

SAI PTF Test

PTF

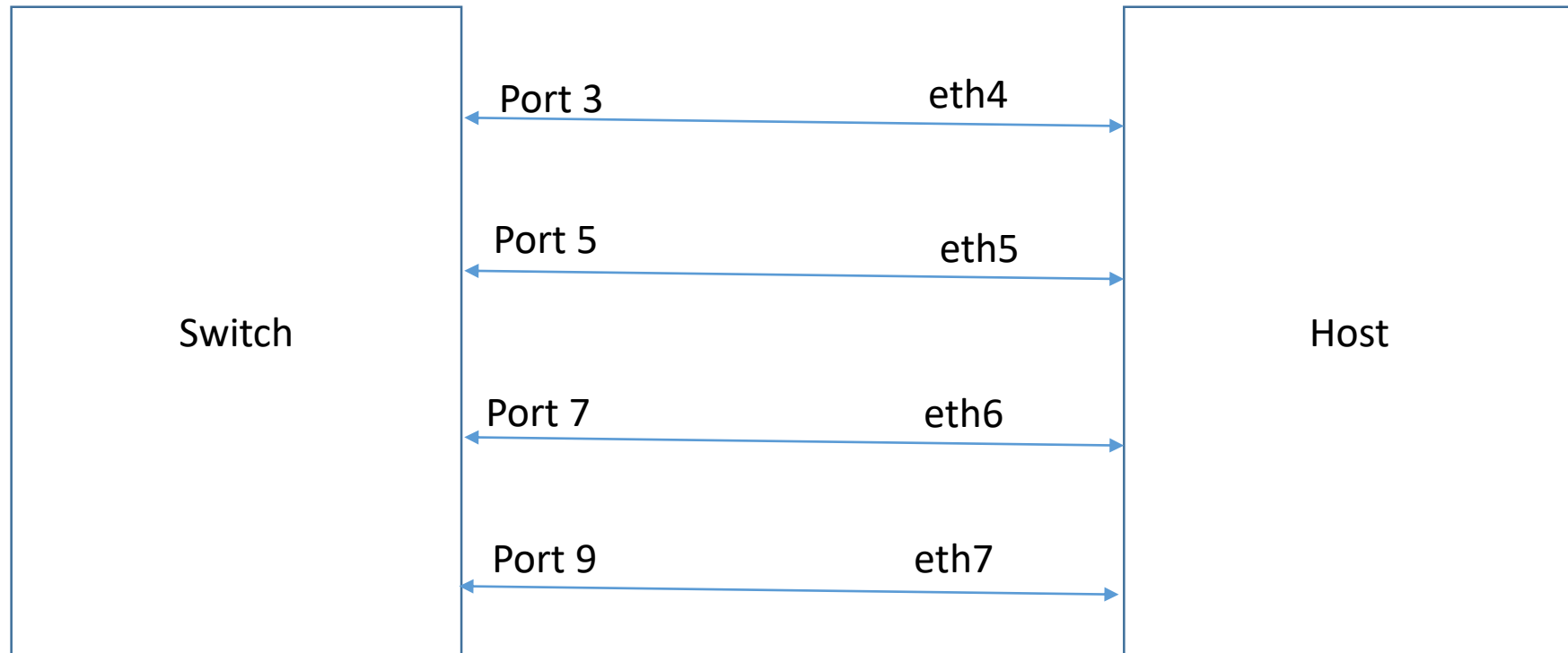
- Packet Testing Framework
- Python based
- Scapy
- <https://github.com/p4lang/ptf/tree/master>

SAI PTF

- SAI
- Thrift
 - Generate functions used for RPC calls
- PTF
 - Build, send and verify packet
- <https://github.com/opencomputeproject/SAI/tree/master/test/saithrift>

Topology

- Example



Configuration file

- Switch
 - portmap.ini
 - profile.ini
 - https://github.com/opencomputeproject/SAI/tree/master/test/saithrift/src/msn_2700

Sample Test Code

```
22 @group('l2')
23 class L2AccessToAccessVlanTest(sai_base_test.ThriftInterfaceDataPlane):
24     def runTest(self):
25         print
26         print "Sending L2 packet port 1 -> port 2 [access vlan=10]"
27         switch_init(self.client)
28         vlan_id = 10
29         port1 = port_list[0]
30         port2 = port_list[1]
31         mac1 = '00:11:11:11:11:11'
32         mac2 = '00:22:22:22:22:22'
33         mac_action = SAI_PACKET_ACTION_FORWARD
34
35         vlan_oid = sai_thrift_create_vlan(self.client, vlan_id)
36         vlan_member1 = sai_thrift_create_vlan_member(self.client, vlan_oid, port1, SAI_VLAN_TAGGING_MODE_UNTAGGED)
37         vlan_member2 = sai_thrift_create_vlan_member(self.client, vlan_oid, port2, SAI_VLAN_TAGGING_MODE_UNTAGGED)
38
39         attr_value = sai_thrift_attribute_value_t(u16=vlan_id)
40         attr = sai_thrift_attribute_t(id=SAI_PORT_ATTR_PORT_VLAN_ID, value=attr_value)
41         self.client.sai_thrift_set_port_attribute(port1, attr)
42         self.client.sai_thrift_set_port_attribute(port2, attr)
43
44         sai_thrift_create_fdb(self.client, vlan_oid, mac1, port1, mac_action)
45         sai_thrift_create_fdb(self.client, vlan_oid, mac2, port2, mac_action)
```

Sample Test Code

```
1270 def sai_thrift_create_vlan(client, vlan_id):
1271     vlan_attr_list = []
1272     attribute_value = sai_thrift_attribute_value_t(u16=vlan_id)
1273     attribute = sai_thrift_attribute_t(id=SAI_VLAN_ATTR_VLAN_ID, value=attribute_value)
1274     vlan_attr_list.append(attribute)
1275     vlan_oid = client.sai_thrift_create_vlan(vlan_attr_list)
1276     return vlan_oid
```

Sample Test Code

```
825 sai_thrift_object_id_t sai_thrift_create_vlan(const std_sai_thrift_attr_vctr_t &thrift_attr_list) {
826     SAI_THRIFT_LOG_DBG("Called.");
827
828     sai_vlan_api_t *vlan_api = nullptr;
829     auto status = sai_api_query(SAI_API_VLAN, reinterpret_cast<void**>(&vlan_api));
830
831     if (status != SAI_STATUS_SUCCESS)
832     { SAI_THRIFT_LOG_ERR("Failed to get VLAN API."); return SAI_NULL_OBJECT_ID; }
833
834     sai_attribute_t *attr_list = nullptr;
835     sai_uint32_t attr_size = thrift_attr_list.size();
836     sai_thrift_alloc_attr(attr_list, attr_size);
837     sai_thrift_parse_vlan_attributes(thrift_attr_list, attr_list);
838
839     sai_object_id_t vlanObjId = 0;
840     status = vlan_api->create_vlan(&vlanObjId, gSwitchId, attr_size, attr_list);
841     sai_thrift_free_attr(attr_list);
842
843     if (status == SAI_STATUS_SUCCESS) { return vlanObjId; }
844
845     SAI_THRIFT_LOG_ERR("Failed to create VLAN.");
846
847     return SAI_NULL_OBJECT_ID;
848 }
```


Sample Test Code

```
22 @group('l2')
23 class L2AccessToAccessVlanTest(sai_base_test.ThriftInterfaceDataPlane):
24     def runTest(self):
25         print
26         print "Sending L2 packet port 1 -> port 2 [access vlan=10]"
27         switch_init(self.client)
28         vlan_id = 10
29         port1 = port_list[0]
30         port2 = port_list[1]
31         mac1 = '00:11:11:11:11:11'
32         mac2 = '00:22:22:22:22:22'
33         mac_action = SAI_PACKET_ACTION_FORWARD
34
35         vlan_oid = sai_thrift_create_vlan(self.client, vlan_id)
36         vlan_member1 = sai_thrift_create_vlan_member(self.client, vlan_oid, port1, SAI_VLAN_TAGGING_MODE_UNTAGGED)
37         vlan_member2 = sai_thrift_create_vlan_member(self.client, vlan_oid, port2, SAI_VLAN_TAGGING_MODE_UNTAGGED)
38
39         attr_value = sai_thrift_attribute_value_t(u16=vlan_id)
40         attr = sai_thrift_attribute_t(id=SAI_PORT_ATTR_PORT_VLAN_ID, value=attr_value)
41         self.client.sai_thrift_set_port_attribute(port1, attr)
42         self.client.sai_thrift_set_port_attribute(port2, attr)
43
44         sai_thrift_create_fdb(self.client, vlan_oid, mac1, port1, mac_action)
45         sai_thrift_create_fdb(self.client, vlan_oid, mac2, port2, mac_action)
```

Sample Test Code

Build a packet

```
47 |  
48 |  
49 |  
50 |  
51 |  
52 |  
53 |  
54 |  
55 |  
  
pkt = simple_tcp_packet(eth_dst='00:22:22:22:22:22',  
                          eth_src='00:11:11:11:11:11',  
                          ip_dst='10.0.0.1',  
                          ip_id=101,  
                          ip_ttl=64)  
  
try:  
    send_packet(self, 0, str(pkt))  
    verify_packets(self, pkt, [1])
```

Send packet from port 0

Verify packet received at port 1

Build a packet

```
56 def simple_tcp_packet(pktlen=100,  
57     eth_dst='00:01:02:03:04:05',  
58     eth_src='00:06:07:08:09:0a',  
59     dl_vlan_enable=False,  
60     vlan_vid=0,  
61     vlan_pcp=0,  
62     dl_vlan_cfi=0,  
63     ip_src='192.168.0.1',  
64     ip_dst='192.168.0.2',  
65     ip_tos=0,  
66     ip_ttl=64,  
67     ip_id=0x0001,  
68     ip_frag=0,  
69     tcp_sport=1234,  
70     tcp_dport=80,  
71     tcp_flags="S",  
72     ip_ihl=None,  
73     ip_options=False,  
74     with_tcp_chksum=True  
75 ):
```

Build a packet

```
101 if MINSIZE > pktlen:
102     pktlen = MINSIZE
103
104 if with_tcp_chksum:
105     tcp_hdr = scapy.TCP(sport=tcp_sport, dport=tcp_dport, flags=tcp_flags)
106 else:
107     tcp_hdr = scapy.TCP(sport=tcp_sport, dport=tcp_dport, flags=tcp_flags, checksum=0)
108
109 # Note Dot1Q.id is really CFI
110 if (dl_vlan_enable):
111     pkt = scapy.Ether(dst=eth_dst, src=eth_src)/ \
112         scapy.Dot1Q(prio=vlan_pcp, id=dl_vlan_cfi, vlan=vlan_vid)/ \
113         scapy.IP(src=ip_src, dst=ip_dst, tos=ip_tos, ttl=ip_ttl, id=ip_id, ihl=ip_ihl)/ \
114         tcp_hdr
115 else:
116     if not ip_options:
117         pkt = scapy.Ether(dst=eth_dst, src=eth_src)/ \
118             scapy.IP(src=ip_src, dst=ip_dst, tos=ip_tos, ttl=ip_ttl, id=ip_id, ihl=ip_ihl, frag=ip_frag)/ \
119             tcp_hdr
120     else:
121         pkt = scapy.Ether(dst=eth_dst, src=eth_src)/ \
122             scapy.IP(src=ip_src, dst=ip_dst, tos=ip_tos, ttl=ip_ttl, id=ip_id, ihl=ip_ihl, frag=ip_frag, options=ip_options)/ \
123             tcp_hdr
124
125 pkt = pkt/("".join([chr(x % 256) for x in xrange(pktlen - len(pkt))]))
126
127 return pkt
```

Sample Test Code

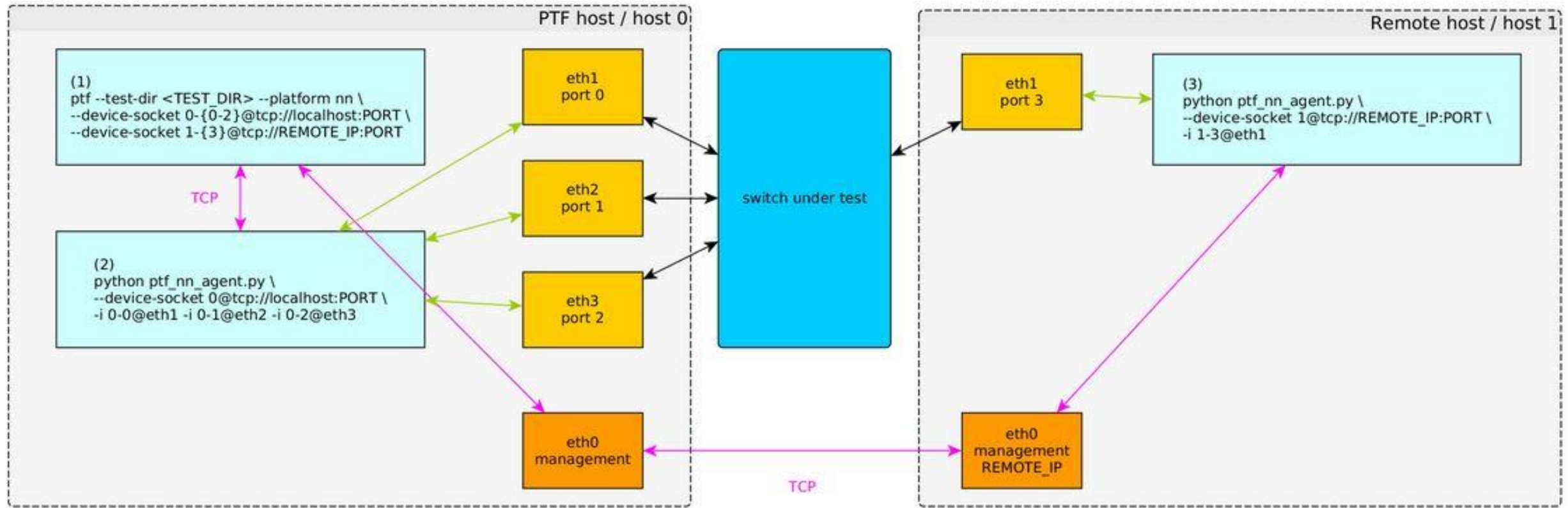
47
48
49
50
51
52
53
54
55

```
pkt = simple_tcp_packet(eth_dst='00:22:22:22:22:22',  
                        eth_src='00:11:11:11:11:11',  
                        ip_dst='10.0.0.1',  
                        ip_id=101,  
                        ip_ttl=64)  
  
try:  
    send_packet(self, 0, str(pkt))  
    verify_packets(self, pkt, [1])
```

Sample Test Code

```
56 finally:
57     sai_thrift_delete_fdb(self.client, vlan_oid, mac1, port1)
58     sai_thrift_delete_fdb(self.client, vlan_oid, mac2, port2)
59
60     attr_value = sai_thrift_attribute_value_t(u16=1)
61     attr = sai_thrift_attribute_t(id=SAI_PORT_ATTR_PORT_VLAN_ID, value=attr_value)
62     self.client.sai_thrift_set_port_attribute(port1, attr)
63     self.client.sai_thrift_set_port_attribute(port2, attr)
64
65     self.client.sai_thrift_remove_vlan_member(vlan_member1)
66     self.client.sai_thrift_remove_vlan_member(vlan_member2)
67     self.client.sai_thrift_remove_vlan(vlan_oid)
```

PTF NN



PTF Demo

- <https://github.com/opencomputeproject/SAI/blob/master/test/saithrift/tests/sail2.py>

PTF NN Demo

- https://github.com/p4lang/ptf/tree/fe3cd89e332d9b0e673230e06a1a7ff29d688df5/ptf_nn

Q & A

Thanks!