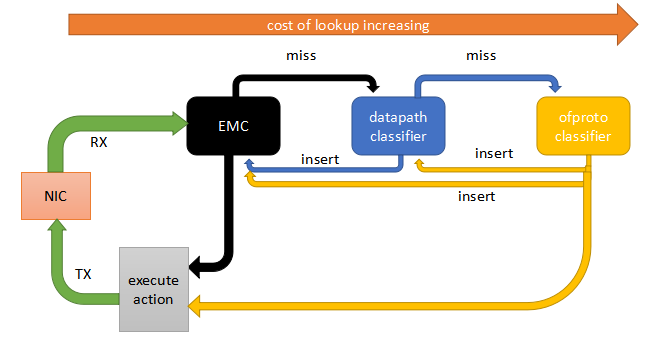
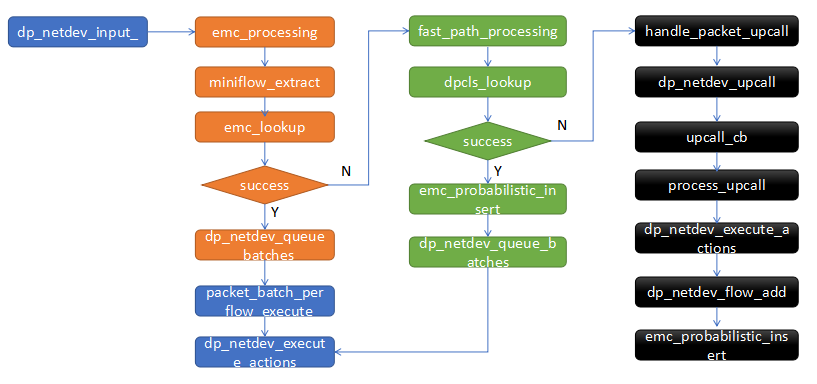
三级流表的查找顺序

microflow在ovs+dpdk代码中，又被称为EMC(exact match cache)。

megaflow在ovs+dpdk代码中，又被称为dpcls(datapath classifer)。

1. 从网卡接收到报文后，首先查找EMC表项，如果命中则直接执行action，如果miss则查找dpcls。
2. 如果查找dpcls命中，则将规则插入EMC，并且执行action。还是miss的话，就查找openflow流表。
3. 如果查找openflow命中，则将规则插入dpcls和EMC，并且执行action。还是miss的话，就drop或者发给controller。actions=NORMAL时，会执行flood而不是drop





emc\_insert

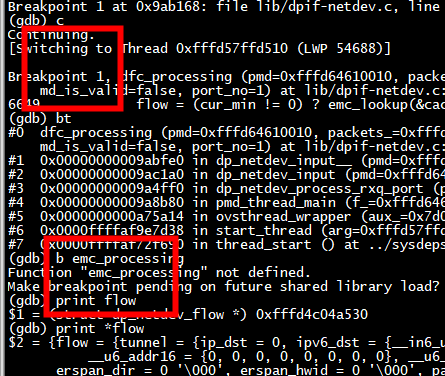
emc\_lookup

dpcls\_lookup

dp\_netdev\_execute\_actions

emc\_probabilistic\_insert

注意：emc\_processing 改成了dfc\_processing

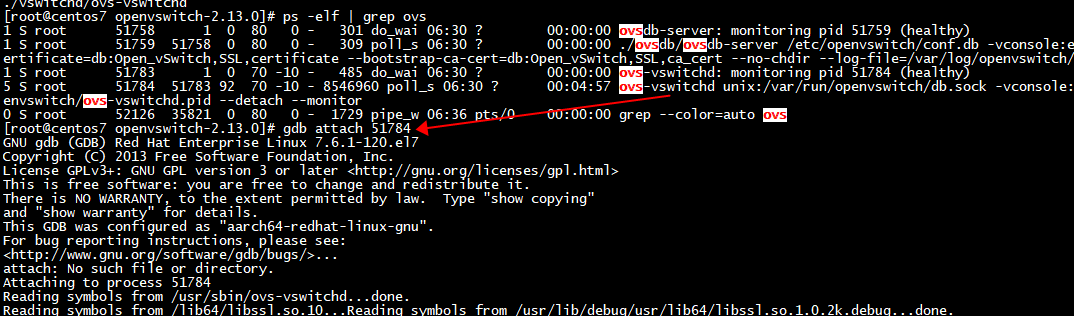


# Flow

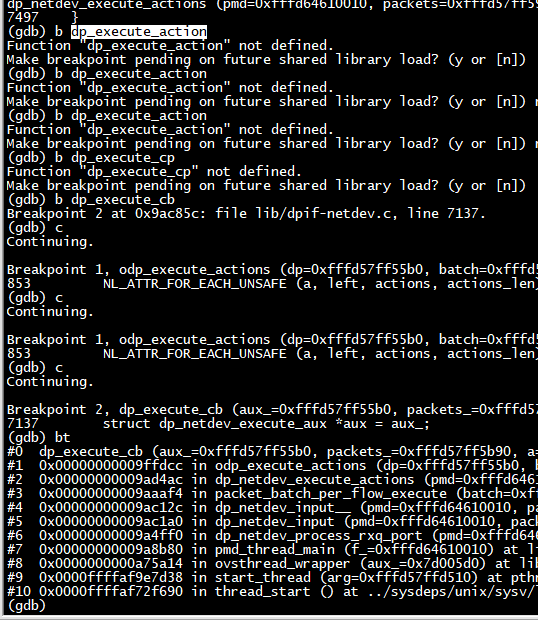


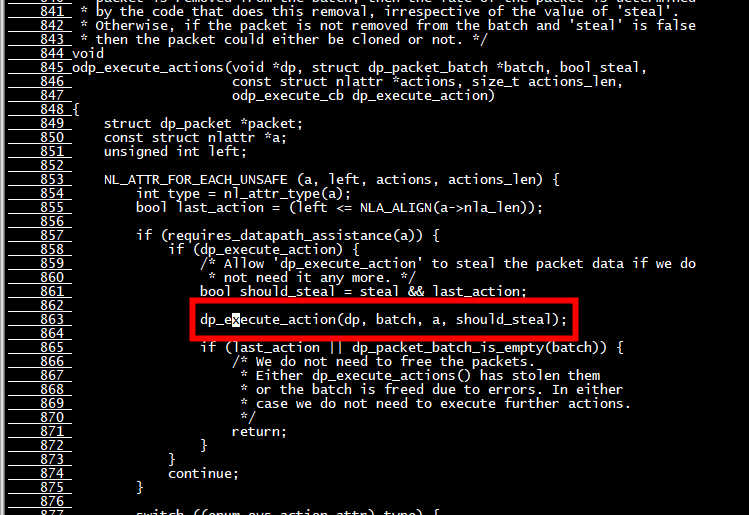
# Gdb ovs-vswitchd

1. Gdb attach ovs-vswitchd 无法 ovs-ofctl dump-flows br0

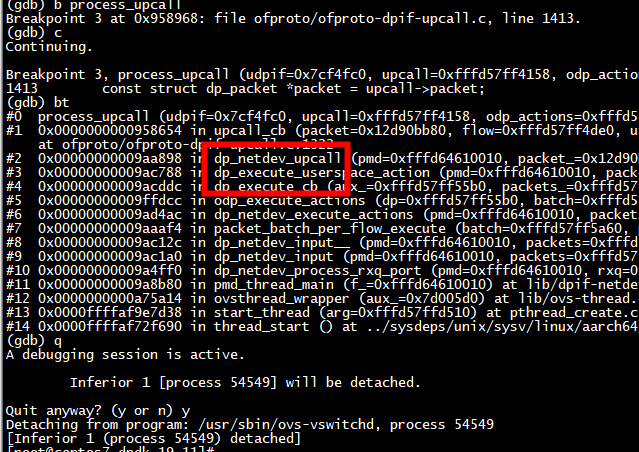


## dp\_execute\_cb and dp\_execute\_action函数

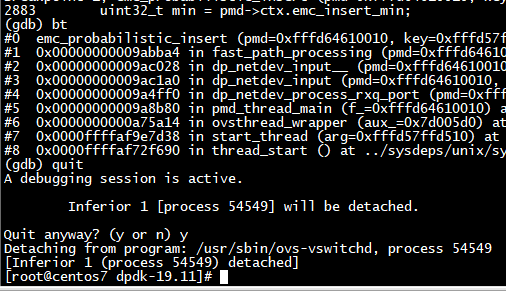




process\_upcall会被调用

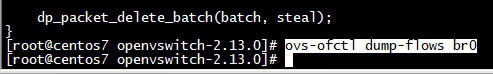


emc\_probabilistic\_insert 也会被调用



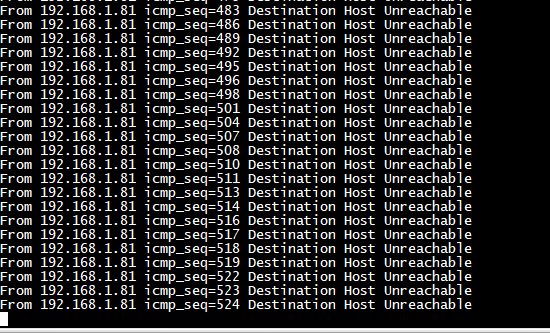
## 测试1

### 删除所有流表

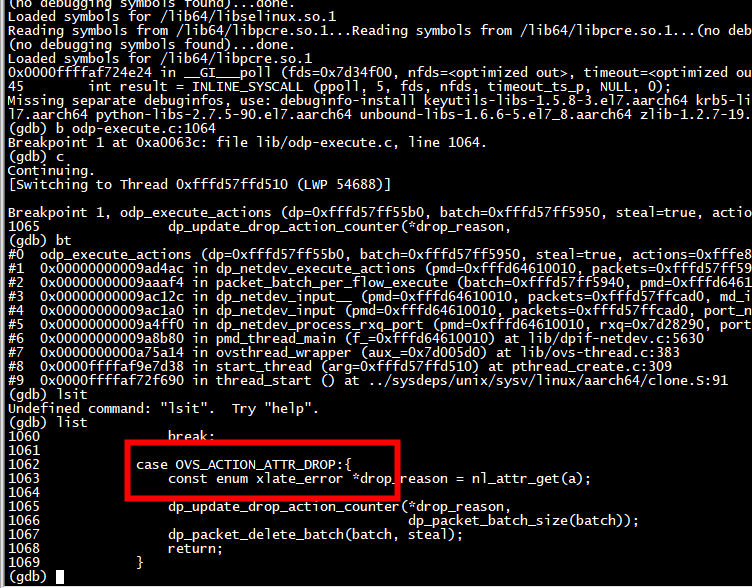


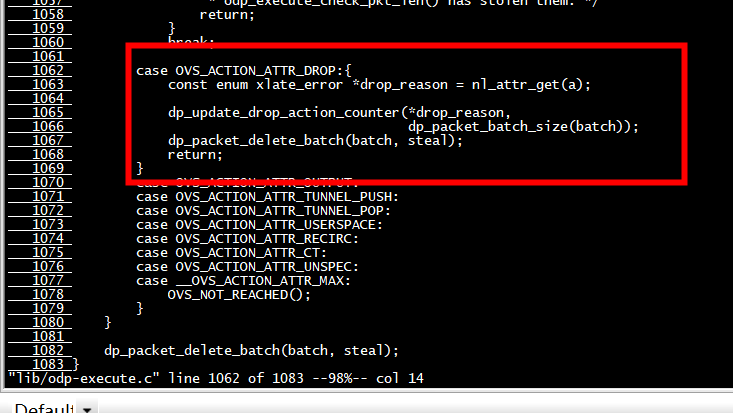
### 结果

Ping 失败



### 数据转发

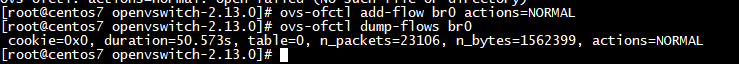




## 测试2

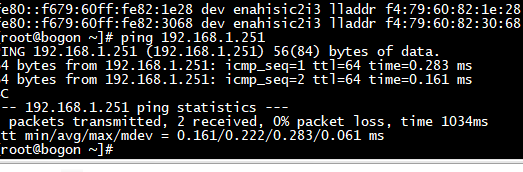
### 添加normal规则

ovs-ofctl add-flow br0 actions=NORMAL



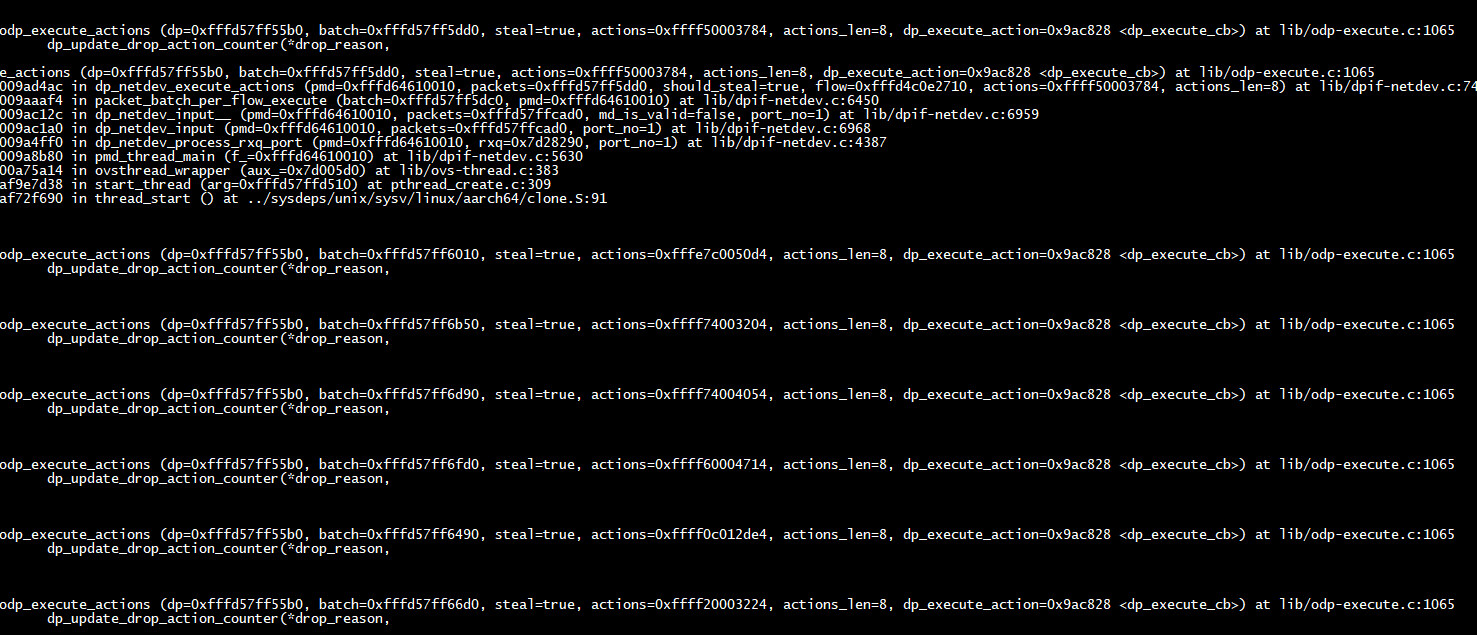
### 结果

Ping 通

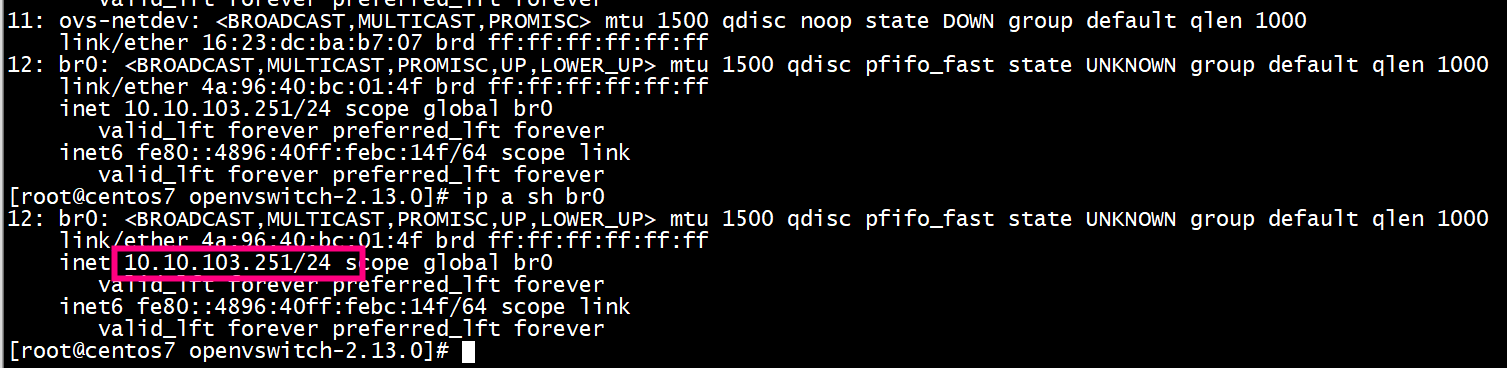


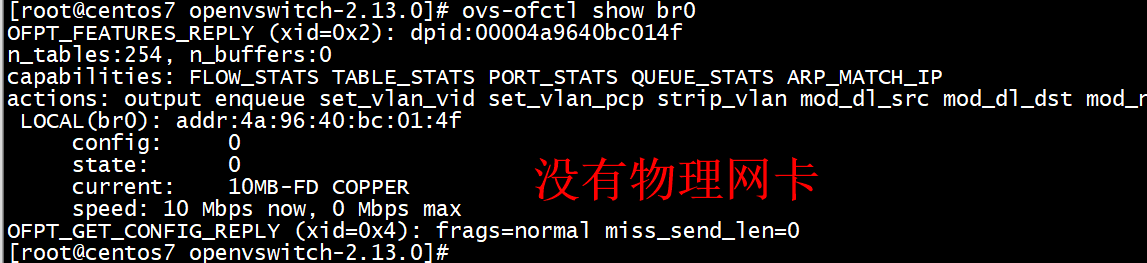
### 数据转发

1. 同样有些drop处理



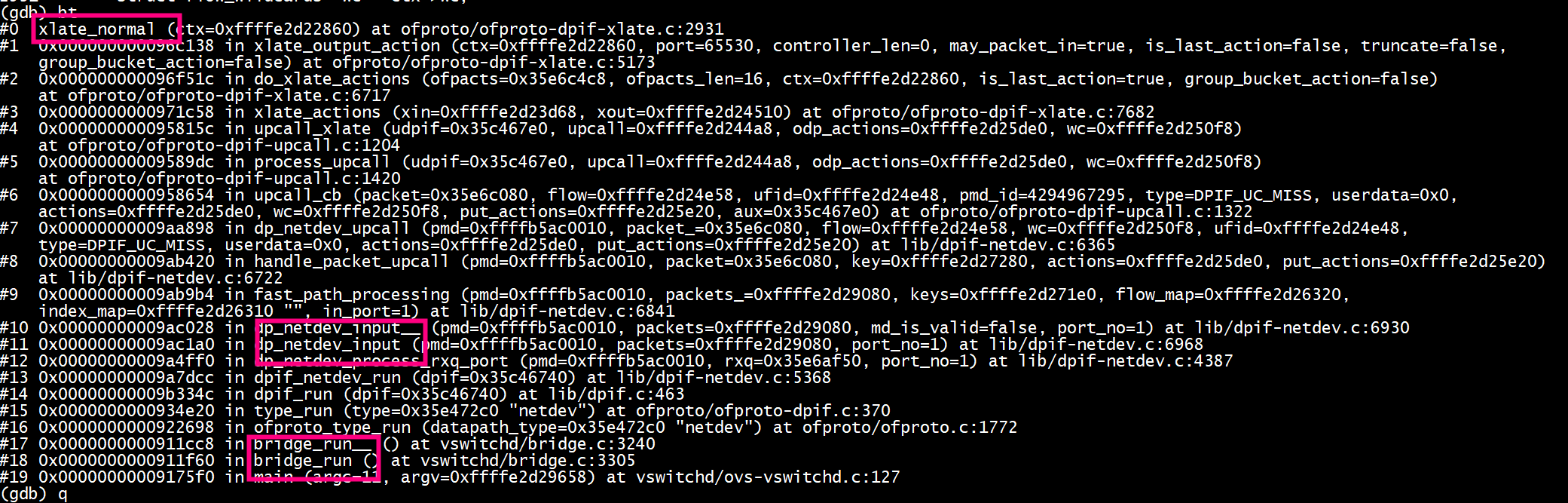
# Linux bridge





* 执行ping

观察 xlate\_normal

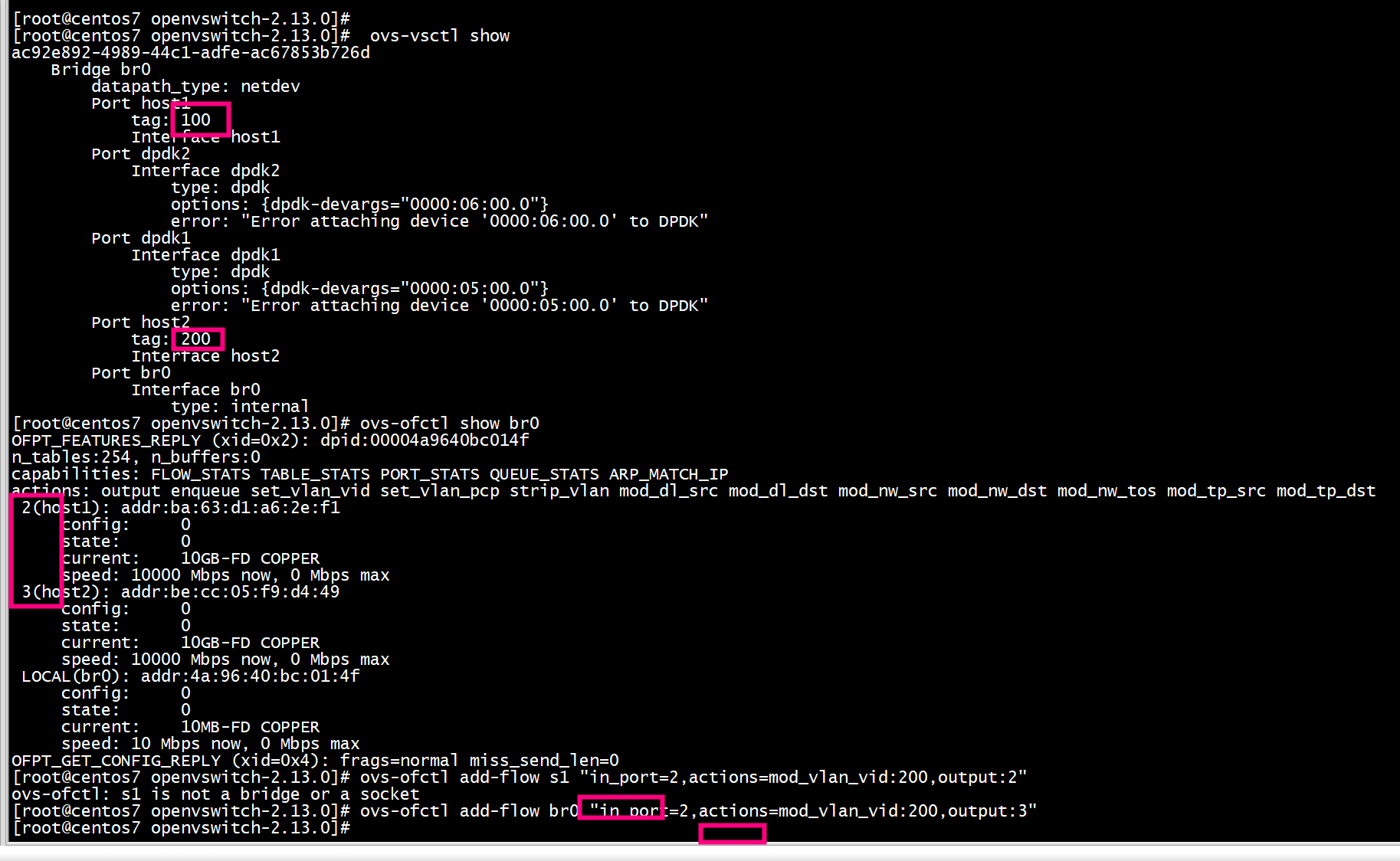


# Vlan

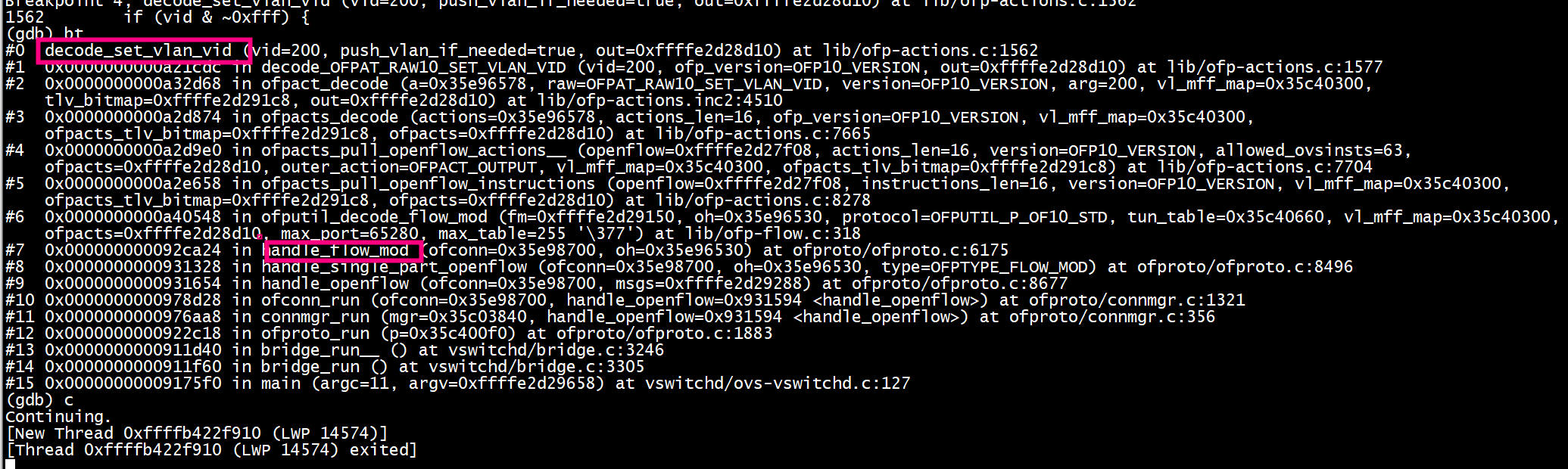
Host1: vlan=100

Host2: vlan=200

ovs-ofctl add-flow br0 "in\_port=2,actions=mod\_vlan\_vid:200,output:3"

**添加mod\_vlan\_vid**

执行ovs-ofctl add-flow br0 "in\_port=2,actions=mod\_vlan\_vid:200,output:3"



# 参考

ovs+dpdk 三级流表(microflow/megaflow/openflow)