**OF-Switch-1.0.0-TestCases detailed testing methodology:**

**Openflow protocol messages**

1. **Features Request**

**Test Description**: Check features request is implemented

**Test mode**: Automated

**Test Title**: Features\_Request

**Ports**: I (Control Plane)

**Initial State**: Default (Clear switch state), Connection Setup

**Test-Field**: Mandatory

**Test Notes:**

a) Send OFPT\_FEATURES\_REQUEST from controller.

c) Verify OFPT\_FEATURES\_REPLY is received without errors

1. **Configuration request**

**Test Description:** Check basic get configuration request is implemented

**Test mode**: Automated

**Test Title**: Configuration\_Request

**Ports**: I (Control Plane)

**Initial State**: Default (Clear switch state), Connection Setup

**Test-Field**: Mandatory

**Test Notes:**

a) Send OFPT\_GET\_CONFIG\_REQUEST

c) Verify OFPT\_GET\_CONFIG\_REPLY is received without errors.

1. **Modify State (ADD)**

**Test Description:** Check basic Flow ADD request is implemented

**Test mode**: Automated

**Test Title**: Modify\_State\_Add

**Ports**: 3 (1 Control Plane 2 dataplane)

**Initial State**: Default (Clear switch state), Connection Setup

**Test-Field**: Mandatory

**Test Notes:**

a) Send OFPT\_FLOW\_MOD, command = OFPFC\_ADD

b) Send ofp\_table\_stats request

c) Verify that active\_count=1 in the reply

1. **Modify State (DELETE)**

**Test Description:** Check basic Flow Delete request is implemented

**Test mode**: Automated

**Test Title**: Modify\_State\_Delete

**Ports**: 3 (1 Control Plane 2 dataplane)

**Initial State**: Default (Clear switch state), Connection Setup

**Test-Field**: Mandatory

**Test Notes:**

a) Send OFPT\_FLOW\_MOD, command = OFPFC\_ADD

b) Send ofp\_table\_stats request

c) Verify that active\_count=1 in the reply

d) Send OFPT\_FLOW\_MOD, command = OFPFC\_DELETE

e) Send ofp\_table\_stats request

f) Verify active \_count=0 in the reply

1. **Modify State (MODIFY)**

**Test Description:** Check basic Flow Modify request is implemented

**Test mode**: Automated

**Test Title**: Modify\_State\_Modify

**Ports**: 3 (1 Control Plane and 2 Dataplane)

**Initial State**: Default (Clear switch state), Connection Setup

**Test-Field**: Mandatory

**Test Notes:**

1. Send OFPT\_FLOW\_MOD , command = OFPFC\_ADD, action A
2. Send ofp\_table\_stats request, Verify active\_count=1
3. Send OFPT\_FLOW\_MOD , command = OFPFC\_MODIFY, action A’
4. Send Test Packet matching the flow
5. Verify packet implements action A’
6. **Read State**

**Test Description:** Check basic Read State is implemented

**Test mode**: Automated

**Test Title**: Read\_State

**Ports**: 3 (1 Control Plane, 2 dataplane)

**Initial State**: Default (Clear switch state), Connection Setup

**Test-Field**: Mandatory

**Test Notes:**

1. Send OFPT\_FLOW\_MOD, command = OFPFC\_ADD
2. Create a OFPC\_FLOW\_STATS message and send it
3. Verify switch replies without errors
4. **Send packet**

**Test Description:** Check basic Send-Packet is implemented.

*Send-Packet: These are used by the controller to send packets out of a specified port on the switch.*

**Test mode**: Automated

**Test Title**: Send\_Packet

**Ports**: 5 (1 Control Plane, 4 Dataplane)

**Initial State**: Default (Clear switch state), Connection Setup

**Test-Field**: Mandatory

**Test Notes:**

a) Send OFPT\_PACKET\_OUT out message from controller to switch for every dataplane port.

b) Verify the packet appears on the each dataplane port

c) Verify sent packet matches the received packet

1. **Barrier Request**

**Test Description:** This test checks that a basic barrier request does not generate an error.

**Test mode**: Automated

**Test Title**: BarrierRequestReply

**Ports**: I (Control Plane)

**Initial State**: Default (Clear switch state), Connection Setup

**Test-Field**: Mandatory

**Test Notes:**

a) Send OFPT\_BARRIER\_REQUEST

c) Verify OFPT\_BARRIER\_REPLY is received on the control plane.

1. **Packet In**

**Test Description**: Check packet\_in is implemented. This test just checks that non matched dataplane packets generate a packet\_in

**Test mode**: Automated

**Test Title**: Packet\_In

**Ports**: 2 (1 Control Plane and 1 Dataplane)

**Initial State**: Default (Clear switch state), Connection Setup

**Test-Field**: Mandatory

**Test Notes:**

1. Send a packet to dataplane port , without inserting a flow entry
2. Verify a OFPT\_PACKET\_IN is generated on the control plane
3. **Hello**

**Test Description**: This test checks for basic Hello message generation with correct version field.

**Test mode**: Automated

**Test Title**: Hello

**Ports**: 1 (Control Plane)

**Initial State**: Default (Clear switch state), Connection Setup

**Test-Field**: Mandatory

**Test Notes:**

1. Send OFPT\_HELLO from controller to switch
2. Verify switch also sends OFPT\_HELLO message in response
3. Verify version field in the hello message is set to Openflow version 1.0.0
4. **Echo**

**Test Description**: This test checks for basic Echo Reply message generation with correct version field with same transaction id.

**Test mode**: Automated

**Test Title**: EchoWithoutBody

**Ports**: 1 (Control Plane)

**Initial State**: Default (Clear switch state), Connection setup

**Test-Field**: Mandatory

**Test Notes:**

1. Send OFPT\_ECHO\_REQUEST from the controller side.
2. Verify switch responds back with OFPT\_ECHO\_REPLY with same xid.
3. Verify Openflow version in header is set to Openflow version 1.0.0.

**Detailed controller to switch messages**

1. **Overlap checking**

**Test Description:** Verify that if overlap check flag is set in the flow entry and an overlapping flow is inserted then an error is generated and switch refuses flow entry

**Test mode**: Automated

**Test Title**: Overlap\_Checking

**Ports**: 3 (1 Control Plane 2 dataplane)

**Initial State**: Default (Clear switch state), Connection setup

**Test-Field**: Mandatory

**Test-Notes:**

1. Generate Flow F1🡪 Wildcard All
2. Send ofp\_table\_stats request , verify active\_count=1
3. Generate overlapping flow F2 🡪 Wildcard All Except Ingress Port ( with flag OFPFF\_CHECK\_OVERLAP set)
4. Verify that switch generates OFPT\_ERROR msg.

Type: OFPET\_FLOW\_MOD\_FAILED code : OFPFMFC\_OVERLAP

1. **No overlap checking**

**Test Description:** Verify that without overlap check flag set, overlapping flows can be created.

**Test mode**: Automated

**Test Title**: No\_Overlap\_Checking

**Ports**: 3 (1 Control Plane, 2 Dataplane)

**Initial State**: Default (Clear switch state), Connection setup

**Test-Field**: Mandatory

**Test Notes:**

1. Generate Flow F1🡪 Wildcard All.
2. Send ofp\_table\_stats request , verify active\_count=1
3. Generate overlapping flow F2 🡪 Wildcard All Except Ingress Port ( without flag OFPFF\_CHECK\_OVERLAP set)
4. Send a ofp\_table\_stats request, verify active\_count=2
5. **Identical flows**

**Test Description:** Verify that adding two identical flows overwrites the existing one and clears counters

**Test mode**: Automated

**Test Title**: Identical\_Flows

**Ports**: 3 (1 Control Plane), (2 dataplane)

**Initial State**: Default (Clear switch state), Connection setup

**Test-Field**: Mandatory

**Test Notes:**

1. Generate Flow F1.
2. Send ofp\_table\_stats request , verify active\_count=1

b) Increment counters (packet\_count, byte\_count) by sending a packet matching flow F1.

C) Send ofp\_flow\_stats request. Verify flow counters: byte\_count and packet\_count

c) Create identical flow F2

d) Send a ofp\_table\_stats request, verify active\_count=1

e) Send ofp\_flow\_stats request. Verify flow counters: byte\_count and packet\_count are reset

1. **No table to add** **(Written in oftest—Need to add to conformance Test-Suites)**

Flow\_Add\_6

1. **Never valid port (TBD)**
2. **Currently not existing port Version A and B (TBD)**
3. **Emergency flow with timeout values**

**Test Description:** Timeout values are not allowed for emergency flows

**Test Title**: Emer\_Flow\_With\_Timeout

**Test mode**: Automated

**Ports**: 3 (1 control plane, 2 Dataplane)

**Initial State**: Default (Clear switch state), Connection setup

**Test Notes:**

a) Generate a flow F with OFPFF\_EMERG set in flag and timeout values assigned.

b) Verify switch generates an OFPT\_ERROR msg, Type: OFPET\_FLOW\_MOD\_FAILED, Code OFPFMFC\_BAD\_EMERG\_TIMEOUT

1. **Missing modify adds**

**Test Description:** If a modify does not match an existing flow, the flow gets added.

**Title**: Missing\_Modify\_Add

**Test mode**: Automated

**Ports**:3 (1 control plane,2 Data Plane)

**Initial State**: Connection setup, Clear Switch State

**Test-Field**: Mandatory

**Test Notes:**

1. Generate a flow-mod , command OFPC\_MODIFY (Note: There should be no flows matching this flow\_mod modify command)
2. Send a ofp\_table\_stats request, verify active\_count=1
3. **Modify changes action, preserves counters**

**Test Description:** A modified flow preserves counters

**Title**: Modify\_Action

**Test mode**: Automated

**Ports**: 4(1 control plane, 3 Dataplane)

**Initial State**: Connection setup, Clear Switch State

**Test-Field**: Mandatory

**Test Notes:**

a) Create a flow\_mod F-1 with command OFPC\_ADD, action A

b) Send a test Packet matching the flow

c) Send an ofp\_flow\_stats request, verify byte\_count and packet\_count

e) Create flow\_mod with command OFPC\_MODIFY ,action A’ and modify action of flow F-1

f) Send a ofp\_flow\_stats request, verify flow counters are preserved

g) Send test packet, verify it implements action A’

1. **Strict Modify changes action, preserves counters**

**Test Description:** Strict Modify Flow also changes action preserves counters

**Title**: Strict\_Modify\_Action

**Test mode**: Automated

**Ports**: 4(1 control plane, 3 Dataplane)

**Initial State**: Connection setup, Clear Switch State

**Test-Field**: Mandatory

**Test Notes:**

1. Create two overlapping flows: F 🡪 Match on all, except one wildcarded (src address). Action A. Priority=100

F’ 🡪 Match on ingress\_port = port [0], wildcarded rest. Action A. Priority=10

1. Send a ofp\_table\_stats request, verify active\_count=2
2. Send Packet (it would have matched both the flows, since they are overlapping flows but it would match Flow-F1 as it has higher priority.)
3. Send ofp\_flow\_stats request for Flow-1 and verify counters packet\_count and byte\_count
4. Create flow\_mod ,command OFPC\_STRICT\_MODIFY,match on all except src address ,priority 100 action A’
5. Send test packet , verify action is modified
6. Send ofp\_flow\_stats request, verify counters are preserved.
7. **Delete non existing flow**

**Test Description:** Request deletion of non-existing flow

**Title**: Delete\_NonExisting\_Flow

**Test mode**: Automated

**Ports**: 1(1 control plane)

**Initial State**: Connection setup, Clear Switch State

**Test-Field**: Mandatory

**Test Notes:**

a) Issue a delete command, with no flows inserted

b) Make sure no error is generated on the control plane

1. **Delete flows with and without removed message**

**Test Description:** Check deletion of flows happens and generates messages as configured.

*I.e. if ‘Send Flow removed message’ Flag is set in the flow entry, the flow deletion of that respective flow should generate the flow removed message, vice versa also exists*

**Test Title**: Send\_Flow\_Rem

**Test mode**: Automated

**Ports**: 3 (1 control plane, 2 Dataplane)

**Initial State**: Connection setup, Default (clear switch state)

**Test-Field**: Mandatory

**Test Notes:**

a) Generate a flow F without OFPFF\_SEND\_FLOW\_REM flag set

b) Issue a delete command OFPFC\_DELETE

c) Verify that OFPT\_FLOW\_REMOVED message is not generated.

c) Generate a flow F’ with OFPFF\_SEND\_FLOW\_REM flag set

d) Issue a delete command OFPFC\_DELETE

e) Verify that OFPT\_FLOW\_REMOVED message is generated

1. **Delete emergency flow**

**Test Description:** Delete emergency flow and verify no message is generated.

*An emergency flow deletion will not generate flow-removed messages even if ‘Send Flow removed message’ flag was set during the emergency flow entry.*

**Title**: Delete\_Emer\_Flow

**Test mode**: Automated

**Ports**: 3 (1 control plane, 2 Dataplane)

**Initial State**: Connection setup, Clear Switch State (default)

**Test-Notes**:

a) Insert a flow with OFPFF\_EMERG flag set.

b) Delete the added flow with OFPFF\_SEND\_FLOW\_REM flag set

c) Test successful if no flow removed message is generated.

1. **Delete and verify strict and non-strict**

**Test Description:** Delete and verify strict and non-strict behaviors

*This test compares the behavior of delete strict and non-strict.*

**Title**: Delete\_Strict\_NonStrict

**Test** **mode**: Automated

**Ports**: 3 (1 control plane, 2 dataplane)

**Initial State**: Connection setup, Clear Switch State

**Test-Notes**:

a) Insert Flow F with an exact match.

b) Issue Non-strict Delete command, verify F gets deleted.

c) Insert F with an exact Match

d) Issue Strict Delete Command, verify F gets deleted.

e) Insert Flow T with match on all, except one wildcarded (say src address).

f) Insert another flow T' with match on ingress\_port, wildcarded rest.

g) Issue Non-strict Delete ( match on ingress\_port). Verify T+T' gets deleted.

h) Insert T and T' again. Issue Strict Delete (match on ingress port), verify only T' gets deleted

i) Insert T, add Priority P (say 100)

j) Insert T' add priority (200).

k) Insert T' again add priority 300 --> T”

l) Issue Non-Strict Delete (match on ingress port). Verify T+T’+T’’ gets deleted.

m) Insert T, T’, T’’ again, Issue Strict Delete (match on ingress\_port) with priority = 200. Verify only T’ gets deleted

1. **Delete flows with constraint out\_port**

**Test Description:** Delete flows filtered by action output

*DELETE and DELETE STRICT commands can be optionally filtered by output port. If the out\_port field contains a value other than OFPP\_NONE, it introduces a constraint when matching. This constraint is that the rule must contain an output action directed at that port.*

**Title**: Delete\_With\_Outport

**Test mode**: Automated

**Ports**: 3 (1 control plane, 2 Dataplanes)

**Initial State**: Connection setup, Clear Switch State

**Test-Field**: Mandatory

**Test-Notes**:

1. Insert a flow F with output action = port x
2. Send a delete command matching flow F ,but out\_port =port y
3. Send a table\_stats request , verify no flow gets deleted i.e. active\_count=1
4. Send a delete command matching flow F ,outport = port x
5. Send a table\_stats request, verify flow F gets deleted.
6. **Add, modify flows with constraint output**

**Test Description:** Add, modify flows with outport set. This field is ignored by ADD, MODIFY, and MODIFY STRICT messages.

**Title**: Add\_Modify\_With\_Outport

**Test mode**: Automated

**Ports**: 4 (1 control plane, 3 Data planes)

**Initial State**: Connection setup, Clear Switch State

**Test-Field**: Mandatory

**Test-Notes**:

1. Insert a flow F with action A (output 🡪 port x) , but out\_port field in the flow mod set as port y
2. Send Table\_Stats\_Request, Verify Flow gets inserted. *( Flow add ignores out\_port field)*
3. Modify the action in flow F , action A’ ( output -->port z ), but out\_port field in the flow mod set as port y
4. Send test packet matching the flow F
5. Verify packet implements action A’ (*flow modify ignores out\_port field)*
6. **Verify that idle timeout is implemented**

**Test Description:** Verify that idle timeout is implemented

**Title**: Idle\_Timeout

**Test mode**: Automated

**Ports**: 3 (1 control plane, 2 Dataplanes)

**Initial State**: Connection setup, Clear Switch State

**Test-Field**: Mandatory

**Test-Notes**:

a) Add a flow with idle timeout set and with OFPP\_SEND\_FLOW\_REM bit set

b) Verify flow removed message is received.

c) Verify flow removed reason was idle\_timeout

d) Verify the duration\_sec field is 1 sec

1. **Verify that hard timeout is implemented**

**Test Description:** Verify that hard timeout is implemented

**Title**: Idle\_Timeout

**Test mode**: Automated

**Ports**: 3 (1 control plane, 2 Dataplanes)

**Initial State**: Connection setup, Clear Switch State

**Test-Field**: Mandatory

**Test-Notes**:

a) Add a flow with hard timeout =1 set and with OFPP\_SEND\_FLOW\_REM bit set

b) Verify flow removed message is received.

c) Verify flow removed reason was hard\_timeout

d) Verify the duration\_sec field is 1 sec

1. **Verify that messages are generated as expected**

**Test Description:** Verify that Flow removed messages are generated as expected

*Since “flow removed messages being generated when flag is set” is already tested in the above tests*

*So here, we test the vice-versa condition.*

**Title**: Flow\_Timeout

**Test mode**: Automated

**Ports**: 3 (1control plane, 1dataplane)

**Initial State**: Connection setup, Clear Switch State

**Test-Field**: Mandatory

**Test-Notes**:

a) Generate and install a flow with idle\_timeout = 1 set no OFPFF\_SEND\_FLOW\_REM flag set.

b) Verify no flow removed message is received.

c) Send table\_stats\_request message and verify active\_count=0