Omni Switch 6450/ 6250 / 6350

Release 6.7.1.108.R04

The following is a list of issues that have been identified and corrected in AOS software release. This document is intended to be used as a pre-upgrade guide and does not replace the Release Notes which are created for every GA release of software.

Important Notice: For a copy of software release not posted on the Web or if you have any question or concern please contact Alcatel's Technical Support Department.

Alcatel·Lucent
Enterprise

Problems Fixed Between Builds 76 and 108

PR **221471** Build:
Summary: OS6250 does not accept redirect-URL which has more than 128 characters.
Explanation: Changes are done to accept 252 characters as the redirect URL length.

PR **222426** Build: 6.7.1.93.R04
Summary: Switch rebooting with pmd file after entering no snmp command station
Explanation: Code changes done to update the community map only if username is present.

PR **221826** Build: 6.7.1.C.21
Summary: OS6450 - LBD auto-recovery timer beyond 300 secs does not work
Explanation: LBD auto recovery timer beyond 300 secs does not work

PR **220749** Build: 6.7.1.C.19
Summary: unable to ssh/console to the switch and mempart alloc warning messages seen in logs
Explanation: Memory leak prevented in IPMS module.

PR **222067** Build: 6.7.1.89.R04
Summary: Primary switch in the 2 unit stack crashed and generated the PMD file..
Explanation: IPMS CMM does a sync of the same flow information repeatedly. When the OMEM buffers are consistently being used and if the sync takes place, more and more memory is allocated which leads to memory depletion. Code changes done to avoid un-necessary flow information sync. .

PR **215398** Build: 6.7.1.C.20
Summary: OS6450 DHCP packets looping on linkagg between 6900-VC and 6450 stack. Inconsistent DHCP issue during lease refresh with Wyse thin client.
Explanation: To bring the logic of identifying if a packet has already been routed in the hardware in 66x like that of 64x.

PR **222017** Build: 6.7.1.97.R04
Summary: OS6450 multicast source timeout is not proper with slow source packet rate.
Explanation: Fix is to timeout the source flows with respect to "ip multicast source timeout" command with more accuracy.

PR **220353** Build: 6.7.1.98.R04
Summary: Issue with authentication for supplicant 802.1x devices caused by 'reason 38 authentication timeout.
Explanation: Removed old authentication context, to enable smooth authentication of the user in fresh context.

PR **222666** Build: 6.7.1.101.R04
Summary: 801x display issue is notice after the takeover.
Explanation: Update the secondary cmm database regarding the flushing of entries when the state changes from connecting to disconnected

PR **220724** Build: 6.7.1.86.R04
Summary: Memory leak notice due to SSH session.
Explanation: Memory leak prevented with SW modification

PR **223300** Build: 6.7.1.108.R04

Summary: Even after the polling interval, Radius server operation status remains down after re-enabling polling with radius-health-check.

Explanation: Display correct status of Radius Server when Radius Health Check is enabled

Known Issues:

PR **222688** Build:

Summary: Traffic is getting dropped while testing open flow with actions drop/out port/set with vlan.

Explanation: Expected behavior: Apply actions are supported only in software, the rate of packets handled in software will depend on the CPU load at that point of time.

PR **222243** Build:

Summary: Set source mac bucket action fails to work (for write actions)

Explanation: Set source mac action cannot be supported in hardware for both write actions and apply action.

PR **222276** Build:

Summary: Untagged frames gets processed with given set of group actions for 'with VLAN tag' flow condition

Explanation: Match criteria set as untagged packets in API mode allows tagged traffic as well.

PR **222521** Build:

Summary: Functionality of IDLE_TIMEOUT is similar to HARD_TIMEOUT. Flow entry gets removed after the IDLE_TIMEOUT.

Explanation: With idle timeout configured, flow is removed from the switch when ideal timeout expires even with continuous traffic flow i.e. Ideal timeout behaves as hard time out.

PR **220338** Build:

Summary: High CPU due to the task "taSLNEvent"

Explanation: Whenever MAC movement (from one port to another) happens, this is processed in software resulting in higher CPU utilization. This is expected SW behavior.

PR **222922** Build:

Summary: In API mode when TOS is set as match criteria , input packet Flows are not matched

Explanation: In API mode TOS value cannot be used as a match criteria along with a match criteria requiring a specific Ethernet type.

New Features:

1. Source Learning Enable/Disable for Non-Metro units

Platforms Supported: OmniSwitch 6450, OmniSwitch 6250 and OmniSwitch 6350

Hosted AOS SW Release: 671.108.R04

Hash Chain Length enhancement feature is to modify the depth of the hash chain length of FBD table (bucket size) used while writing MAC addresses in ASIC in the AOS platform.

Collisions while writing MAC addresses in ASIC occurs mainly due to poor hashing algorithm being used. Hashing mode XOR has more chances of collision, i.e. same hash-index being assigned to different set of source/destination address. Changing the hash-mode to CRC is a more efficient technique than XOR mode. Since the mac-collision is hardware and algorithm dependent, we may further reduce the probability of mac-collision by using the provision in hardware to increase the bucket size from current set value of 4 to a higher value of 8. This setting can be controlled from our software.

This feature requirement is to change the Hash Length of the FDB table from DEFAULT (bucket size is 4) to EXTEND (bucket size is 8) through a configuration command with mode options as DEFAULT or EXTEND. By default, the Hash Length value will be DEFAULT i.e. bucket size as 4.

The modification of hash length done at runtime will be effective only during boot-up and hence when this command is executed by the user, a warning message is displayed on the console that the change will be effective only after the next reload / reboot of the switch / stack.

Usage:

hash-control chain-length {DEFAULT | *EXTEND}*

This command configures the hash chain length in the HW. Depending upon this configuration, the hashing bucket size for the hardware table will be decided.

The allowed CLI combinations are as follows:

hash-control chain-length default
hash-control chain-length extend

Syntax Definitions:

DEFAULT	If configured, Hash Chain Length will be set to 4.
EXTEND	If configured, Hash Chain Length will be set to 8.

Defaults:
DEFAULT is the default value for this CLI

Usage Guidelines:

1. Use this command to change the hash chain length from EXTEND to DEFAULT mode and vice-versa.
2. After using this command, save the configurations using “write memory” command and reload the switch to reflect the hash length changes in the switch.

Display Commands
show hash-control chain-length

This CLI displays the configured value for the depth of the hashing bucket

Usage Guidelines
The * symbol displayed in the show output (FDB Hash Chain Length = EXTEND*) indicates that the configured hash chain length will be applied only after reloading the switch.

Example

```
/* sample output */
    ➔ show hash-control chain-length
FDB Hash Chain Length = DEFAULT

    ➔ show hash-control chain-length
(*=new hash chain length config will be applied after reboot)
FDB Hash Chain Length = EXTEND*
```

Configuration Snapshot

➔ Show configuration snapshot chassis

```
! Chassis :
hash-control chain-length EXTEND
```

Limitations:

If any modification in hash chain length is made by the user, it is important to reload/reboot of the switch/stack. Only the configured value will be displayed for SNMP and Webview. Any configuration change made with respect to hash chain length in CLI/SNMP/Webview requires switch reboot to get the configuration changes applied in the switch.

Without doing reboot (after change in hash length), actions like inserting a new NI or doing takeover should not be done. Therefore a warning message is reflected in console to indicate that the user must reboot/reload the switch after the change in hash chain length

2. Source Learning Enable / disable for Non-Metro units

Platforms Supported: OmniSwitch 6450, OmniSwitch 6250 and OmniSwitch 6350

Hosted AOS SW Release: 671.108.R04

Currently we have a limitation in AOS, that enabling / disabling of source-learning can be done only in Metro OmniSwitch units. This feature implementation now allows source-learning enable / disable for non-metro units also.

This provides the user an option to enable or disable source MAC learning on a specified port or linkagg. Any port, except those already activated with ‘software’ learning, can be set to source learning enabled / disabled by users. This feature is restricted to maximum of 48 ports (including Linkagg ports) across system.

No new CLI is added. Same CLI earlier used for metro units can be now used for non-metro units as well.

Limitations:
None

3. Support for packet modification actions for group type ALL in OpenFlow

Platforms Supported: OmniSwitch 6450 and OmniSwitch 6250

Hosted AOS SW Release: 671.108.R04

Current design of QOS can execute a set of actions for a packet when there is a match for one condition. In order to support a flow which can execute more than one set of actions for a particular condition, we use the concept of Groups and Buckets in OpenFlow. The limitation earlier in Groups and Buckets is that, packet modification actions were not supported. This feature implementation enables the support of packet modification action for group type ALL.

Usage:

debug OpenFlow flow-id <value>

Output will change to reflect the new flow information in the form of buckets

Example;

```
debug OpenFlow flow-id 5

Flow ID: 5
       Logical Switch: LS_1
       Priority: 0
       Idle Timeout: 0
       Hard Timeout: 0
      Flow Type: Wildcard
Match:  Ingress Port: 0/5, Src MAC: 00:00:00:00:00:01/ff:ff:ff:ff:ff:ff, Dst MAC: 00:00:0d:00:00:01/ff:ff:ff:ff:ff:ff,
VLAN Priority: 3, Ether-type: 800
Apply Action(s): VLAN: 142, VLAN Priority: 0x300, Dst MAC: 00:00:00:00:0d:01, Out: 1/1;,
Write Action(s): Group: 1
Bucket: 1
Action: VLAN: 143, VLAN Priority: 0x4, Drop
Bucket: 2
Action: VLAN: 144, VLAN Priority: 0x5, Out: 1/1;
Bucket: 3
Action: VLAN: 145, Out: 1/3; 1/5;
```

For the single flow match criteria different bucket actions can be applied using Write actions.

Use case:

OpenFlow controller attempts to push the below flow to OpenFlow enabled 6450.

- *in_port=1, actions=output:2,push_vlan:0x8100,set_field:5->vlan_vid,output:3.*

This flow means that: The untagged frame from port 1, should be sent out of port 2 without any modification to the packet. But as a tagged frame out of port 3. For the frame sent out of port 3 having tag id 5 (VLAN 5).

Earlier, in the switch side, validation of the flow fails because packet modification action is not allowed for group type ALL (SET_VLAN of VLAN 5) and the below errors are seen in the switch,

```
>>>>>>
          OFCMM_LOG_GROUP   ofcmm_group_validate:222   OFLS ID: 1 gid: 1 type: 0 bkts count: 96
          OFCMM_LOG_GROUP   ofcmm_bucket_validate:174   OFPGMFC_BAD_BUCKET return
          OFCMM_LOG_PROTO   ofcmm_proto_send_error:108   OFP error msg. Type: 6 code: 12
          OFCMM_LOG_PROTO   ofcmm_proto_send:48   send 4 1 108 0x4
          OFCMM_LOG_PROTO   ofcmm_protov4_rcv_group_mod:1327
          OFPETV4_GROUP_MOD_FAILED. cmd: 0 xid: 4 len: 96
<<<<<<
```

As per Openflow 1.3.1 specification, ALL is a required group type. With this new implementation above use case would be accepted and traffic would work functionally as expected in the use-case.

Limitations:

1. The support for different types of actions can be done only in software so, packets which exceeds the predefined rate limit for OpenFlow (1024 pps) will be dropped.
2. The concept of group type ALL does not work when we try to send the packets to non-primary Nis.

4. Change in the order of actions in reply messages

Platforms Supported: OmniSwitch 6450 and OmniSwitch 6250

Hosted AOS SW Release: 671.108.R04

Multipart flow statistics messages and Multipart group descriptor messages show the list of actions associated with the flow and group respectively. The current implementation fills the actions in the multipart flow statistics reply messages and multipart group descriptor messages in the same order the actions were received in the ‘flow mod’ message from the OpenFlow controller.

Usage:
The output of “*debug OpenFlow flow-id all*” is expected to display the actions in the order it was received in the ‘flow mod’ message from the OpenFlow controller.

Example:
Debug OpenFlow flow-id 7.
Flow ID: 7
Logical Switch: LS_1
Priority: 0
Idle Timeout: 0
Hard Timeout: 0
Flow Type: Wildcard
Match: Ingress Port: 1/27, Src MAC: 00:00:0a:00:00:18/ff:ff:ff:ff:ff:ff, Dst MAC: 00:00:00:00:00:18/ff:ff:ff:ff:ff:ff, VLAN: 140, VLAN Priority: 5,
Ether-type: 800, Src IP Address: 172.16.0.0/16;
*Apply Action(s): Strip VLAN, Push VLAN, VLAN: 145, VLAN Priority: 0x500, Dst MAC: 00:00:00:00:0b:53, IP TOS: 0x28, Out: 1/3*7;

Actions mentioned under Apply actions need to follow the order mentioned in the controller and the Reply part messages should possess the same order of actions

Limitations:
Openflow version 1.0 does not support concept of Multipart reply for groups. But for individual flows statistics message (OFPST_FLOW) we send out the list of actions associated with the flow in the actual order it was received

5. Support for PUSH_VLAN

Platforms Supported: OmniSwitch 6450 and OmniSwitch 6250

Hosted AOS SW Release: 671.108.R04

PUSH VLAN action inserts new VLAN tags to the incoming packets. It is similar to the Q-in-Q service. Newly pushed tags should always be inserted as the outermost tag in the outermost valid location for that tag i.e. when a new VLAN tag is pushed, it should be the outermost tag inserted, immediately after the Ethernet header and before other tags.

Related functions done as part of this feature:

1. OmniSwitch has to count the number of VLAN tags, number of SET_VLAN and VLAN_PCP actions that the flow is trying to push. When the controller tries to insert a flow which has more than 1 tag, the actions get replaced if it is already present.
2. In a case where there are PUSH_VLAN actions set without SET_VLAN, the switch should respond with an error OFPBACV4_BAD_SET_ARGUMENT.
3. PUSH VLAN cannot be supported in hardware. So, VLAN headers needs to be pushed by the software only.

4. A priority of zero and the tag of zero are used for the new tag inserted using PUSH VLAN.
5. Modifies the output of ‘debug openflow flow-id all” if needed.

Usage:
The output of “debug OpenFlow flow-id all” is expected to display the PUSH VLAN actions received from the OpenFlow controller.

```
debug OpenFlow flow-id 7.
Flow ID: 7
Logical Switch: LS_1
Priority: 0
Idle Timeout: 0
Hard Timeout: 0
Flow Type: Wildcard
Match:  Ingress Port: 1/27, Src MAC: 00:00:0a:00:00:18/ff:ff:ff:ff:ff:ff, Dst MAC: 00:00:00:00:00:18/ff:ff:ff:ff:ff:ff,
VLAN: 140, VLAN Priority: 5, Ether-type: 800, Src IP Address: 172.16.0.0/16;
Apply Action(s): Strip VLAN, Push VLAN, VLAN: 145, VLAN Priority: 0x500, Dst MAC: 00:00:00:00:0b:53, IP
TOS: 0x28, Out: 1/37;
```

Limitations:

1. Ether types other than 0x8100 is not supported
2. PUSH_VLAN only adds VLAN tags. The other fields in VLAN tag cannot be modified with PUSH_VLAN.
3. For every push VLAN action there has to be a corresponding set or modify VLAN action field.
4. A maximum of 8 VLAN tags can be inserted to a packet. If we have a case where packets coming in with VLAN tags, and if we try to add more tags which leads to the number of tags in the packet becoming greater than 8, we stop adding more tags to the packets

6. Support for Apply Actions

Platforms Supported: OmniSwitch 6450 and OmniSwitch 6250

Hosted AOS SW Release: 671.108.R04

Earlier, AOS fully supported only the Write Actions instruction type in OpenFlow. Instructions of type *Apply-Actions* were also considered as *Write-Actions* when a flow is saved in the database and action structure used Write Actions. Now with this feature implementation, AOS look at these OpenFlow actions as two separate entities and supports the actions separately for both these types.

Related functions done as part of this feature:

1. Output of “debug OpenFlow flow-id all” should reflect the Apply-Actions also.
2. Multipart reply packets for Apply-Actions maintains the order in which the flow was received.
3. Incoming packets to OF-NI matching a flow which has Apply-Actions applies the actions to the incoming packet in the order the actions were received in the first place.
4. If we have a flow with both Apply-Actions and Write-Actions, we start with Apply-Actions first and then the Write-Actions.
5. If there are more than 2 instructions of the same type, Switch responds with error to controller.

Limitations:

1. Support a maximum of 32 actions for Apply-Actions type in every flow. If the number of actions are greater than 32, switch should respond with an error.
2. Support for Apply-Actions can be done only in software so, packets which exceeds the predefined rate limit for OpenFlow will be dropped.