Testing-Interop Working Group

Interoperability Event Technical Issues Report

May 12th- 16th 2014

Version 0.4

ONF TR-503

ONF Document Type: SDN Library
ONF Document Name: Interoperability Event Technical Issues Report – May 2014

Disclaimer

Open Networking Foundation
2275 E. Bayshore Road, Suite 103, Palo Alto, CA 94303
www.opennetworking.org

1 Introduction

The bi-annual interoperability event was conducted in the SDN labs at INCNTRE and BII from May 12th- 16th 2014. For the first time the event was hosted in two labs across different geographies. Several hardware and software switch vendors, controller vendors and test tool developers were part of the interoperability plug fest.

The focus of the interoperability event was predominantly to uncover potential incompatibilities across switch and controller implementations based on the Open Flow 1.3 version of the switch protocol. Test tool vendors take the opportunity to make the switch and controller implementations of the test tools more robust.

A common test plan for the interoperability event was developed that included test cases from the 1.3 conformance test suite, switches with multiple controllers in simple and complex topologies. The test plan was used as a baseline for all tests executed by the stake holders as part of the event. All the issues uncovered were assessed with respect to compliance with the 1.3.2 version of OpenFlow switch specification.

This document summarizes the list of major issues uncovered as part of the plug fest, reasons for the same, resolutions, recommendations for future action and status of issues being worked on within the ONF. The purpose of this document is to increase awareness among the OpenFlow community and to help avoid such interoperability issues in the future.

In line with the policy of how the plug fests are conducted, vendor names are not explicitly referred to as part of this report to protect the confidentiality of participants.

Please refer to the whitepaper for more details on the event.

2 Switch and controller Issues

There was a wide range of issues that were identified at the plug fest event. The reason for the problems uncovered include hardware and software bugs on the products, features not being implemented, misunderstanding of switch specification and faulty implementation due to ambiguity in switch specification. The impact due to interoperability issues found vary from incorrect or partially correct behavior, crashes and stability issues.

The interoperability issues are categorized into the following sections depending on the key factor that contributed to the interoperability issue:

- Bugs on the Open Flow switches or test tools implementing switch functionality
- Bugs on the Open Flow controllers or test tools implementing controller functionality
- Incorrect or ambiguous Open Flow switch specification

While all the sections in this document is likely to be relevant for implementers and the ONF standards groups, the section on specifications is to be reviewed in-depth by the standards groups to help bring in sufficient clarity in future versions of the switch

specification. As they proceed with their product development efforts, implementers are encouraged to engage within the test and interoperability group to seek additional clarity if some of these issues are still unresolved.

2.1 Open Flow Switch Issues

The section below captures the list of issues uncovered that were attributed to bugs found on the open flow switch. Switch developers should watch out for these interoperability issues when they develop their product. Controller vendors should be aware of potential abnormal behavior of switches and have means of handling these issues gracefully should these scenarios occur in the field.

Impact level is an indicator of the severity of the bug should it be uncovered in a real world deployment. Where reasons are not obvious, additional comments elaborate the rationale for the impact level of the bug.

The extensibility team is additionally working on potential enhancements to the OpenFlow switch specification (EXT-510-513), that should further help switch implementers avoid similar interoperability issues in future. Appendix-A has more details.

2.1.1 Session establishment

Unsupported multipart table messages: Impact level: High
When establishing the control channel, controller sends OFPMP_TABLE_FEATURES multipart request message to switches and inspects the reply message strictly. Some switches fail to establish the control channel due to not support multipart table feature message. Other switches' reply messages lack OFPTFPT_MATCH and OFPTFPT_WILDCARDS information which makes the switches believe the control channel has been established but the controller does not.

Incorrect port advertisement: Impact level: High

Some switches report incorrect port description to the controller on connection establishment. As a result of this, the packets are sent to the wrong port when the controller eventually tries to send the packets.

Default flow for table-miss: Impact level: High

The default action of Table-miss flow is drop. After the establishment of the control channel, the controller does not install Table-miss flow actively nor a flow entry to match topology discovery packets (Eth_type=0x88cc). Because of these reasons, the discovery packets are dropped and the controller is unable to complete topology discovery.

It is recommended that the controllers install a Table-miss entry actively after the establishment of the control channel.

OF 1.0 and 1.3 negotiation: Impact level: Med

During initial connection with OF v1.0, the switch disconnected and attempts to reconnect with OF v1.3 while the controller is still attempting to queue an OF v1.0

default flow. This results in failure of the controller not being able to connecting to switch. The issue was seen only while testing with multiple switches.

TCP/IP request flooding on control channel: Impact level: Med

In a test scenario, it was observed that the switch was sending TCP SYNs too fast. By the time the test tool replies with a SYN ACK, the DUT was sending more SYNs. This causes the two sides to not agree and do a TCP reset, causing the test to time out and fail on control connection establishment. It is recommended that the vendors define the prerequisites and life cycle of the establishment of the control channel more explicitly.

2.1.2 Tables

Group_mod delete un-supported: Impact level: High
Switch was unable to remove table group with bucket.

Additional comments: *When the group_mod DELETE message include buckets, an error INVALID_GROUP was generated by the switch.*

Loose table definition: Impact level: High

Switches that support multiple tables have limitations on what each table can accomplish. For instance, Table 0 only support Goto Instruction or only support match Destination MAC and Ingress Port fields. Making table 0 as read only can also have cause discovery to fail.

Additional comments: *Inconsistent implementation of tables can cause interoperability issues. A recommendation is also being made to enhance the OpenFlow specification in order to make implementation consistent across all vendors.*

Incorrect reporting of number of tables supported: Impact level: Med

Switch reports incorrect number of tables supported. It caused issues where the packets from the switch do not reach the controller.

Persistence of table definition: Impact level: Low

Some switches only support single table in the normal operational mode. In order to support multi tables, the device should switch to another operational mode and reboot.

Multi-table support: Impact level: Info

Some switches do not support multi tables.

Group table support: Impact level: Info

Some devices do not support Group Tables.

Support for AUX channel: Impact level: Info

It was found that not all switches implement auxiliary connections (implementing auxiliary connections is not mandatory). No interoperability issues were found however with respect to the same.

2.1.3 Actions

Write action support: Impact level: High
A switch that supports write action crashes after controller sends a flow_mod message with WriteAction Instruction.

Write action support: Impact level: High

Some switches do not support WriteAction Instruction. After controller sends a flow_mod message with WriteAction Instruction, switches reply with an error message with error code OFPBIC_UNSUP_INST.

Flow entry with Group action: Impact level: Med

Controller could install Group Table entry but could not install flow entry with Group Action.

Null bucket action: Impact level: Low

Switch replies with error message when controller attempts to install a Type ALL Group Table entry with the bucket action is null.

2.1.4 Matching

Packet_matching_basic: Impact level: High
DUT ignores matching on ICMPv4 type.

Additional comments: *Since the DUT supports field matching only on specific table types this caused a mismatch in expectation between the switch and controller.*

Packet match_multiple: Impact level: High

Flow does not get modified correctly according to the match field. In some cases the switch was unable to correctly program and push VLANID flows.

ARP packet forward: Impact level: High

Controller sent ARP reply via a packet out for the switch to forward out of a specific port. ARP packets were not forwarded by the switch. Controller used group table to flood ARP which worked.

Packet forwarding to wrong port: Impact level: High

When controller send packet out message to switch, the packets are sent to the wrong port.

Flow_mod priority: Impact level: High

The priority of Flow mod messages sent by the controllers is zero. Some switches treat this flow entry as Table-miss entry and this leads to some mistakes with the expected behavior.

Additional comments: *According to the spec, only the flow entry wildcards all match fields and has the priority zero should be treated as Table-miss entry, so the switch should fix this bug.*

Incorrect handling of flow_mod message: Impact level: High

When implementing L3 forwarding, the controller's flow_mod message did not have the destination MAC included in the match fields leading to the switch replying with an error message. The spec does not require a pre-requisite for the destination MAC and it does not need to be present in the match.

Multiple match: Impact level: Low

Some switches are not able to implement Pop VLAN and Push MPLS at the same time.

2.1.5 Message handling

Table query using multi-part message: Impact level: High
When the controller sends a multipart_request to query table feature and ports, the switch disconnects.

Barrier messages: Impact level: High

Controller uses Port_Mod to change port status to up, also uses barrier message for each port mod. Switch replied to first barrier but did not continue to respond after the first one.

Padding field for messages: Impact level: High

Padding field of OFPMP_TABLE_FEATURES request message sent by some controllers is not zero. Some switches are not able to recognize the message and give no response to the request message.

***Additional comments:** OpenFlow 1.3 specification defines the padding field could be non-zero value, hence switches need to recognize the message.*

Queue config request: Impact level: High

When controller sends a get queue config for OFPP_MAX_PORT, the switch crashes.

Invalid length on feature request response: Impact level: Med

Switch response to feature request with invalid length and leads to TCP timeouts.

***Additional comments:** Bad length message seems to cause the time out, and leaves sessions hanging. New TCP port has to be used if systems not entirely cleaned.*

2.1.6 Meters

Ungraceful handling of bad value for meter mod: Impact level: High

Controller sent meter mod with a bad band type leading to switch crash.

***Additional comments:** Switch should have gracefully replied with an error message*

OFPET_METER_MOD_FAILED with OFPMMFC_BAD_BAND code.

Invalid response for meter query: Impact level: Low
Switch provides an invalid response to get_meter status from the controller

Unsupported meter_rate: Impact level: Low

Switch didn't accept meter add with rate 100,000 kbps, it returned with meter_mod failed bad band value.

Switch support for metering: Impact level: Info

Some devices do not support Metering

2.1.7 Switch implementation

Mandatory port type support: Impact level: High
Some switches do not support reserved port type ALL. Reserved ports ALL, CONTROLLER, TABLE, IN_PORT, ANY are required to be implemented for all types of open flow switches (both open flow only and hybrid switches).

Incorrect handling of OFPP_ANY: Impact level: Med

If an in_port is set to 0xFFFFFFFF in packet_out message, switch does not handle correctly leading to a crash.

2.1.8 Error handling

Meter_mod with unknown type: Impact level: Med
When test tool sends a meter mod message with a band of unknown type, the switch crashed.

Handling of multiple OF headers: Impact level: Med

When a test tool sends a message with two OF headers, the switch crashed.

***Additional comments:** This happened because of problem parsing message with bad length in OF header.*

2.1.9 Multiple controllers

Flow entry not preserved on switchover: Impact level: High
When a controller becomes a new master, all the flow entries installed by the old master are deleted by the switch.

Forwarding stops on controller switchover: Impact level: High

With multiple controllers, switch successfully recognized master/slave roles for multiple controllers, but forwarding stopped once detection is done.

Invalid generation id: Impact level: Med

When a controller generates an invalid generation_id, some switches do not examine this field nor reply with an error message.

Multiple controller support: Impact level: Info

Some devices do not support multi controllers.

2.2 Open Flow Controller Issues

The section below captures the list of issues uncovered that were attributed to bugs found on the open flow controller. Controller vendors should watch out for these interoperability issues when they develop their product. Switch vendors should be aware of potential abnormal behavior by the controller and have means of handling these issues gracefully should these issues occur in the field.

Impact level is an indicator of the severity of the bug should it be uncovered in a real world deployment. Where reasons are not obvious, additional comments elaborate the rationale for the impact level of the bug.

2.2.1 Session establishment

Malformed data path ID: Impact level: High
Malformed data path ID was sent by controller.

Connection establishment with OF 1.0: Impact level: Med

During initial connection with OF v1.0, the switch disconnected and attempts to reconnect with OF v1.3 while the controller is still attempting to queue an OF v1.0 default flow and this results in failure of connecting to switch. This issue happens only when the controller is attempting to communicate with multiple switches.

Invalid output port setting on startup: Impact level: Med

When the controller has not learned any MAC addresses of the hosts, it would try to install a flow entry on the switch with action Output to PORT_ANY which is incorrect.

***Additional comments:** Spec declares PORT_ANY cannot be applied as an output port*

Incorrect handling for s/w that don't support fragmentation: Impact level: Med

The switch announced that it does not support reassemble IP fragments in the OFP_SWITCH_FEATURE reply message. After receiving this message, controller still sets OFPC_FRAG_REASM = 1 in the OFPT_SET_CONFIG message which makes the switch reply back with error.

Incorrect handling for pure OF switches: Impact level: Med

Dynamic path paving uses port type Flood which is optional for pure OF switches. Because of this the switch failed to process the ARP packet.

2.2.2 Matching

Incorrect implementation: Impact level: High
Controller did not support Ingress Port in the match field (basic bug).

2.2.3 Message handling

Port type local not handled properly: Impact level: High

Controller crashed on get port stats message while testing with a switch that supports port type local. Port type "local" was not expected.

Incorrect parsing when device description is none: Impact level: High

If the switch returns none for device description request, the controller processes the message with an error. In a few other cases, the controller crashes on error with message parsing.

Incorrect parsing: Impact level: Med

Controller parser crashed parsing get flow stats reply. Flows with complex matches were left over from the previous test. It is likely that the controller parser is buggy and could not process the matches in the flows.

Unable to install group entry if multipart table request fails: Impact level: Med

If there's error in the multipart table feature request message, the controller may not be able to install group entry.

2.2.4 Meters

Unable to install flow with meter instruction: Impact level: Med

Controller could install Meter Table Entry but could not install flow entry with meter instruction.

2.2.5 Multiple controllers

Incorrect incrementing of generation_id: Impact level: High

Controller hand off could not be completed because the generation_id of the OFP_ROLE_REQUEST message sent by controller does not change (increase).

The switch replies with an error type OFPET_ROLE_REQUEST_FAILED error code OFPRRFC_STALE message and could not complete controller hand-off.

3 Specification issues

The following section captures interoperability issues identified as part of the plug fest due to ambiguity or lack of clarity in the switch specification. All the scenarios listed below might have resulted in bugs and unpredictable behavior on switch, controller or tools during the plug fest.

These issues have subsequently been discussed in the test and interoperability and the extensibility groups within ONF. As of the date of publishing this document, some of the enhancements to the switch specification are already in place and addressed in a later version of the 1.3 switch specification. Appendix-A in this document captures more details on the recommendations and issue tracking details for proposed enhancements.

3.1 Matching

3.1.1 Matching order for prerequisites

Description:

When multiple match fields are specified in a flow, the prerequisites defined in the specification are insufficient to crisply define behavior. The prerequisites themselves are clear enough, but ordering of prerequisites in flow-mod messages are not clearly defined. It is also true that the specification uses language that allows vendors to choose to use a relaxed standard of enforcement. This is a problem for interoperability.

Impact level: High

Spec wording:

Section 7.2.3.6 and 7.2.3.7 of Open Flow Specification 1.3.2

Recommendations:

Open a ticket with extensibility to clarify order of prerequisite prior to match field that relies on the pre-requisite with three recommendations

1. Switches should be able to parse all fields and determine correct pre-requisites independent of order
2. Could consider specifying specific device prerequisite order in a feature response
3. Specification should specify the order of match fields.

3.1.2 Relaxed implementations

Description:

The specification suggests to send a bad_prerequisite error when prerequisite are not respected by a flow_mod. One switch vendor however used a “relaxed” implementation that allows controllers to add flows without respecting prerequisites. Relaxed implementations in general might pose a problem for the conformance programs; additionally the specification seems to be a bit contradictory in a couple of places leading to incorrect interpretation.

Impact level: High

Spec wording:

“Most match fields have prerequisites, these restrictions are noted in specifications for individual fields (see 7.2.3.7). The prerequisites are cumulative, a match field inherit all the restrictions of its prerequisites (see examples above), and all the chains of prerequisites must be present in the match”

“A switch may implement relaxed versions of these restrictions. For example, a switch may accept no prerequisite at all. A switch must reject an attempt to set up a flow entry that violates its restrictions(see 6.4), and must deal with inconsistent matches created by the lack of prerequisite (for example matching both a TCP source port and a UDP destination port)”

Recommendations:

The controller is allowed to add flows while skipping prerequisites but the switch is expected to implement prerequisites. There are some ambiguity with respect to prerequisites which are already tracked within the extensibility group. Need a recommendation from the extensibility group on the expected behavior while dealing with a switch that has relaxed implementation.

3.2 Meters

3.2.1 DSCP precedence value

Description:

The specification says the switch should increase the DSCP drop precedence value by configured precedence value, but there is clarity required on how to increment the value. Also the definition of "rate" as defined in OF standard is different from rate implemented by some vendors “information rate / line rate”. It is common for SP device vendors to use line rate in rate limiting.

Impact level: High

Spec wording:

"The prec_level field indicates by which amount the drop precedence of the packet should be increased if the band is exceeded. This band increases the encoded drop precedence by this amount, not the raw DSCP value; it always result in a valid DSCP value, and DSCP values that do not encode a drop precedence are not modified".

“For dscp marking to take effect, current rate should be higher than specified meter band rate. The rate value is in kilobits per second, unless the flags field includes OFPMF_PKTPS, in which case the rate is in packets per second. The burst value is in kilobits, unless the flags field includes OFPMF_PKTPS, in which case the burst value is in packets”

Recommendations:

Need clarification on what DSCP values or increments to use. Updating the specification with a DSCP table and recommendation would provide better clarity.

3.3 Tables

3.3.1 Switch table implementation

Description:

There are no specific recommendations on how the switch tables need to be implemented resulting in various switches implementing special behavior. For instance, table 0 is used only for "go to" instructions. Subsequent tables 50, 100, etc. are specialized and difficult to support all possible table structures for controller and test vendors.

Wording in the specification is not explicit. In the context of the specification, it's pretty clear that the table should be writeable. But it isn't explicit, allowing at least one vendor to make it read-only.

Impact level: High

Spec wording:

Sample ambiguous wording from Section 5.1: The flow tables of an OpenFlow switch are sequentially numbered, starting at 0. Pipeline processing always starts at the first flow table: the packet is first matched against flow entries of flow table 0. Other flow tables may be used depending on the outcome of the match in the first table.

Recommendations:

Some recommendations about how to implement tables. TTP when implemented can help with this. Continue discussions in T & I on the expected behavior for switch table implementation until TTP is implemented.

3.4 Error messages

3.4.1 Multiple write actions of the same type

Description:

While testing for behavior where more than one write action instruction of the same type is sent to the switch, it was found that the expected behavior is not clearly specified. Some devices may use the first action of same type, other devices may use second action of same type.

Impact level: Med

Spec wording:

When multiple actions of the same type are required, the apply actions instruction should be used.

Recommendations:

Specification should indicate that an error be sent by the switch when two actions of the same type are sent in a write action instruction.

3.4.2 Multiple instructions in flow_mod message

Description:

The specification requires that the flow entry can have only one instruction of each type, whereas it is possible to construct a flow_mod message where the same instruction is used more than once. This can cause ambiguity if a controller sends multiple instructions as part of the same flow_mod message. No error type/code seems suitable for a situation where the same instruction type is used more than once in a flow_mod message.

Impact level: Med

Spec wording:

"The instruction set associated with a flow entry contains a maximum of one instruction of each type"

Recommendations:

Specification should indicate an error be sent by the switch when multiple instructions of the same type are sent in the instruction set of a flow_mod message.

3.4.3 Multiple violations

Description:

The error messages to be generated when multiple violations occur simultaneously can be ambiguous leading to mismatched expectations around actions to be taken on error code behavior for open flow controllers.

Following are some of the issues in this category reported during the plug fest:

1. If a goto action in table 255 is set to table 254, it is not clear if the switch should generate an invalid table ID or an invalid action error.
2. A group_mod message is sent with a bucket which includes an output action to a bad port. The switch sent a BAD_ACTION, BAD_OUT_PORT error while the testing is expecting a GROUP_MOD_FAILED error with BAD_BUCKET code.

Impact level: Med

Spec wording:

"If the instructions requested contain a Goto-Table and the next-table-id refers to an invalid table the switch must return and ofp_error_msg with OFPET_BAD_INSTRUCTION type and OFPBIC_BAD_TABLE_ID code"

"The action set for each bucket must be validated using the same rules as those for flow mods (Section6.4), with additional group specific checks. If an action in one of the

buckets is invalid or unsupported, the switch should return an ofp_error_msg with OFPET_BAD_ACTION type and code corresponding to the error (see 6.4)"

Recommendations:

Need more clarity and specific recommendations from the extensibility group on expected behavior for the above error conditions and a general recommendation on the default behavior when multiple violations occur simultaneously.

3.5 Session establishment

3.5.1 Handshakes

Description:

While verifying a fail secure scenario that defined flows are not deleted when controller connection is lost. Controller closed the connection, waited 10 second then waited for switch to connect. The switch was sending ECHO_REQUEST messages before the handshake was completed (right after the HELLO message).

The handshake should be completed before switch sends messages (except HELLO) to the controller. The initial handshake is not 3-way which could be contributing to this issue.

Impact level: Med

Spec wording:

After the switch and the controller have exchanged OFPT_HELLO messages and successfully negotiated a common version number, the connection setup is done and standard OpenFlow messages can be exchanged over the connection.

Recommendations:

Seek clarification from extensibility team.

3.6 Message handling

3.6.1 Packet_in messages

Description:

The specification is not clear whether the packet_in headers should duplicate information that is in the header of the data part of the packet_in. The spec seems to allow a minimum of including only meta-data fields, which would not be in the data portion of the packet_in.

This caused failure for a few cases during the interop, where the packet in message does not include all mapped fields.

Impact level: High

Spec wording:

"The OXM TLVs must include context fields, that is, fields whose values cannot be determined from the packet data. The standard context fields are OFPXMT_OFB_IN_PORT, OFPXMT_OFB_IN_PHY_PORT, OFPXMT_OFB_METADATA and OFPXMT_OFB_TUNNEL_ID. Fields whose values are all-bits-zero should be omitted. Optionally, the OXM TLVs may also include packet header fields that were previously extracted from the packet, including any modifications of those in the course of the processing".

Recommendations:

This should be raised with extensibility to define controller guidelines since presently the switch may need to make some assumptions about how a collection of controllers behave.

3.7 Multiple controllers

3.7.1 Master slave behavior

Description:

The OpenFlow specification needs to elaborate on the expectations with respect to interactions and cooperation between multiple active controllers. This ought to be a focus for ONF because of high-availability requirements driven by a controller being a single point of failure.

For example, Controller 1 can make a role request for Master and becomes the Master. Controller 2 can then make a role request for Master. Controller 2 is made master and Controller 1 is silently made slave. If controller 1 then tries to execute a flow_mod it will receive an error back from the switch.

Impact level: High

Spec wording: Section 6.3.4 of Open Flow Specification 1.3.2

Recommendations:

Get in touch with extensibility team to clarify intent.

4 Appendix – A: Extensibility comments and issue tracking

This annexure summarizes the list of issues being worked on by the Extensibility team and comments on the interoperability issues identified in this document.

ONF members can view the progress and status of the enhancements being considered at https://rs.opennetworking.org/bugs/browse/<Ticket-ID>

- **Section 2.1.2**

  "When the group_mod DELETE message include buckets, an error INVALID_GROUP was generated by the switch"

  EXT-510 - Error for group delete with buckets

- **Section 2.1.3**

  "Switch replies with error message when controller attempts to install a Type ALL Group Table entry with the bucket action is null"

  EXT-511 - Clarify the empty bucket is acceptable

- **Section 2.1.4**

  "The priority of Flow mod messages sent by the controllers is zero. Some switches treat this flow entry as Table-miss entry and this leads to some mistakes with the expected behavior"

  EXT-512 - Zero priority and table-miss flow entry

- **Section 2.1.5**

  "Controller uses Port_Mod to change port status to up, also uses barrier message for each port mod. Switch replied to first barrier but did not continue to respond after the first one"

  EXT-513 - Each barrier request must get a reply

- **Section 3.1.1**

  "When multiple match fields are specified in a flow, the prerequisites defined in the specification are insufficient to crisply define behavior"

  Extensibility comments:

  The spec indicates "An OXM TLV that has prerequisite restrictions must appear after the OXM TLVs for its prerequisites. Ordering of OXM TLVs within an OpenFlow

match is not otherwise constrained." Most switches and controller do support this without any issue.

Additional tickets listed below might help further clarify the expectations.

EXT-466 - Canonical order of match fields
EXT-377 - Match relaxation rewording
EXT-428 - Clarify descriptions of OXM prerequisites

- **Section 3.1.2**
  "One switch vendor however used a "relaxed" implementation that allows controllers to add flows without respecting prerequisites"

  Extensibility comments:

  It is a valid implementation.

  EXT-377 - Match relaxation rewording

- **Section 3.2.1**

  "The specification says the switch should increase the DSCP drop precedence value by configured precedence value, but there is clarity required on how to increment the value"

  Extensibility comments:

  DSCP encoding is governed by IETF RFCs, we won't duplicate those specification into OpenFlow because:
  1) We may introduce bugs or inconsistency with RFCs
  2) IETF may change its mind

  EXT-416 - Clarify DSCP remark meter band (Included in 1.3.4)

  "Also the definition of "rate" as defined in OF standard is different from rate implemented by some vendors "information rate / line rate". It is common for SP device vendors to use line rate in rate limiting"

  Most implementations are in hardware, we don't want to reject existing implementations, which is why we currently don't define rate and allow all implementations. We could have suggestions on how to calculate rate, but I can't see us mandating one way or another.

  EXT-495 - Specification unclear w.r.t. which fields are included in packet byte lengths used in metering + statistics

- **Section 3.3.1**

  "There are no specific recommendations on how the switch tables need to be implemented resulting in various switches implementing special behavior"

Extensibility comments:

Solution likely has to be Table Features or TTP.

- **Section 3.4.1**

  "While testing for behavior where more than one write action instruction of the same type is sent to the switch, it was found that the expected behavior is not clearly specified"

  Extensibility comments:

  EXT-421 - Behavior on Duplicate Action in Action Set (Included in 1.3.4)

- **Section 3.4.2**

  "The specification requires that the flow entry can have only one instruction of each type, whereas it is possible to construct a flow_mod message where the same instruction is used more than once"

  Extensibility comments:

  EXT-260 - Add error code for duplicate instruction (Included in 1.4.0 and available for 1.3.X as ONF Extension EXT-260)

- **Section 3.4.3**

  "The error messages to be generated when multiple violations occur simultaneously can be ambiguous leading to mismatched expectations around actions to be taken on error code behavior for open flow controllers"

  Extensibility comments:

  We cannot be too prescriptive when there is multiple violations in a single message, because it will impact implementation. We just want one error to be generated. Having said that, we have been disambiguating what error message need to be generated for specific violations.

  A general recommendation on the default behavior when multiple violations occur is tracked at EXT-514 - Multiple errors in single message

  "If a goto action in table 255 is set to table 254, it is not clear if the switch should generate an invalid table ID or an invalid action error"

  ⇨ Bad table-id is the expected error message.

  " A group_mod message is sent with a bucket which includes an output action to a bad port. The switch sent a BAD_ACTION,BAD_OUT_PORT error while the testing is expecting a GROUP_MOD_FAILED error with BAD_BUCKET code"

  => Bad output port is expected error message.

- **Section 3.5.1**

  "The switch was sending ECHO_REQUEST messages before the handshake was completed (right after the HELLO message). The handshake should be completed before switch sends messages (except HELLO) to the controller"

  Extensibility comments:

  You can't calculate version number if you have not received any Hello message. But on the other hand, if the switch is slow, the controller may receive the hello and an echo request messages back to back. It's not an issue because after processing the hello, the connection is set up and the echo request can be processed.

  In other words, sending an echo-request just after a hello may not be a violation and should not be an issue. It's only an issue if the echo-request is received before the party has sent its hello.

- **Section 3.6.1**

  "Specification is not clear whether the packet_in headers should duplicate information that is in the header of the data part of the packet_in. The spec seems to allow a minimum of including only meta-data fields, which would not be in the data portion of the packet_in"

  Extensibility comments:

  EXT-432 - Clarify presence of data fields in Packet-In OXM headers (Included in 1.3.4)

- **Section 3.7.1**
  "The OpenFlow specification needs to elaborate on the expectations with respect to interactions and cooperation between multiple active controllers"

  Extensibility comments:

  The multi-controller support in OpenFlow assumes some level of coordination between controllers, and is mostly controller driven. For example, generation-ids can't be managed if the controller don't talk to each other. We don't want to be prescriptive to allow various multi-controller models to be implemented.

  EXT-191 - Role Change Event (Included in 1.4.0 and also available for 1.3.X as ONF Extension EXT-191)

5 Appendix – B: References

- Open Flow 1.3.2 Switch Specification - https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.2.pdf
- Plug fest 1.3 interoperability test plan
- ONF-Testing-Interoperability Event white paper
- https://rs.opennetworking.org/wiki/display/EXT/ONF+Extensions#ONFExtensions-WGapprovedONFExtensionsforOpenFlow1.3.X

6 Appendix – C: Revision History

Version	Date	Notes
0.1	28th August 2014	Initial version – Kumar Jayaprakash, CNLabs
0.2	30th October 2014	Includes review comments from Pan Zhang (BII) and Ron Milford (INCNTRE)
0.3	10th December 2014	Incorporated inputs from Jean Tourrilhes (Extensibility) and adding issue tracking details
0.4	20th December 2014	Included plugfest dates in index