14:04:35 debug kernel: [fe]: send ip session from sync 10.155.33.24/62804 <-> 119.147.241.43/80, 6, mac sess dir (reverse)  2012-12-11 14:04:35 debug kernel: [fe]: send mac-session (id 22) 80f6:2e3e:9005 (zone backhaul) <-> 2477:0304:aeb0 (zone access), ageout -711233708 ms  2012-12-11 14:04:35 debug kernel: [fe]: send mac-session (id 19) ffff:ffff:ffff (zone backhaul) <-> 2477:0304:aeb0 (zone access), ageout -711233708 ms  2012-12-11 14:04:35 debug kernel: [fe]: send sess sync msg, instance 2 mac 2477:0304:aeb0  2012-12-11 14:04:35 debug kernel: [fe]: fe get sync session  AP2 syslog:  2012-12-11 14:04:35 debug kernel: [fe]: created ip session from sync 10.155.33.24/62794 <-> 119.147.241.43/80, 6, mac sess dir (reverse) flag 0x220  2012-12-11 14:04:35 debug kernel: [fe]: create new IP session, id [6] ageout 300 sec, alg n/a flags 0x220  2012-12-11 14:04:35 debug kernel: [fe]: l7fw ingress aclgrp ce713000 acl cef98200 fwst 0 rc 1  2012-12-11 14:04:35 debug kernel: [fe]: permitted by ACL  2012-12-11 14:04:35 debug kernel: [fe]: ACL: matched upid [1] dir [ingress (from-air)] group [L7Verify] id [2] [permit] mirror [no]  2012-12-11 14:04:35 debug kernel: [fe]: doing from-air IP ACL filter for upid 1  2012-12-11 14:04:35 debug kernel: [fe]: doing IP filter check: mr\_flag 0xa1210 mr\_acl 0x0/0x0 mf\_flg 0x25000 mf\_acl 0x0/0x0  2012-12-11 14:04:35 debug kernel: [fe]: no IP sess found  2012-12-11 14:04:35 debug kernel: [fe]: receive ip session from sync 10.155.33.24/62794 <-> 119.147.241.43/80, 6, mac sess dir (reverse) flag 0x0  2012-12-11 14:04:35 debug kernel: [fe]: created ip session from sync 10.155.33.24/62796 <-> 119.147.241.43/80, 6, mac sess dir (reverse) flag 0x220  2012-12-11 14:04:35 debug kernel: [fe]: create new IP session, id [5] ageout 300 sec, alg n/a flags 0x220  2012-12-11 14:04:35 debug kernel: [fe]: l7fw ingress aclgrp ce713000 acl cef98200 fwst 0 rc 1  2012-12-11 14:04:35 debug kernel: [fe]: permitted by ACL  2012-12-11 14:04:35 debug kernel: [fe]: ACL: matched upid [1] dir [ingress (from-air)] group [L7Verify] id [2] [permit] mirror [no]  2012-12-11 14:04:35 debug kernel: [fe]: doing from-air IP ACL filter for upid 1  2012-12-11 14:04:35 debug kernel: [fe]: doing IP filter check: mr\_flag 0xa1210 mr\_acl 0x0/0x0 mf\_flg 0x25000 mf\_acl 0x0/0x0  2012-12-11 14:04:35 debug kernel: [fe]: no IP sess found  2012-12-11 14:04:35 debug kernel: [fe]: receive ip session from sync 10.155.33.24/62796 <-> 119.147.241.43/80, 6, mac sess dir (reverse) flag 0x0  2012-12-11 14:04:35 debug kernel: [fe]: created ip session from sync 10.155.33.24/62800 <-> 119.147.241.43/80, 6, mac sess dir (reverse) flag 0x220  2012-12-11 14:04:35 debug kernel: [fe]: create new IP session, id [4] ageout 300 sec, alg n/a flags 0x220  2012-12-11 14:04:35 debug kernel: [fe]: l7fw ingress aclgrp ce713000 acl cef98200 fwst 0 rc 1  2012-12-11 14:04:35 debug kernel: [fe]: permitted by ACL  2012-12-11 14:04:35 debug kernel: [fe]: ACL: matched upid [1] dir [ingress (from-air)] group [L7Verify] id [2] [permit] mirror [no]  2012-12-11 14:04:35 debug kernel: [fe]: doing from-air IP ACL filter for upid 1  2012-12-11 14:04:35 debug kernel: [fe]: doing IP filter check: mr\_flag 0xa1210 mr\_acl 0x0/0x0 mf\_flg 0x25000 mf\_acl 0x0/0x0  2012-12-11 14:04:35 debug kernel: [fe]: no IP sess found  2012-12-11 14:04:35 debug kernel: [fe]: receive ip session from sync 10.155.33.24/62800 <-> 119.147.241.43/80, 6, mac sess dir (reverse) flag 0x0  2012-12-11 14:04:35 debug kernel: [fe]: created ip session from sync 10.155.33.24/62801 <-> 119.147.241.43/80, 6, mac sess dir (reverse) flag 0x220  2012-12-11 14:04:35 debug kernel: [fe]: create new IP session, id [3] ageout 300 sec, alg n/a flags 0x220  2012-12-11 14:04:35 debug kernel: [fe]: l7fw ingress aclgrp ce713000 acl cef98200 fwst 0 rc 1  2012-12-11 14:04:35 debug kernel: [fe]: permitted by ACL  2012-12-11 14:04:35 debug kernel: [fe]: ACL: matched upid [1] dir [ingress (from-air)] group [L7Verify] id [2] [permit] mirror [no]  2012-12-11 14:04:35 debug kernel: [fe]: doing from-air IP ACL filter for upid 1  2012-12-11 14:04:35 debug kernel: [fe]: doing IP filter check: mr\_flag 0xa1210 mr\_acl 0x0/0x0 mf\_flg 0x25000 mf\_acl 0x0/0x0  2012-12-11 14:04:35 debug kernel: [fe]: no IP sess found  2012-12-11 14:04:35 debug kernel: [fe]: receive ip session from sync 10.155.33.24/62801 <-> 119.147.241.43/80, 6, mac sess dir (reverse) flag 0x0  2012-12-11 14:04:35 debug kernel: [fe]: created ip session from sync 10.155.33.24/62804 <-> 119.147.241.43/80, 6, mac sess dir (reverse) flag 0x220  2012-12-11 14:04:35 debug kernel: [fe]: create new IP session, id [2] ageout 300 sec, alg n/a flags 0x220  2012-12-11 14:04:35 debug kernel: [fe]: l7fw ingress aclgrp ce713000 acl cef98200 fwst 0 rc 1  2012-12-11 14:04:35 debug kernel: [fe]: permitted by ACL  2012-12-11 14:04:35 debug kernel: [fe]: ACL: matched upid [1] dir [ingress (from-air)] group [L7Verify] id [2] [permit] mirror [no]  2012-12-11 14:04:35 debug kernel: [fe]: doing from-air IP ACL filter for upid 1  2012-12-11 14:04:35 debug kernel: [fe]: doing IP filter check: mr\_flag 0xa1210 mr\_acl 0x0/0x0 mf\_flg 0x25000 mf\_acl 0x0/0x0  2012-12-11 14:04:35 debug kernel: [fe]: no IP sess found  2012-12-11 14:04:35 debug kernel: [fe]: receive ip session from sync 10.155.33.24/62804 -> 119.147.241.43/80, 6, mac sess dir (reverse) flag 0x0  Step 10. Laptop1 roams from AP2 to AP1, no AP crash. And ip session is synchronized to AP1 without App ID.  AP1 syslog:  Show log buffer  2012-12-11 13:54:13 debug kernel: [fe]: created ip session from sync 10.155.33.24/63468 <-> 10.155.3.222/53, 17, mac sess dir (reverse) flag 0x220  2012-12-11 13:54:13 debug kernel: [fe]: create new IP session, id [3] ageout 60 sec, alg n/a flags 0x220  2012-12-11 13:54:13 debug kernel: [fe]: l7fw ingress aclgrp cf35c800 acl cefdf560 fwst 0 rc 1  2012-12-11 13:54:13 debug kernel: [fe]: permitted by ACL  2012-12-11 13:54:13 debug kernel: [fe]: ACL: matched upid [1] dir [ingress (from-air)] group [L7Verify] id [2] [permit] mirror [no]  2012-12-11 13:54:13 debug kernel: [fe]: doing from-air IP ACL filter for upid 1  2012-12-11 13:54:13 debug kernel: [fe]: doing IP filter check: mr\_flag 0xa1210 mr\_acl 0x0/0x0 mf\_flg 0x25000 mf\_acl 0x0/0x0  2012-12-11 13:54:13 debug kernel: [fe]: no IP sess found  2012-12-11 13:54:13 debug kernel: [fe]: receive ip session from sync 10.155.33.24/63468 <-> 10.155.3.222/53, 17, mac sess dir (reverse) flag 0x0  2012-12-11 13:54:13 debug kernel: [fe]: created ip session from sync 10.155.33.24/62758 <-> 61.174.61.175/80, 6, mac sess dir (reverse) flag 0x220  2012-12-11 13:54:13 debug kernel: [fe]: create new IP session, id [2] ageout 300 sec, alg n/a flags 0x220  2012-12-11 13:54:13 debug kernel: [fe]: l7fw ingress aclgrp cf35c800 acl cefdf560 fwst 0 rc 1  2012-12-11 13:54:13 debug kernel: [fe]: permitted by ACL  2012-12-11 13:54:13 debug kernel: [fe]: ACL: matched upid [1] dir [ingress (from-air)] group [L7Verify] id [2] [permit] mirror [no]  2012-12-11 13:54:13 debug kernel: [fe]: doing from-air IP ACL filter for upid 1  2012-12-11 13:54:13 debug kernel: [fe]: doing IP filter check: mr\_flag 0xa1210 mr\_acl 0x0/0x0 mf\_flg 0x25000 mf\_acl 0x0/0x0  2012-12-11 13:54:13 debug kernel: [fe]: no IP sess found  2012-12-11 13:54:13 debug kernel: [fe]: receive ip session from sync 10.155.33.24/62758 <-> 61.174.61.175/80, 6, mac sess dir (reverse) flag 0x0 | | |
| Test Result |  | | |
| Comment | 1. At step 8, check if AP2 loads App ID associated with ip session. 2. Two APs should create IP sessions during roaming.   In order to keep IP session, should set IP policy or alg after disable L7 engine. | | |

#### L7\_Engine\_Function\_11

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Function\_11 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP1(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  -----(wifi0/wifi1)AP2(eth0)\_\_\_\_\_\_\_|  |  Internet | | |
| Description | Verify roaming between APs which one supports L7 engine and another one doesn’t support it. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | Load new image to AP1 to make it support L7 engine.  Load old image to AP2 to make it doesn’t support L7 engine.  AP1 and AP2 are managed by HM.  Create a SSID and bind it with APs sub-interface, which is set as access mode.  Laptop1 associates with AP1 by SSID. | | |
| Test procedure | * + 1. Turn on L7 engine at AP1.   CLI: No application identification shutdown   * + 1. Set application reporting mode as “auto” at AP1.   CLI: application reporting auto   * + 1. ~~Enable application HTTP, TCP and IP for reporting.~~     2. Laptop1 accesses Internet via HTTP session.     3. Check ip-sessions for HTTP at AP1.     4. Laptop1 roams from AP1 to AP2.     5. Check ip-session for HTTP at AP2.     6. Clear HTTP session at AP2, and launch it again at Laptop1.     7. Repeat step 5 to 7 again, but at this time roams from AP2 to AP1. | | |
| Expect result | Step 1. Turn on L7 engine at AP1 successfully.  ~~Step 3. Enable application for reporting successfully.~~  Step 5. The HTTP session is classified at AP1.  Step 6. Laptop1 roams from AP1 to AP2. No AP crash.  Step 7. The HTTP session is synchronized to AP2. AP2 ignores App ID associated with ip sessions.  Laptop1: 10.155.33.6 0024:d7c0:0308  FTP server: 10.155.33.189 000c:2940:6dbc  Show log buffer at AP1:  2012-11-20 20:30:36 debug kernel: [fe]: send ip-session (id 5) 10.155.33.6/49556 <-> 10.155.33.189/21, proto 6 qos 2, mac-sess dir (reverse) app 618 flag 0x0  2012-11-20 20:30:36 debug kernel: [fe]: send ip session from sync 10.155.33.6/49556 <-> 10.155.33.189/21, 6, mac sess dir (reverse)  2012-11-20 20:30:36 debug kernel: [fe]: send mac-session (id 28) 000c:2940:6dbc (zone backhaul) <-> 0024:d7c0:0308 (zone access), ageout 6 ms  Show log buffer at AP2:  2012-11-20 20:30:35 debug kernel: [fe]: created ip session from sync 10.155.33.6/49556 <-> 10.155.33.189/21, 6, mac sess dir (reverse) flag 0x4420  2012-11-20 20:30:35 debug kernel: [fe]: create new IP session, id [4] ageout 1800 sec, alg n/a flags 0x420  2012-11-20 20:30:35 debug kernel: [fe]: permitted by ACL  2012-11-20 20:30:35 debug kernel: [fe]: ACL: matched upid [1] dir [ingress (from-air)] group [L7Verify] id [2] [permit] mirror [no]  2012-11-20 20:30:35 debug kernel: [fe]: doing from-air IP ACL filter  2012-11-20 20:30:35 debug kernel: [fe]: doing IP filter check: mr\_flag 0xa1210 mr\_acl 0x0/0x0 mf\_flg 0x25000 mf\_acl 0x0/0x0  2012-11-20 20:30:35 debug kernel: [fe]: no IP sess found  2012-11-20 20:30:35 debug kernel: [fe]: receive ip session from sync 10.155.33.6/49556 -> 10.155.33.189/21, 6, mac sess dir (reverse) flag 0x0  2012-11-20 20:30:35 debug kernel: [fe]: QoS: first pkt L2 session (eth0->wifi1.1) upid=1 src=0(0) dst=1  2012-11-20 20:30:35 debug kernel: [fe]: mac session(id 33) 000c:2940:6dbc->0024:d7c0:0308, flags 0xa1210/0x25000, nf->acl 0x0/0x0 nr->acl 0x0/0x0  2012-11-20 20:30:35 debug kernel: [fe]: mac sess id [33] created, ageout time 300 sec  2012-11-20 20:30:35 debug kernel: [fe]: dst route lookup 000c:2940:6dbc: GET\_ROUTE dev(eth0) nhop(0019:770e:e440) mp (0019:770e:e440) flag (0xc11) upid (0)  2012-11-20 20:30:35 debug kernel: [fe]: dst route lookup 0024:d7c0:0308: GET\_ROUTE dev(wifi1.1) nhop(0019:770e:e440) mp (0019:770e:e440) flag (0x1c03) upid (1)  2012-11-20 20:30:35 debug kernel: [fe]: receive mac sess from sync 000c:2940:6dbc (zone backhaul) vid 0 <-> 0024:d7c0:0308 (zone access) vid 0  Check ip-session at AP2:  Id:4; Ageout:1797983; Flags:0x4420; QOS:2; Up: 0 min 2 sec; InPol:L7Verify/2;  10.155.33.6/49556 -> 10.155.33.189/21; Proto 6; Flg:0x102; Pkts:0 Bytes:0 Parent-MAC-Sess: 33  10.155.33.189/21 -> 10.155.33.6/49556; Proto 6; Flg:0x0; Pkts:0 Bytes:0  Step 9. Laptop1 roams from AP2 to AP1, no AP crash. And ip session is synchronized to AP1 without App ID.  Laptop1: 10.155.33.6 0024:d7c0:0308  FTP server: 10.155.33.189 000c:2940:6dbc  Syslog:  Nov 20 20:18:44 10.155.33.1 kernel: [fe]: fe get sync session  Nov 20 20:18:44 10.155.33.1 kernel: [fe]: send sess sync msg, instance 3 mac 0024:d7c0:0308  Nov 20 20:18:44 10.155.33.1 kernel: [fe]: send mac-session (id 26) ffff:ffff:ffff (zone backhaul) <-> 0024:d7c0:0308 (zone access), ageout -1070578496 ms  Nov 20 20:18:44 10.155.33.1 kernel: [fe]: send ip session from sync 10.155.33.6/8235 <-> 10.155.33.255/8235, 17, mac sess dir (reverse)  Nov 20 20:18:44 10.155.33.1 kernel: [fe]: send ip-session (id 6) 10.155.33.6/8235 <-> 10.155.33.255/8235, proto 17 qos 2, mac-sess dir (reverse) flag 0x0  Nov 20 20:18:44 10.155.33.1 kernel: [fe]: send mac-session (id 29) 0024:d7c0:0308 (zone access) <-> 80f6:2e3e:9005 (zone backhaul), ageout 17 ms  Nov 20 20:18:44 10.155.33.1 kernel: [fe]: send ip session from sync 10.155.30.28/59536 <-> 10.155.33.6/3389, 6, mac sess dir (reverse)  Nov 20 20:18:44 10.155.33.1 kernel: [fe]: send ip-session (id 7) 10.155.30.28/59536 <-> 10.155.33.6/3389, proto 6 qos 2, mac-sess dir (reverse) flag 0x0  Nov 20 20:18:44 10.155.33.1 kernel: [fe]: send mac-session (id 30) 000c:2940:6dbc (zone backhaul) <-> 0024:d7c0:0308 (zone access), ageout 6 ms  Nov 20 20:18:44 10.155.33.1 kernel: [fe]: send ip session from sync 10.155.33.6/49544 <-> 10.155.33.189/21, 6, mac sess dir (reverse)  Nov 20 20:18:44 10.155.33.1 kernel: [fe]: send ip-session (id 1) 10.155.33.6/49544 <-> 10.155.33.189/21, proto 6 qos 2, mac-sess dir (reverse) flag 0x0  Nov 20 20:18:44 10.155.33.1 kernel: [fe]: send sess sync of size 360, instance 3  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: fe set sync session  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: recv session sync msg, size 360 version 0  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: receive mac sess from sync ffff:ffff:ffff (zone backhaul) vid 0 <-> 0024:d7c0:0308 (zone access) vid 0  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: dst route lookup 0024:d7c0:0308: GET\_ROUTE dev(wifi1.1) nhop(08ea:440e:e400) mp (08ea:440e:e400) flag (0x1c03) upid (1)  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: dst route lookup ffff:ffff:ffff: GET\_ROUTE dev(eth0) nhop(08ea:440e:e400) mp (08ea:440e:e400) flag (0xc11) upid (0)  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: MAC session [24] already exists  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: QoS: first pkt L2 session (eth0->wifi1.1) upid=1 src=0(0) dst=1  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: receive ip session from sync 10.155.33.6/8235 -> 10.155.33.255/8235, 17, mac sess dir (reverse) flag 0x0  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: no IP sess found  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: doing IP filter check: mr\_flag 0xa9210 mr\_acl 0x0/0x0 mf\_flg 0x25000 mf\_acl 0x0/0x0  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: doing from-air IP ACL filter for upid 1  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: ACL: matched upid [1] dir [ingress (from-air)] group [L7Verify] id [2] [permit] mirror [no]  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: permitted by ACL  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: l7fw ingress aclgrp cf38a800 acl cbb47b00 fwst 0 rc 1  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: create new IP session, id [1] ageout 100 sec, alg n/a flags 0x220  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: created ip session from sync 10.155.33.6/8235 <-> 10.155.33.255/8235, 17, mac sess dir (reverse) flag 0x4220  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: receive mac sess from sync 0024:d7c0:0308 (zone access) vid 0 <-> 80f6:2e3e:9005 (zone backhaul) vid 0  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: dst route lookup 80f6:2e3e:9005: GET\_ROUTE dev(eth0) nhop(08ea:440e:e400) mp (08ea:440e:e400) flag (0xc11) upid (0)  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: dst route lookup 0024:d7c0:0308: GET\_ROUTE dev(wifi1.1) nhop(08ea:440e:e400) mp (08ea:440e:e400) flag (0x1c03) upid (1)  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: mac sess id [26] created, ageout time 300 sec  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: mac session(id 26) 0024:d7c0:0308->80f6:2e3e:9005, flags 0x25200/0xa1010, nf->acl 0x0/0x0 nr->acl 0x0/0x0  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: QoS: first pkt L2 session (wifi1.1->eth0) upid=1 src=1(1) dst=0  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: receive ip session from sync 10.155.30.28/59536 -> 10.155.33.6/3389, 6, mac sess dir (reverse) flag 0x0  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: no IP sess found  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: doing IP filter check: mr\_flag 0x25200 mr\_acl 0x0/0x0 mf\_flg 0xa1010 mf\_acl 0x0/0x0  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: doing to-air IP ACL filter for upid 1  Nov 20 20:18:44 10.155.33.4 kernel: [fe]: ACL: matched upid [1] dir [egress (to-air)] group [L7Verify] id [2] [permit] mirror [no]  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: permitted by ACL  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: conn 0 : l7fw egress aclgrp cf38a800 acl cbb47b00 fwst 0  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: create new IP session, id [2] ageout 300 sec, alg n/a flags 0x220  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: created ip session from sync 10.155.30.28/59536 <-> 10.155.33.6/3389, 6, mac sess dir (reverse) flag 0x4220  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: receive mac sess from sync 000c:2940:6dbc (zone backhaul) vid 0 <-> 0024:d7c0:0308 (zone access) vid 0  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: dst route lookup 0024:d7c0:0308: GET\_ROUTE dev(wifi1.1) nhop(08ea:440e:e400) mp (08ea:440e:e400) flag (0x1c03) upid (1)  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: dst route lookup 000c:2940:6dbc: GET\_ROUTE dev(eth0) nhop(08ea:440e:e400) mp (08ea:440e:e400) flag (0xc11) upid (0)  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: mac sess id [28] created, ageout time 300 sec  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: mac session(id 28) 000c:2940:6dbc->0024:d7c0:0308, flags 0xa1210/0x25000, nf->acl 0x0/0x0 nr->acl 0x0/0x0  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: QoS: first pkt L2 session (eth0->wifi1.1) upid=1 src=0(0) dst=1  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: receive ip session from sync 10.155.33.6/49544 -> 10.155.33.189/21, 6, mac sess dir (reverse) flag 0x0  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: no IP sess found  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: doing IP filter check: mr\_flag 0xa1210 mr\_acl 0x0/0x0 mf\_flg 0x25000 mf\_acl 0x0/0x0  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: doing from-air IP ACL filter for upid 1  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: ACL: matched upid [1] dir [ingress (from-air)] group [L7Verify] id [2] [permit] mirror [no]  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: permitted by ACL  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: l7fw ingress aclgrp cf38a800 acl cbb47b00 fwst 0 rc 1  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: create new IP session, id [3] ageout 1800 sec, alg n/a flags 0x220  Nov 20 20:18:45 10.155.33.4 kernel: [fe]: created ip session from sync 10.155.33.6/49544 <-> 10.155.33.189/21, 6, mac sess dir (reverse) flag 0x4220  Check ip-session at AP1:  Id:3; Ageout:1797437; Flags:0x4220; QOS:2; L7 ID:65535; Up: 0 min 2 sec; InPol:L7Verify/2;  10.155.33.6/49544 -> 10.155.33.189/21; Proto 6; Flg:0x102; Pkts:0 Bytes:0 Parent-MAC-Sess: 28  10.155.33.189/21 -> 10.155.33.6/49544; Proto 6; Flg:0x0; Pkts:0 Bytes:0 | | |
| Test Result |  | | |
| Comment | For more information, please check bug 20311.  Two APs should create IP sessions during roaming. | | |

#### L7\_Engine\_Function\_12

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Function\_12 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP1(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  -----(wifi0/wifi1)AP2(eth0)\_\_\_\_\_\_\_|  |  Internet | | |
| Description | Verify roaming between APs which one supports L7 engine but disable it and another one doesn’t support it. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | Load new image to AP1 to make it support L7 engine.  Load old image to AP2 to make it doesn’t support L7 engine.  AP1 and AP2 are managed by HM.  Create a SSID and bind it with APs sub-interface, which is set as access mode.  Laptop1 associates with AP1 by SSID. | | |
| Test procedure | * + 1. Turn off L7 engine at AP1.     2. Keep default setting of application reporting at AP1.     3. ~~Enable application HTTP, TCP and IP for reporting.~~     4. Laptop1 accesses Internet via HTTP session.     5. Check ip-sessions for HTTP at AP1.     6. Laptop1 roams from AP1 to AP2.     7. Check ip-session for HTTP at AP2.     8. Clear HTTP session at AP2, and launch it again at Laptop1.     9. Repeat step 5 to 7 again, but at this time roams from AP2 to AP1. | | |
| Expect result | Step 1. Turn on L7 engine at AP1 successfully.  ~~Step 3. Enable application for reporting successfully.~~  Step 5. AP1 does NOT classifiy application HTTP.  Id:3; Ageout:1797334; Flags:0x24400; QOS:2; L7 ID:65535; Up: 0 min 7 sec; InPol:L7Verify/2;  10.155.33.6/50520 -> 10.155.33.189/21; Proto 6; Flg:0x112; Pkts:10 Bytes:446 Parent-MAC-Sess: 42  10.155.33.189/21 -> 10.155.33.6/50520; Proto 6; Flg:0x110; Pkts:7 Bytes:536  Step 6. Laptop1 roams from AP1 to AP2. No AP crash.  Laptop1: 10.155.33.6 0024:d7c0:0308  FTP server: 10.155.33.189 000c:2940:6dbc  Show log buffer at AP1:  2012-11-21 13:22:39 debug kernel: [fe]: send sess sync of size 820, instance 1  2012-11-21 13:22:39 debug kernel: [fe]: send ip-session (id 3) 10.155.33.6/50520 <-> 10.155.33.189/21, proto 6 qos 2, mac-sess dir (reverse) app 65535 flag 0x0  2012-11-21 13:22:39 debug kernel: [fe]: send ip session from sync 10.155.33.6/50520 <-> 10.155.33.189/21, 6, mac sess dir (reverse)  2012-11-21 13:22:39 debug kernel: [fe]: send mac-session (id 42) 000c:2940:6dbc (zone backhaul) <-> 0024:d7c0:0308 (zone access), ageout 6 ms  2012-11-21 13:22:39 debug kernel: [fe]: send mac-session (id 36) 00e0:1e5d:bf43 (zone backhaul) <-> 0024:d7c0:0308 (zone access), ageout 6 ms  2012-11-21 13:22:39 debug kernel: [fe]: send ip-session (id 1) 10.155.30.28/59822 <-> 10.155.33.6/3389, proto 6 qos 2, mac-sess dir (reverse) app 65535 flag 0x0  2012-11-21 13:22:39 debug kernel: [fe]: send ip session from sync 10.155.30.28/59822 <-> 10.155.33.6/3389, 6, mac sess dir (reverse)  2012-11-21 13:22:39 debug kernel: [fe]: send mac-session (id 32) 0024:d7c0:0308 (zone access) <-> 80f6:2e3e:9005 (zone backhaul), ageout 17 ms  2012-11-21 13:22:39 debug kernel: [fe]: send ip-session (id 4) 10.155.33.6/8235 <-> 10.155.33.255/8235, proto 17 qos 2, mac-sess dir (reverse) app 65535 flag 0x0  2012-11-21 13:22:39 debug kernel: [fe]: send ip session from sync 10.155.33.6/8235 <-> 10.155.33.255/8235, 17, mac sess dir (reverse)  2012-11-21 13:22:39 debug kernel: [fe]: send ip-session (id 2) 10.155.33.6/4112 -> 224.0.0.252/4112, proto 2 qos 2, mac-sess dir (reverse) app 65535 flag 0x0  2012-11-21 13:22:39 debug kernel: [fe]: send ip session from sync 10.155.33.6/4112 -> 224.0.0.252/4112, 2, mac sess dir (reverse)  2012-11-21 13:22:39 debug kernel: [fe]: send mac-session (id 29) ffff:ffff:ffff (zone backhaul) <-> 0024:d7c0:0308 (zone access), ageout -711241588 ms  2012-11-21 13:22:39 debug kernel: [fe]: send sess sync msg, instance 1 mac 0024:d7c0:0308  2012-11-21 13:22:39 debug kernel: [fe]: fe get sync session  Show log buffer at AP2:  2012-11-21 13:22:39 debug [fe]: created ip session from sync 10.155.33.6/50520 <-> 10.155.33.189/21, 6, mac sess dir (reverse) flag 0x4420  2012-11-21 13:22:39 debug [fe]: create new IP session, id [4] ageout 1800 sec, alg n/a flags 0x420  2012-11-21 13:22:39 debug [fe]: permitted by ACL  2012-11-21 13:22:39 debug [fe]: ACL: matched upid [1] dir [ingress (from-air)] group [L7Verify] id [2] [permit] mirror [no]  2012-11-21 13:22:39 debug [fe]: doing from-air IP ACL filter  2012-11-21 13:22:39 debug [fe]: doing IP filter check: mr\_flag 0xa1210 mr\_acl 0x0/0x0 mf\_flg 0x25000 mf\_acl 0x0/0x0  2012-11-21 13:22:39 debug [fe]: no IP sess found  2012-11-21 13:22:39 debug [fe]: receive ip session from sync 10.155.33.6/50520 -> 10.155.33.189/21, 6, mac sess dir (reverse) flag 0x0  2012-11-21 13:22:39 debug [fe]: QoS: first pkt L2 session (eth0->wifi1.1) upid=1 src=0(0) dst=1  2012-11-21 13:22:39 debug [fe]: mac session(id 21) 000c:2940:6dbc->0024:d7c0:0308, flags 0xa1210/0x25000, nf->acl 0x0/0x0 nr->acl 0x0/0x0  2012-11-21 13:22:39 debug [fe]: mac sess id [21] created, ageout time 300 sec  2012-11-21 13:22:39 debug [fe]: dst route lookup 000c:2940:6dbc: GET\_ROUTE dev(eth0) nhop(0019:770e:e440) mp (0019:770e:e440) flag (0xc11) upid (0)  2012-11-21 13:22:39 debug [fe]: dst route lookup 0024:d7c0:0308: GET\_ROUTE dev(wifi1.1) nhop(0019:770e:e440) mp (0019:770e:e440) flag (0x1c03) upid (1)  2012-11-21 13:22:39 debug [fe]: receive mac sess from sync 000c:2940:6dbc (zone backhaul) <-> 0024:d7c0:0308 (zone access)  Step 7. The IP session without App ID is synchronized to AP2.  Id:4; Ageout:1794590; Flags:0x4420; QOS:2; Up: 0 min 5 sec; InPol:L7Verify/2;  10.155.33.6/50520 -> 10.155.33.189/21; Proto 6; Flg:0x102; Pkts:0 Bytes:0 Parent-MAC-Sess: 21  10.155.33.189/21 -> 10.155.33.6/50520; Proto 6; Flg:0x0; Pkts:0 Bytes:0  Step 9. Laptop1 roams from AP2 to AP1, no AP crash. And ip session without App ID is synchronized to AP1.  IP session at AP2  Id:4; Ageout:1797467; Flags:0x24400; QOS:2; Up: 0 min 7 sec; InPol:L7Verify/2;  10.155.33.6/50529 -> 10.155.33.189/21; Proto 6; Flg:0x112; Pkts:10 Bytes:446 Parent-MAC-Sess: 21  10.155.33.189/21 -> 10.155.33.6/50529; Proto 6; Flg:0x110; Pkts:7 Bytes:536  Show log buffer at AP2:  2012-11-21 13:31:16 debug [fe]: send ip-session (id 4) 10.155.33.6/50529 <-> 10.155.33.189/21, proto 6 qos 2, mac-sess dir (reverse) flag 0x0  2012-11-21 13:31:16 debug [fe]: send ip session from sync 10.155.33.6/50529 <-> 10.155.33.189/21, 6, mac sess dir (reverse)  2012-11-21 13:31:16 debug [fe]: send mac sess 000c:2940:6dbc (zone backhaul) -> 0024:d7c0:0308 (zone access)  2012-11-21 13:31:16 debug [fe]: send mac-session (id 21) 000c:2940:6dbc (zone backhaul) <-> 0024:d7c0:0308 (zone access), ageout -820905436 ms  2012-11-21 13:31:16 debug [fe]: send ip-session (id 3) 10.155.30.28/59822 <-> 10.155.33.6/3389, proto 6 qos 2, mac-sess dir (reverse) flag 0x0  2012-11-21 13:31:16 debug [fe]: send ip session from sync 10.155.30.28/59822 <-> 10.155.33.6/3389, 6, mac sess dir (reverse)  2012-11-21 13:31:16 debug [fe]: send mac sess 0024:d7c0:0308 (zone access) -> 80f6:2e3e:9005 (zone backhaul)  2012-11-21 13:31:16 debug [fe]: send mac-session (id 20) 0024:d7c0:0308 (zone access) <-> 80f6:2e3e:9005 (zone backhaul), ageout -917650908 ms  2012-11-21 13:31:16 debug [fe]: send ip-session (id 1) 10.155.33.6/4112 -> 224.0.0.252/4112, proto 2 qos 2, mac-sess dir (reverse) flag 0x0  2012-11-21 13:31:16 debug [fe]: send ip session from sync 10.155.33.6/4112 -> 224.0.0.252/4112, 2, mac sess dir (reverse)  2012-11-21 13:31:16 debug [fe]: send ip-session (id 2) 10.155.33.6/8235 <-> 10.155.33.255/8235, proto 17 qos 2, mac-sess dir (reverse) flag 0x0  2012-11-21 13:31:16 debug [fe]: send ip session from sync 10.155.33.6/8235 <-> 10.155.33.255/8235, 17, mac sess dir (reverse)  2012-11-21 13:31:16 debug [fe]: send mac sess ffff:ffff:ffff (zone backhaul) -> 0024:d7c0:0308 (zone access)  2012-11-21 13:31:16 debug [fe]: send mac-session (id 17) ffff:ffff:ffff (zone backhaul) <-> 0024:d7c0:0308 (zone access), ageout -710487192 ms  2012-11-21 13:31:16 debug [fe]: send sess sync msg, instance 1 mac 0024:d7c0:0308  2012-11-21 13:31:16 debug [fe]: fe get sync session  Show log buffer at AP1:  2012-11-21 13:31:16 debug kernel: [fe]: created ip session from sync 10.155.33.6/50529 <-> 10.155.33.189/21, 6, mac sess dir (reverse) flag 0x4220  2012-11-21 13:31:16 debug kernel: [fe]: create new IP session, id [6] ageout 1800 sec, alg n/a flags 0x220  2012-11-21 13:31:16 debug kernel: [fe]: l7fw ingress aclgrp cdc57400 acl cd5e8480 fwst 0 rc 1  2012-11-21 13:31:16 debug kernel: [fe]: permitted by ACL  2012-11-21 13:31:16 debug kernel: [fe]: ACL: matched upid [1] dir [ingress (from-air)] group [L7Verify] id [2] [permit] mirror [no]  2012-11-21 13:31:16 debug kernel: [fe]: doing from-air IP ACL filter for upid 1  2012-11-21 13:31:16 debug kernel: [fe]: doing IP filter check: mr\_flag 0xa1210 mr\_acl 0x0/0x0 mf\_flg 0x25000 mf\_acl 0x0/0x0  2012-11-21 13:31:16 debug kernel: [fe]: no IP sess found  2012-11-21 13:31:16 debug kernel: [fe]: receive ip session from sync 10.155.33.6/50529 -> 10.155.33.189/21, 6, mac sess dir (reverse) flag 0x0  2012-11-21 13:31:16 debug kernel: [fe]: QoS: first pkt L2 session (eth0->wifi1.1) upid=1 src=0(0) dst=1  2012-11-21 13:31:16 debug kernel: [fe]: mac session(id 42) 000c:2940:6dbc->0024:d7c0:0308, flags 0xa1210/0x25000, nf->acl 0x0/0x0 nr->acl 0x0/0x0  IP session at AP1:  Id:6; Ageout:1796493; Flags:0x4220; QOS:2; L7 ID:65535; Up: 0 min 3 sec; InPol:L7Verify/2;  10.155.33.6/50529 -> 10.155.33.189/21; Proto 6; Flg:0x102; Pkts:0 Bytes:0 Parent-MAC-Sess: 42  10.155.33.189/21 -> 10.155.33.6/50529; Proto 6; Flg:0x0; Pkts:0 Bytes:0 | | |
| Test Result |  | | |
| Comment | Two APs should create IP sessions during roaming. | | |

### Interface Application Reporting Test

#### ApplicationReporting\_Function\_01

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_01 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP access wifi0.x. (Auto Mode) | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and only bind it with AP’s sub-interface wifi0.x, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.     2. Enable application ICMP for reporting.   CLI: application reporting app-id 285 enable   * + 1. Laptop1 launch ICMP ping.     2. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify ICMP session and report application traffic, which goes through wifi access sub-interface binding with SSID, with correct TLV field to HM correctly.     3. Set application reporting mode as “disable” globally at AP.     4. Clear previous http session at AP, and then launch ICMP again at laptop1.     5. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify ICMP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1's MAC | | AppType | ICMP App ID | | IntName4Client | AP wifi0.x | | PeerIntName | AP backhaul eth | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   For other applications which are not enabled for reporting, HiveOS report them as Unknown to HM.  Syslog:  Show log buffer  Dec 04 16:32:01 10.155.33.4 l7d: High resolution timer expired, start to report  Dec 04 16:32:01 10.155.33.4 l7d: A new report start: from 29 to 31, span 3  Dec 04 16:32:01 10.155.33.4 l7d: A new report file created: "08EA440EE4000423.hpr"  Dec 04 16:32:01 10.155.33.4 l7d: [rpt\_basic]: Entry: cmac 3619685128 app 285 if 1 pif 0 pt 0 ts 1354609741 size 180 app-size 3 dn-pkt 179 dn-byte 10740 up-pkt 181 up-byte 10916  Dec 04 16:32:01 10.155.33.4 l7d: A MAC entry block by app-id filter: 618  Dec 04 16:32:01 10.155.33.4 l7d: A MAC entry block by app-id filter: 365  Dec 04 16:32:01 10.155.33.4 l7d: A MAC entry block by app-id filter: 275  Dec 04 16:32:01 10.155.33.4 l7d: [rpt\_basic]: Entry: cmac 3619685128 app 0 if 1 pif 0 pt 0 ts 1354609741 size 180 app-size 3 dn-pkt 132 dn-byte 143547 up-pkt 123 up-byte 8933  Dec 04 16:32:01 10.155.33.4 l7d: Totally 47KB report files occupied  Dec 04 16:32:01 10.155.33.4 capwap: receive event application report notification capwap: eventid = 106: length = 42  Dec 04 16:32:01 10.155.33.4 capwap: CAPWAP: receive application report notification capwap event!, length:42  Dec 04 16:32:01 10.155.33.4 l7d: Send notification via capwap successfully  Dec 04 16:32:01 10.155.33.4 l7d: Receive AH\_KEVENT\_FE\_L7\_APP with 2 entries  Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as AP access sub-interface wifi0.x. | | |
| Test Result |  | | |
| Comment | 1. Repeate test case for 3 times to verify wifi sub-interface: wifi0.1, wifi0.8 and wifi0.16.  2. HiveOS Dev will provide a test tool to covert HiveOS application reporting file, which will be retrieved by HM for application reporting statistics, to readable format. Check application reporting file to make sure AP and BR can classify application and report it to HM correctly.  3. Quotes from HiveOS Yufeng Wu mail:  We currently don't report filtered traffic (traffic belonging to applications that are not enabled by CLI) to HM, which is not correct. We need to report those as UNKNOWN to HM so that they can know the total amount of traffic going through the box. I have opened the bug 21618 and checked in a fix. | | |

#### ApplicationReporting\_Function\_02

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_02 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Laptop1-----(wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP access wifi1.x. (Auto Mode) | | |
| PlatformDependence | AP: AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and only bind it with AP’s sub-interface wifi1.x, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify and report application traffic, which goes through wifi access sub-interface binding with SSID, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at AP.     6. Clear previous http session at AP, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | AP wifi1.1 | | PeerIntName | AP backhaul eth | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as AP access sub-interface wifi0.x. | | |
| Test Result |  | | |
| Comment | Repeate test case for 3 times to verify wifi sub-interface: wifi1.1, wifi1.2 (set wifi1.1 as backhaul), wifi1.3 and wifi1.16. | | |

#### ApplicationReporting\_Function\_03

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_03 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi1.2)MP(wifi1.1)---------(wifi1.1)AP(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for MP backhaul wifi1.1. (Auto Mode) | | |
| PlatformDependence | AP: AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth0 mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Create a SSID and bind it with MP wifi sub-interface, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if AP can classify and report application traffic, which goes through MP wifi1.1, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at MP.     6. Clear previous http session at MP, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. No TLV with field “IntName4Client” as MP backhaul wifi1.1.  Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. HiveOS does NOT generate application reporting file. | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_04

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_04 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth0)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP independent backhaul eth0. (Auto Mode) | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth0 as independent backhaul.  Create a SSID and bind it with AP’s wifi sub-interface, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.   CLI:  no application identification shutdown  application reporting auto   * + 1. Enable application HTTP, TCP and IP for reporting.   CLI: application reporting app-id 275 enable   * + 1. Laptop1 accesses Internet web site via http session.     2. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify and report application traffic, which goes through AP backhaul eth0, with correct TLV field to HM correctly.     3. Set application reporting mode as “disable” globally at AP.   CLI: no application reporting auto   * + 1. Clear previous http session at AP, and then launch http session again at laptop1.     2. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. No TLV with field “IntName4Client” as AP backhaul eth0.  Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as AP backhaul eth0. | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_05

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_05 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth1)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP independent backhaul eth1. (Auto Mode) | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth1 as independent backhaul.  Create a SSID and bind it with AP’s wifi sub-interface, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.   CLI:  no application identification shutdown  application reporting auto   * + 1. Enable application HTTP, TCP and IP for reporting.   CLI: application reporting app-id 275 enable   * + 1. Laptop1 accesses Internet web site via http session.     2. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify and report application traffic, which goes through AP backhaul eth1, with correct TLV field to HM correctly.     3. Set application reporting mode as “disable” globally at AP.   CLI: no application reporting auto   * + 1. Clear previous http session at AP, and then launch http session again at laptop1.     2. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. No TLV with field “IntName4Client” as AP backhaul eth1.  Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as AP backhaul eth1. | | |
| Test Result |  | | |
| Comment | In order to get test result immediately, can set application reporting interval as 60 seconds. | | |

#### ApplicationReporting\_Function\_06

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_06 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(agg0)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP aggregate backhaul agg0 (Auto Mode). | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP is managed by HM.  Bind AP eth0 and eth1 with an aggregate backhaul.  Create a SSID and bind it with AP’s wifi sub-interface, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.   CLI:  no application identification shutdown  application reporting auto   * + 1. Enable application HTTP, TCP and IP for reporting.   CLI: application reporting app-id 275 enable   * + 1. Laptop1 accesses Internet web site via http session.     2. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify and report application traffic, which goes through AP aggregate backhaul agg0, with correct TLV field to HM correctly.     3. Set application reporting mode as “disable” globally at AP.   CLI: no application reporting auto   * + 1. Clear previous http session at AP, and then launch http session again at laptop1.     2. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. No TLV with field “IntName4Client” as AP aggregate backhaul agg0.  Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as AP aggregate backhaul eth0. | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_07

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_07 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(red0)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP redundant backhaul red0. (Auto Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP is managed by HM.  Bind AP eth0 and eth1 with a redundant backhaul.  Create a SSID and bind it with AP’s wifi sub-interface, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify and report application traffic, which goes through AP redundant backhaul red0, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at AP.     6. Clear previous http session at AP, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. No TLV with field “IntName4Client” as AP redundant backhaul red0.  Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. HiveOS doesn’t generate application reporting file. | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_08

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_08 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_SW(trunk)\_\_\_\_\_(eth0)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP independent bridge-802.1q eth0. (Auto Mode) | | |
| PlatformDependence | AP: AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Set MP eth0 mode as independent bridge-802.1q  SW and MP allow all vlans.  Laptop1 connects with Internet via SW. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at MP.     2. Enable application ICMP for reporting.     3. Laptop1 launches ICMP ping.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP eth0, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at MP.     6. Clear previous http session at MP, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | ICMP APP ID | | IntName4Client | MP eth0 | | PeerIntName | MP wii1.1 | | PassThrough | 2 (unknown) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   For other applications which are not enabled for reporting, HiveOS report them as Unknown to HM.  Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. HiveOS does NOT generate application reporting file. | | |
| Test Result |  | | |
| Comment | In order to get test result immediately, can set application reporting interval as 60 seconds. | | |

#### ApplicationReporting\_Function\_09

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_09 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_SW(trunk)\_\_\_\_\_(eth1)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP independent bridge-802.1q eth1. (Auto Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Set MP eth1 mode as independent bridge-802.1q  SW and MP allow all vlans.  Laptop1 connects with Internet via SW. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP eth1, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at MP.     6. Clear previous http session at MP, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP eth1 | | PeerIntName | MP wii1.1 | | PassThrough | 2 (unknown) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   For other applications which are not enabled for reporting, HiveOS report them as Unknown to HM.  "clientMac","appType","iniName4Client","peerIntName","passthrough","radioType","timeStamp","sampleSize","appSampleSize","downlinkPacketDelta","downlinkByteDelta","uplinkPacketDelta","uplinkByteDelta"  "D0:67:E5:34:EF:13","ICMP","eth1","wifi0.1","unknown",3,"06-12-12 05:45:01",60,1,26,1560,26,1560  "D0:67:E5:34:EF:13","Unknown","eth1","wifi0.1","unknown",3,"06-12-12 05:45:01",60,1,10,780,11,1244  AP330\_L7\_2#show log buffered | in 13:46:02  2012-12-06 13:46:02 debug l7d: Send notification via capwap successfully  2012-12-06 13:46:02 warn capwap: CAPWAP: discard event message (type:6213) because CAPWAP status is not RUN or event flag is not enable  2012-12-06 13:46:02 info capwap: CAPWAP: receive application report notification capwap event!, length:42  2012-12-06 13:46:02 info capwap: receive event application report notification capwap: eventid = 106: length = 42  2012-12-06 13:46:02 info l7d: Totally 20KB report files occupied  2012-12-06 13:46:02 debug l7d: [rpt\_basic]: Entry: cmac 19770EE440 app 0 if 3 pif 16 pt 0 ts 1354772701 size 60 app-size 1 dn-pkt 73 dn-byte 3832 up-pkt 106 up-byte 15700  2012-12-06 13:46:02 debug l7d: A MAC entry block by app-id filter: 618  2012-12-06 13:46:02 debug l7d: A MAC entry block by app-id filter: 645  2012-12-06 13:46:02 debug last message repeated 14 times  2012-12-06 13:46:02 debug l7d: A MAC entry block by app-id filter: 306  2012-12-06 13:46:02 debug l7d: [rpt\_basic]: Entry: cmac D067E534EF13 app 0 if 5 pif 16 pt 2 ts 1354772701 size 60 app-size 1 dn-pkt 10 dn-byte 780 up-pkt 11 up-byte 1244  2012-12-06 13:46:02 debug l7d: [rpt\_basic]: Entry: cmac D067E534EF13 app 285 if 5 pif 16 pt 2 ts 1354772701 size 60 app-size 1 dn-pkt 26 dn-byte 1560 up-pkt 26 up-byte 1560  2012-12-06 13:46:02 debug l7d: A MAC entry block by app-id filter: 618  2012-12-06 13:46:02 debug l7d: [rpt\_basic]: New MAC entry list with timestamp 1354772701, samplesize 60  2012-12-06 13:46:02 info l7d: A new report file created: "0019770EE4400180.hpr"  2012-12-06 13:46:02 debug l7d: A new report start: from 45:01 to 46:01, span 1  2012-12-06 13:46:02 debug l7d: [rpt\_basic]: On reporting thread: c\_intv 1, span 1, lead\_tm 0, type 1  2012-12-06 13:46:02 debug l7d: High resolution timer expired, start to report  Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. HiveOS does NOT generate application reporting file. | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_10

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_10 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_SW(trunk)\_\_\_\_\_(agg0)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP aggregate bridge-802.1q agg0. (Auto Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Bind MP eth0 and eth1 with aggregate bridge-802.1q agg0  SW and MP allow all vlans.  Laptop1 connects with Internet via SW. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP aggregate bridge-802.1q agg0, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at MP.     6. Clear previous http session at MP, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP agg0 | | PeerIntName | MP wii1.1 | | PassThrough | 2 (unknown) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as MP agg0. | | |
| Test Result |  | | |
| Comment | Dec 10. 2012:  According to HiveOS Dev response, red0/agg0 as bridge-802.1q/bridge-access is not supported cunnently. | | |

#### ApplicationReporting\_Function\_11

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_11 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_SW(trunk)\_\_\_\_\_(red0)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP redundant bridge-802.1q red0. (Auto Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Bind MP eth0 and eth1 with redundant bridge-802.1q red0  SW and MP allow all vlans.  Laptop1 connects with Internet via SW. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP redundant bridge-802.1q red0, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at MP.     6. Clear previous http session at MP, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP red0 | | PeerIntName | MP wii1.1 | | PassThrough | 2 (unknown) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as MP red0. | | |
| Test Result |  | | |
| Comment | Dec 10. 2012:  According to HiveOS Dev response, red0/agg0 as bridge-802.1q/bridge-access is not supported cunnently. | | |

#### ApplicationReporting\_Function\_12

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_12 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_(eth0)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP independent bridge-access eth0 (Auto Mode). | | |
| PlatformDependence | AP: AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Set MP eth0 mode as independent bridge-access  Laptop1 connects with Internet via MP. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP independent bridge-access eth0, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at MP.     6. Clear previous http session at MP, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP eth0 | | PeerIntName | MP wii1.1 | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as MP eth0. | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_13

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_13 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_(eth1)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP independent bridge-access eth1. (Auto Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Set MP eth1 mode as independent bridge-access  Laptop1 connects with Internet via MP. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP bridge-access eth1, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at MP.     6. Clear previous http session at MP, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP eth1 | | PeerIntName | MP wii1.1 | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as MP eth1. | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_14

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_14 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_(agg0)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP aggregate bridge-access agg0. (Auto Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Bind MP eth0 and eth1 with aggregate bridge-access agg0.  Laptop1 connects with Internet via MP. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP aggregate bridge-access agg0, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at MP.     6. Clear previous http session at MP, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP agg0 | | PeerIntName | MP wii1.1 | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as MP eth1. | | |
| Test Result |  | | |
| Comment | Dec 10. 2012:  According to HiveOS Dev response, red0/agg0 as bridge-802.1q/bridge-access is not supported cunnently. | | |

#### ApplicationReporting\_Function\_15

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_15 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_(red0)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP redundant bridge-access red0. (Auto Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Bind MP eth0 and eth1 with redundant bridge-access agg0.  Laptop1 connects with Internet via MP. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP redundant bridge-access red0, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at MP.     6. Clear previous http session at MP, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP red0 | | PeerIntName | MP wii1.1 | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as MP eth1. | | |
| Test Result |  | | |
| Comment | Dec 10. 2012:  According to HiveOS Dev response, red0/agg0 as bridge-802.1q/bridge-access is not supported cunnently. | | |

#### ApplicationReporting\_Function\_16

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_16 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0.x)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for BR access wifi0.x. (Auto Mode) | | |
| PlatformDependence | BR: BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth0 mode as wan, and wifi0.x as access.  Create a SSID and bind it with BR sub-interface wifi0.x, which is set as access mode.  Laptop1 connects with SSID.  The Laptop1’s GW should be the one of BR mgt0.x. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at BR.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify http session and report application traffic, which goes through wifi access sub-interface binding with SSID, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at BR.     6. Clear previous http session at BR, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | BR wifi0.x | | PeerIntName | Interface as Laptop1’s GW | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as BR access sub-interface wifi0.x. | | |
| Test Result |  | | |
| Comment | Repeate test case for 3 times to verify wifi sub-interface: wifi0.1, wifi0.2 (wifi0.1 as backhaul), wifi0.8 and wifi0.16. | | |

#### ApplicationReporting\_Function\_17

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_17 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi1.x)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for BR access wifi1.x. (Auto Mode) | | |
| PlatformDependence | BR: BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth0 mode as wan, and wifi1.x as access.  Create a SSID and bind it with BR sub-interface wifi0.x, which is set as access mode.  Laptop1 connects with SSID.  The Laptop1’s GW should be the one of BR mgt0.x. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at BR.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify http session and report application traffic, which goes through wifi access sub-interface binding with SSID, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at BR.     6. Clear previous http session at BR, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application repoting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | BR wifi1.x | | PeerIntName | Interface as Laptop1’s GW | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as BR access sub-interface wifi0.x. | | |
| Test Result |  | | |
| Comment | Repeate test case for 3 times to verify wifi sub-interface: wifi1.1, wifi1.2 (wifi1.1 as backhaul), wifi1.8 and wifi1.16. | | |

#### ApplicationReporting\_Function\_18

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_18 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi1.2)MP(wifi1.1)---------(wifix.1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for BR backhaul wifix.1. (Auto Mode) | | |
| PlatformDependence | BR: BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR and MP are managed by HM.  Set MP wifi1.1 and BR wifix.1 as backhaul. MP connects with BR via wifi1.1 backhaul interface.  Create a SSID and bind it with MP wifi sub-interface, which is set as access mode.  Laptop1 connects with SSID to access Internet.  The Laptop1’s GW should be the one of BR mgt0.x. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify and report application traffic, which goes through BR backhaul wifix.1, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at BR.     6. Clear previous http session at BR, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if AP can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. No TLV with field “IntName4Client” as BR backhaul wifix.1.  Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. HiveOS does NOT generate application reporting file. | | |
| Test Result |  | | |
| Comment | For BR200-WP, only can set wifi0.1 as wifi backhaul interface.  For BRAP330 and BRAP350, can set wifi0.1 and wifi1.1 as wifi backhaul. | | |

#### ApplicationReporting\_Function\_19

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_19 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_SW(trunk)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for BR bridge-802.1q ethx interface. (Auto Mode) | | |
| PlatformDependence | BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth0 as WAN mode, and eth1 as bridge-802.1q mode.  BR eth1 and SW trunk interface allow all vlans.  Laptop1 connects with Internet via SW.  Laptop1’s GW should be one of BR mgt0.x sub-interfaces. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at BR.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify and report application traffic, which goes through BR bridge-802.1q eth1, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at BR.     6. Clear previous http session at BR, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | BR eth1 | | PeerIntName | Interface as Laptop1’s GW | | PassThrough | 2 (unknown) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as BR eth1. | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_20

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_20 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for BR bridge-access ethx interface. (Auto Mode) | | |
| PlatformDependence | BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth0 as wan mode, and eth1 as bridge-access mode.  Laptop1 connects with Internet via BR.  Laptop1’s GW should be one of BR mgt0.x sub-interfaces. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at BR.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify and report application traffic, which goes through BR bridge-access eth1, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at BR.     6. Clear previous http session at BR, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | BR eth1 | | PeerIntName | Interface as Laptop1’s GW | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “IntName4Client” as BR eth1. | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_21

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_21 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for BR subinterfaces mgt0.x. (Auto Mode) | | |
| PlatformDependence | BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth0 as wan mode, and eth1 as bridge-access mode.  Laptop1 connects with Internet via BR.  Laptop1’s GW should be one of BR mgt0.x sub-interfaces. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at BR.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify and report application traffic, which goes through BR mgt0.x, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at BR.     6. Clear previous http session at BR, and then launch http session again at laptop1.     7. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify HTTP session and report application traffic with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. No TLV with field “IntName4Client” as BR mgt0.x.  Step 5. Set application reporting mode as “disable” globally successfully.  Step 7. HiveOS does NOT generate application reporting file. | | |
| Test Result |  | | |
| Comment | **Repeate test case for 3 times to verify sub-interface: mgt0, mgt0.1 and mgt0.16.**  **Check functional requirement description.** | | |

#### ApplicationReporting\_Function\_22

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_22 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP access wifi0.x. (Enable Mode) | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and only bind it with AP’s sub-interface wifi0.x, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at AP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify http session and report application traffic, which goes through wifi access sub-interface binding with SSID, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1's MAC | | AppType | HTTP APP ID | | IntName4Client | AP wifi0.x | | PeerIntName | AP backhaul eth | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Syslog:  Show log buffer  2012-12-10 19:02:02 debug l7d: [rpt\_basic]: Entry: cmac 2477304AEB0 app 0 if 1 pif 0 pt 0 radio 1 ts 1355137261 size 60 app-size 1 dn-pkt 10 dn-byte 792 up-pkt 14 up-byte 1582  2012-12-10 19:02:02 debug l7d: A MAC entry block by app-id filter: 618  2012-12-10 19:02:02 debug l7d: [rpt\_basic]: Entry: cmac 2477304AEB0 app 285 if 1 pif 0 pt 0 radio 1 ts 1355137261 size 60 app-size 1 dn-pkt 29 dn-byte 1740 up-pkt 29 up-byte 1740  2012-12-10 19:02:02 debug l7d: A MAC entry block by app-id filter: 645  2012-12-10 19:02:02 debug l7d: [rpt\_basic]: Entry: cmac FFFFFFFFFFFF app 0 if 0 pif 1 pt 3 radio 3 ts 1355137261 size 60 app-size 1 dn-pkt 3 dn-byte 234 up-pkt 0 up-byte 0  2012-12-10 19:02:02 debug l7d: A MAC entry block by app-id filter: 645  2012-12-10 19:02:02 debug l7d: [rpt\_basic]: Entry: cmac 80711F32DF4B app 0 if 0 pif 0 pt 3 radio 3 ts 1355137261 size 60 app-size 1 dn-pkt 11 dn-byte 1891 up-pkt 0 up-byte 0  2012-12-10 19:02:02 debug last message repeated 5 times  2012-12-10 19:02:02 debug l7d: A MAC entry block by app-id filter: 306  2012-12-10 19:02:02 debug l7d: A MAC entry block by app-id filter: 645  2012-12-10 19:02:02 debug l7d: [rpt\_basic]: Entry: cmac 80F62E3E9005 app 0 if 0 pif 1 pt 3 radio 3 ts 1355137261 size 60 app-size 1 dn-pkt 11 dn-byte 1348 up-pkt 10 up-byte 792  2012-12-10 19:02:02 debug l7d: A MAC entry block by app-id filter: 618  2012-12-10 19:02:02 debug l7d: [rpt\_basic]: Entry: cmac 80F62E3E9005 app 285 if 0 pif 1 pt 3 radio 3 ts 1355137261 size 60 app-size 1 dn-pkt 29 dn-byte 1740 up-pkt 29 up-byte 1740  2012-12-10 19:02:02 debug l7d: [rpt\_basic]: New GMT hour 11 min 1 sec 1 ts 1355137261 idx\_min 2 ts\_hour 11 ts\_min 1  2012-12-10 19:02:02 debug l7d: [rpt\_basic]: New MAC entry list with timestamp 1355137261, samplesize 60  2012-12-10 19:02:02 info l7d: A new report file created: "08EA440EE4000024.hpr"  2012-12-10 19:02:02 debug l7d: A new report start: from 01:01 to 02:01, span 1  2012-12-10 19:02:02 debug l7d: [rpt\_basic]: On reporting thread: c\_intv 1, span 1, lead\_tm 0, type 1  2012-12-10 19:02:02 debug l7d: High resolution timer expired, start to report | | |
| Test Result |  | | |
| Comment | 1. Repeate test case for 3 times to verify wifi sub-interface: wifi0.1, wifi0.8 and wifi0.16.  2. HiveOS Dev will provide a test tool to covert HiveOS application reporting file, which will be retrieved by HM for application reporting statistics, to readable format. Check application reporting file to make sure AP and BR can classify application and report it to HM correctly. | | |

#### ApplicationReporting\_Function\_23

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_23 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Laptop1-----(wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP access wifi1.x. (Enable Mode) | | |
| PlatformDependence | AP: AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and only bind it with AP’s sub-interface wifi1.x, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at AP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify and report application traffic, which goes through wifi access sub-interface binding with SSID, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | AP wifi1.1 | | PeerIntName | AP backhaul eth | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment | Repeate test case for 3 times to verify wifi sub-interface: wifi1.1, wifi1.2 (set wifi1.1 as backhaul), wifi1.8 and wifi1.16. | | |

#### ApplicationReporting\_Function\_24

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_24 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi1.2)MP(wifi1.1)---------(wifi1.1)AP(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for MP backhaul wifi1.1. (Enable Mode) | | |
| PlatformDependence | AP: AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth0 mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Create a SSID and bind it with MP wifi sub-interface, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if AP can classify and report application traffic, which goes through MP wifi1.1, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1 GW MAC | | AppType | HTTP APP ID | | IntName4Client | MP wifi1.1 | | PeerIntName | MP wifi sub-interface | | PassThrough | 1 (yes) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_25

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_25 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth0)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP independent backhaul eth0. (Enable Mode) | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth0 as independent backhaul.  Create a SSID and bind it with AP’s wifi sub-interface, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at AP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify and report application traffic, which goes through AP backhaul eth0, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1's MAC | | AppType | HTTP APP ID | | IntName4Client | AP eth0 | | PeerIntName | AP wifi sub-interface | | PassThrough | 3 (backhaul) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_26

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_26 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth1)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP independent backhaul eth1. (Enable Mode) | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth1 as independent backhaul.  Create a SSID and bind it with AP’s wifi sub-interface, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at AP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify and report application traffic, which goes through AP backhaul eth1, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1 GW MAC | | AppType | HTTP APP ID | | IntName4Client | AP eth1 | | PeerIntName | AP wifi sub-interface | | PassThrough | 3 (backhaul) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_27

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_27 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(agg0)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP aggregate backhaul agg0 (Enable Mode). | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP is managed by HM.  Bind AP eth0 and eth1 with an aggregate backhaul.  Create a SSID and bind it with AP’s wifi sub-interface, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at AP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify and report application traffic, which goes through AP aggregate backhaul agg0, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1 GW MAC | | AppType | HTTP APP ID | | IntName4Client | AP agg0 | | PeerIntName | AP wifi sub-interface | | PassThrough | 3 (backhaul) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_28

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_28 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(red0)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP redundant backhaul red0. (Enable Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP is managed by HM.  Bind AP eth0 and eth1 with a redundant backhaul.  Create a SSID and bind it with AP’s wifi sub-interface, which is set as access mode.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at AP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can classify and report application traffic, which goes through AP redundant backhaul red0, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1 GW MAC | | AppType | HTTP APP ID | | IntName4Client | AP red0 | | PeerIntName | AP wifi sub-interface | | PassThrough | 3 (backhaul) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_29

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_29 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_SW(trunk)\_\_\_\_\_(eth0)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP independent bridge-802.1q eth0. (Enable Mode) | | |
| PlatformDependence | AP: AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Set MP eth0 mode as independent bridge-802.1q  SW and MP allow all vlans.  Laptop1 connects with Internet via SW. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP eth0, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP eth0 | | PeerIntName | MP wii1.1 | | PassThrough | 2 (unknown) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Syslog:  AP330\_L7\_2#show log buffered | in 13:51:02  2012-12-07 13:51:02 debug l7d: Send notification via capwap successfully  2012-12-07 13:51:02 info capwap: CAPWAP: receive application report notification capwap event!, length:42  2012-12-07 13:51:02 info capwap: receive event application report notification capwap: eventid = 106: length = 42  2012-12-07 13:51:02 info l7d: Totally 14KB report files occupied  2012-12-07 13:51:02 debug l7d: [rpt\_basic]: Entry: cmac 19770EE440 app 0 if 3 pif 16 pt 0 ts 1354859341 size 120 app-size 2 dn-pkt 186 dn-byte 9384 up-pkt 202 up-byte 33912  2012-12-07 13:51:02 debug l7d: A MAC entry block by app-id filter: 618  2012-12-07 13:51:02 debug l7d: A MAC entry block by app-id filter: 645  2012-12-07 13:51:02 debug last message repeated 13 times  2012-12-07 13:51:02 debug l7d: A MAC entry block by app-id filter: 306  2012-12-07 13:51:02 debug l7d: [rpt\_basic]: Entry: cmac 19778C8200 app 0 if 16 pif 3 pt 3 ts 1354859341 size 120 app-size 2 dn-pkt 34 dn-byte 2506 up-pkt 22 up-byte 1834  2012-12-07 13:51:02 debug l7d: A MAC entry block by app-id filter: 645  2012-12-07 13:51:02 debug last message repeated 13 times  2012-12-07 13:51:02 debug l7d: A MAC entry block by app-id filter: 306  2012-12-07 13:51:02 debug l7d: [rpt\_basic]: Entry: cmac D067E5437D2D app 0 if 16 pif 3 pt 3 ts 1354859341 size 120 app-size 2 dn-pkt 168 dn-byte 31406 up-pkt 164 up-byte 7550  2012-12-07 13:51:02 debug l7d: A MAC entry block by app-id filter: 618  2012-12-07 13:51:02 debug l7d: [rpt\_basic]: Entry: cmac D067E534EF13 app 0 if 5 pif 16 pt 0 ts 1354859341 size 120 app-size 2 dn-pkt 20 dn-byte 1560 up-pkt 22 up-byte 2488  2012-12-07 13:51:02 debug l7d: [rpt\_basic]: Entry: cmac D067E534EF13 app 285 if 5 pif 16 pt 0 ts 1354859341 size 120 app-size 1 dn-pkt 35 dn-byte 2100 up-pkt 35 up-byte 2100  2012-12-07 13:51:02 debug l7d: A MAC entry block by app-id filter: 618  2012-12-07 13:51:02 debug l7d: [rpt\_basic]: Entry: cmac 19778C8200 app 0 if 16 pif 5 pt 3 ts 1354859341 size 120 app-size 2 dn-pkt 22 dn-byte 2488 up-pkt 20 up-byte 1560  2012-12-07 13:51:02 debug l7d: [rpt\_basic]: Entry: cmac 19778C8200 app 285 if 16 pif 5 pt 3 ts 1354859341 size 120 app-size 1 dn-pkt 35 dn-byte 2100 up-pkt 35 up-byte 2100  2012-12-07 13:51:02 debug l7d: A MAC entry block by app-id filter: 618  2012-12-07 13:51:02 debug l7d: [rpt\_basic]: New MAC entry list with timestamp 1354859341, samplesize 120  2012-12-07 13:51:02 info l7d: A new report file created: "0019770EE4400083.hpr"  2012-12-07 13:51:02 debug l7d: A new report start: from 49:01 to 51:01, span 2  2012-12-07 13:51:02 debug l7d: [rpt\_basic]: On reporting thread: c\_intv 2, span 2, lead\_tm 0, type 1  2012-12-07 13:51:02 debug l7d: High resolution timer expired, start to report  2012-12-07 13:51:02 info ah\_cli: admin:<show log buffered > | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_30

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_30 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_SW(trunk)\_\_\_\_\_(eth1)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP independent bridge-802.1q eth1. (Enable Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Set MP eth1 mode as independent bridge-802.1q  SW and MP allow all vlans.  Laptop1 connects with Internet via SW. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP eth1, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP eth1 | | PeerIntName | MP wii1.1 | | PassThrough | 2 (unknown) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_31

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_31 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_SW(trunk)\_\_\_\_\_(agg0)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP aggregate bridge-802.1q agg0. (Enable Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Bind MP eth0 and eth1 with aggregate bridge-802.1q agg0  SW and MP allow all vlans.  Laptop1 connects with Internet via SW. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP aggregate bridge-802.1q agg0, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP agg0 | | PeerIntName | MP wii1.1 | | PassThrough | 2 (unknown) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment | Dec 10. 2012:  According to HiveOS Dev response, red0/agg0 as bridge-802.1q/bridge-access is not supported cunnently. | | |

#### ApplicationReporting\_Function\_32

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_32 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_SW(trunk)\_\_\_\_\_(red0)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP redundant bridge-802.1q red0. (Enable Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Bind MP eth0 and eth1 with redundant bridge-802.1q red0  SW and MP allow all vlans.  Laptop1 connects with Internet via SW. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP redundant bridge-802.1q red0, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP red0 | | PeerIntName | MP wii1.1 | | PassThrough | 2 (unknown) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment | In order to get correct test result: set reporting interval as 120s, and only send ping in 1 minute slot.  Dec 10. 2012:  According to HiveOS Dev response, red0/agg0 as bridge-802.1q/bridge-access is not supported cunnently. | | |

#### ApplicationReporting\_Function\_33

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_33 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_(eth0)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP independent bridge-access eth0 (Enable Mode). | | |
| PlatformDependence | AP: AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Set MP eth0 mode as independent bridge-access  Laptop1 connects with Internet via MP. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP independent bridge-access eth0, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP eth0 | | PeerIntName | MP wii1.1 | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   AP330\_L7\_2#show log buffered | in 13:33:02  2012-12-07 13:33:02 debug l7d: Send notification via capwap successfully  2012-12-07 13:33:02 info capwap: CAPWAP: receive application report notification capwap event!, length:42  2012-12-07 13:33:02 info capwap: receive event application report notification capwap: eventid = 106: length = 42  2012-12-07 13:33:02 info l7d: Totally 13KB report files occupied  2012-12-07 13:33:02 debug l7d: [rpt\_basic]: Entry: cmac 19770EE440 app 0 if 3 pif 16 pt 0 ts 1354858261 size 120 app-size 2 dn-pkt 168 dn-byte 8550 up-pkt 183 up-byte 30456  2012-12-07 13:33:02 debug l7d: A MAC entry block by app-id filter: 618  2012-12-07 13:33:02 debug l7d: A MAC entry block by app-id filter: 645  2012-12-07 13:33:02 debug last message repeated 13 times  2012-12-07 13:33:02 debug l7d: A MAC entry block by app-id filter: 306  2012-12-07 13:33:02 debug l7d: [rpt\_basic]: Entry: cmac 19778C8200 app 0 if 16 pif 3 pt 3 ts 1354858261 size 120 app-size 2 dn-pkt 56 dn-byte 4994 up-pkt 42 up-byte 3394  2012-12-07 13:33:02 debug l7d: [rpt\_basic]: Entry: cmac 19778C8200 app 285 if 16 pif 3 pt 3 ts 1354858261 size 120 app-size 1 dn-pkt 28 dn-byte 1680 up-pkt 28 up-byte 1680  2012-12-07 13:33:02 debug l7d: A MAC entry block by app-id filter: 618  2012-12-07 13:33:02 debug l7d: A MAC entry block by app-id filter: 645  2012-12-07 13:33:02 debug last message repeated 13 times  2012-12-07 13:33:02 debug l7d: A MAC entry block by app-id filter: 306  2012-12-07 13:33:02 debug l7d: [rpt\_basic]: Entry: cmac D067E5437D2D app 0 if 16 pif 3 pt 3 ts 1354858261 size 120 app-size 2 dn-pkt 149 dn-byte 27950 up-pkt 146 up-byte 6716  2012-12-07 13:33:02 debug l7d: A MAC entry block by app-id filter: 618  2012-12-07 13:33:02 debug l7d: [rpt\_basic]: Entry: cmac D067E534EF13 app 0 if 3 pif 16 pt 0 ts 1354858261 size 120 app-size 2 dn-pkt 20 dn-byte 1560 up-pkt 22 up-byte 2488  2012-12-07 13:33:02 debug l7d: [rpt\_basic]: Entry: cmac D067E534EF13 app 285 if 3 pif 16 pt 0 ts 1354858261 size 120 app-size 1 dn-pkt 28 dn-byte 1680 up-pkt 28 up-byte 1680  2012-12-07 13:33:02 debug l7d: A MAC entry block by app-id filter: 618  2012-12-07 13:33:02 debug l7d: [rpt\_basic]: New MAC entry list with timestamp 1354858261, samplesize 120  2012-12-07 13:33:02 info l7d: A new report file created: "0019770EE4400074.hpr"  2012-12-07 13:33:02 debug l7d: A new report start: from 31:01 to 33:01, span 2  2012-12-07 13:33:02 debug l7d: [rpt\_basic]: On reporting thread: c\_intv 2, span 2, lead\_tm 0, type 1  2012-12-07 13:33:02 debug l7d: High resolution timer expired, start to report | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_34

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_34 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_(eth1)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP independent bridge-access eth1. (Enable Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Set MP eth1 mode as independent bridge-access  Laptop1 connects with Internet via MP. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP bridge-access eth1, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP eth1 | | PeerIntName | MP wii1.1 | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_35

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_35 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_(agg0)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP aggregate bridge-access agg0. (Enable Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Bind MP eth0 and eth1 with aggregate bridge-access agg0.  Laptop1 connects with Internet via MP. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP aggregate bridge-access agg0, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP agg0 | | PeerIntName | MP wii1.1 | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_36

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_36 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_(red0)MP(wifi1.1)--------(wifi1.1)AP(eth)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for AP redundant bridge-access red0. (Enable Mode) | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP/AP wifi1.1 and AP eth interface mode as backhaul. MP connects with AP via wifi1.1 backhaul interface.  Bind MP eth0 and eth1 with redundant bridge-access agg0.  Laptop1 connects with Internet via MP. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at MP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in MP reporting file. Check if MP can classify and report application traffic, which goes through MP redundant bridge-access red0, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | MP red0 | | PeerIntName | MP wii1.1 | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_37

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_37 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0.x)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for BR access wifi0.x. (Enable Mode) | | |
| PlatformDependence | BR: BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth0 mode as wan, and wifi0.x as access.  Create a SSID and bind it with BR sub-interface wifi0.x, which is set as access mode.  Laptop1 connects with SSID.  The Laptop1’s GW should be the one of BR mgt0.x. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at BR.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify http session and report application traffic, which goes through wifi access sub-interface binding with SSID, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | BR wifi0.x | | PeerIntName | Interface as Laptop1’s GW | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment | Repeate test case for 3 times to verify wifi sub-interface: wifi0.1, wifi0.2 (wifi0.1 as backhaul), wifi0.8 and wifi0.16. | | |

#### ApplicationReporting\_Function\_38

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_38 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi1.x)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for BR access wifi1.x. (Enable Mode) | | |
| PlatformDependence | BR: BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth0 mode as wan, and wifi1.x as access.  Create a SSID and bind it with BR sub-interface wifi0.x, which is set as access mode.  Laptop1 connects with SSID.  The Laptop1’s GW should be the one of BR mgt0.x. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at BR.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify http session and report application traffic, which goes through wifi access sub-interface binding with SSID, with correct TLV field to HM correctly.     5. Set application reporting mode as “disable” globally at BR. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application repoting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | BR wifi1.x | | PeerIntName | Interface as Laptop1’s GW | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment | Repeate test case for 3 times to verify wifi sub-interface: wifi1.1, wifi1.2 (wifi1.1 as backhaul), wifi1.8 and wifi1.16. | | |

#### ApplicationReporting\_Function\_39

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_39 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi1.2)MP(wifi1.1)---------(wifix.1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for BR backhaul wifix.1. (Enable Mode) | | |
| PlatformDependence | BR: BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR and MP are managed by HM.  Set MP wifi1.1 and BR wifix.1 as backhaul. MP connects with BR via wifi1.1 backhaul interface.  Create a SSID and bind it with MP wifi sub-interface, which is set as access mode.  Laptop1 connects with SSID to access Internet.  The Laptop1’s GW should be the one of BR mgt0.x. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at AP.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify and report application traffic, which goes through BR backhaul wifix.1, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | BR wifix.1 | | PeerIntName | Interface as Laptop1’s GW | | PassThrough | 1 (yes) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment | For BR200-WP, only can set wifi0.1 as wifi backhaul interface.  For BRAP330 and BRAP350, can set wifi0.1 and wifi1.1 as wifi backhaul. | | |

#### ApplicationReporting\_Function\_40

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_40 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_SW(trunk)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for BR bridge-802.1q ethx interface. (Enable Mode) | | |
| PlatformDependence | BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth0 as WAN mode, and eth1 as bridge-802.1q mode.  BR eth1 and SW trunk interface allow all vlans.  Laptop1 connects with Internet via SW.  Laptop1’s GW should be one of BR mgt0.x sub-interfaces. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at BR.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify and report application traffic, which goes through BR bridge-802.1q eth1, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 10. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | BR eth1 | | PeerIntName | Interface as Laptop1’s GW | | PassThrough | 2 (unknown) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_41

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_41 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for BR bridge-access ethx interface. (Enable Mode) | | |
| PlatformDependence | BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth0 as wan mode, and eth1 as bridge-access mode.  Laptop1 connects with Internet via BR.  Laptop1’s GW should be one of BR mgt0.x sub-interfaces. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at BR.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify and report application traffic, which goes through BR bridge-access eth1, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application reporting file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | HTTP APP ID | | IntName4Client | BR eth1 | | PeerIntName | Interface as Laptop1’s GW | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_42

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_42 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Laptop1\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_SW\_\_\_\_\_HM  |  |  Internet | | |
| Description | The application reporting verification for BR subinterfaces mgt0.x. (Enable Mode) | | |
| PlatformDependence | BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth0 as wan mode, and eth1 as bridge-access mode.  Laptop1 connects with Internet via BR.  Laptop1’s GW should be one of BR mgt0.x sub-interfaces. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “enable” globally at BR.     2. Enable application HTTP, TCP and IP for reporting.     3. Laptop1 accesses Internet web site via http session.     4. Making use of test tool provided by Dev, search corresponding TLV in BR reporting file. Check if BR can classify and report application traffic, which goes through BR mgt0.x, with correct TLV field to HM correctly. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode as “enable” globally successfully.  Step 2. Enable application HTTP, TCP and IP for reporting successfully.  Step 4. Check TLV fields in application file:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | BR mgt0.x MAC | | AppType | HTTP APP ID | | IntName4Client | BR mgt0.x | | PeerIntName | BR eth1 | | PassThrough | 3 (backhaul) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment | Repeate test case for 3 times to verify wifi sub-interface: mgt0.1, mgt0.8 and mgt0.16. | | |

### Application Reporting Accuracy Test

#### ApplicationReporting\_Function\_43

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_43 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  FTP Server | | |
| Description | The application reporting accuracy verification for applications which have a “Monitored” state. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP is managed by HM.  Set AP eth0 as backhaul mode.  Create a SSID and bind it with AP’s wifi interface wifi0.x or wifi1.x.  Laptop1 connects with Internet via AP. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.     2. ~~Keep default settings of application reporting for every individual interface at AP~~     3. Enable application FTP, TCP and IP for reporting.     4. Launch FTP file download at laptop1.     5. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can report number of application packet and byte correctly to HM. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. The default mode of application reporting at interfaces is “auto”.  Step 3. Enable application FTP, TCP and IP for reporting successfully.  Step 5. Check TLV fields: ClientMAC, AppType, TimeStamp, SampleSize, AppSampleSize, DownLinkPacketDelta, DownLinkByteDelta, UpLinkPacketDelta, UpLinkByteDelta   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | AP wifi sub-interface’s MAC | | AppType | FTP APP ID | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   Based on Test result, there are 2 IP sessions for FTP control and FTP data respectively, so it includes 2 reporting entry in reporting files:  "clientMac","appType","iniName4Client","peerIntName","passthrough","radioType","timeStamp","sampleSize","appSampleSize","downlinkPacketDelta","downlinkByteDelta","uplinkPacketDelta","uplinkByteDelta"  "24:77:03:04:AE:B0","FTP Control","wifi0.1","eth0","no",1,"05-12-12 08:59:01",180,3,29,2085,40,1800  "24:77:03:04:AE:B0","FTP Data","wifi0.1","eth0","no",1,"05-12-12 08:59:01",180,1,206,215326,114,4788, | | |
| Test Result |  | | |
| Comment | For some applications like FTP, there is a middle state, “Monitored”, before it can be classified.  These applications include more than 1 ip-session. Make sure HiveOS reports right value of application reporting TLV fields to HM. | | |

#### ApplicationReporting\_Function\_44

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_44 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  FTP Server1 and FTP server2 | | |
| Description | Verify HiveOS can summary data for same kind of application into same one application reporting TLV. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP is managed by HM.  Set AP eth0 as backhaul mode.  Create a SSID and bind it with AP’s wifi interface wifi0.x or wifi1.x.  Laptop1 connects with Internet via AP. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.     2. Keep default settings of application reporting for every individual interface at AP     3. Enable application FTP, TCP and IP for reporting.     4. Launch FTP file download from FTP server1 and server2.     5. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can report number of application packet and byte correctly to HM. | | |
| Expect result | Step 1. Turn on L7 engine and set application reporting mode globally successfully.  Step 2. The default mode of application reporting at interfaces is “auto”.  Step 3. Enable application FTP, TCP and IP for reporting successfully.  Step 5. Check TLV fields: ClientMAC, AppType, TimeStamp, SampleSize, AppSampleSize, DownLinkPacketDelta, DownLinkByteDelta, UpLinkPacketDelta, UpLinkByteDelta   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | AP wifi sub-interface’s MAC | | AppType | FTP APP ID | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_45

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_45 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | (eth1)AP(eth0)\_\_\_\_SW\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | The application reporting accuracy verification at maximum interface data rate. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP is managed by HM.  Set AP eth0 mode as backhaul and eth1 as bridge-access.  IXIA connects with AP eth0. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.     2. Keep default settings of application reporting for every individual interface at AP     3. Enable application FTP, TCP and IP for reporting.     4. IXIA simulates FTP server and client.     5. Launch FTP file download at data rate 1Gbps for 5 minute.     6. Making use of test tool provided by Dev, search corresponding TLV in AP reporting file. Check if AP can report number of application packet and byte correctly to HM. | | |
| Expect result | Step 6. Check TLV fields: ClientMAC, AppType, TimeStamp, SampleSize, AppSampleSize, DownLinkPacketDelta, DownLinkByteDelta, UpLinkPacketDelta, UpLinkByteDelta   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | IXIA MAC1 as FTP Client | | AppType | FTP APP ID | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | Sample size in seconds (multiple of 60 seconds interval) | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment | AP320/340 base on 64bit for Packet/Byte record, other platforms base on 32bit.  HiveOS gets delta packet/byte number from kernel every minute, the max number is 4294967296. | | |

#### ApplicationReporting\_Function\_46

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_46 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Verify HiveOS device layered application reporting. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR.     2. ~~Keep default settings of application reporting for every individual interface at AP and BR~~     3. Enable application FTP, TCP and IP for reporting with CSV format.     4. Connect FTP server from Laptop1.     5. Check if AP can classify FTP and report it as FTP to HM correctly.     6. Check if BR can classify FTP and report it as FTP to HM correctly.     7. Clear FTP session, and then disable application FTP for reporting.     8. Launch FTP session again, check if AP and BR can classify this session and report it as TCP to HM correctly.     9. Clear FTPsession again, and then disable application TCP for reporting.     10. Launch FTP session again, check if AP and BR can classify this session and report it as IP to HM correctly.     11. Clear FTPsession again, and then disable application IP for reporting.     12. Launch FTP session again. | | |
| Expect result | Step 3. Enable application for reporting with CSV format successfully.  Step 5. AP can classify YouTube and report it as YouTube to HM correctly.  Step 6. BR can classify YouTube and report it as YouTube to HM correctly.  Step 8. AP and BR can classify session and report it as HTTP to HM correctly.  Step 10. AP and BR can classify session and report it as TCP to HM correctly.  Step 12 AP and BR can classify session and report it as IP to HM correctly.  Step 14. AP and BR can does NOT report application to HM. | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_47

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_47 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Check HiveOS device reports layered application correctly. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR.     2. Keep default settings of application reporting for every individual interface at AP and BR     3. Enable application YouTube, and disable application HTTP, TCP and IP for reporting.     4. Launch YouTube session, check if AP and BR can classify YouTube and report it to HM correctly. | | |
| Expect result | Step 4. AP and BR can classify session as YouTube and report it to HM correctly. | | |
| Test Result |  | | |
| Comment | In order to simplify test case, replace YouTube with FTP. | | |

#### ApplicationReporting\_Function\_58

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_58 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or  Laptop1-----(ethx)BR(WAN)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Verify accuracy of HiveOS hourly application report. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID or with BR’s ethx. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR.     2. Enable application ICMP for reporting.     3. Launch ICMP session at laptop1, capture live traffic for more than 1 hour.     4. After 1 hour, check HiveOS hourly application reporting file. | | |
| Expect result | Step 4. The data in AP/BR HiveOS hourly application reporting file should be consistent with captured live traffic.  For AP:   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1's MAC | | AppType | ICMP App ID | | IntName4Client | AP wifix | | PeerIntName | AP backhaul eth | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | 3600 | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report |   For BR   |  |  | | --- | --- | | TLV Filed | Expected Value | | ClientMAC | Laptop1’s MAC | | AppType | ICMP APP ID | | IntName4Client | BR ethx | | PeerIntName | Interface as Laptop1’s GW | | PassThrough | 0 (no) | | TimeStamp | Precision in seconds (epoch time) | | SampleSize | 3600 | | AppSampleSize | Number of 60 second intervals that have "app" traffic in this sample period | | DownLinkPacketDelta | Number of packets transmitted to client since last report | | DownLinkByteDelta | Number of bytes transmitted to client since last report | | UpLinkPacketDelta | Number of packets received from client since last report | | UpLinkByteDelta | Number of bytes received from client since last report | | | |
| Test Result |  | | |
| Comment | The accuracy of minutely report verification has been covered by other cases.  During previous L7 application function test, miss hourly application report accuracy verification. | | |

#### ApplicationReporting\_Function\_59

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_59 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or  Laptop1-----(ethx)BR(WAN)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Verify accuracy of CLI "show application reporting app-stats" minutely output. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID or with BR’s ethx. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR.     2. Enable application ICMP for reporting.     3. Execute CLI “Clear application reporting app-stats” to clear previous stats data.     4. Launch ICMP session at laptop1, capture live traffic within 1 minute.     5. After 1 minute, check output of “show application reporting app-stats”.     6. Does not clear application reporting app-stats, repeat step 4 and 5 again. Check output of “sholw application reporting app-stats” is consistent with live captured traffic of 2 minutes. | | |
| Expect result | Step 5.The output of “show application reporting app-stats” should be consistent with live captured traffic of 1 minute.  Step 6. The output of “show application reporting app-stats” should be consistent with live captured traffic of 2 minutes. | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_60

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_60 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or  Laptop1-----(ethx)BR(WAN)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Verify accuracy of CLI "show application reporting app-stats" hourly output. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID or with BR’s ethx. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR.     2. Enable application ICMP for reporting.     3. Save config and reboot HiveOS device to clear previous stats data.     4. Launch ICMP session at laptop1, capture live traffic within 1 hour.     5. After 1 hour, check output of “show application reporting app-stats”.     6. Does not clear application reporting app-stats, repeat step 4 and 5 again. Check output of “sholw application reporting app-stats” is consistent with live captured traffic of 2 hours. | | |
| Expect result | Step 5.The output of “show application reporting app-stats” should be consistent with live captured traffic of 1 hour or hourly application reporting file.  Step 6. The output of “show application reporting app-stats” should be consistent with live captured traffic of 2 hours or hourly application reporting file. | | |
| Test Result |  | | |
| Comment |  | | |

### Collection-interval and report-interval Test

#### ApplicationReporting\_Function\_48

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_48 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Verify background application reporting collection interval and report interval | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 and laptop2 connect with SSID. | | |
| Test procedure | 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR.   CLI: application reporting auto   1. ~~Keep default settings of application reporting for every individual interface at AP and BR~~ 2. Enable application for reporting and lauch this application.   CLI: application reporting app-id 285 enable   1. Keep AP and BR power on for more than 2 hour after enable application for reporting. 2. Login AP and BR shell, check application reporting file at “/tmp/L7\_report”. | | |
| Expect result | Step 5. The 1st background application reporting file is for the 1st complete hour after it is powered on. And it indicates background application reporting collection interval and report interval are 3600s. | | |
| Test Result |  | | |
| Comment | * + 1. From 1 hour data, the actual time range of traffic is from 00:59:00 to 01:59:00, since HiveOS reports previous 1 hour data at around 01:59:02. In order to be easy to understand, still asume data cover from 01:00:00 to 02:00:00. It is the same for finer granularity reporting.     2. Application reporting file only records complete time period. For example, it only records data from 00:00:00 to 01:00:00 instead of 00:10:00 to 01:10:00, which misses 10 minutes from a complete hour. HiveOS will ignore record the data of incomplete time slot.     3. Since 6.1r1, HiveOS makes enhancement that HiveOS genereates reports at integral point, like 01:00:00 instead of 00:59:00. | | |

#### ApplicationReporting\_Function\_49

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_49 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Verify enable fine granularity of collection interval and report interval when AP or BR already has had application reporting data for more than 1 hour. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID. | | |
| Test procedure | 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR. 2. ~~Keep default settings of background application reporting for every individual interface at AP and BR~~ 3. Enable applications for reporting. 4. Launch application at laptop1. 5. After enable applications for reporting for more than 1 hour, set finer granularity of collection interval and report interval less than the default setting.   For example, set both collection interval and report interval as 10 minutes.   1. Login AP or BR shell, check application reporting file at “/tmp/L7\_report”. 2. Keep AP or BR power on for more than 1 report interval. 3. Login AP or BR shell, check application reporting file at “/tmp/L7\_report”. 4. Keep AP or BR power on for more than 1 hour 5. Check background application reporting still works as common, and finer granularity reporting doesn’t intrupt background reporting. | | |
| Expect result | Step 5. Set collection interval and report interval successfully.  Step 6. AP or BR reports previous 1 hour data with finer granularity.  2012-12-11 16:46:17 info l7d: A new report file created: "08EA440EE4000082.hpr"  2012-12-11 16:46:17 debug l7d: A new report start: from 46:01 to 46:01, span 60  2012-12-11 16:46:17 debug l7d: [rpt\_basic]: On reporting thread: c\_intv 1, span 60, lead\_tm 0, type 1  2012-12-11 16:46:17 debug l7d: Start an immediate report (high resolution) ending at 46 min for the past hour  2012-12-11 16:46:17 debug l7d: L7D set timer for high resolution to 47 min, 46 second remain  2012-12-11 16:46:17 info l7d: Start timers in period CLI (112)  2012-12-11 16:46:17 info ah\_cli: admin:<application reporting collection-period 60 report-period 60>  Step 8. AP or BR report data with finer granularity on the new report interval.  Show log buffer:  Step 10. The background reporting and finer granularity reporting co-exists, no disturbance between each other. | | |
| Test Result |  | | |
| Comment | Explaination for step 6:  For example:   1. If HiveOS stores previous 1 hour data from 00:12:00 to 01:12:00.   Issue “application reporting collection-period 600 report-period 600” at 01:12:00.  At step 6. HiveOS only reports data from 00:20:00 to 01:10:00, and ignore data from 00:12:00 to 00:20:00. The data from 01:10:00 to 01:12:00 will be reported next time.   1. This case verifies the 1st time of finer granularity reporting enabling. In order to verify another common scenario that disable finer granularity reporting and enable it again as following:   After step 10, disable finer granularity reporting, and repeat test case step 4 to 10.  Confirm with HiveOS Dev, these 2 scenarios apply the same code. If time and resource limitation, can only verify one of 2 sceanrios.  No GUI option at HM to set collection-period, it sets collection-period as 60s by default. | | |

#### ApplicationReporting\_Function\_50

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_50 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Verify enable fine granularity of collection interval and report interval when AP or BR already has had application reporting data for less than 1 hour. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID. | | |
| Test procedure | 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR. 2. ~~Keep default settings of background application reporting for every individual interface at AP and BR~~ 3. Enable applications for reporting. 4. Launch application at laptop1. 5. After enable applications for reporting for less than 1 hour, set finer granularity of collection interval and report interval than the default setting.   For example, set both collection interval and report interval as 10 minutes.   1. Login AP or BR shell, check application reporting file at “/tmp/L7\_report”. 2. Keep AP or BR power on for more than 1 report interval. 3. Login AP or BR shell, check application reporting file at “/tmp/L7\_report”. 4. Keep AP or BR power on for more than 1 hour 5. Check background application reporting still works as common, and finer granularity reporting doesn’t intrupt background reporting. | | |
| Expect result | Step 5. Set collection interval and report interval successfully.  Step 6. AP or BR reports previous data with finer granularity. The reporting content is from time point when enable applications for reporting.  Show log buffer:  2012-12-12 09:55:44 debug l7d: [rpt\_basic]: New MAC entry list with timestamp 1355276821, samplesize 120  2012-12-12 09:55:44 info l7d: A new report file created: "08EA440EE4000000.hpr"  2012-12-12 09:55:44 debug l7d: A new report start: from 47:01 to 55:01, span 8  2012-12-12 09:55:44 debug l7d: [rpt\_basic]: On reporting thread: c\_intv 2, span 60, lead\_tm 0, type 1  2012-12-12 09:55:44 debug l7d: Start an immediate report (high resolution) ending at 55 min for the past hour  2012-12-12 09:55:44 debug l7d: L7D set timer for high resolution to 57 min, 79 second remain  2012-12-12 09:55:44 info l7d: Start timers in period CLI (112)  2012-12-12 09:55:44 info ah\_cli: admin:<application reporting collection-period 120 report-period 120>  Step 8. AP or BR report data with finer granularity on the new report interval.  Show log buffer  2012-12-12 09:57:02 debug l7d: [rpt\_basic]: New MAC entry list with timestamp 1355277301, samplesize 120  2012-12-12 09:57:02 info l7d: A new report file created: "08EA440EE4000001.hpr"  2012-12-12 09:57:02 debug l7d: A new report start: from 55:01 to 57:01, span 2  2012-12-12 09:57:02 debug l7d: [rpt\_basic]: On reporting thread: c\_intv 2, span 2, lead\_tm 0, type 1  2012-12-12 09:57:02 debug l7d: High resolution timer expired, start to report  Step 10. The background reporting and finer granularity reporting co-exists, no disturbance between each other.  Step 8. HiveOS reports application data of past 1 reporting interval. | | |
| Test Result |  | | |
| Comment | This case verifies the 1st time of finer granularity reporting enabling. In order to verify more common scenario that disable finer granularity reporting and enable it again as following:  After step 10, disable finer granularity reporting, and repeat test case step 4 to 10.  Confirm with HiveOS Dev, these 2 scenarios apply the same code. If time and resource limitation, can only verify one of 2 sceanrios. | | |

#### ApplicationReporting\_Function\_51

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_51 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Verify collection period and report interval setting and function. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID. | | |
| Test procedure | 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR. 2. ~~Keep default settings of background application reporting for every individual interface at AP and BR~~ 3. Enable applications for reporting. 4. Launch application at laptop1. 5. Set the minimum value for both collection interval and report interval. It should be 60s. 6. Keep AP or BR power on for more than 5 minutes. 7. Login AP or BR shell, check application reporting file at “/tmp/L7\_report”. 8. Set a middle value, like 600s, for both collection interval and report interval. 9. Keep AP or BR power on for more than 20 minutes. 10. Login AP or BR shell, check application reporting file at “/tmp/L7\_report”. 11. Set the maximum value for both collection interval and report interval. It should be 3600s. 12. Keep AP or BR power on for more than 1 hour. 13. Check the finer granularity reporting has been disabled after setting maximum value for collection and report interval. And background reporting works as common. | | |
| Expect result | Step 5. Set mimimum value for both collection interval and report interval successfully.  Step 7. HiveOS reports application with finer granularity as setting.  Step 8. Set middle value for both collection interval and report interval successfully.  Step 10. HiveOS reports application with finer granularity as setting.  Step 11. Set maximum value for both collection interval and report interval successfully.  Step 13. The finer granularity reporting has been disabled after setting maximum value for collection and report interval. And background reporting works as common. | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_52

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_52 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Currently, HM only supports setting same value for collection interval and report interval.  Design this case for verify setting different value for collection interval and report interval. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID. | | |
| Test procedure | 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR. 2. Keep default settings of background application reporting for every individual interface at AP and BR 3. Enable applications for reporting. 4. Launch application at laptop1. 5. Set the different value for both collection interval and report interval.   For example, set collection interval as 300s and report interval as 600s.   1. Keep AP or BR power on for more than 20 minutes. 2. Login AP or BR shell, check application reporting file at “/tmp/L7\_report”. | | |
| Expect result | Step 5. Set collection interval and report interval successfully.  Step 7. HiveOS reports application with finer granularity as setting. | | |
| Test Result |  | | |
| Comment | Currently, HM only support setting same value for collection interval and report interval. | | |

#### ApplicationReporting\_Function\_53

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_53 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Verify HiveOS will delete obsolete application reporting files when the size of them achieves maximum limitation. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID. | | |
| Test procedure | 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR. 2. Keep default settings of background application reporting for every individual interface at AP and BR 3. Enable applications for reporting. 4. Launch a large number of applications at laptop1. 5. Keep AP or BR power on for long time to generate large size of application reporting file. 6. Login AP or BR shell and enter file directory “/tmp/L7\_report”. Check HiveOS action when the size of application reporting files is achieve 2M. | | |
| Expect result | Step 6. HiveOS will delete old application reporting files. After file delete, the size of directory is close to 1M. If delete one more file, the size of directory would be less than 1 M. | | |
| Test Result | application reporting max-report-space [numbers] safety-report-space [numbers]  AP330-L7:/tmp/l7\_report$ ll  drwxrwxrwx 2 root root 140 Jan 17 18:11 ./  drwxrwxrwt 22 root root 900 Jan 17 18:11 ../  -rw-r----- 1 root root 460 Jan 17 18:07 08EA440EE4000391.hpr  -rw-r----- 1 root root 414 Jan 17 18:08 08EA440EE4000392.hpr  -rw-r----- 1 root root 414 Jan 17 18:09 08EA440EE4000393.hpr  -rw-r----- 1 root root 368 Jan 17 18:10 08EA440EE4000394.hpr  -rw-r----- 1 root root 414 Jan 17 18:11 08EA440EE4000395.hpr  AP330-L7:/tmp/l7\_report$ ll  drwxrwxrwx 2 root root 100 Jan 17 18:12 ./  drwxrwxrwt 22 root root 900 Jan 17 18:12 ../  -rw-r----- 1 root root 368 Jan 17 18:10 08EA440EE4000394.hpr  -rw-r----- 1 root root 414 Jan 17 18:11 08EA440EE4000395.hpr  -rw-r----- 1 root root 368 Jan 17 18:12 08EA440EE4000396.hpr  AP330-L7:/tmp/l7\_report$ exit  Welcome back to CLI console!  AP330-L7#show application reporting report-space  max-report-space: 2KB  safety-report-space: 1KB | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_56

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_56 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or Laptop1-----(ethx)BR(WAN)\_\_\_\_\_\_Switch\_\_\_\_\_HM |  |  Internet | | |
| Description | Verify HiveOS can generate correct hourly report after time drifting. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID or with BR’s ethx. | | |
| Test procedure | 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP/BR. 2. Enable applications for reporting. 3. Launch application at laptop1 for 30 minutes since 01:00. 4. Change HiveOS local time to 02:40 5. After more than 20 minutes, HiveOS local time should pass 03:00. Login AP/BR shell, check application reporting file at “/tmp/L7\_report”. 6. After more than 1 reporting interval, check application reporting file accuracy after time change. | | |
| Expect result | Step 5. HiveOS application reporting file covers following data:  00:50-01:30 and 02:40-03:00.  And the timestamp should be 02:00  Step 6. After time drifting, HiveOS should report application data correctly. | | |
| Test Result |  | | |
| Comment | HiveOS 6.0r2 makes enhancement to overcome time drifting issue, add this case to verify HiveOS enhancement. | | |

#### ApplicationReporting\_Function\_57

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_57 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or  Laptop1-----(ethx)BR(WAN)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Verify HiveOS can generate correct minutely report after time drifting. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID or with BR’s ethx. | | |
| Test procedure | 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP/BR. 2. Enable applications for reporting. 3. Set both collection interval as 1 minute and reporting interval as 10 minutes. 4. Launch application at laptop1 for 3 minutes since 00:10. 5. Change HiveOS local time to 01:29. 6. After more than 1 minute, HiveOS local time should pass 01:30. Login AP/BR shell, check application reporting file at “/tmp/L7\_report”. 7. After more than 1 reporting interval, check application reporting file accuracy after time change. | | |
| Expect result | Step 5. HiveOS application reporting file covers following data:  00:04-00:13 and 01:29-01:30.  And the timestamp should be 01:20  Step 7. After time drifting, HiveOS should report application data correctly. | | |
| Test Result |  | | |
| Comment | HiveOS 6.0r2 makes enhancement to overcome time drifting issue, add this case to verify HiveOS enhancement.  In order to check reporting accuracy, laptop pings outside network. Capture ping traffic. Reporting file should cover these ping packets.  If the time drifting is less than 18 seconds, HiveOS keeps original reporting data slot and does not reset reporting timer. At next reporting interval, HiveOS reports at time point based on time drifting. (会根据时钟漂移，做相应的漂移). At the 2nd reporting interval, it will reports at expected timer.(整点生成报告)  If the time drifting is more than 1 minute, HiveOS moves 1 reporting data slot and reset reporting timer, and following new data slot after time drifting. (整点生成报告，但在报告中丢失一分钟，这一分钟的数据被报在前一个slot中。)  If the time drifting is more than 18 seconds and less than 1 minute, HiveOS keeps original reporting data slot but reset reporting timer. At next reporting interval, HiveOS reports at expected time.(整点生成报告)  Due to time drifting fix is not a test project, just draft one case for minutely reporting verification.  In order to verify it more carefully, we’d better [separate](app:ds:separate) it [into](app:ds:into) 3 cases to verify all scenarios.  Update on Jul 1 2013:  Due to bug 27521 and 27617, HiveOS changes time drifting fix as following:  It makes time drifting check every 1 second, and will handle time drifting as following 2 kinds of scenario:  Time drifting is less than 1 minute.  Time drifting is more than 1 minute. | | |

### Disable L7 Engine/Application reporting Test

#### ApplicationReporting\_Function\_54

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_54 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Verify application reporting when disable L7 engine. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s sub-interface, which is set as access mode.  Laptop1 connects with SSID.  Laptop1’s gateway should be BR. | | |
| Test procedure | * + 1. Disable L7 engine and set application reporting mode as “auto” globally at AP.     2. ~~Keep default settings of application reporting for AP and BR interfaces which traffic will go through.~~     3. Enable application HTTP, TCP and IP for reporting.     4. Laptop1 accesses Internet web site via http session.     5. Check if AP and BR can classify HTTP session and report it to HM correctly.     6. Set application reporting mode as “disable” globally at AP.     7. Clear previous http session at AP, and then launch http session again at laptop1.     8. Check if AP and BR can classify HTTP session and report application traffic to HM correctly.     9. Set application reporting mode as “enable” globally at AP.     10. Clear previous http session at AP, and then launch http session again at laptop1.     11. Check if AP and BR can classify HTTP session and report application traffic to HM correctly. | | |
| Expect result | Step 1. Turn off L7 engine and set application reporting mode globally successfully.  ~~Step 2. The default mode of application reporting at interfaces is “auto”.~~  Step 3. Enable application HTTP, TCP and IP for reporting successfully.  Step 5. No TLV with field “AppType” as HTTP App ID at BR and AP.  Step 6. Set application reporting mode as “disable” globally successfully.  Step 7. No TLV with field “AppType” as HTTP App ID at BR and AP.  Step 9. Set application reporting mode as “enable” globally successfully.  Step 11. No TLV with field “AppType” as HTTP App ID at BR and AP. | | |
| Test Result |  | | |
| Comment |  | | |

#### ApplicationReporting\_Function\_55

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_55 | | |
| Priority | High | Automation Flag | No |
| Topology to use | Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_(eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet | | |
| Description | Turn off specific application for reporting | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, and AP eth as backhaul.  Create a SSID and bind it with AP’s wifi interface.  Laptop1 connects with SSID. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP and BR.     2. ~~Keep default settings of application reporting for every individual interface at AP and BR~~     3. Disable application YouTube, Facebook, HTTP, TCP and IP for reporting with CSV format.     4. Launch YouTube and play veido, check if AP and BR can classify YouTube and report it to HM correctly.     5. Launch Facebook, check if AP and BR can classify Facebook and report it to HM correctly. | | |
| Expect result | Step 4. AP and BR do NOT report application to HM.  Step 5. AP and BR do NOT report application to HM. | | |
| Test Result |  | | |
| Comment |  | | |

### HiveOS application discovery

#### ApplicationReporting\_Function\_61

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_61 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | For AP  Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  For BR  Laptop1-----(eth1/WiFi)BR(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or we can meger AP and BR into same one topology. | | |
| Description | Verify HiveOS reports original minutely application data correctly. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  Create a SSID and bind it with AP’s sub-interface, which is set as access mode.  Laptop1 connects with SSID, or with BR. | | |
| Test procedure | * + 1. Enable L7 engine and set application reporting mode as “auto” globally at AP.     2. Enable HiveOS original minutely application report.     3. Capture live traffic at Laptop1.     4. Laptop1 Ping outside network. Last this step more than 1 minutely repor interval until HiveOS generates minutely application report.     5. Compare HiveOS minutely application report with captured live traffic. | | |
| Expect result | Step 5. The minutely application report should be consistent withcaptured live traffic.  Should check all items of record on ICMP in minutely application report file. | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release.  It is the baseline for HiveOS application discovery verification. | | |

#### ApplicationReporting\_Function\_62

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_62 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | For AP  Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  For BR  Laptop1-----(eth1/WiFi)BR(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or we can meger AP and BR into same one topology. | | |
| Description | Verify HiveOS reports original hourly application data correctly. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  Create a SSID and bind it with AP’s sub-interface, which is set as access mode.  Laptop1 connects with SSID, or with BR. | | |
| Test procedure | * + 1. Enable L7 engine and set application reporting mode as “auto” globally at AP.     2. Enable HiveOS original hourly application report.     3. Capture live traffic at Laptop1.     4. Laptop1 Ping outside network. Last this step more than 1 hourly repor interval until HiveOS generates hourly application report.     5. Compare HiveOS hourly application report with captured live traffic.     6. Get last hour application usage via output of “show application reporting app-stats”. Compare HiveOS hourly application report with it. | | |
| Expect result | Step 5. The hourly application report should be consistent with captured live traffic.  Should check all items of record on ICMP in hourly application report file.  Step 6. The hourly application report should be consistent with output of “show application reporting app-stats”. | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release.  It is the baseline for HiveOS application discovery verification.  The step 5 is to verify data accuracy of a specific application. It can cover all items of application record.  The step 6 is to verify data accuracy of all applications. It can cover application usage of all applications. | | |

#### ApplicationReporting\_Function\_63

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_63 | | |
| Priority | High | Automation Flag | No |
| Topology to use | For AP  Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  For BR  Laptop1-----(eth1/WiFi)BR(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or we can meger AP and BR into same one topology. | | |
| Description | Verify HiveOS reports correct hourly application data with application discovery. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  Create a SSID and bind it with AP’s sub-interface, which is set as access mode.  Laptop1 connects with SSID, or with BR. | | |
| Test procedure | * + 1. Enable L7 engine and set application reporting mode as “auto” globally at AP.     2. Enable HiveOS original hourly application report.     3. Make sure application discovery is enabled     4. Laptop1 access outside network to simulate common usage, like office scenario, retail scenario, education scenario, healthcare scenario and so on. Last this step more than 1 hourly report interval until HiveOS generates hourly application report.     5. Get last hour application usage via output of “show application reporting app-stats”. Compare HiveOS hourly application report with it.     6. Compare HiveOS hourly application report with original hourly application report. | | |
| Expect result | Step 5. The hourly application report should be consistent with output of “show application reporting app-stats”.  Step 6. The minutely application report filtered by application discovery algorithm should be consistent with orignail minutely application report.  Both step 5 and step 6: Check application report data accuracy and application coverage.  Partial applications whose percentage is less than "percentage screening granularity" should be reported as “Unknown” applications.  For more info on HiveOS application discovery and definition of "percentage screening granularity", please check following site:  <https://wiki.aerohive.com/wiki/pages/viewpage.action?pageId=7768314> | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release.  The output of “show application reporting app-stats” and original application hourly report are the baseline for application discovery verification. Its accuracy was verified at previous release.  In order to verify HiveOS algorithm branch as complete as possible, QA may need to repeate this test for a few times.If we can simulate application traffic by BPS, it is may be not necessary. | | |

#### ApplicationReporting\_Function\_64

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_64 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | For AP  Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  For BR  Laptop1-----(eth1/WiFi)BR(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or we can meger AP and BR into same one topology. | | |
| Description | Verify HiveOS reports correct minutely application data with application discovery. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  Create a SSID and bind it with AP’s sub-interface, which is set as access mode.  Laptop1 connects with SSID, or with BR. | | |
| Test procedure | * + 1. Enable L7 engine and set application reporting mode as “auto” globally at AP.     2. Make sure application discovery is enabled.     3. Enable orignail minutely application report.     4. Laptop1 access outside network to simulate common usage, like office scenario, retail scenario, education scenario, healthcare scenario and so on. Last this step more than 1 minutely report interval until HiveOS generates minutely application report.     5. Compare HiveOS minutely application report with original minutely application report. | | |
| Expect result | Step 5. The minutely application report filtered by application discovery algorithm should be consistent with orignail minutely application report.  Check application report data accuracy and application coverage.  Partial applications whose percentage is less than "percentage screening granularity" should be reported as “Unknown” applications.  For more info on HiveOS application discovery and definition of "percentage screening granularity", please check following site:  <https://wiki.aerohive.com/wiki/pages/viewpage.action?pageId=7768314> | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release.  The original minutely application report is the baseline for application discovery verification.  In order to verify HiveOS algorithm branch as complete as possible, QA may need to repeate this test for a few times. If we can simulate application traffic by BPS, it is may be not necessary. | | |

#### ApplicationReporting\_Function\_65

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_65 | | |
| Priority | Accept | Automation Flag | No |
| Topology to use | For AP  Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  For BR  Laptop1-----(eth1/WiFi)BR(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or we can meger AP and BR into same one topology. | | |
| Description | Verify HiveOS reports application which is in “Constant Watch List” when its usage percentage is NOT more than initial “Percentage Screening Granularity”. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  Create a SSID and bind it with AP’s sub-interface, which is set as access mode.  Laptop1 connects with SSID, or with BR. | | |
| Test procedure | * + 1. Enable L7 engine and set application reporting mode as “auto” globally at AP.     2. Make sure application discovery is enabled.     3. Add ICMP into “Constant Watch List”     4. Capture live traffic at Laptop1.     5. Laptop1 ping outside network. Only ping few packets to make sure its usage percentage is NOT more than initial “Percentage Screening Granularity”. In order to make ICMP usage percentage is less enough to be filtered, generate huge background traffic like http video, and ftp download large files. And try to generate applications as many as possible, at least should be more than 15. | | |
| Expect result | After Step 5, HiveOS generates application reporting file.  It should report ICMP correctly. | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release.  When application usage percentage is not more than initial “Percentage Screening Granularity”, HiveOS may not classify it as “Unknown” with very low possibility.  The following case may be a better verification, but the BPS should be available:  In case ApplicationReporting\_Function\_70, add #2 applications into watchlist, make sure #2 applications will be reserved. | | |

#### ApplicationReporting\_Function\_66

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_66 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | For AP  Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  For BR  Laptop1-----(eth1/WiFi)BR(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or we can meger AP and BR into same one topology. | | |
| Description | Verify HiveOS reports application which is in “Constant Watch List” when its usage percentage is more than “Percentage Screening Granularity”. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  Create a SSID and bind it with AP’s sub-interface, which is set as access mode.  Laptop1 connects with SSID, or with BR. | | |
| Test procedure | * + 1. Enable L7 engine and set application reporting mode as “auto” globally at AP.     2. Make sure application discovery is enabled.     3. Add ICMP or http video into “Constant Watch List”     4. Capture live traffic at Laptor1.     5. Laptop1 ping outside network or watch http video. Make sure ICMP or http video usage percentage is more than “Percentage Screening Granularity”. | | |
| Expect result | After Step 5, HiveOS generates application reporting file.  It should not report ICMP or http video as “Unknown”. | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release.  The following case may be a better verification, but the BPS should be available:  In case ApplicationReporting\_Function\_70, add #1 application into watchlist, make sure #1 application will be reserved. | | |

#### ApplicationReporting\_Function\_67

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_67 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | For AP  BPS--------------(eth1)AP(eth0)\_\_\_\_\_HM  | |  |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|  Internet  For BR  BPS--------------(eth1)BR(eth0)\_\_\_\_\_HM  | |  |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|  Or we can meger AP and BR into same one topology. | | |
| Description | Verify HiveOS application discovery algorithm branch1. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  BPS connects with AP or BR. | | |
| Test procedure | * + 1. Enable L7 engine and set application reporting mode as “auto” globally at AP.     2. Make sure application discovery is enabled.     3. Simulates following application usage percentage:  |  |  |  | | --- | --- | --- | | Item | Number of App | Usage Percetage | | 1 | 3 | 33% | | 2 | 5 | 0.2% | | | |
| Expect result | After Step 3, HiveOS generates application reporting file.  It should only reports #2 applications as “Unknown” and #1 as correct application. | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release. | | |

#### ApplicationReporting\_Function\_68

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_68 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | For AP  BPS--------------(eth1)AP(eth0)\_\_\_\_\_HM  | |  |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|  Internet  For BR  BPS--------------(eth1)BR(eth0)\_\_\_\_\_HM  | |  |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|  Or we can meger AP and BR into same one topology. | | |
| Description | Verify HiveOS application discovery algorithm branch2. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  BPS connects with AP or BR. | | |
| Test procedure | * + 1. Enable L7 engine and set application reporting mode as “auto” globally at AP.     2. Make sure application discovery is enabled.     3. Simulates following application usage percentage:  |  |  |  | | --- | --- | --- | | Item | Number of App | Usage Percetage | | 1 | 5 | 7% | | 2 | 13 | 5% | | | |
| Expect result | After Step 3, HiveOS generates application reporting file.  It should only reports three of #2 applications as “Unknown” and the rest as correct application. | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release. | | |

#### ApplicationReporting\_Function\_69

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_69 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | For AP  BPS--------------(eth1)AP(eth0)\_\_\_\_\_HM  | |  |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|  Internet  For BR  BPS--------------(eth1)BR(eth0)\_\_\_\_\_HM  | |  |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|  Or we can meger AP and BR into same one topology. | | |
| Description | Verify HiveOS application discovery algorithm branch3. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  BPS connects with AP or BR. | | |
| Test procedure | * + 1. Enable L7 engine and set application reporting mode as “auto” globally at AP.     2. Make sure application discovery is enabled.     3. Simulates following application usage percentage:  |  |  |  | | --- | --- | --- | | Item | Number of App | Usage Percetage | | 1 | 4 | 24% | | 2 | 1 | 3% | | 3 | 1 | 1% | | | |
| Expect result | After Step 3, HiveOS generates application reporting file.  It should reports #2 and #3 applications as “Unknown”, and #1 as correct application. | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release. | | |

#### ApplicationReporting\_Function\_70

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_70 | | |
| Priority | High | Automation Flag | No |
| Topology to use | For AP  BPS--------------(eth1)AP(eth0)\_\_\_\_\_HM  | |  |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|  Internet  For BR  BPS--------------(eth1)BR(eth0)\_\_\_\_\_HM  | |  |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|  Or we can meger AP and BR into same one topology. | | |
| Description | Verify HiveOS application watchlist with Application discovery. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  BPS connects with AP or BR. | | |
| Test procedure | * + 1. Enable L7 engine and set application reporting mode as “auto” globally at AP.     2. Make sure application discovery is enabled.     3. Add #2 applications which usage percentage is 1% into watchlist.     4. Simulates following application usage percentage:  |  |  |  | | --- | --- | --- | | Item | Number of App | Usage Percetage | | 1 | 3 | 33% | | 2 | 1 | 1% | | | |
| Expect result | After Step 4, HiveOS generates application reporting file.  It should reports both #1 and #2 as correct applications. | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release. | | |

#### ApplicationReporting\_Function\_71

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_71 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | For AP  Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  For BR  Laptop1-----(eth1/WiFi)BR(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or we can meger AP and BR into same one topology. | | |
| Description | Verify HiveOS removes all applications in watchlist after it upgrades from obsolete version to new version which supports application discovery. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  Create a SSID and bind it with AP’s sub-interface, which is set as access mode.  Laptop1 connects with SSID, or with BR. | | |
| Test procedure | * + 1. HiveOS loads previous version like 6.0r2 or 6.1r1.     2. Add applications into watchlist.     3. Upgrade from obsolete version to new version which supports application discovery.     4. Reboot HiveOS device to load new version | | |
| Expect result | Step 4, After HiveOS image upgrade, it removes all applications in watchlist. | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release. | | |

#### ApplicationReporting\_Function\_72

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_72 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | For AP  Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  For BR  Laptop1-----(eth1/WiFi)BR(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or we can meger AP and BR into same one topology. | | |
| Description | Verify Geneva HiveOS gets watchlist correctly by new CLI1.  Enable application reporting by obsolete CLI | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  BPS connects with AP or BR. | | |
| Test procedure | * + 1. Enable L7 engine and set application reporting mode as “auto” globally at AP.     2. Make sure application discovery is enabled.     3. Add ICMP into watchlist via obsolete CLI: application reporting app-id 285 enable. But does not add it via new CLI.     4. Issue CLIs “Show running configuration” and “show application reporting configuration”     5. Capture live traffic at Laptop1.     6. Laptop1 ping outside network. Only ping few packets to make sure its usage percentage is NOT more than initial “Percentage Screening Granularity”. | | |
| Expect result | Step 4. No “application reporting app-id 285 enable” in running configuration. And no app-id 285 in application reporting configuration.  After Step 6, HiveOS generates application reporting file.  It should report ICMP as “Unknown”. | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release.  When application usage percentage is not more than initial “Percentage Screening Granularity”, HiveOS may not classify it as “Unknown” with very low possibility.  The following case may be a better verification, but the BPS should be available:  In case ApplicationReporting\_Function\_70, add #2 applications into watchlist, make sure #2 applications will not be reserved. | | |

#### ApplicationReporting\_Function\_73

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_73 | | |
| Priority | Low | Automation Flag | No |
| Topology to use | For AP  Laptop1-----(wifi0/wifi1)AP(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  For BR  Laptop1-----(eth1/WiFi)BR(eth)\_\_\_\_\_\_Switch\_\_\_\_\_HM  |  |  Internet  Or we can meger AP and BR into same one topology. | | |
| Description | Verify Geneva HiveOS gets watchlist correctly by new CLI2.  Disable application reporting by obsolete CLI. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  BPS connects with AP or BR. | | |
| Test procedure | * + 1. Enable L7 engine and set application reporting mode as “auto” globally at AP.     2. Make sure application discovery is enabled.     3. Add ICMP into watchlist via obsolete CLI: “no application reporting app-id 658 enable”.     4. Issue CLIs “Show running configuration”.     5. Laptop1 play http video from outside network. Make sure its usage percentage is more than 50%. | | |
| Expect result | Step 4. No “no application reporting app-id 658 enable” in running configuration.  After Step 5, HiveOS generates application reporting file.  It should report http video correctly instead of “Unknown”. | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release. | | |

#### ApplicationReporting\_Function\_74

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_Function\_74 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Office Network | | |
| Description | Capture office network live traffic for DEV analysis. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  BPS connects with AP or BR. | | |
| Test procedure | * + 1. Capture office network live traffic for DEV analysis. | | |
| Expect result |  | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release.  As per DEV requirement to capture office network traffic for their analysis. | | |

## Stress Test Case

### L7\_Engine\_Stress\_01

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Stress\_01 | | |
| Priority | High | Automation Flag | No |
| Topology to use | (eth0)MP(wifi1.1)-----AP330\_\_\_\_SW\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify AP’s stress capacity.  Simulate traffic close to system IP session capacity for long time.  This case is specific for AP with only one eth interface. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170, | | |
| Pre-condition | AP330 and MP are managed by HM.  Set MP wifi1.1 as backhaul mode.  IXIA connects with MP eth0. | | |
| Test procedure | IXIA simulates traffic close to system's ip session capacity for 24 hours or over a week.  Check application reporting accuracy of stress test period. | | |
| Expect result | No HiveOS device crash. And record how much time HiveOS takes to resume to normal after stress test. | | |
| Test Result |  | | |
| Comment | As design, the AP maximum system IP session capacity is 8191. | | |

### L7\_Engine\_Stress\_02

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Stress\_02 | | |
| Priority | High | Automation Flag | No |
| Topology to use | (eth0)MP(wifi1.1)-----AP330\_\_\_\_SW\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify AP’s stress load.  Simulate traffic with traffic rate close to maximum tcp connection rate measured in performance test.  This case is specific for AP with only one eth interface. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170, | | |
| Pre-condition | AP330 and MP are managed by HM.  Set MP wifi1.1 as backhaul mode.  IXIA connects with MP eth0. | | |
| Test procedure | * + 1. Simulates HTTP1.1 without persistence by IXIA, and set HTTP page size as 1b.     2. IXIA simulates traffic close to maximum tcp connection rate for 24 hours or over a week.     3. Check application reporting accuracy of stress test period. | | |
| Expect result | No HiveOS device crash. And record how much time HiveOS takes to resume to normal after stress test. | | |
| Test Result |  | | |
| Comment |  | | |

### L7\_Engine\_Stress\_03

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Stress\_03 | | |
| Priority | High | Automation Flag | No |
| Topology to use | (eth1)AP(eth0)\_\_\_\_SW\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify AP’s stress capacity.  Simulate traffic close to system IP session capacity for long time.  This case is specific for AP with 2 eth interfaces. | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP is managed by HM.  IXIA connects with AP eth0. | | |
| Test procedure | IXIA simulates traffic close to system's ip session capacity for 24 hours or over a week | | |
| Expect result | No HiveOS device crash. And record how much time HiveOS takes to resume to normal after stress test. | | |
| Test Result |  | | |
| Comment | As design, the AP maximum system IP session capacity is 8191. | | |

### L7\_Engine\_Stress\_04

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Stress\_04 | | |
| Priority | High | Automation Flag | No |
| Topology to use | (eth1)AP(eth0)\_\_\_\_SW\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify AP’s stress load.  Simulate traffic with traffic rate close to maximum tcp connection rate measured in performance test.  This case is specific for AP with 2 eth interfaces. | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP is managed by HM.  IXIA connects with AP eth0. | | |
| Test procedure | * + 1. Simulates HTTP1.1 without persistence by IXIA, and set HTTP page size as 1b.     2. IXIA simulates traffic close to maximum tcp connection rate for 24 hours or over a week.     3. Check application reporting accuracy of stress test period. | | |
| Expect result | No HiveOS device crash. And record how much time HiveOS takes to resume to normal after stress test. | | |
| Test Result |  | | |
| Comment |  | | |

### L7\_Engine\_Stress\_05

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Stress\_05 | | |
| Priority | High | Automation Flag | No |
| Topology to use | (eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify BR’s stress capacity.  Simulate traffic close to system IP session capacity for long time. | | |
| PlatformDependence | BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth1 mode as bridge-access, and eth0 as wan. | | |
| Test procedure | IXIA simulates traffic close to system's ip session capacity for 24 hours or over a week | | |
| Expect result | No HiveOS device crash. And record how much time HiveOS takes to resume to normal after stress test. | | |
| Test Result |  | | |
| Comment | No data on maximum system IP session capacity of BR | | |

### L7\_Engine\_Stress\_06

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Stress\_06 | | |
| Priority | High | Automation Flag | No |
| Topology to use | (eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify BR’s stress load.  Simulate traffic with traffic rate close to maximum tcp connection rate measured in performance test. | | |
| PlatformDependence | BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth1 mode as bridge-access, and eth0 as wan. | | |
| Test procedure | * + 1. Simulates HTTP1.1 without persistence by IXIA, and set HTTP page size as 1b.     2. IXIA simulates traffic close to maximum tcp connection rate for 24 hours or over a week.     3. Check application reporting accuracy of stress test period. | | |
| Expect result | No HiveOS device crash. And record how much time HiveOS takes to resume to normal after stress test. | | |
| Test Result |  | | |
| Comment |  | | |

### L7\_Engine\_Performance\_07

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Perfremane\_07 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | Office Network | | |
| Description | Enable application discovery at office network, and check whether it will impacts other feature. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition | AP and BR are managed by HM.  Set BR eth1 mode as bridge-802.1q, AP eth as backhaul.  Create a SSID and bind it with AP’s sub-interface, which is set as access mode.  Laptop1 connects with SSID, or with BR. | | |
| Test procedure | * + 1. Enable L7 engine and set application reporting mode as “auto” globally at AP.     2. Enable application reporting and application discovery.     3. Add some application into watchlist. | | |
| Expect result | Trigger application minutely report frequently, maybe 3 to 5 times every hour.  Observe office network for long time, like 1 or 2 weeks.  Application discovery should not impact other features. | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release. | | |

## Longevity Test Case

NA

## Performance Test Case

### L7\_Engine\_Performance\_01

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Perfremane\_01 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | (eth0)MP(wifi1.1)-----AP330\_\_\_\_SW\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify maximum TCP connection rate of AP and the maximum concurrent applications which AP can classify. (Based on HTTP session)  This case is specific for AP with only one eth interface. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170, | | |
| Pre-condition | AP330 and MP are managed by HM.  Set MP wifi1.1 as backhaul mode.  IXIA connects with MP eth0. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.     2. Keep default settings of application reporting for every individual interface at AP     3. Simulates HTTP1.1 without persistence by IXIA, and set HTTP page size as 1b.     4. Check the maximum TCP connection rate (connections/second). And the maximum concurrent applications AP L7 engine can classify (applications/second), which may be much less than maximum TCP connection rate.     5. Turn off L7 engine and repeat step 4 again to check maximum TCP connection rate. | | |
| Expect result |  | | |
| Test Result |  | | |
| Comment | 1. Due to lack of actual product performace target from PLM or developer, can NOT set quantitative data, like data rate and number of application sessions, for performance test case. We need to repeat test step for many times to check actual performance. 2. Record maximum TCP connection rate, and the percentage of application classification at maximum TCP connection rate, which may be not 100%. 3. Record maximum concurrent applications AP L7 engine can classify and the TCP connection rate at maximum concurrent applications. 4. Check reporting accuracy during test, the calculation of percentage of application classification is based on it. 5. Compare test results of enable L7 engine and disable it. 6. Set test perorid as 5 minutes and overnight. The result may be different. 7. Recorde performace test result for every platform. | | |

### L7\_Engine\_Performance\_02

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Perfremane\_02 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | (eth0)MP(wifi1.1)-----AP330\_\_\_\_SW\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify maximum TCP throughput of AP and maximum concurrent applications which AP can classify. (Based on HTTP session)  This case is specific for AP with only one eth interface. | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170, | | |
| Pre-condition | AP330 and MP are managed by HM.  Set MP wifi1.1 as backhaul mode.  IXIA connects with MP eth0. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.     2. Keep default settings of application reporting for every individual interface at AP     3. Simulates HTTP1.1 with persistence by IXIA, and set HTTP page size as 5k.     4. Check the maximum TCP throughput. And the maximum concurrent applications AP L7 engine can classify (applications/second).     5. Set HTTP page size as 64k, repeate step 4 again.     6. Turn off L7 engine and repeat step 4 and step 5 again to check maximum TCP throughput. | | |
| Expect result |  | | |
| Test Result |  | | |
| Comment | 1. Due to lack of actual product performace target from PLM or developer, can NOT set quantitative data, like data rate and number of application sessions, for performance test case. We need to repeat test step for many times to check actual performance. 2. Record maximum TCP throughput and the percentage of application classification at maximum TCP throughput, which may be not 100%. 3. Record maximum concurrent applications AP L7 engine can classify and the TCP throughput at maximum concurrent applications. 4. Check reporting accuracy during test, the calculation of percentage of application classification is based on it. 5. Compare test results of enable L7 engine and disable it. 6. Set test perorid as 5 minutes and overnight. The result may be different. 7. Recorde performace test result for every platform. | | |

### L7\_Engine\_Performance\_03

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Perfremane\_03 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | (eth1)AP(eth0)\_\_\_\_SW\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify maximum TCP connection rate of AP and the maximum concurrent applications which AP can classify. (Based on HTTP session)  This case is specific for AP with 2 eth interfaces. | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP is managed by HM.  IXIA connects with AP eth0. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.     2. ~~Keep default settings of application reporting for every individual interface at AP~~     3. Simulates HTTP1.1 without persistence by IXIA, and set HTTP page size as 1b.     4. Check the maximum TCP connection rate (connections/second). And the maximum concurrent applications AP L7 engine can classify (applications/second), which may be much less than maximum TCP connection rate.     5. Turn off L7 engine and repeat step 4 again to check the maximum TCP connection rate. | | |
| Expect result |  | | |
| Test Result |  | | |
| Comment | 1. Due to lack of actual product performace target from PLM or developer, can NOT set quantitative data, like data rate and number of application sessions, for performance test case. We need to repeat test step for many times to check actual performance. 2. Record maximum TCP connection rate, and the percentage of application classification at maximum TCP connection rate, which may be not 100%. 3. Record maximum concurrent applications AP L7 engine can classify and the TCP connection rate at maximum concurrent applications. 4. Check reporting accuracy during test, the calculation of percentage of application classification is based on it. 5. Compare test results of enable L7 engine and disable it. 6. Set test perorid as 5 minutes and overnight. The result may be different. 7. Recorde performace test result for every platform. | | |

### L7\_Engine\_Performance\_04

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Perfremane\_04 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | (eth1)AP(eth0)\_\_\_\_SW\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify maximum TCP throughput of AP and the maximum concurrent applicationswhich AP can classify. (Based on HTTP session)  This case is specific for AP with 2 eth interfaces. | | |
| PlatformDependence | AP: AP320,AP340,AP330,AP350, | | |
| Pre-condition | AP is managed by HM.  IXIA connects with AP eth0. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at AP.     2. Keep default settings of application reporting for every individual interface at AP     3. Simulates HTTP1.1 with persistence by IXIA, and set HTTP page size as 5k.     4. Check the maximum TCP throughput. And the maximum concurrent applications AP L7 engine can classify (applications/second).     5. Set HTTP page size as 64k, repeate step 4 again.     6. Turn off L7 engine and repeat step 4 and step 5 again to check the maximum TCP throughput. | | |
| Expect result |  | | |
| Test Result |  | | |
| Comment | 1. Due to lack of actual product performace target from PLM or developer, can NOT set quantitative data, like data rate and number of application sessions, for performance test case. We need to repeat test step for many times to check actual performance. 2. Record maximum TCP throughput and the percentage of application classification at maximum TCP throughput, which may be not 100%. 3. Record maximum concurrent applications AP L7 engine can classify and the TCP throughput at maximum concurrent applications. 4. Check reporting accuracy during test, the calculation of percentage of application classification is based on it. 5. Compare test results of enable L7 engine and disable it. 6. Set test perorid as 5 minutes and overnight. The result may be different. 7. Recorde performace test result for every platform. | | |

### L7\_Engine\_Performance\_05

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Perfremane\_05 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | (eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify maximum TCP connection rate of BR, and the maximum concurrent applications which BR can classify. (Based on HTTP session) | | |
| PlatformDependence | BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth1 mode as bridge-access, and eth0 as wan. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at BR.     2. Keep default settings of application reporting for every individual interface at BR     3. Simulates HTTP1.1 without persistence by IXIA, and set HTTP page size as 1b.     4. Check the maximum TCP connection rate (connections/second). And the maximum concurrent applications AP L7 engine can classify (applications/second), which may be much less than maximum TCP connection rate.     5. Turn off L7 engine and repeat step 4 again. | | |
| Expect result |  | | |
| Test Result |  | | |
| Comment | 1. Due to lack of actual product performace target from PLM or developer, can NOT set quantitative data, like data rate and number of application sessions, for performance test case. We need to repeat test step for many times to check actual performance. 2. Record maximum TCP connection rate and the percentage of application classification at maximum TCP connection rate which may be not 100%. 3. Record maximum concurrent applications AP L7 engine can classify and the TCP connection rate at maximum concurrent applications. 4. Check reporting accuracy during test, the calculation of percentage of application classification is based on it. 5. Compare test results of enable L7 engine and disable it. 6. Set test perorid as 5 minutes and overnight. The result may be different. 7. Recorde performace test result for every platform. | | |

### L7\_Engine\_Performance\_06

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Perfremane\_06 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | (eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify maximum TCP throughput of BR and the maximum concurrent applications which BR can classify. (Based on HTTP session) | | |
| PlatformDependence | BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | BR is managed by HM.  Set BR eth1 mode as bridge-access, and eth0 as wan. | | |
| Test procedure | * + 1. Turn on L7 engine and set application reporting mode as “auto” globally at BR.     2. ~~Keep default settings of application reporting for every individual interface at BR~~     3. Simulates HTTP1.1 with persistence by IXIA, and set HTTP page size as 5k.     4. Check the maximum TCP throughput. And the maximum concurrent applications AP L7 engine can classify (applications/second).     5. Set HTTP page size as 64k, repeate step 4 again.     6. Turn off L7 engine and repeat step 4 and step 5 again to check maximum TCP throughput. | | |
| Expect result |  | | |
| Test Result |  | | |
| Comment | 1. Due to lack of actual product performace target from PLM or developer, can NOT set quantitative data, like data rate and number of application sessions, for performance test case. We need to repeat test step for many times to check actual performance. 2. Record maximum TCP throughput and the percentage of application classification at maximum TCP throughput, which may be not 100%. 3. Record maximum concurrent applications AP L7 engine can classify and the TCP throughput at maximum concurrent applications. 4. Check reporting accuracy during test, the calculation of percentage of application classification is based on it. 5. Compare test results of enable L7 engine and disable it. 6. Set test perorid as 5 minutes and overnight. The result may be different. 7. Recorde performace test result for every platform. | | |

## Capacity Test Case

NA

## Compatibility Test Case

NA

## Negative Test Case

### L7\_Engine\_Negative\_01

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_Negative\_01 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use | For AP only 1 eth interface:  (eth0)MP(wifi1.1)-----AP\_\_\_\_SW\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|  For AP with 2 eth interfaces:  (eth1)AP(eth0)\_\_\_\_SW\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_|  For BR:  (eth1)BR(eth0)\_\_\_\_\_Switch\_\_\_\_\_HM  | |  IXIA\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| | | |
| Description | Verify scenario as following:  Pump HiveOS device with heavy traffic load over TCP connection rate measured in performance test, HiveOS device should be able to resume after traffic load back to normal | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition | AP and MP are managed by HM.  Set MP eth0 as bridge-access mode, and wifi1.1 as backhaul mode.  Create a SSID and bind it with AP’s sub-interface, which is set as access mode. | | |
| Test procedure | * + 1. Pump HiveOS device with heavy traffic load over TCP connection rate measured in performance.     2. Traffic load returns back to normal. Check HiveOS memory and CPU utilization. | | |
| Expect result | Step 2. HiveOS device can resume to normal. | | |
| Test Result |  | | |
| Comment | Should record how much time HiveOS takes to resume to normal. | | |

## Other Test Case

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

<Firstly, list all cli that this feature has one by one>

<CLI test case>

**Enable L7 engine**

*[no] application identification shutdown*

**Enable application reporting globally**

*[no] application reporting enable* (enable/disable/auto) (Default to "disable")

**Enable application for reporting**

*[no] application reporting app-id <string> enable*

**Set application collection period and report period**

*application reporting collection-period <num> report-period <num>*

**Show application reporting configuration**

*show application reporting configuration*

### [no] application identification shutdown

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | L7\_Engine\_CLI\_01 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use |  | | |
| Description | CLI verification: [no] application identification shutdown | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition |  | | |
| Test procedure | 1. Check default setting of L7 engine 2. Turn off L7 engine:   CLI: application identification shutdown   1. Show running configuration to check L7 engine is disabled. 2. Save running configuration, and reboot HiveOS device. 3. After reboot, show running configuration again to check L7 engine is disabled. 4. Turn on L7 engine:   CLI: no application identification shutdown   1. Show running configuration to check L7 engine is enabled. 2. Save running configuration, and reboot HiveOS device. 3. After reboot, show running configuration again to check L7 engine is enabled. | | |
| Expect result | Step 1: The default setting of L7 engine is “enable”.  Step 3: As per running configuration, the L7 engine has been disabled.  Step 4. As per running configuration, the L7 engine has been disabled.  Step 7. As per running configuration, the L7 engine has been enabled.  Step 9. As per running configuration, the L7 engine has been enabled. | | |
| Test Result |  | | |
| Comment |  | | |

### [no] application reporting enable

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_CLI\_01 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use |  | | |
| Description | CLI verification: [no] application reporting enable (enable/disable/auto) | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition |  | | |
| Test procedure | 1. Check default setting of application reporting global mode. 2. Set application reporting global mode as “enable”   Application reporting enable   1. Show running configuration to check applicaton reporting global mode is “enable”. 2. Save running configuration, and reboot HiveOS device. 3. After reboot, show running configuration again to check application reporting global mode is “enable”. 4. Issue CLI “no application reporting enable” to disable application reporting. 5. Show running configuration to check application reporting is disabled. 6. Save running configuration, and reboot HiveOS device. 7. After reboot, show running configuration again to check application reporting is disabled. 8. Set application reporting global mode as “disable”:   application reporting disable   1. Show running configuration to check application reporting global mode is “disable”. 2. Save running configuration, and reboot HiveOS device. 3. After reboot, show running configuration again to check application reporting global mode is “disable”. 4. Set application reporting global mode as “auto”:   application reporting auto   1. Show running configuration to check application reporting global mode is “auto”. 2. Save running configuration, and reboot HiveOS device. 3. After reboot, show running configuration again to check application reporting global mode is “auto”. | | |
| Expect result | Step1: The default value of application reporting global mode is “disable”  Step 3: As per running configuration, the application reporting global mode is “enable”.  Step 5. As per running configuration, the application reporting global mode is “enable”.  Step 7. As per running configuration, the application reporting is disabled.  Step 9. As per running configuration, the application reporting is disabled.  Step 11. As per running configuration, the application reporting global mode is “disable”.  Step 13. As per running configuration, the application reporting global mode is “disable”.  Step 15. As per running configuration, the application reporting global mode is “auto”.  Step 17. As per running configuration, the application reporting global mode is “auto”. | | |
| Test Result |  | | |
| Comment |  | | |

### [no] application reporting app-id <string> enable

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_CLI\_02 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use |  | | |
| Description | CLI verification: [no] application reporting app-id <string> enable | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition |  | | |
| Test procedure | 1. Enable a application for reporting   application reporting app-id <string> enable   1. Show running configuration to check application has been enabled for reporting. 2. Save running configuration, and reboot HiveOS device. 3. After reboot, show running configuration again to check application has been enabled for reporting. 4. Disable a application for reporting   no application reporting app-id <string> enable   1. Show running configuration to check application has been disabled for reporting. 2. Save running configuration, and reboot HiveOS device. 3. After reboot, show running configuration again to check application has been disabled for reporting. 4. Repeat step 1 to 8 with CSV format. | | |
| Expect result | Step 2: As per running configuration, the application has been enabled for reporting.  Step 4: As per running configuration, the application has been enabled for reporting.  Step 5: As per running configuration, the application has been disabled for reporting.  Step 7: As per running configuration, the application has been disabled for reporting.  Step 9: As previous expected results. | | |
| Test Result |  | | |
|  |  | | |

### application reporting collection-period <num> report-period <num>

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_CLI\_03 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use |  | | |
| Description | CLI verification: application reporting collection-period <num> report-period <num> (Default: 3600 seconds, Range: [60 – 3600]) | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition |  | | |
| Test procedure | 1. Check default setting of collection period and report period. 2. Set application reporting both collection period and report period as 60   Application reporting collection-period 60 report-period 60   1. Show running configuration to check application reporting collection period and report period. 2. Save running configuration, and reboot HiveOS device. 3. After reboot, show running configuration again to check application reporting collection period and report period. 4. Set application reporting both collection period and report period as 3600   Application reporting collection-period 3600 report-period 3600   1. Show running configuration to check application reporting collection period and report period. 2. Save running configuration, and reboot HiveOS device. 3. After reboot, show running configuration again to check application reporting collection period and report period. 4. Set application reporting both collection period and report period as 600   Application reporting collection-period 600 report-period 600   1. Show running configuration to check application reporting collection period and report period. 2. Save running configuration, and reboot HiveOS device. 3. After reboot, show running configuration again to check application reporting collection period and report period. 4. Set application reporting both collection period and report period as 700   Application reporting collection-period 700 report-period 700   1. Show running configuration to check application reporting collection period and report period. 2. Set collection period as 600 and report period as 700   Application reporting collection-period 600 report-period 700   1. Show running configuration to check application reporting collection period and report period. 2. Set application reporting both collection period and reporting period as 90.   Application reporting collection-period 90 report-period 90   1. Show running configuration to check application reporting collection period and report period. | | |
| Expect result | Step 1: The default setting of both collection period and report period are 3600.  Step 3: As per running configuration, both application reporting collection period and report period are 60.  Step 5: As per running configuration, both application reporting collection period and report period are 60.  Step 7: As per running configuration, both application reporting collection period and report period are 3600.  Step 9: As per running configuration, both application reporting collection period and report period are 3600.  Step 11: As per running configuration, both application reporting collection period and report period are 600.  Step 13: As per running configuration, both application reporting collection period and report period are 600.  Step 14: HiveOS only receive integer divisor of 3600 as collection and report interval, it prompts this is an invalid parameter.  Step 15: As per running configuration, both application reporting collection period and report period are 600.  Step 16: The collection interval should be the integer divisor of report interval, HiveOS prompts this is an invalid parameter.  Step 17: As per running configuration, both application reporting collection period and report period are 600.  Step 18: The collection interval and report interval should be the multiple of 60. HiveOS prompts this is an invalid parameter.  Step 19: As per running configuration, both application reporting collection period and report period are 600. | | |
| Test Result |  | | |
|  |  | | |

### Show application reporting configuration

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_CLI\_04 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use |  | | |
| Description | CLI verification: show application reporting configuration | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200-WP,BRAP330,BRAP350, | | |
| Pre-condition |  | | |
| Test procedure | 1. Display L7 application reporting configuration   Show application reporting configuration | | |
| Expect result | Step 1: Display L7 application reporting configuration like following:  AP330\_L7#show application reporting configuration  Application reporting: automated  Application reporting collection-interval: 600  Application reporting report-interval: 600  Application enabled:  1: 12306.cn  2: 126.com  3: 2345.com  4: 39.net  **5: 3COM-TSMUX** | | |
| Test Result |  | | |
| Comment |  | | |

### [no] application reporting watch-list <string\_256> enable

|  |  |  |  |
| --- | --- | --- | --- |
| Case ID | ApplicationReporting\_CLI\_05 | | |
| Priority | Middle | Automation Flag | No |
| Topology to use |  | | |
| Description | CLI verification: application reporting watch-list <string\_256> enable | | |
| PlatformDependence | AP: AP110,AP120,AP121,AP141,AP170,AP320,AP340,AP330,AP350,  BR: BR200,BR200-WP,BRAP330,BRAP350,  Note: when draft this test case, HiveOS Millau 6.0r2c AP370/390 does not support application discovery. And HiveOS Geneva release is not related with AP370/390. As per plan, AP370/390 will support it by 2013 fall. | | |
| Pre-condition |  | | |
| Test procedure | 1. Set a watch list for application discovery to report relevant app-ids forcibly   application reporting watch-list 1 enable   1. Remove application from watch list   no application reporting watch-list 1 enable   1. Check the maximum number of applications which watchlist can afford | | |
| Expect result | Step1. Add a new application into watchlist successfully.  Step 2. Remove application from watchlist successfully.  Step 3. Watchlist supports up to 7 applications. | | |
| Test Result |  | | |
| Comment | Draft this case for application discovery which is a L7 application enhancement involved into HiveOS since Geneva release. | | |

## GUI Management-HiveManager

<List HM test case or test log>

## GUI Management-HiveUI

<List HiveUI test case or test log>

## Typical issue Test Case