

P4源码运行步骤

1.在基于Tofino架构的P4交换机上编译运行IBLT.p4源码文件

这里以Tofino X312P-T交换机为例，说明我们的编译步骤

```
//1.在IBLT.p4所在目录下(为了统一起见，以下所有命令请均在IBLT.p4所在目录下执行)输入以下命令  
编译源代码--IBLT.p4  
p4_build-9.x.y.sh IBLT.p4
```

2.启动switchd

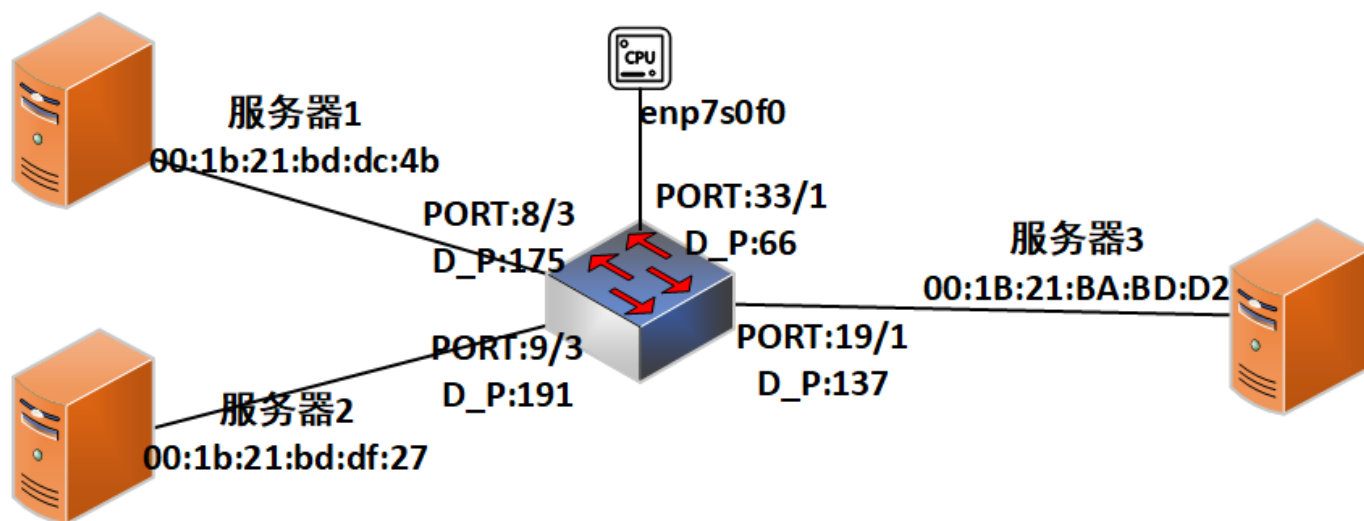
```
//2.输入以下命令启动switchd--启动输出参数会存放在当前目录下nohup.out文件中  
nohup run_switchd.sh -p IBLT &
```

3.启动bfshell 并配置交换机相应端口

```
//3.输入以下命令启动bfshell 并配置交换机相应端口  
bfshell -f port_up_command.txt
```

重点说明

启动端口需要根据实际P4交换机的端口进行相应的启动。这里说明一下我们的实际网络拓扑图。
(由于我们的P4交换机连接有其余的服务器等，所以我们port_uo_command.txt启动的不仅仅只有下图中的端口，可以选择删除不启动除了下述网络拓扑图的其余端口，步骤四中的SwitchIngress.L2_fwd的流表项的添加也同理)



4.使用controller_test.py脚本向数据面下发转发流表

```
//4.输入以下命令进行流表项的添加  
python controller_test.py
```

重点说明

转发流表中SwitchIngress.L2_fwd中的mac地址和端口信息需要根据实际的网络拓扑和交换机的实际启用端口信息进行相应的配置。

目前所使用的端口信息和步骤三中的网络拓扑图相对应。

5.使用back_and_test.py脚本从P4交换机数据面接收数据包解码提取拥塞流信息并保存为json文件

```
//5.输入以下命令从P4交换机数据面接收数据包提取拥塞流信息  
python3 back_end_test.py enp7s0f0
```

重点说明

enp7s0f0为P4交换机镜像传输拥塞包的端口，需要根据实际情况进行替换

其中解码提取的中拥塞流信息会保存在 IBLT.p4所在目录/medium_dict/xxxx.json文件中

其中解码提取的高拥塞流信息会保存在 IBLT.p4所在目录/high_dict/xxxx.json文件中

其中镜像传输所使用的镜像数据包个数会保存在 IBLT.p4所在目录下的count.txt文件中

6.在服务器1使用DPDK-Pktgen进行发包

6.1 在服务器1中安装DPDK版本19.11.11 pktgen版本19.12.0

6.2 启动DPDK和pktgen进行发包

```
//服务器1往服务器3发包 发包参数如下
Pktgen:/> set 0 dst mac 00:1b:21:ba:bd:d2
Pktgen:/> set 0 size 650
Pktgen:/> set 0 proto udp
Pktgen:/> start 0
```

7.在服务器2中使用tcpreplay重放微突发流数据集

7.1 安装tcpreplay

7.2 重放微突发流数据集

```
//在数据集所在目录下（以下统一在数据集所在目录下）输入以下命令间隔1s重放所有微突发流数据集
./timingSend.sh 1
//输入以下命令只重放一个微突发流数据集
./sendone.sh pt1_copy.pca
//实时性对比时所输入的命令
//镜像传输方式以5s为间隔进行重放
./timeSend.sh 1 5
//grpc传输方式以40s为间隔进行重放（grpc方式所使用的P4源码和控制器获取信息的脚本不一样，以下
会说明GRPC代码的使用方式）
./timeSend.sh 2 40
```

GRPC交互方式所使用的P4源码和控制器代码（源码均在GRPC文件夹内）使用说明

1.编译P4代码（这里的P4代码为GRPC文件下的P4代码，编译方法同上）

2.启动switchd（同上）

3.启动bfshell 并配置交换机相应端口（同上）

4.使用controller_test.py脚本向数据面下发转发流表（同上）

5.运行register_read_test_new.py获取P4交换机内存存储的拥塞流信息

```
//输入以下命令运行  
python register_read_test_new.py
```

6.运行resolve_IBLT_new.py解码并存储步骤5获取的拥塞流信息

```
//输入以下命令运行  
python3 resolve_IBLT_new.py
```

说明：GRPC交互方式中所使用的网络拓扑和发包方式均同上，可参考上述说明进行运行测试