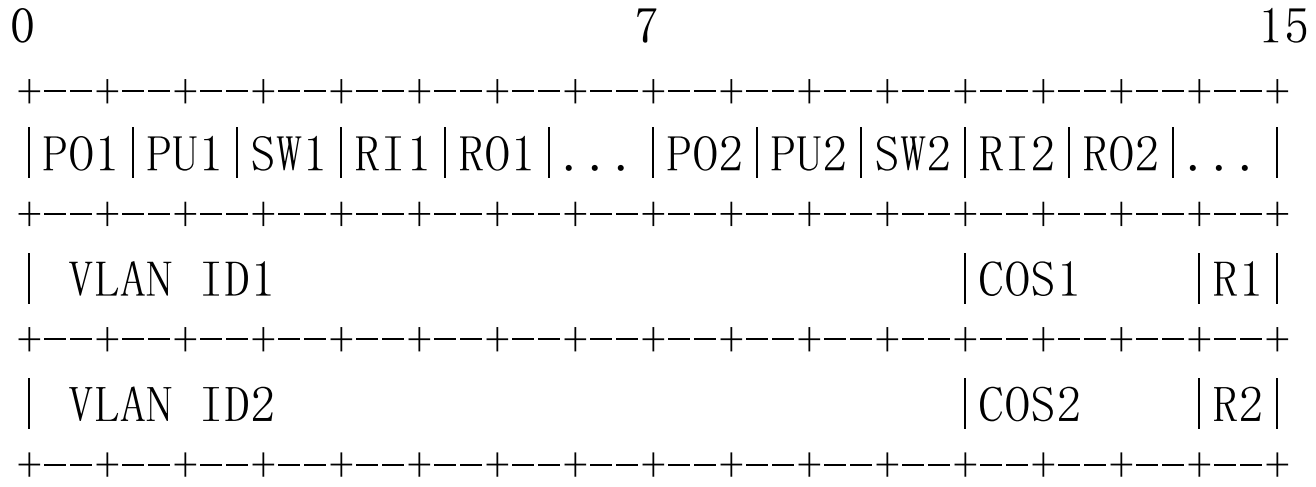
Dessimination of Flow Specification Rules for L2 VPN

draft-ietf-idr-flowspec-l2vpn-04

Weiguo Hao Huawei
Qiandeng Liang Huawei
Feng Dong Huawei
Jim Uttaro AT&T
S. Litkowski Orange
Shunwan Zhuang Huawei
IETF 97, Seoul, Korea

New flags for (VLAN/QinQ Action)



Added:

PO1: Pop action.

PU1: Push action.

SW1: Swap action - outer VLAN and inner VLAN should be swapped

PO2: Pop action - outmost VLAN should be removed)

PU2: Push action - VLAN ID2 will be added the associated COS is COS2.

SW2: Swap action - the outer VLAN and inner VLAN should be swapped.

RI1 and RI2: Rewrite inner VLAN action.

RO1 and RO2: Rewrite outer VLAN action

Updates Flow spec L2VPN Component Types)

type	RFC or Draft	discription
1	RFC5575	Destination Prefix
1	draft-ietf-idr-flow-spec-v6-06	Destination IPv6 Prefix
2	RFC5575	Source Prefix
2	draft-ietf-idr-flow-spec-v6-06	Source IPv6 Prefix
3	RFC5575	IP Protocol
…		
13	draft-ietf-idr-flow-spec-v6-06	Flow Label
TBD	This draft	Ethernet Type
TBD	This draft	Source MAC
TBD	This draft	Destination MAC
TBD	This draft	DSAP in LLC

Next Steps

- Plan to implement
  - L2VPN flowspec server in ONOS
  - Implement L2VPN flowspec client in Huawei product
- Report back results to IDR WG

Updates con. Add Ordering of Traffic Filtering Rules

```
Pseudocode:
flow_rule_L2_cmp (a, b)
   {
       comp1 = next_component(a);
       comp2 = next_component(b);
       while (comp1 || comp2) {
           // component_type returns infinity on end-of-list
           if (component_type(comp1) < component_type(comp2)) {
               return A_HAS_PRECEDENCE;
           }
           if (component_type(comp1) > component_type(comp2)) {
               return B_HAS_PRECEDENCE;
           }

           if (component_type(comp1) == MAC_DESTINATION || MAC_SOURCE) {
               common = MIN(MAC Address length (comp1),
                     MAC Address length (comp2));
               cmp = MAC Address compare(comp1, comp2, common);
               // not equal, lowest value has precedence
               // equal, longest match has precedence
           } else {
               common =
                  MIN(component_length(comp1), component_length(comp2));
               cmp = memcmp(data(comp1), data(comp2), common);
               // not equal, lowest value has precedence
               // equal, longest string has precedence
           }
       }
       return EQUAL;
   }
```

The original definition for the order of traffic filtering rules can be reused with new consideration for the MAC Address offset. As long as the offsets are equal, the comparison is the same, retaining longest-prefix-match semantics. If the offsets are not equal, the lowest offset has precedence, as this flow matches the most significant bit.