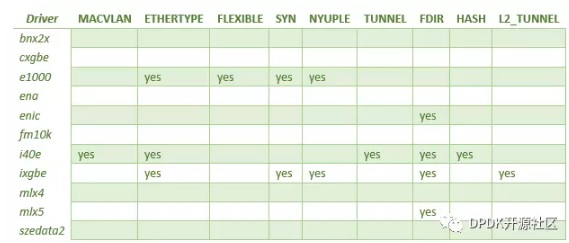
www.sysight.com/index.php?qa=471&qa\_1=generic-flow-api%E7%AE%80%E4%BB%8B

**Classification功能**

Classification功能是指网卡在收包时，将符合某种规则的包放入指定的队列。网卡一般支持一种或多种classification功能，以intel 700系列网卡为例，其支持MAC/VLAN filter、Ethertype filter、Cloud filter、flow director等等。不同的网卡可能支持不同种类的filter，例如intel 82599系列网卡支持n-tuple、L2\_tunnel、flow director等，下表列出了DPDK中不同驱动对filter的支持。即使intel 82599系列网卡和intel 700系列网卡都支持flow director，它们支持的方式也不一样。那DPDK是如何支持不同网卡的filter的呢？



**现有flow API**

DPDK定义所有网卡的filter类型以及每种filter的基本属性，并提供相应的接口给上层应用，所以要求用户对filter属性有一定的概念。以flow director为例，其中有两个数据结构为：

/\*\*

 \* A structure used to contain extend input of flow

 \*/

struct rte\_eth\_fdir\_flow\_ext {

        uint16\_t vlan\_tci;

        uint8\_t flexbytes[RTE\_ETH\_FDIR\_MAX\_FLEXLEN];

        /\*\*< It is filled by the flexible payload to match. \*/

        uint8\_t is\_vf;   /\*\*< 1 for VF, 0 for port dev \*/

        uint16\_t dst\_id; /\*\*< VF ID, available when is\_vf is 1\*/

};

/\*\*

 \* A structure used to define the input for a flow director filter entry

 \*/

struct rte\_eth\_fdir\_input {

        uint16\_t flow\_type;

        union rte\_eth\_fdir\_flow flow;

        /\*\*< Flow fields to match, dependent on flow\_type \*/

        struct rte\_eth\_fdir\_flow\_ext flow\_ext;

        /\*\*< Additional fields to match \*/

};

用户在添加flow director流规则的时候，需要填写上述信息。但并不是所有的网卡都关注flow\_type 和is\_vf，比如intel 82599系列网卡不需要flow\_type及is\_vf。其它类型的filter也有类似的情况。

从上可以看出，现有方案有很多缺陷：首先，DPDK为所有网卡抽象出统一的属性，但是某些属性只对一种网卡有意义；其次，随着DPDK支持的网卡越来越多，DPDK需要定义的filter类型要增加，网卡filter功能升级也需要DPDK作相应修改，这样很容易导致API/ABI的破坏；

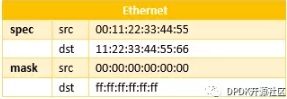
另外，从应用角度来看，现有的方案也有诸多不便，使得API比较难用，不够友好：那些可选或者可缺省的属性容易让用户产生疑惑；经常在某种filter类型中随意插入一些某个网卡特有的属性；设计复杂，也没有比较详细的说明文档。

鉴于上述原因，一个generic flow API必不可少。

**Generic flow API**

DPDK v17.02 推出了generic flow API方案，DPDK把一条流规则抽象为pattern和actions两部分。

Pattern由一定数目的item组成。Item主要和协议相关，支持ETH, IPV4, IPV6, UDP, TCP, VXLAN等等。item也包括一些标志符，比如PF, VF, END等，目前DPDK支持的item类型定义在rte\_flow.h的enum rte\_flow\_item\_type。在描述一个item的时候可以添加spec和mask，告诉驱动哪些需要匹配。下面以以太网包的流规则为例，该item描述的是精确匹配二层包头的目的地址11:22:33:44:55:66。

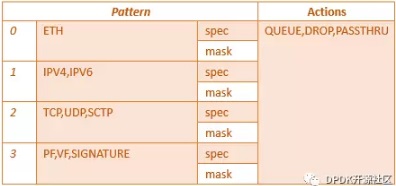


Actions表示流规则的动作，比如QUEUE, DROP, PF, VF,END等等，DPDK支持的action类型定义在rte\_flow.h的enum rte\_flow\_action\_type。以下action表示符合某种pattern的包放入队列3。

http://sysight.com/?qa=blob&qa_blobid=9059864200635835811

该方案把复杂的区分filter类型的事情交给驱动处理，用户再也不需要关注硬件的能力，这样使得上层应用能够方便添加或者删除流规则。

如果想添加一条流规则，上层应用只需要调用rte\_flow\_create()这个接口，并填好相应的pattern和actions；如果要删除一条流规则，上层应用只需要调用rte\_flow\_destroy()。还是以flow director为例，流规则定义如下所示，对于用户来说，这种方式更易操作。



综上所述，generic flow API明显方便很多。