

Open vSwitch - Virtual switching reporting of interface telemetry via sFlow - High Level Design

- [Requirement](#)
- [Overview](#)
- [Design/Development](#)
 - [OVS architecture \(see: <http://docs.openvswitch.org/en/latest/topics/porting/>\)](http://docs.openvswitch.org/en/latest/topics/porting/)
 - [netdev component](#)
 - [ofproto-dpif provider for sFlow](#)
 - [sFlow generic counter type](#)
 - [sFlow ethernet counter type](#)
 - [Full patch](#)
- [User-guide](#)

Requirement

Exposure of the "Counter Data Types" from the sFlow RFC (http://www.sflow.org/sflow_version_5.txt) in Open vSwitch (OVS).

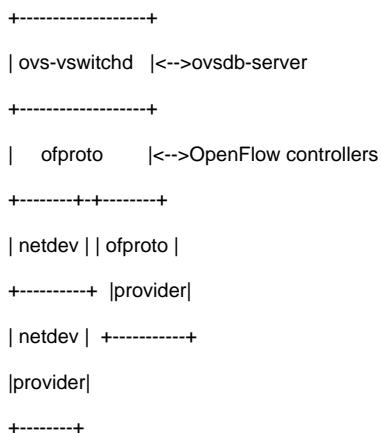
Overview

Open vSwitch (OVS) supports performance monitoring using sFlow protocol (see: <http://docs.openvswitch.org/en/latest/howto/sflow/>).

The sFlow monitoring system consists of an sFlow Agent (embedded in OVS) and a central sFlow Collector. The sFlow Agent uses two forms of sampling: statistical packet-based sampling of switched or routed packet flows, and time-based sampling of counters. The latter mechanism can be extended by additional counters using data provided by DPDK Extended Statistics API (see: http://dpdk.org/doc/guides/prog_guide/poll_mode_drv.html) and this document describes how to do it.

Design/Development

OVS architecture (see: <http://docs.openvswitch.org/en/latest/topics/porting/>)



netdev component

It abstracts interacting with network interfaces (physical and virtual). All network statistics are collected and provided by this component (netdev-dpdk provider uses DPDK Extended Statistics API). sFlow monitoring uses data provided by netdev for exposing particular counters.

ofproto-dpif provider for sFlow

There is implemented in OVS ofproto-dpif-sflow component for handling sFlow performance monitoring. In this component there are prepared sFlow counters basing on the data provided by netdev (functions: sflow_agent_get_counters, sflow_agent_get_global_counters).

sFlow generic counter type

sFlow RFC structure

/* Generic Interface Counters - see RFC 2233 */

/* opaque = counter_data; enterprise = 0; format = 1 */

```
struct if_counters {
    unsigned int ifIndex;
    unsigned int ifType;
    unsigned hyper ifSpeed;
    unsigned int ifDirection; /* derived from MAU MIB (RFC 2668)
                               0 = unknown, 1=full-duplex, 2=half-duplex,
                               3 = in, 4=out */
    unsigned int ifStatus; /* bit field with the following bits assigned
                           bit 0 = ifAdminStatus (0 = down, 1 = up)
                           bit 1 = ifOperStatus (0 = down, 1 = up) */
    unsigned hyper ifInOctets;
    unsigned int ifInUcastPkts;
    unsigned int ifInMulticastPkts;
    unsigned int ifInBroadcastPkts;
    unsigned int ifInDiscards;
    unsigned int ifInErrors;
    unsigned int ifInUnknownProtos;
    unsigned hyper ifOutOctets;
    unsigned int ifOutUcastPkts;
    unsigned int ifOutMulticastPkts;
    unsigned int ifOutBroadcastPkts;
    unsigned int ifOutDiscards;
    unsigned int ifOutErrors;
    unsigned int ifPromiscuousMode;
}
```

Implementation

In OVS there was already available implementation of this counter type. However it has been extended by ifInBroadcastPkts, ifOutMulticastPkts and ifOutBroadcastPkts counters.

sFlow ethernet counter type

sFlow RFC structure

```
/* Ethernet Interface Counters - see RFC 2358 */  
/* opaque = counter_data; enterprise = 0; format = 2 */  
  
struct ethernet_counters {  
    unsigned int dot3StatsAlignmentErrors;  
    unsigned int dot3StatsFCSErrors;  
    unsigned int dot3StatsSingleCollisionFrames;  
    unsigned int dot3StatsMultipleCollisionFrames;  
    unsigned int dot3StatsSQETestErrors;  
    unsigned int dot3StatsDeferredTransmissions;  
    unsigned int dot3StatsLateCollisions;  
    unsigned int dot3StatsExcessiveCollisions;  
    unsigned int dot3StatsInternalMacTransmitErrors;  
    unsigned int dot3StatsCarrierSenseErrors;  
    unsigned int dot3StatsFrameTooLongs;  
    unsigned int dot3StatsInternalMacReceiveErrors;  
    unsigned int dot3StatsSymbolErrors;  
}
```

Implementation

This counter type was not supported. It has been added, but only following counters are exposed dot3StatsAlignmentErrors, dot3StatsFCSErrors and dot3StatsFrameTooLongs, because only those statistics are provided by netdev component. It means that for other counters there is sent maximum counter value to indicate that the counter is not available.

Full patch

Patchwork: <https://patchwork.ozlabs.org/patch/732122/>

User-guide

<http://docs.openvswitch.org/en/latest/howto/sflow/>