


openvswitch 2.16.1 源码注释

写在前面










本项目是加了注释版的openvswitch 2.16.1源码，原始代码可以在网址<https://www.openvswitch.org/download/>处下载。如果你想查看本项目的全套代码可以通过jumpserver登录云桌面 gitlab.dpu.tech 网站查看ovs代码的 `code_annotation_branch` 分支在线查看，或者通过命令行 `git clone http://gitlab.dpu.tech/nebula-matrix-sw/ovs.git` 下载到本地进行查看(PS:我没看懂的地方我标注了**toknow**，我之所以不提交md源文档是因为考虑到信息安全就没有弄图床，所以统一转了pdf格式，如果你想要md源文件可以私聊我)。

code_annotation... v ovs / + v

HistoryFind fileWeb IDEDownloadClone

**src code commit with annotation**
polo.li authored 32 minutes ago

db50d369

Name	Last commit	Last update
 .ci	python: Replace pyOpenSSL with ssl.	3 months ago
 .github/workflows	github: Stick to python 3.9.	4 months ago
 openvswitch-2.16.1	src code commit with annotation	32 minutes ago
 .cirrus.yml	python: Replace pyOpenSSL with ssl.	3 months ago
 .gitignore	Remove manpages.mk from git.	1 year ago
 .mailmap	AUTHORS: Update Simon Horman	5 months ago
 .travis.yml	python: Replace pyOpenSSL with ssl.	3 months ago
 OVS源码学习.md	test1	5 days ago
 OVS镜像业务分析.pdf	test1	5 days ago

在注释的过程中，除了少量空格和空行方面的调整外，没有对原始代码进行任何其他改动，最大程度地保证了代码的“原汁原味”。但是由于一个人精力有限，仍然有很多地方没有加上注释，很多业务没有分析，而且肯定有很多不足之处，十分希望有人加入进来，并通过提交代码注释、业务文档、纠错等各种方式为公司的OVS知识积累添金加玉。目前已增加注释的文件如下(部分c文件只是注释了部分核心函数)：

文件	作用
/datapath/linux/action.c	动作执行与fifo相关内容
/datapath/linux/datapath.c	sw_flow初始化与生成相关内容
/datapath/linux/flow.c	流表查询、flow提取key信息等内容
/datapath/linux/flow_netlink.c	解析流表卸载参数、提取metadata
/datapath/linux/flow_table.c	sw_flow流表初始化与查询等内容
/datapath/linux/meter.c	meter链表创建与初始化相关内容
/datapath/linux/vport.c、vport-internal_dev.c、vport-netdev.c	虚拟端口收发包、初始化、销毁等
/lib/conntrack.c	ct下conn链接相关内容
/lib/dpctl.c	dpctl命令行注册
/lib/dpdk.c	dpdk_init初始化、解析参数等内容
/lib/dpif-netdev.c	端口添加、流表查询核心代码流程
/lib/dpif.c	dpif网桥初始化相关内容
/lib/dpif-netdev-private-dfc.c	emc、smc流表初始化、更新、插入等相关内容
/lib/dp-packet.c	报文拷贝
/lib/netdev.c、netdev-dpdk.c、netdev-offload-dpdk.c	卸载rte-flow主流程
/lib/netdev-vport.c	网桥虚拟端口获取等内容
/lib/odp-execute.c、odp-util.c	设置端口执行和flush端口报文
/lib/ovs-numa.c	设置cpu亲和性相关内容
/lib/ovs-router.c、ovs-thread.c、reconnect.c、seq.c、tnl-neigh-cache.c	路由、线程、重连、序号、邻接表
/ofproto/ofproto.c、ofproto-dpif.c、ofproto-dpif-upcall.c、ofproto-dpif-xlate.c	of层交换机初始化、首包上送查openflow流表等内容
/vswitchd/bridge.c	网桥建立、配置和更新相关内容
/vswitchd/ovs-vswitchd.c	交换机初始化流程，包括启守护进程、注册回调、解析参数等

注释风格大致如下：

```

82: .....log_file=/var/log/openvswitch/ovs-vswitchd.log
83: .....pidfile=/var/run/openvswitch/ovs-vswitchd.pid
84: .....detach---monitor
85: -输出参数:»
86: -返回-值-:-»无
87: *****/
88: int
89: main(int argc, char *argv[])
90: {
91:     ..../*unixctl路径*/
92:     ....char *unixctl_path=NULL;
93:     ....struct unixctl_server *unixctl;/*unixctl服务*/
94:     ....char *remote;
95:     ....bool exiting, cleanup;
96:     ....struct ovs_vswitchd_exit_args exit_args={&exiting, &cleanup};
97:     ....int retval;
98:
99:     ....set_program_name(argv[0]);/*设置程序名称、版本、编译日期等信息*/
100:     ....ovsthread_id_init();/*线程id初始化*/
101:
102:     ....dns_resolve_init(true);/*..空函数..toknow*/
103:
104:     ..../*复制出输入的参数列表到新的存储中，让argv指向这块内存，
105:     ....主要是为了后面的proctitle_set()函数（在daemonize_start()->monitor_daemon()中调用，
106:     ....为可能修改原argv存储）做准备*/
107:     ....ovs_cmdl_proctitle_init(argc, argv);
108:
109:     ..../*注册回调和服务管理器出现故障错误时操作的配置*/
110:     ....service_start(&argc, &argv);
111:
112:     ..../*解析参数
113:     .....1.unixctl_path存储unixctl域的sock名，作为接收外部控制命令的渠道；
114:     .....2.而remote存储连接到ovsdb的信息，即连接到配置数据库的sock名
115:
116:     ....ovs有两大进程vswitchd和ovsdb-server，remote用于这两个进程的IPC，即进程间socket通信。
117:     ....remote其实是一个socket文件地址，由ovsdb-server服务端绑定监听时产生，
118:     .....作用类似于网络socket的Ip+Port地址，
119:     ....remote格式如unix:/usr/local/var/run/openvswitch/db.sock。后面创建网桥时会使用。
120:     ....*/
121:     ....remote = parse_options(argc, argv, &unixctl_path);
122:
123:     ..../*忽略pipe读信号的结束*/
124:     ....fatal_ignore_sigpipe();
125:
126:     ..../*让进程变为守护程序
127:     ..... "守护进程"（daemon）就是一一直在后台运行的进程（daemon）。
128:     ....*/

```

```

/*****
·函数名称: dpif_netdev_init
·功能描述: dpif命令行函数注册
·输入参数:
·输出参数:
·返回值: 无
*****/
static int
dpif_netdev_init(void)
{
    static enum pmd_info_type show_aux = PMD_INFO_SHOW_STATS,
                                clear_aux = PMD_INFO_CLEAR_STATS,
                                poll_aux = PMD_INFO_SHOW_RXQ;

    /*流量统计*/
    unixctl_command_register("dpif-netdev/pmd-stats-show", "-pmd-core-[dp]",
                             0, 3, dpif_netdev_pmd_info,
                             (void *)&show_aux);

    /*流量清除*/
    unixctl_command_register("dpif-netdev/pmd-stats-clear", "-pmd-core-[dp]",
                             0, 3, dpif_netdev_pmd_info,
                             (void *)&clear_aux);

    /*pmd-rx队列信息*/
    unixctl_command_register("dpif-netdev/pmd-rxq-show", "-pmd-core-[dp]",
                             0, 3, dpif_netdev_pmd_info,
                             (void *)&poll_aux);

    /*pmd-流量速率*/
    unixctl_command_register("dpif-netdev/pmd-perf-show",
                             "[-nh] [-it-iter-history-len]
                             [-ms-ms-history-len]
                             [-pmd-core-[dp]",
                             0, 8, pmd_perf_show_cmd,
                             NULL);













    /*队列重新调整*/
    unixctl_command_register("dpif-netdev/pmd-rxq-rebalance", "[dp]",
                             0, 1, dpif_netdev_pmd_rebalance,
                             NULL);

    /*log*/
    unixctl_command_register("dpif-netdev/pmd-perf-log-set",
                             "on|off [-b-before] [-a-after] [-e|-ne]
                             [-us-usec] [-q-qlen]",
                             0, 10, pmd_perf_log_set_cmd,
                             NULL);

    /*bond信息*/
    unixctl_command_register("dpif-netdev/bond-show", "[dp]",

```

目前已分析的业务文档如下：

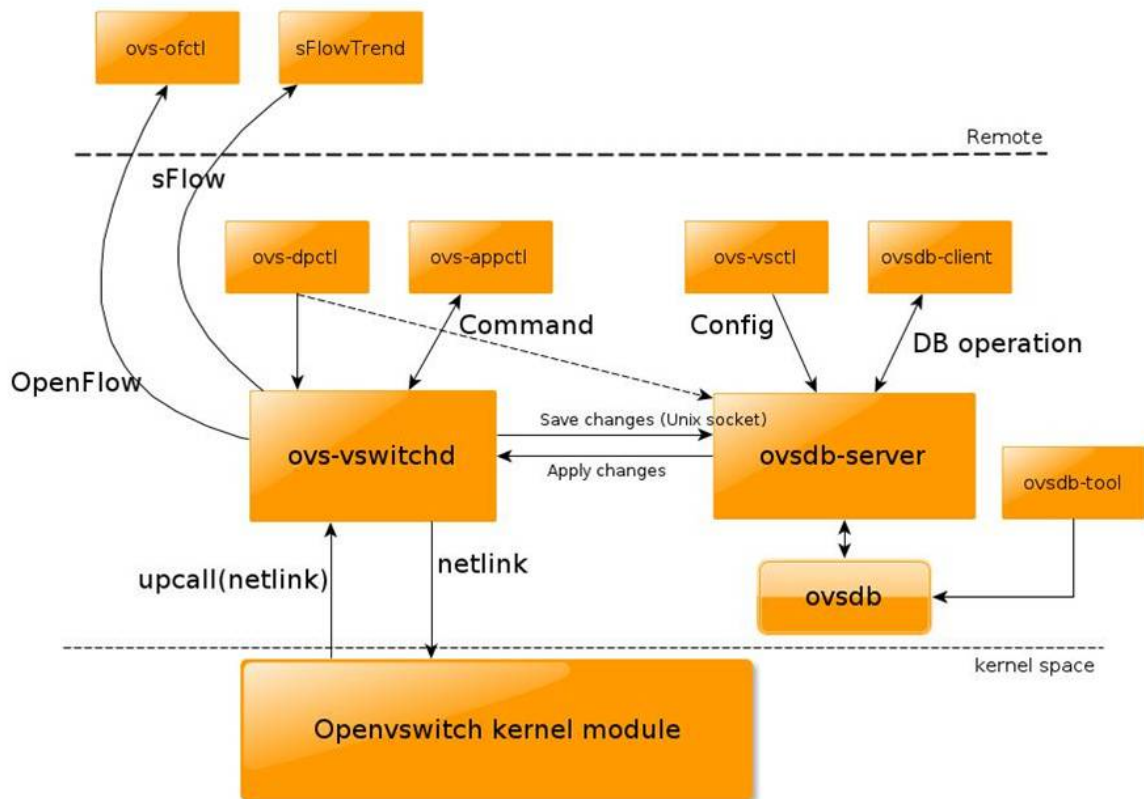
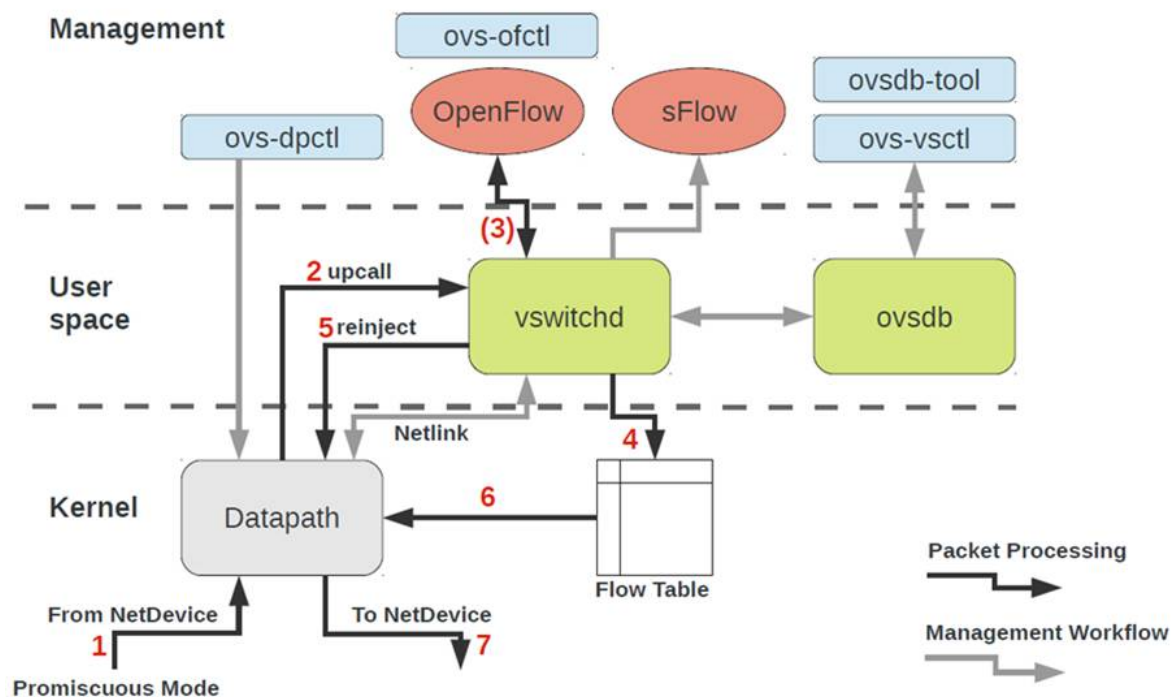
 openvswitch 2.16.1 源码注释.pdf	2022/2/23 15:49	Foxit PDF Reade...	732 KB
 OVS 流表分类器.pdf	2022/2/23 15:27	Foxit PDF Reade...	701 KB
 OVS rte.pdf	2022/2/23 15:36	Foxit PDF Reade...	491 KB
 OVS-DPDK命令行总结.pdf	2022/2/23 15:37	Foxit PDF Reade...	425 KB
 OVS概念学习.pdf	2022/2/23 15:41	Foxit PDF Reade...	410 KB
 OVS镜像业务分析.pdf	2022/2/23 15:42	Foxit PDF Reade...	655 KB
 OVS收包到卸载函数分析.pdf	2022/2/23 15:38	Foxit PDF Reade...	291 KB
 OVS收发包和流表卸载源码学习 .pdf	2022/2/23 15:44	Foxit PDF Reade...	1,030 KB
 qos学习心得.pdf	2022/2/23 15:44	Foxit PDF Reade...	358 KB
 revalidator与handler.pdf	2022/2/23 15:45	Foxit PDF Reade...	251 KB
 几个你也许不知道的ovs知识.pdf	2022/2/23 15:32	Foxit PDF Reade...	387 KB
 流表删除机制.pdf	2022/2/23 15:46	Foxit PDF Reade...	357 KB

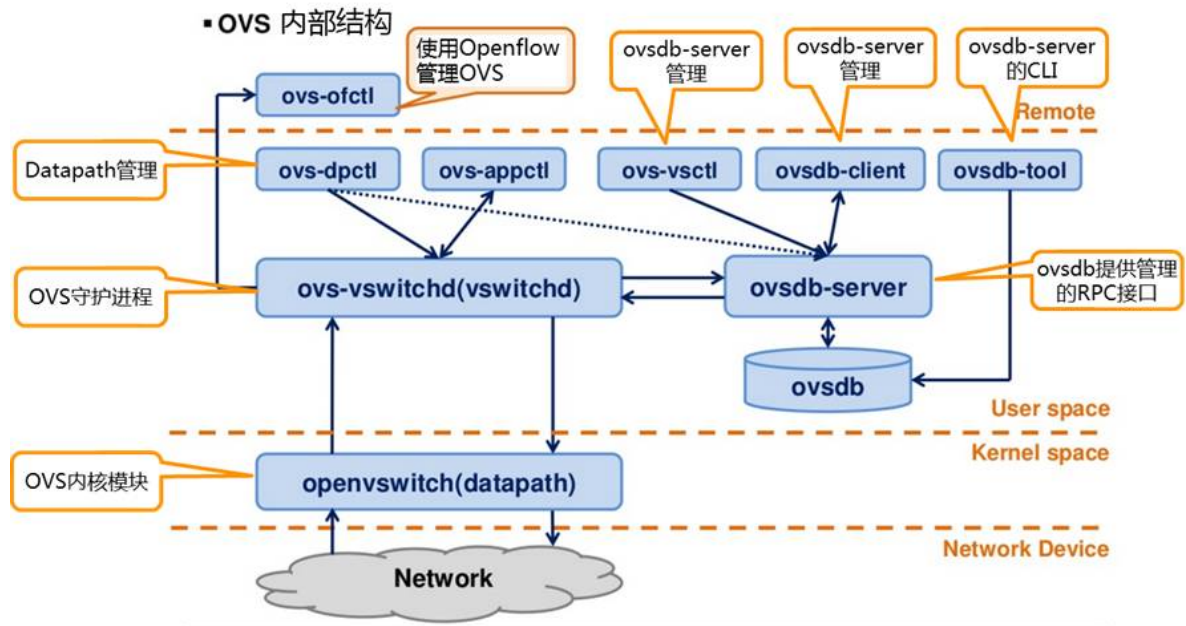
目前仍然需要补充的部分业务列举如下：

名称	进展
bond	目前damon在做相关业务代码，后期如果他有输出文档我会同步上来
aes等	目前我们组无人分析相关加密算法

总体架构

Openvswitch架构:





- `ovs-vswitchd`为主要模块，实现交换机的守护进程daemon。
- `openvswitch.ko`为linux内核模块，支持数据流在内核的交换。

```
1 ~# lsmod | grep openvswitch
2 openvswitch 66901 0
3 gre 13808 1 openvswitch
4 vxlan 37619 1 openvswitch
5 libcrc32c 12644 2 btrfs,openvswitch
```

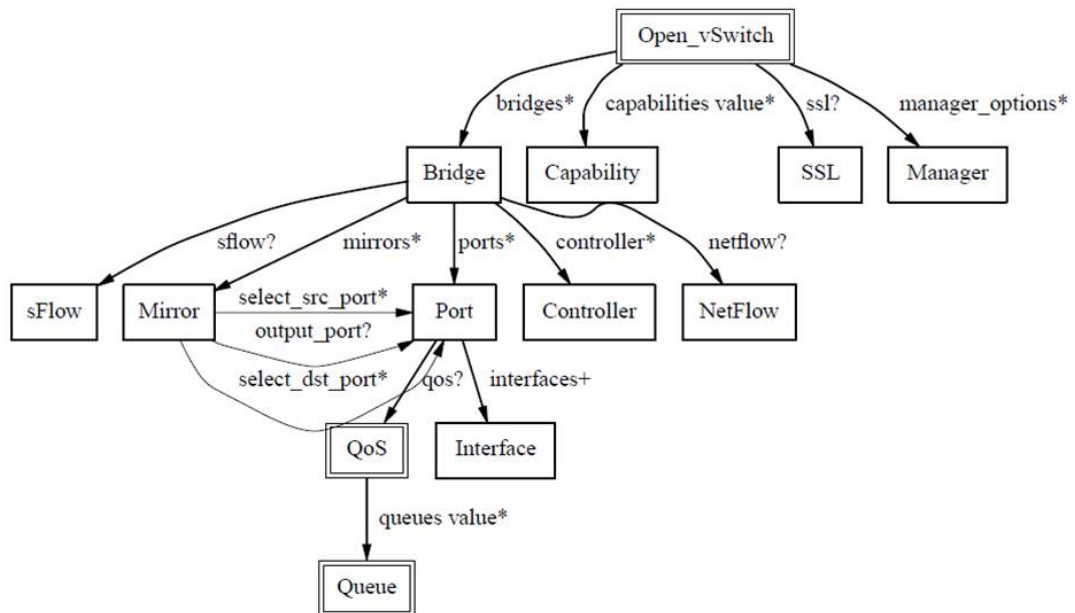
- `ovsdb-server` 轻量级数据库服务器，保存配置信息，`ovs-vswitchd`通过这个数据库获取配置信息。可以通过`ps aux`看到对应进程。

```
1 root 985 0.0 0.0 21172 2120 ? S< Aug06 1:20 ovsdb-server
   /etc/openvswitch/conf.db -vconsole:emer -vsyslog:err -vfile:info --
   remote=punix:/var/run/openvswitch/db.sock --private-
   key=db:Open_vSwitch,SSL,private_key --
   certificate=db:Open_vSwitch,SSL,certificate --bootstrap-ca-
   cert=db:Open_vSwitch,SSL,ca_cert --no-chdir --log-
   file=/var/log/openvswitch/ovsdb-server.log --
   pidfile=/var/run/openvswitch/ovsdb-server.pid --detach -monitor
```

`ovsdb-server`将配置信息保存在`conf.db`中，并通过`db.sock`提供服务，`ovs-vswitchd`通过这个`db.sock`从这个进程读取配置信息。`/etc/openvswitch/conf.db`是json格式的，可以通过命令`ovsdb-client dump`将数据库结构打印出来。

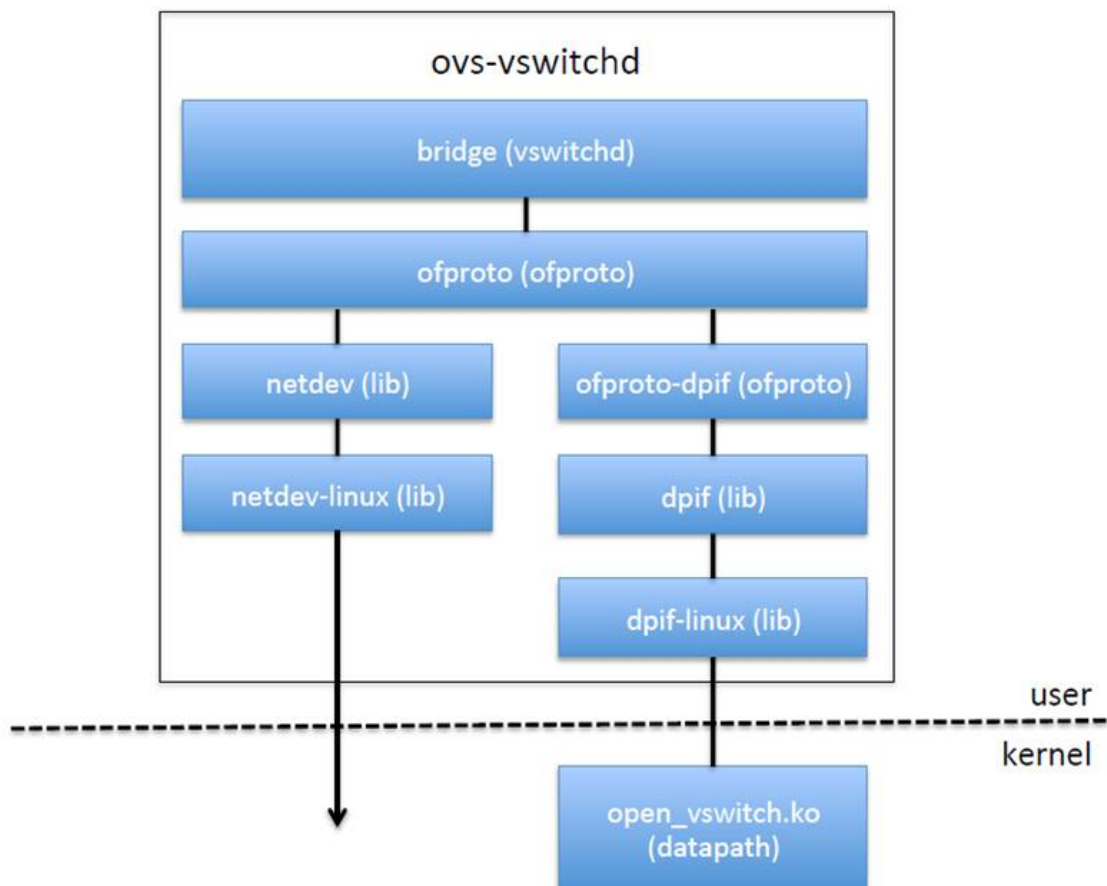
```
1 //数据库结构主要包含如下表格
2 Port、Manager、Bridge、interface、SSL、IPFIX、NetFlow、Qos、sFlow、FlowTab等
```

数据库结构如下：



- 1 | `ovs-vsctl` 创建的所有的网桥，网卡，都保存在数据库里面，`ovs-vswitchd`会根据数据库里面的配置创建真正的网桥，网卡。
- 2 | `ovs-dpctl` 用来配置switch内核模块。
- 3 | `ovs-vsctl` 查询和更新`ovs-vswitchd`的配置。
- 4 | `ovs-appctl` 发送命令消息，运行相关daemon。
- 5 | `ovs-ofctl` 查询和控制OpenFlow交换机和控制器。

代码架构



- 1 ovs-vswitchd会从ovsdb-server读取配置，然后调用ofproto层进行虚拟网卡的创建或者流表的操作。
- 2 ofproto是一个库，实现了软件的交换机和对流表的操作。
- 3 Netdev层抽象了连接到虚拟交换机上的网络设备。
- 4 Dpif层实现了对于流表的操作。

对于OVS来讲，有以下几种网卡类型：

- 1 1). netdev: 通用网卡设备 eth0 veth(OVS+DPDK 用户态流表，新建bridge的时候指定type为netdev)
接收：一个netdev在L2收到报文后回直接通过ovs接收函数处理，不会再走传统内核协议栈。
发送：ovs中的一条流指定从该netdev发出的时候就通过该网卡设备发送
- 2 2). internal: 一种虚拟网卡设备
接收：当从系统发出的报文路由查找通过该设备发送的时候，就进入ovs接收处理函数
发送：ovs中的一条流制定从该internal设备发出的时候，该报文被重新注入内核协议栈
- 3 3). gre device: gre设备。不管用户态创建多少个gre tunnel，在内核态有且只有一个gre设备
接收：当系统收到gre报文后，传递给L4层解析gre header，然后传递给ovs接收处理函数
发送：ovs中的一条流制定从该gre设备发送，报文会根据流表规则加上gre头以及外层包裹ip，查找路由发送

此电脑 > 本地磁盘 (D:) > pre_analy_code > ovs_2_16_1 > openvswitch-2.16.1

名称	修改日期	类型	大小
.ci	2021/12/1 12:57	文件夹	
.github	2021/12/1 12:57	文件夹	
build-aux	2021/12/1 12:57	文件夹	
datapath	2021/12/1 12:57	文件夹	
datapath-windows	2021/12/1 12:57	文件夹	
debian	2021/12/1 12:57	文件夹	
Documentation	2022/1/10 16:48	文件夹	
include	2021/12/1 12:57	文件夹	
ipsec	2021/12/1 12:57	文件夹	
lib	2021/12/1 12:57	文件夹	
m4	2021/12/1 12:57	文件夹	
ofproto	2021/12/1 12:57	文件夹	
ovsdb	2021/12/1 12:57	文件夹	
poc	2021/12/1 12:57	文件夹	
python	2021/12/1 12:57	文件夹	
rhel	2021/12/1 12:57	文件夹	
selinux	2021/12/1 12:57	文件夹	
tests	2021/12/1 12:57	文件夹	
third-party	2021/12/1 12:57	文件夹	
tutorial	2021/12/1 12:57	文件夹	
utilities	2021/12/1 12:57	文件夹	
vswitchd	2021/12/1 12:57	文件夹	
vtep	2021/12/1 12:57	文件夹	
windows	2021/12/1 12:57	文件夹	
xenserver	2021/12/1 12:57	文件夹	
.cirrus.yml	2021/10/22 6:14	YML 文件	1 KB
.mailmap	2021/10/22 6:16	MAILMAP 文件	5 KB

- 1 | `vswitchd`中就是`ovs-vswitchd`的入口代码，
- 2 | `ovsdb`就是`ovsdb-server`的代码，
- 3 | `ofproto`即上述的中间抽象层，
- 4 | `lib`下面有`netdev`，`dpif`的实现，
- 5 | `datapath`里面就是内核模块`openvswitch.ko`的代码

核心网桥初始化：

