**ODL yangman下发流表**

Feng Wei 2018.4

**说明**：

1.我使用的是OpenDayLight(Nitrogen SR2)版本，使用其他的版本也没问题，只要能正确安装好features即可。

2.获取流表项/流表/组表/计量表等等信息使用operational, 进行相关设置则使用config,以下只介绍config部分，一般情况下默认流表为0号流表。

3. GET从OpenDaylight获取数据，

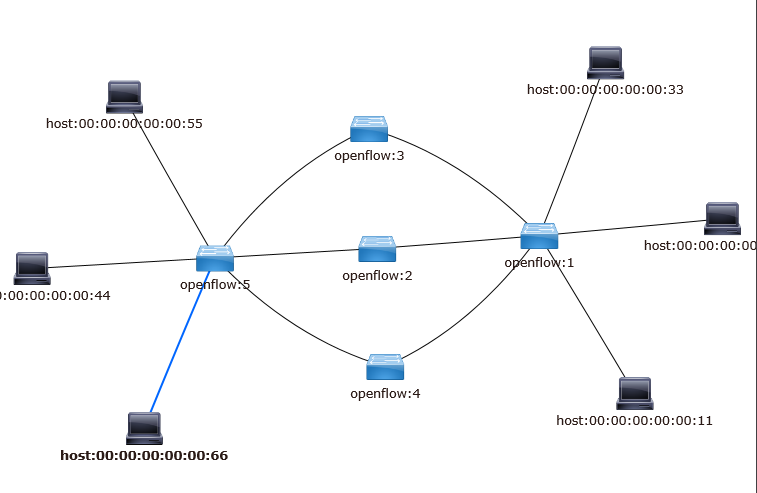
PUT和POST将数据发送到OpenDaylight进行保存，

DELETE将数据发送到OpenDaylight进行删除。

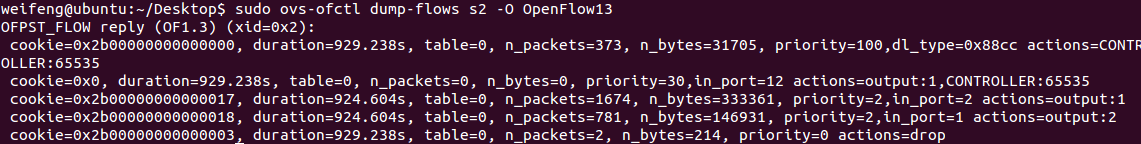
4. yangman删除/修改流表与下发流表类似，不再累述。

5.可参考[OpenDaylight用户指南](http://docs.opendaylight.org/en/stable-nitrogen/user-guide/index.html) [OpenDaylight简单流表操作](https://www.sdnlab.com/15173.html)

**测试拓扑（任意搭建拓扑测试）：**



拓扑搭建时可以ping通，可查看流表，例如查看s2流表项：



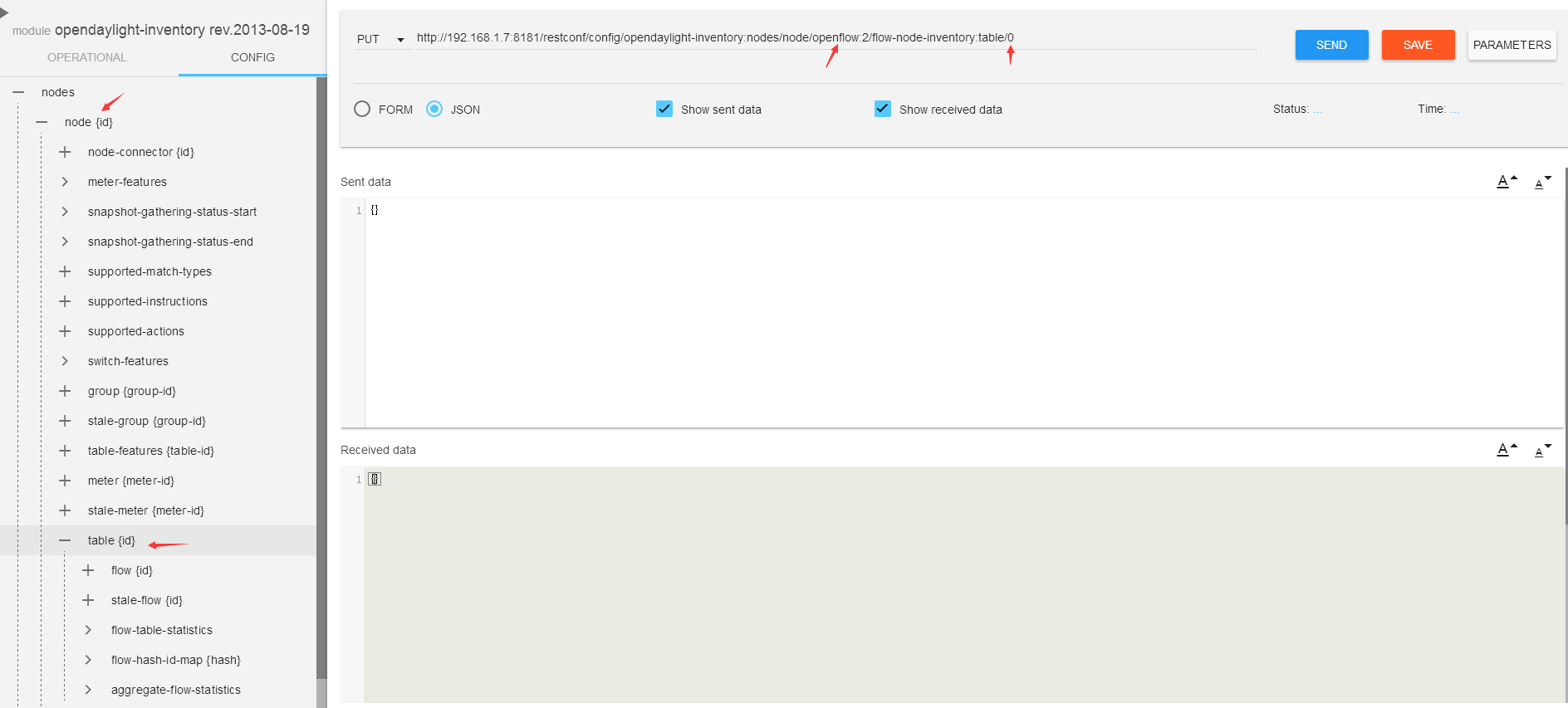
**向s2下发流表：**

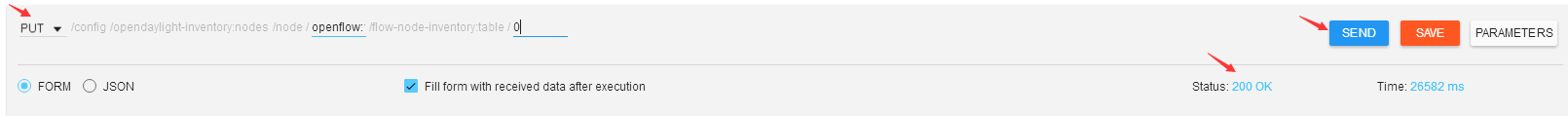
注：需要说明的是，在测试的时发现，此ODL版本中下发flow entry时的instruction字段中的apply-action-case所列举的action并没有得到很好的支持，即并不是每个action都可以被下发的交换机中，例如controller-action-case、flood-action-case等等。

**I NetConfig下发流表**

OpenDayLight Flows功能：该模块的功能主要实现静态流表添加，下发，删除及修改等功能。ODL下发flow table的界面也在Yang UI中，具体节点为：Yang UI API opendaylight-inventory config nodes node{id} table{id} flow{id}

举例：向s2交换机添加流表：匹配项为：in\_port:12 动作是：output:1,output:CONTROLLER





**With JSON:**

{

"table": [

{

"id": "0",

"flow": [

{

"id": "0",

"match": {

"in-port": "12"

},

"instructions": {

"instruction": [

{

"order": "0",

"apply-actions": {

"action": [

{

"order": "0",

"output-action": {

"output-node-connector": "2",

"max-length": "65535"

}

},

{

"order": "1",

"output-action": {

"output-node-connector": "CONTROLLER",

"max-length": "65535"

}

}

]

}

}

]

},

"installHw": "true",

"barrier": "true",

"strict": "true",

"priority": "30",

"idle-timeout": "0",

"hard-timeout": "0",

"table\_id": "0"

}

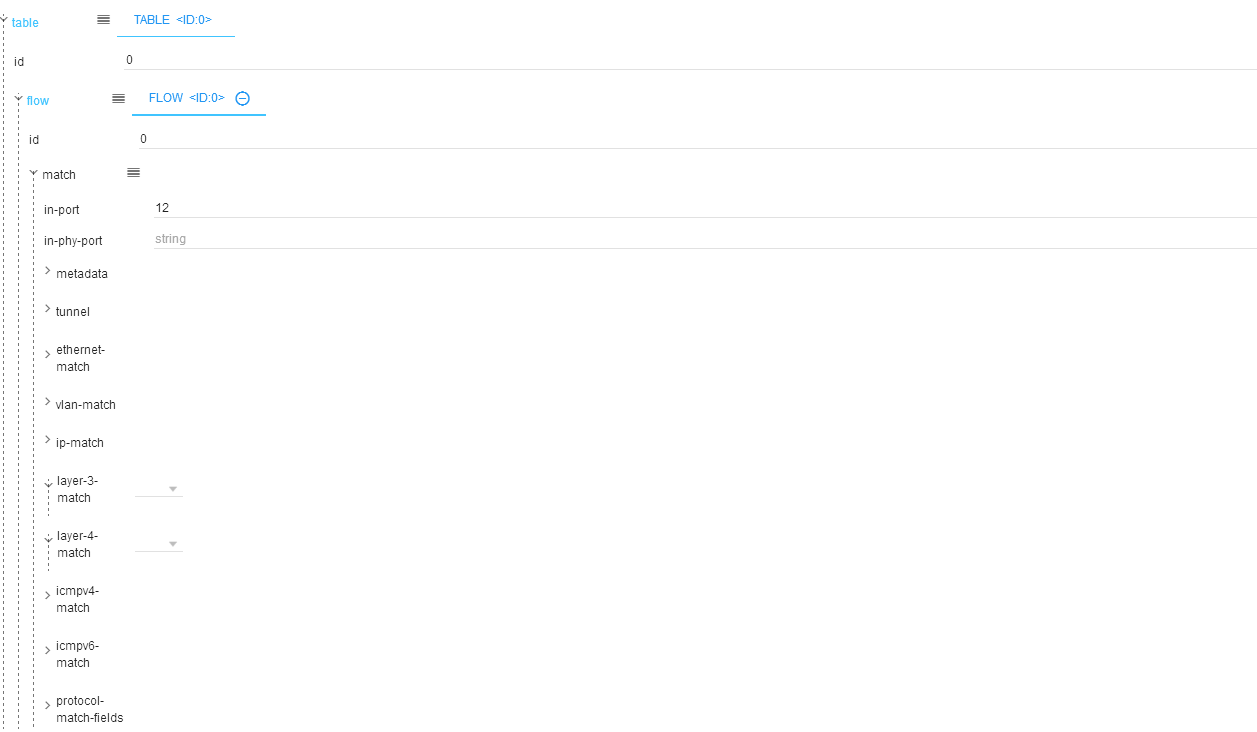
]

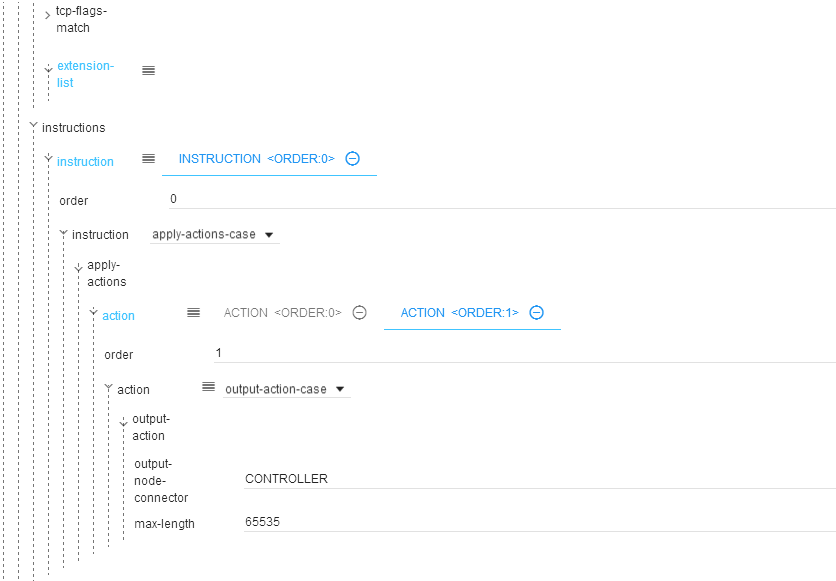
}

]

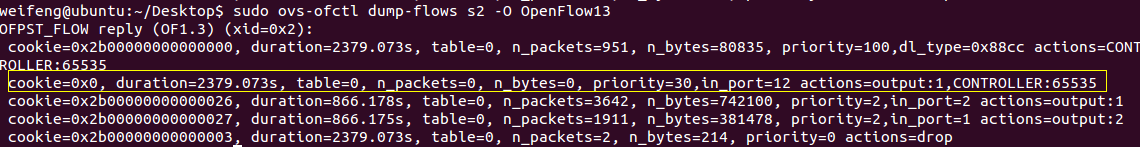
}

**With FORM:**





采取任意一种方式下发流表，我们来查看流表：



**II RPC下发流表(测试)**

举例：向s2交换机添加流表：匹配项为：in\_port:20 动作是：NORMAL

同样可通过表单和JSON两种形式下发，以下仅说明JSON格式，表单可仿上自行填写，可选的只有POST

填写表单处注意：



流表下发成功

