

ovs的qos具体应用

- meter
待定
- Policing管制
Policing用于控制接口上接收分组（ingress）的速率，是一种简单的QoS的功能，通过简单的丢包机制实现接口速率的限制，它既可以作用于物理接口，也可以作用于虚拟接口；
- Shaping整形
Shaping是作用于接口上的出口流量（egress）策略，可以实现多个QoS队列，不同队列里面处理不同策略；

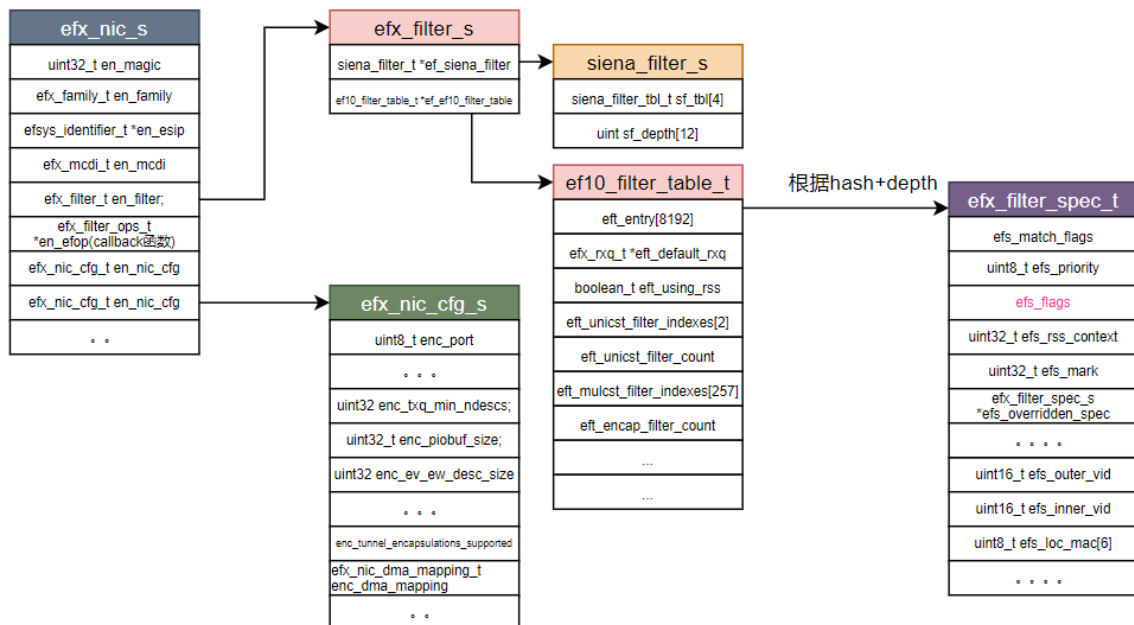
配置

配置命令如下：

```
1 ovs-vsctl set interface <devname> ingress_policing_rate=<kbits_rate>  
  ingress_policing_burst=<kbits_burst>  
2 //配置ingress，这个地方只是限制整个端口的，目前不能细分，跟实现有关。他的filter是all。  
3  
4 ovs-vsctl set port <devname> qos=@qosname -- --id=@qosname create qos  
  type=linux-htb queues=queue_id=@queue_name -- --id=@queue_name create queue  
  other-config:max-rate=<bytes_rate>  
5 //配置出口HTB类型的一个队列，需要ovs-ofctl配置将需要的报文导入该队列  
6 ovs-ofctl add-flow brname in_port=portno,actions=set_queue:queue_id,normal  
7 //最简单的倒流量的方式，可以在流表添加过滤条件。
```

这里有个注意点就是这里的端口一定是 `ovs-vsctl show` 命令展示出来的端口；

主要代码流程如下：



代码流程

qos主要使用的算法如下：

```

1  static const struct tc_ops *const tcs[] = {
2      &tc_ops_htb,                /* Hierarchy token bucket (see tc-htb(8)).
   */
3      &tc_ops_hfsc,                /* Hierarchical fair service curve. */
4      &tc_ops_code1,              /* Controlled delay */
5      &tc_ops_fqcode1,            /* Fair queue controlled delay */
6      &tc_ops_sfq,                /* Stochastic fair queueing */
7      &tc_ops_netem,              /* Network Emulator */
8      &tc_ops_noop,               /* Non operating qos type. */
9      &tc_ops_default,            /* Default qdisc (see tc-pfifo_fast(8)). */
10     &tc_ops_other,              /* Some other qdisc. */
11     NULL
12 };
  
```

限速的实现点有以下几个

ingress限速: **netif_receive_skb**-->netif_receive_skb_core-->handle_ing(常用，配置了CONFIG_NET_CLS_ACT)。

egress限速: dev_queue_xmit-->__dev_xmit_skb。

vhost网口处的限速调用为

ingress限速: vhost_worker-->handle_tx_kick-->handle_tx(sock->ops->sendmsg)-->tun_sendmsg-->tun_get_user(内部的tun_alloc_skb?)-->netif_rx_ni(netif_rx没看到多占cpu)-->do_softirq-->call_softirq-->**do_softirq-->net_rx_action-->process_backlog-->netif_receive_skb-->netif_receive_skb_core-->handle_ing**

egress限速: **ovs_vport_send-->netdev_send-->dev_queue_xmit-->dev_xmit_skb**