

OPEN VSWITCH OPENFLOW 性能测试报告

全球 SDN 测试认证中心 SDNCTC

2016 年 10 月 14 日

目 录

一、引言.....	1
二、测试环境配置.....	1
2.1 待测设备	1
2.2 测试工具	1
2.3 测试拓扑	2
三、测试建议.....	2
3.1 隔离建议	2
3.2 连接建议	2
3.3 重复建议	3
四、性能测试例.....	3
4.1 PACKET IN 性能.....	3
4.1.1 TABLE-MISS 表项的 PACKET IN 性能.....	3
4.1.2 非 TABLE-MISS 表项的 PACKET_IN 性能.....	5
4.2 流表性能	6
4.2.1 流表查询时间.....	6
4.2.2 流表安装时间	7
4.2.3 流表删除时间	10
4.2.4 流表修改时间	11
4.2.5 流表容量	13
4.3 PACKET OUT 性能.....	14

4.3.1 PACKET OUT 消息吞吐/时延.....	14
4.4 交换机统计性能测试	15
4.4.1 流统计性能	15
4.4.2 端口统计性能	16
4.5 交换机定时性能测试	18
4.5.1 HARD 定时误差	18
4.5.2 IDLE 定时误差	19

一、引言

目前，交换机厂商已经陆续推出了多款 OpenFlow 交换机，并且已经通过了 OpenFlow 1.3 的一致性认证。

但是对于网络用户来说，交换机设备性能是否达到相关要求，这些仍然是未知的。

为此，全球 SDN 测试认证中心开发了交换机性能测试工具 OFsuite_Performance(switch)。

本文档以开源交换机 Open vSwitch 作为待测交换机，使用 OFsuite_Performance(switch)执行性能测试，汇总测试结果得出性能测试报告。报告中也以具体测试例的形式介绍了一些 OpenFlow 交换机性能测试的方法和步骤。

二、测试环境配置

2.1 待测设备

待测交换机为 Open vSwitch v2.5.0，运行于 Linux 服务器上，服务器配置如下：

- 服务器型号: Dell PowerEdge R720
- 处理器： Intel(R) Xeon(R) CPU E5-2609 v2 @ 2.50GHz 4 核
- 内存： 16GB 1600MHz
- 操作系统： Ubuntu server 14.04.1 LTS 64 位
- 控制通道网卡： 1Gbps × 1
- 数据通道网卡： 1Gbps × 4

2.2 测试工具

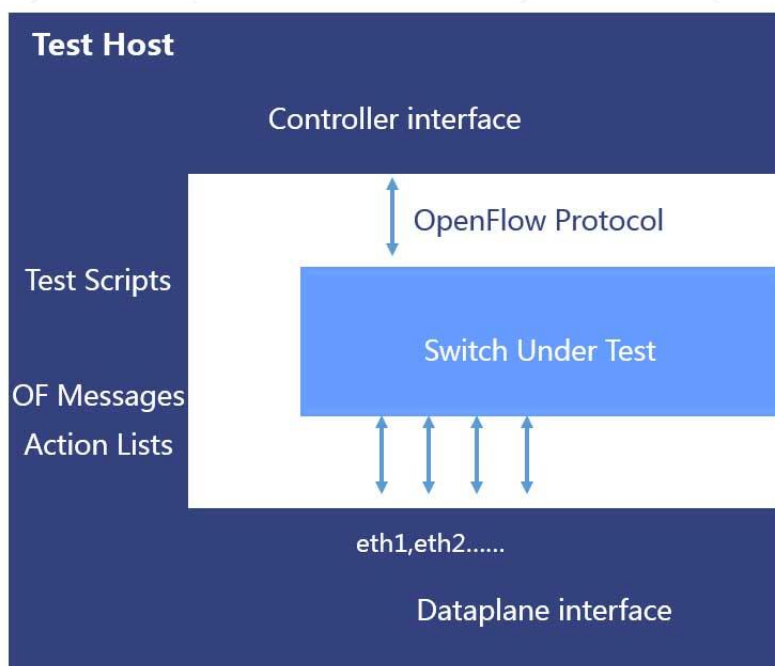
测试工具为 OFsuite_Performance(switch)，运行于 Linux 服务器上，服务器配置如下：

- 处理器： Intel(R) Xeon(R) E3-1230 @ 3.20GHz 4 核
- 内存： 8GB 1333MHz
- 操作系统： Ubuntu server 12.04 LTS 64 位

- 控制通道网卡：1Gbps × 1
- 数据通道网卡：1Gbps × 4

2.3 测试拓扑

待测设备与测试工具通过一个控制通道和四个数据通道相互连接。OpenFlow 消息通过控制通道传输，模拟的网络流量通过数据通道传输。



三、测试建议

3.1 隔离建议

建议性能测试中所有的测试例在隔离的实验室环境中进行测试以保证测试结果的准确性；

在实验室中进行测试时，建议阻止任何与测试无关的网络流量进入测试环境。

3.2 连接建议

建议待测设备直接连接到测试工具以避免其他中间设备带来的时延；

建议对 Linux 服务器的 TCP 协议栈进行调优。

3.3 重复建议

所有性能测试例建议单独执行至少 10 次测试以消除性能不稳定带来的负面影响从而保证统计结果的可信度。

四、性能测试例

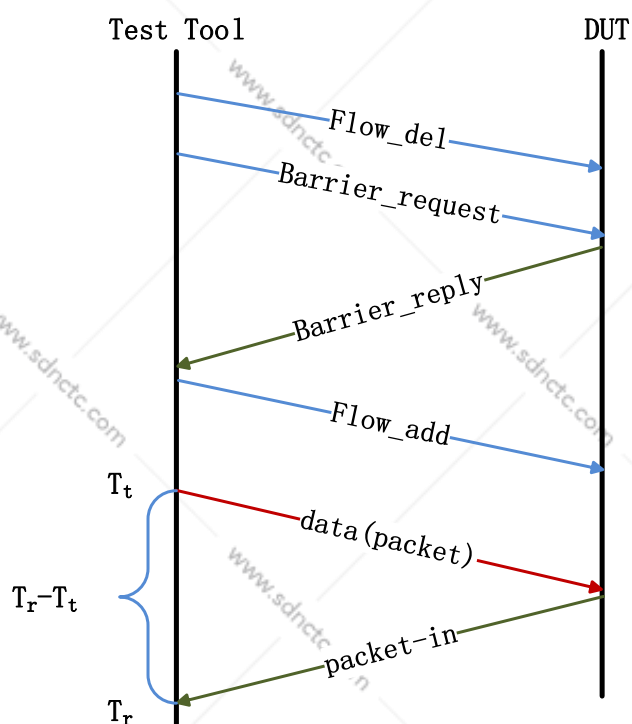
4.1 PACKET IN 性能

4.1.1 TABLE-MISS 表项的 PACKET IN 性能

- 测试目的

OpenFlow 1.3 中新增了 TABLE-MISS 表项，通过 TABLE-MISS，当交换机收到没有匹配流表的网络包时，会向控制器发送 PACKET_IN 消息。本测试例测试交换机收到未匹配的包时，上报 PACKET_IN 所花费的时间。

- 测量方法



如上图所示，测试时首先删除交换机的所有流表，确认删除成功后再下发一条 TABLE-MISS 表项，待下

发成功并且该条流表生效后，在数据通道发送一个包到交换机。记此时间为 T_t ，待交换机上报 PACKET_IN 时，记此时间为 T_r ，则测试结果为 $T_{result}=T_r - T_t$ 。

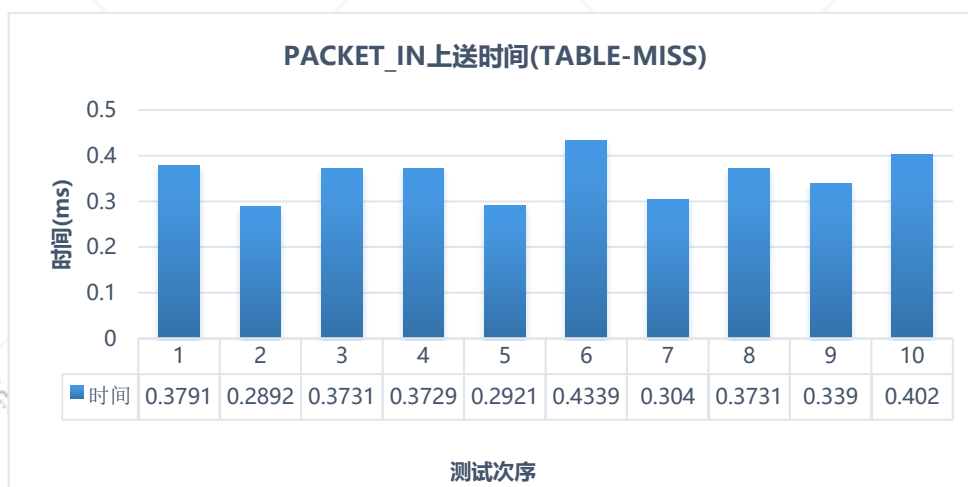
- **测试前提条件**

1. 待测交换机兼容 OpenFlow 1.3 协议
2. 待测交换机上送 PACKET_IN 消息不使用 buffer 进行缓存

- **测试步骤**

1. 设置待测交换机连接到测试工具
2. 删除交换机中的所有流表
3. 下发 TABLE-MISS 流表项，匹配项为通配，动作为转发给控制器
4. 等待足够长的时间（例如 5 秒），以使该条流表生效
5. 在数据通道发送一个包到待测交换机，记录此时间为 T_t
6. 当测试工具收到 PACKET_IN 消息时，记录此时间为 T_r
7. 测试结果即为 $T_{result}=T_r - T_t$
8. 改变数据通道封包的类型和长度重复进行测试

- **测试结果**



上图测试结果是数据通道封包类型为 ARP、长度为 60 字节时，OVS 上报 PACKET_IN 所用的时间。在

TABLE-MISS 表项下，OVS 上报 PACKET_IN 时间平均值为 0.356 毫秒。

4.1.2 非 TABLE-MISS 表项的 PACKET_IN 性能

- **测试目的**

在普通的流表中，通过设置动作为转发到控制器，交换机会将匹配到的数据包封装成 PACKET_IN 上送给控制器。本测试例测试交换机收到匹配的包时，上报 PACKET_IN 花费的时间。

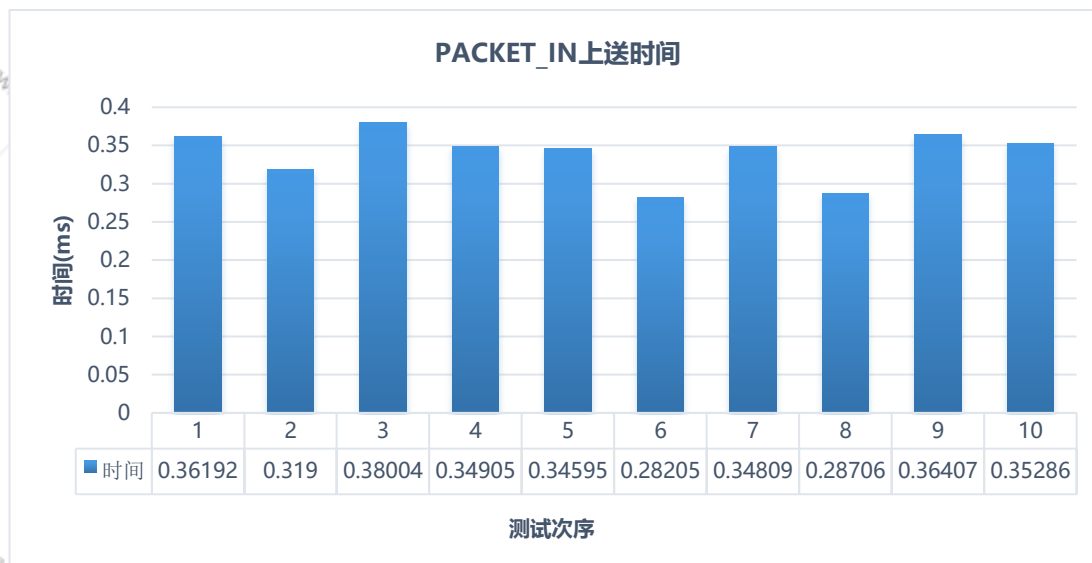
- **测试前提条件**

1. 待测交换机兼容 OpenFlow 1.3 协议
2. 待测交换机上送 PACKET_IN 消息不使用 buffer 缓冲

- **测试步骤**

1. 设置待测交换机连接到测试工具
2. 删除交换机中的所有流表
3. 下发一条流表，流表匹配一个入端口，动作为转发到控制器
4. 等待足够长的时间（例如 5 秒），以使流表生效
5. 从数据通道发送一个包到待测交换机，记录此时间为 T_t
6. 记录收到待测交换机发送 PACKET_IN 的时间为 T_r
7. 测试结果即为 $T_{result} = T_r - T_t$
8. 改变数据通道封包的类型和长度重复进行测试

- **测试结果**



上图测试结果是数据通道封包类型为 ARP、长度为 60 字节时，OVS 上报 PACKET_IN 所用的时间。普通流表项的 PACKET_IN 上送时间与 TABLE-MISS 表项相差不多，平均值为 0.339 毫秒。

4.2 流表性能

4.2.1 流表查询时间

- **测试目的**

测试待测交换机查询一条流表的时间。

- **测试前提条件**

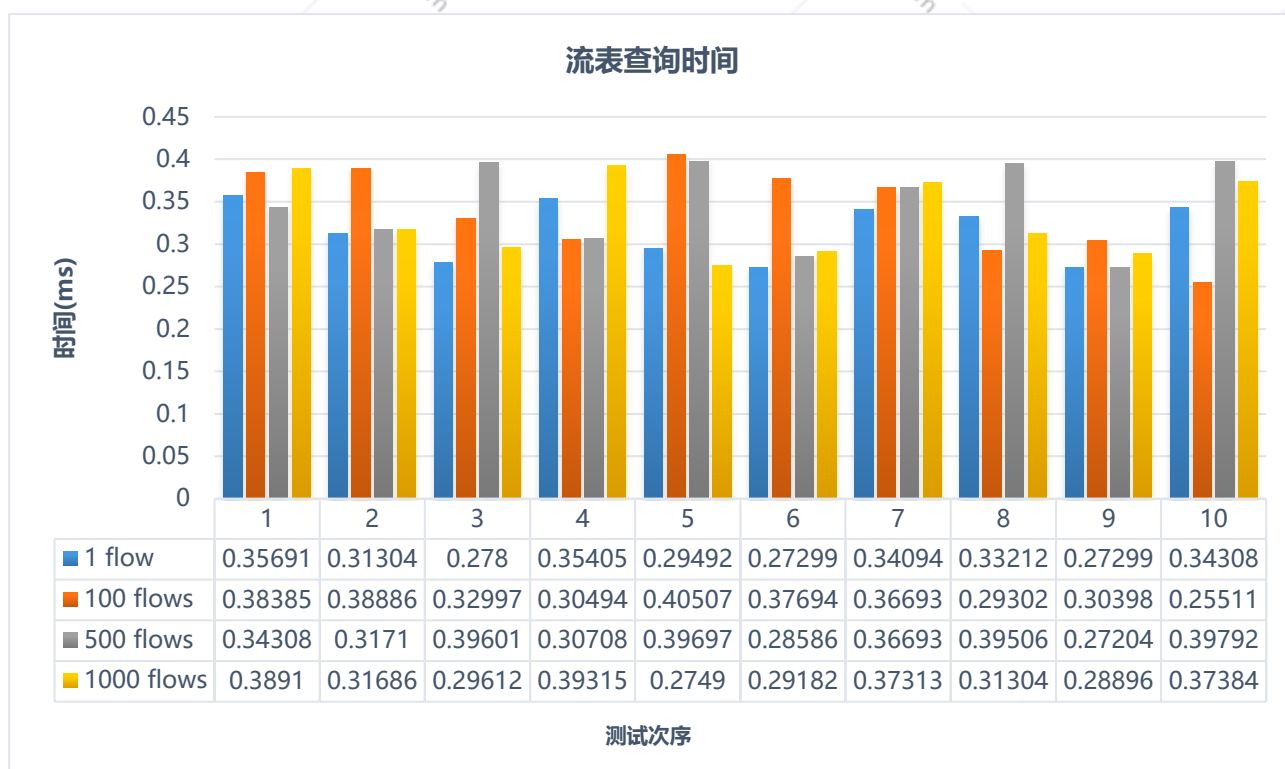
1. 待测交换机兼容 OpenFlow 1.3 协议
2. 待测交换机最大流表项数量为已知

- **测试步骤**

1. 设置待测交换机连接到测试工具
2. 删除待测交换机的所有流表
3. 向待测交换机添加一条流表，动作为转发到某一个端口
4. 等待足够长的时间（例如 5 秒），以使该流表项生效

5. 从数据通道发送一个 TCP 包，记录此时间为 T_t
6. 记录该 TCP 包从相应端口被转发出来的时间为 T_r
7. 测试结果即为 $T_{result} = T_r - T_t$
8. 改变安装流表项的数量（不多于已知的最大流表项数量）重复进行测试

• 测试结果



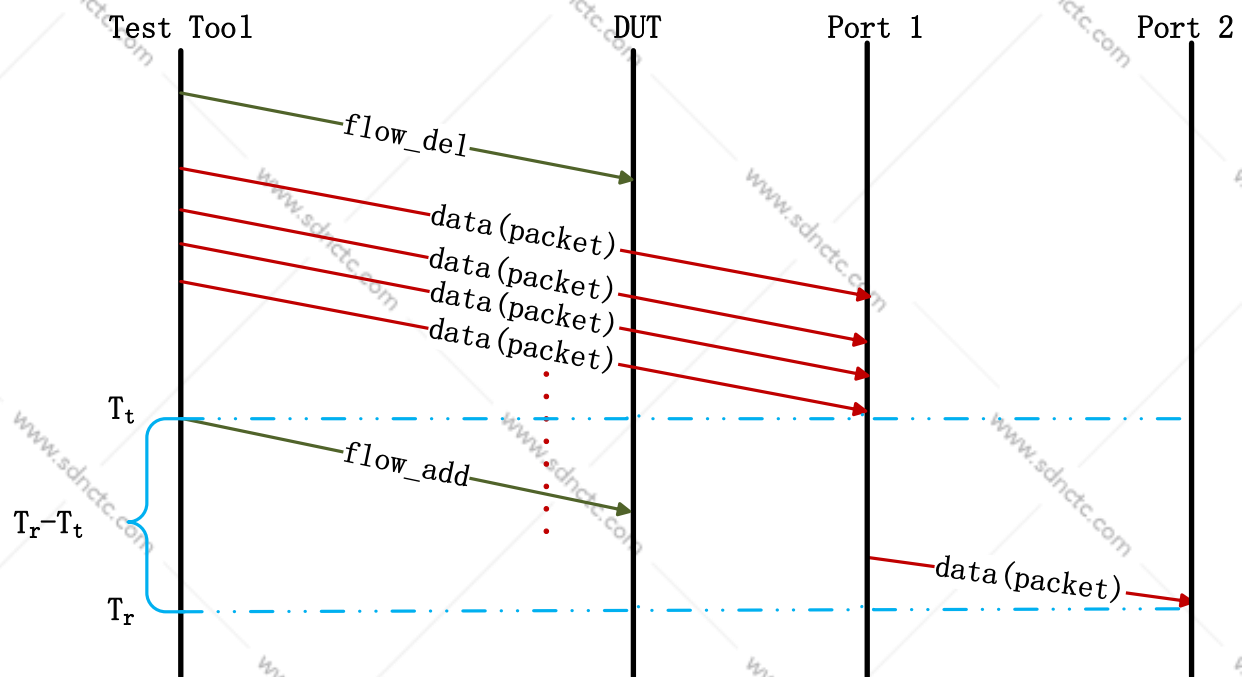
流表查询时间性能对于 OpenFlow 交换机来说是一个很重要的衡量指标。从上图测试结果中可以看到，OVS 在 1000 条流表中精确查询一条表项的时间在 0.4 毫秒以内。同时也可以看到，随着流表项数量的增加，OVS 精确查询其中一条流表的时间并没有明显增长。

4.2.2 流表安装时间

• 测试目的

测试待测交换机安装一条流表并使之生效的时间

- 测量方法



- 测试前提条件

1. 待测交换机兼容 OpenFlow 1.3 协议
2. 待测交换机最大流表项数量为已知
3. 待测交换机支持 PUSH_VLAN 和 SET_FIELD 动作

- 测试步骤

1. 设置待测交换机连接到测试工具
2. 删除待测交换机上的所有流表
3. 持续发送 TCP 流量到待测交换机的一个入端口（例如图中的端口 1），此时没有流量被转发出去
4. 向 table 0 添加一条流表，该流表能够匹配到步骤三中发送的 TCP 流量，动作为 PUSH_VLAN, SET_FIELD 并 OUTPUT 到一个出端口（图中端口 2），记录下发此流表的时间为 T_t
5. 流表生效后会有流量从出端口被转发出去，记录第一个被转发出去的包的的时间的 T_r

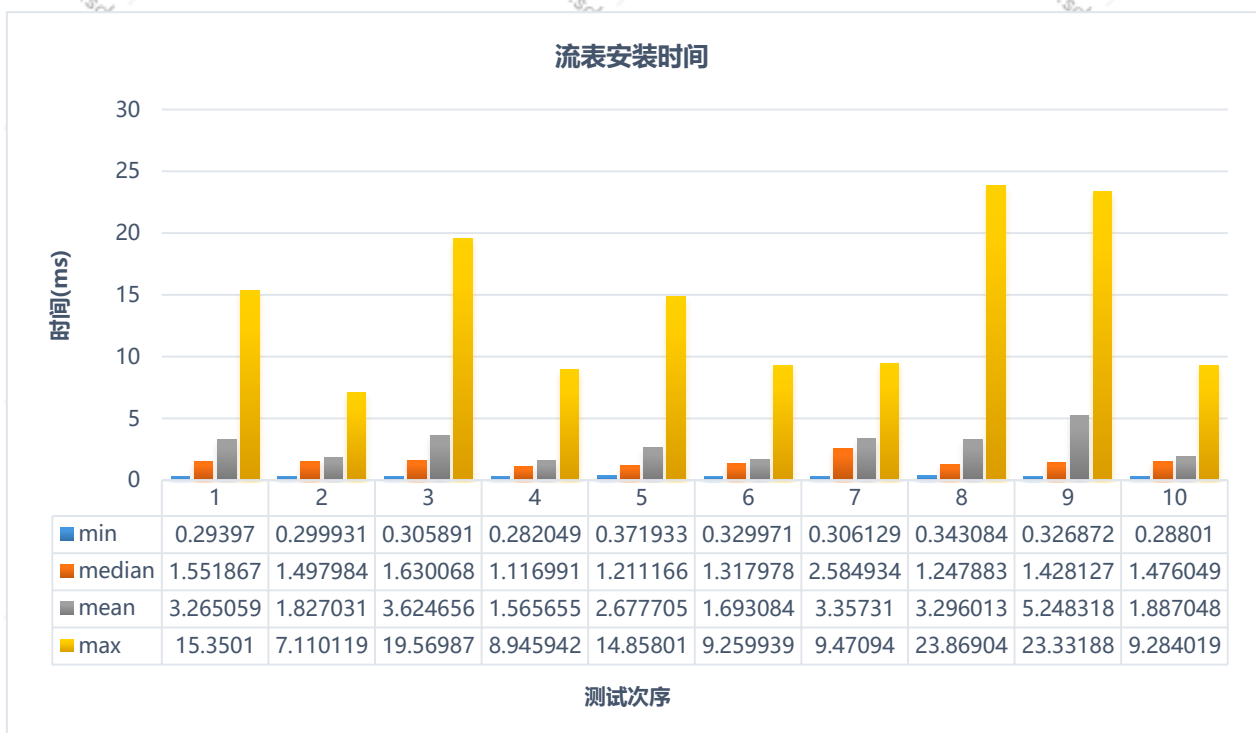
6. 测试结果为 $T_{\text{result}} = T_r - T_t$

7. 以不同的吞吐量下发多条不同的流表重复进行测试，在数据通道也发送不同的数据流以匹配所有下发的流表，测试每一条流表的安装时间

- **注意**

下发流表种类很多吞吐很大时，数据通道也要发送相应的数据流，此时数据通道发送流量的速率可能引起测试误差。

- **测试结果**



流表安装时间是从测试工具下发流表开始，直到流表生效，数据包根据流表被转发到相应端口为止的时间，其中包括了流表的查询过程。本测试例中，测试了下发 1000 条流表时，每条流表的安装时间，并计算出流表安装时间的最小值、中位值、平均值和最大值，如上图所示。与 4.2.1 的流表查询时间测试结果相对比，平均流表安装时间明显长于流表查询时间。

4.2.3 流表删除时间

- **测试目的**

测试待测交换机删除一条流表的时间

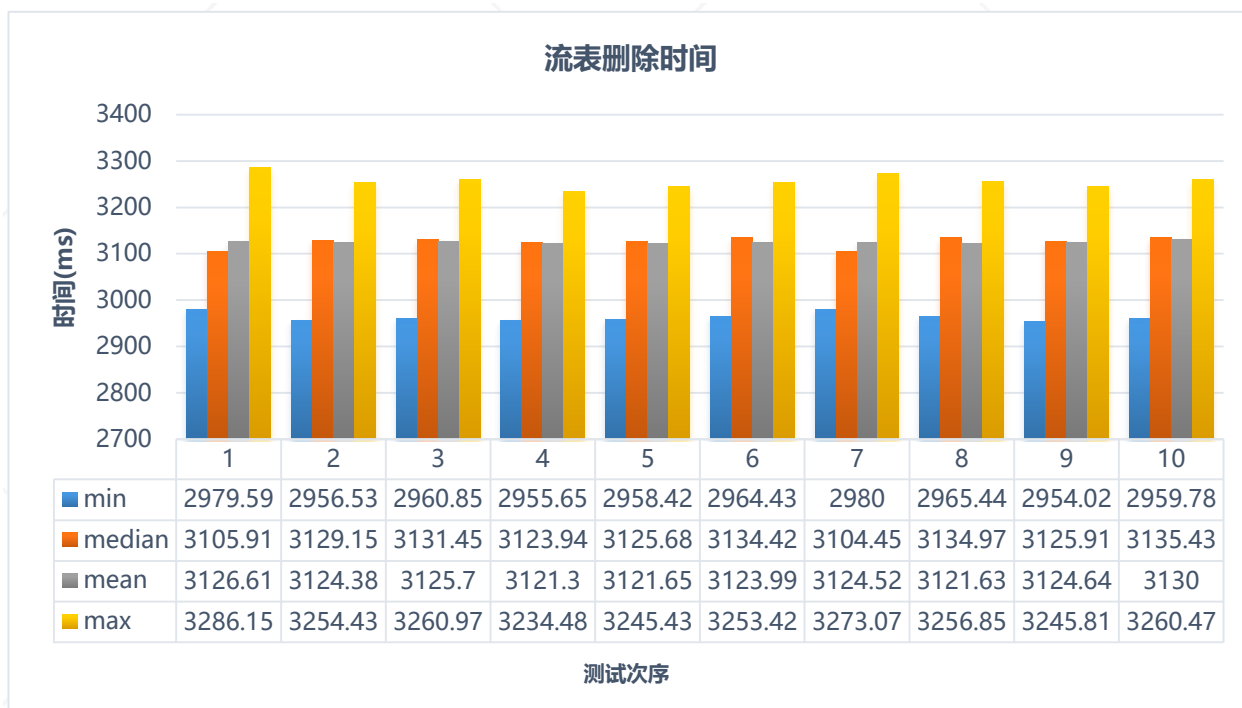
- **测试前提条件**

1. 待测交换机兼容 OpenFlow 1.3 协议
2. 待测交换机的最大流表项数量为已知

- **测试步骤**

1. 设置待测交换机连接到测试工具
2. 删除待测交换机上的所有流表
3. 向 table 0 添加一条流表，动作为 OUTPUT 到某一个端口
4. 在数据通道持续发送能够匹配到步骤 3 的流表的数据流到某一个端口
5. 等待足够长的时间（例如 3 秒）直到有数据包被转发出去
6. 发送删除流表消息，记录此时间为 T_t
7. 等待一段时间直到没有数据包再被转发出去，记录最后一个被转发的数据包的时间为 T_r
8. 测试结果为 $T_{result} = T_r - T_t$
9. 向待测交换机下发 100 条或者 1000 条或者已知的最大流表项数量的流表反复进行测试

- **测试结果**



流表删除时间测试的是从下发删除流表消息开始，到数据包不再被转发而止的时间。本测试例中，我们测试了删除 1000 条流表时每条流表的删除时间，计算了其中的最小值、中位值、平均值和最大值，测试结果如上图所示。从上图中可以看到，OVS 删除每条流表的时间为 3 秒左右。

4.2.4 流表修改时间

- **测试目的**

测试待测交换机修改一条流表的时间

- **测试前提条件**

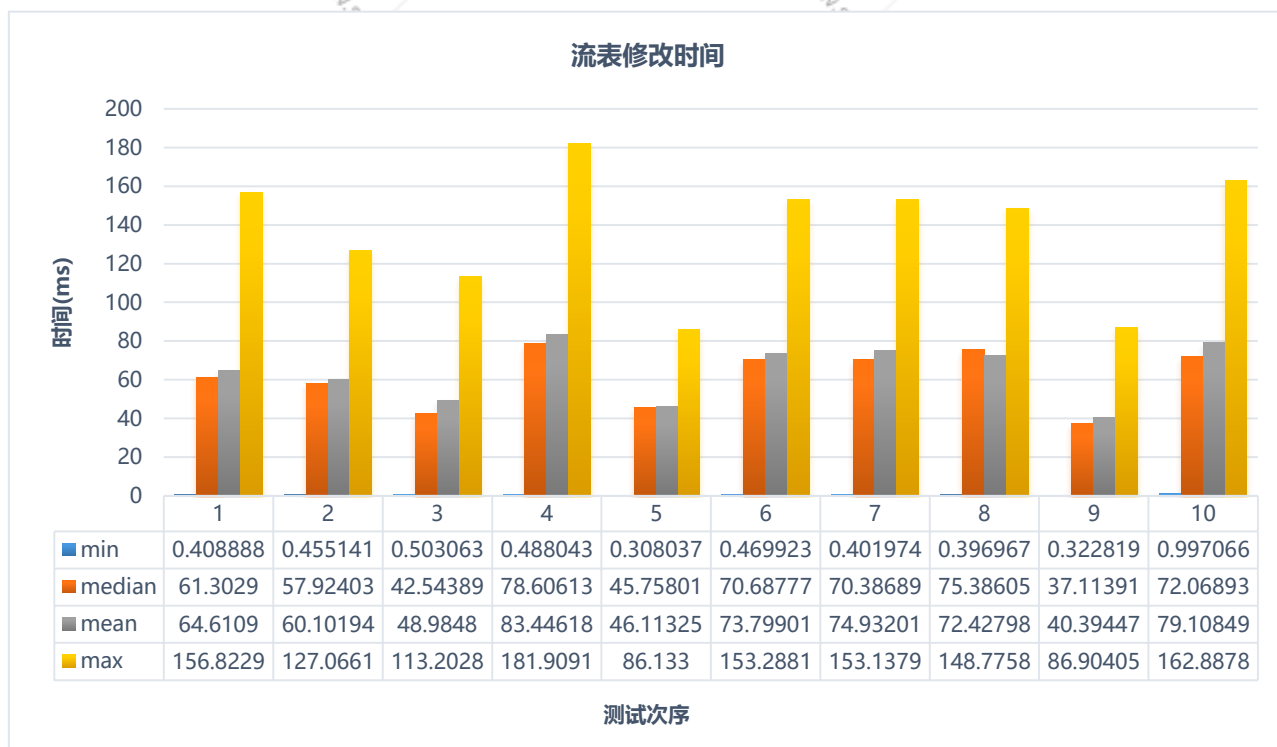
1. 待测交换机兼容 OpenFlow 1.3 协议
2. 待测交换机的最大流表项数量为已知
3. 待测交换机支持 PUSH_VLAN 和 SET_FIELD 动作

- **测试步骤**

1. 设置待测交换机连接到测试工具

2. 删除待测交换机上的所有流表
3. 添加一条流表，流表动作为 OUTPUT 到某一个端口
4. 在数据通道持续发送能够匹配到步骤 3 下发的流表的数据流量
5. 等待足够长的时间（例如 3 秒）直到数据包被转发到相应端口
6. 发送修改流表的消息，将数据包转发到另一个端口，记录下发送修改流表消息的时间为 T_t
7. 待数据包被转发到相应端口后，记录第一个包转发出去的时间为 T_r
8. 测试结果为 $T_{result} = T_r - T_t$
9. 在已安装 100 条流表或者 1000 条流表或已知的最大流表项数量下重复进行测试，得到不同情况下每条流表修改的时间

• 测试结果



同流表安装时间，流表修改时间测试的是从下发流表开始，到流表生效有数据包被转发到相应端口为止的时间，其中也是包含了流表查询时间。本测试例测试了修改 1000 条流表时，每条流表修改生效的时

间，并计算出了流表修改时间的最小值、中位值、平均值和最大值。从上图测试结果中可以看到，OVS 修改 1000 条流表的时间明显大于流表查询和流表安装时间。

4.2.5 流表容量

- **测试目的**

测试待测交换机指定流表的流表项容量

- **测试前提条件**

待测交换机兼容 OpenFlow 1.3 协议

- **测试步骤**

1. 设置待测交换机连接到测试工具
2. 删除待测交换机上的所有流表
3. 发送 OFPMP_TABLE_FEATURES 类型的 MULTIPART_REQUEST 消息以得到指定流表的最大流表项数量，记为 Fmax
4. 向待测交换机添加 Fmax 条流表，流表项具有不同的优先级和匹配项
5. 继续向待测交换机添加流表，等待待测交换机发送错误消息，错误类型为 OFPET_FLOW_MOD_FAILED，错误码为 OFPFMFC_TABLE_FULL
6. 记录收到错误消息前发送的流表项数量即为测试结果

- **测试结果**

通过 OFPMP_TABLE_FEATURES 得到 OVS 每个表最大流表项数量为 1000000 条。下发 1000000 条流表后，继续下发流表，直到下发了 2000000 条流表，OVS 依然没有返回任何错误，考虑到 OVS 使用链表存储流表项，可能在耗尽内存资源之前可以无限制的增加流表项数量，故本测试例没有得到有效测试结果。

4.3 PACKET_OUT 性能

4.3.1 PACKET_OUT 消息吞吐/时延

- **测试目的**

测试待测交换机处理 PACKET_OUT 消息的吞吐和时延

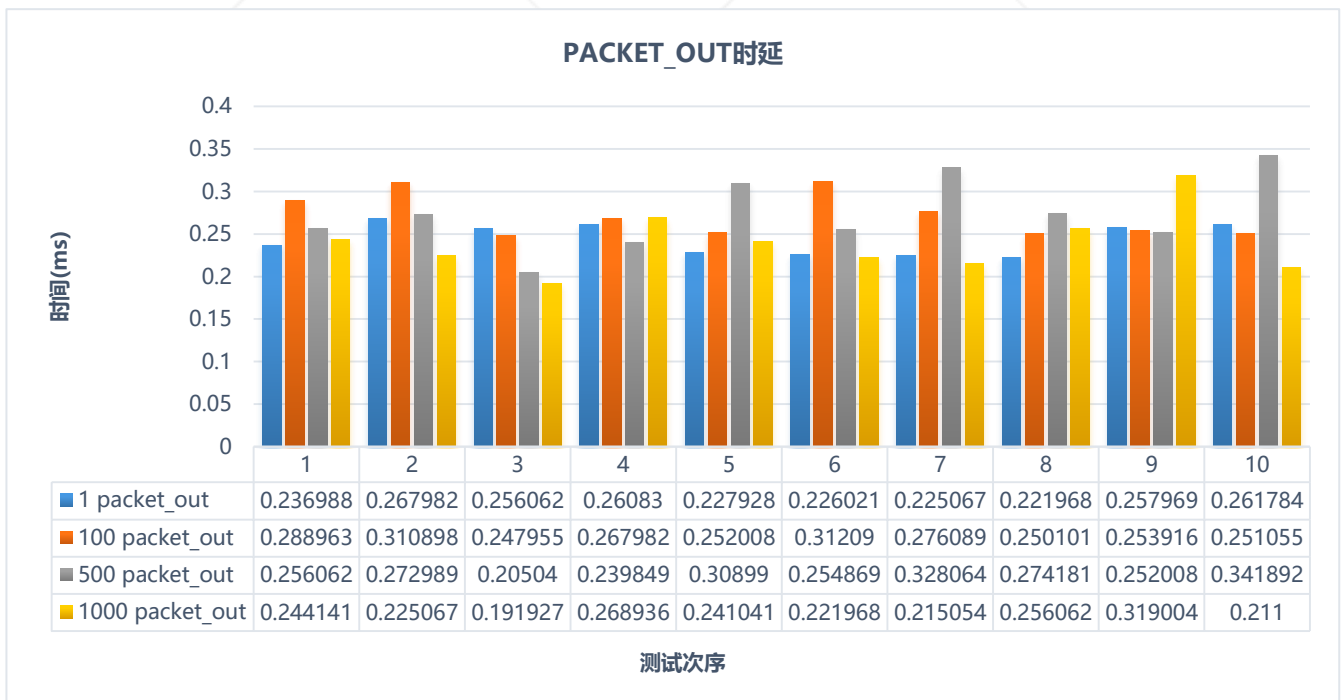
- **测试前提条件**

1. 待测交换机兼容 OpenFlow 1.3 协议
2. 设置交换机上报 PACKET_IN 消息不使用 buffer 缓存

- **测试步骤**

1. 设置待测交换机连接到测试工具
2. 发送 PACKET_OUT 消息 其中 data 字段包含一定长度的数据包 动作为 OUTPUT 到某一个端口 , 记录此时间为 T_t
3. 在相应端口收到步骤 2 中 data 字段的包后 , 记录此时间为 T_r
4. 测试结果为 $T_{result} = T_r - T_t$
5. 在 PACKET_OUT 消息的 data 字段中添加不同长度的数据包 , 并发送不同数量的 PACKET_OUT 消息反复进行测试

- **测试结果**



PACKET_OUT 性能测试的是从测试工具下发 PACKET_OUT 开始，到数据包被转发到相应端口为止的时间。从上图测试结果中可以看到，OVS 执行 1000 条 PACKET_OUT 动作的时间在 0.35 毫秒以内。此外，随着 PACKET_OUT 数量的增加，OVS 处理 PACKET_OUT 的时间并没有明显增加。

4.4 交换机统计性能测试

4.4.1 流统计性能

- **测试目的**

测试待测交换机更新流统计的时间

- **测试前提条件**

待测交换机兼容 OpenFlow 1.3 协议

- **测试步骤**

1. 设置待测交换机连接到测试工具
2. 删除待测交换机上的所有流表

3. 向 table0 添加一条流表，动作为转发到某一个端口

4. 持续发送 OFPMP_FLOW 类型的 OFPT_MULTIPART_REQUEST 消息，请求步骤 3 下发的流表的统计信息

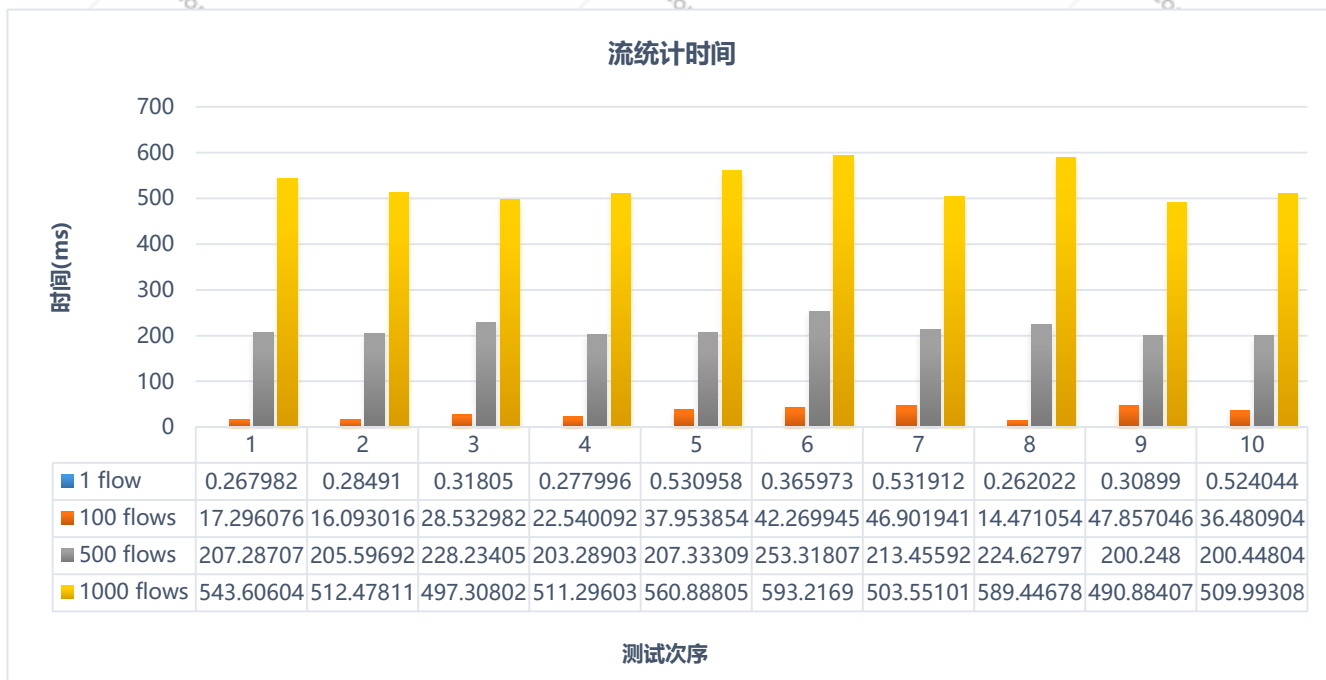
5. 在数据通道发送一个能够匹配步骤 3 流表的数据包，记录此时间为 T_t

6. 根据 OFPT_MULTIPART_REPLY 消息判断流统计是否更新，记录更新的时间为 T_r

7. 测试结果为 $T_{result} = T_r - T_t$

8. 反复测试不同流的更新时间

• 测试结果



从上图测试结果可以看到，OVS 更新流统计信息的时间随着流数量的增加而明显增长。

4.4.2 端口统计性能

• 测试目的

测试端口统计信息更新时间

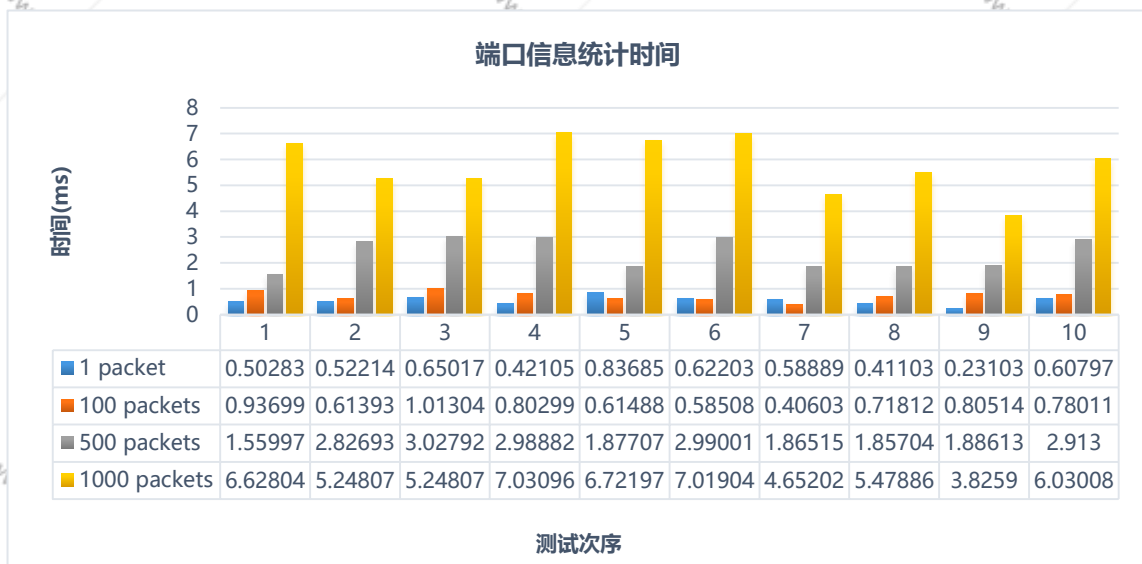
- **测试前提条件**

待测交换机兼容 OpenFlow 1.3 协议

- **测试步骤**

1. 设置待测交换机连接到测试工具
2. 删除待测交换机上的所有流表
3. 发送 OFPMP_PORT_STATS 类型的 OFPT_MULTIPART_REQUEST 消息，请求当前端口的统计信息，记录下 rx_packets 和 rx_bytes
4. 持续发送步骤 3 的 OFPT_MULTIPART_REQUEST 请求
5. 在数据通道发送一个数据包到相应端口，记录此时间为 T_t
6. 根据 OFPT_MULTIPART_REPLY 消息判断端口统计信息更新情况，记录更新的时间为 T_r
7. 测试结果为 $T_{result} = T_r - T_t$
8. 重复测试发送不同类型和不同数量的数据包时端口统计信息的更新时间

- **测试结果**



从上图测试结果中可以看到，随着数据通道发送的数据包数量的增加，OVS 更新端口统计信息的时间也在增长。

4.5 交换机定时性能测试

4.5.1 HARD 定时误差

- **测试目的**

测试流表 HARD 定时的准确性

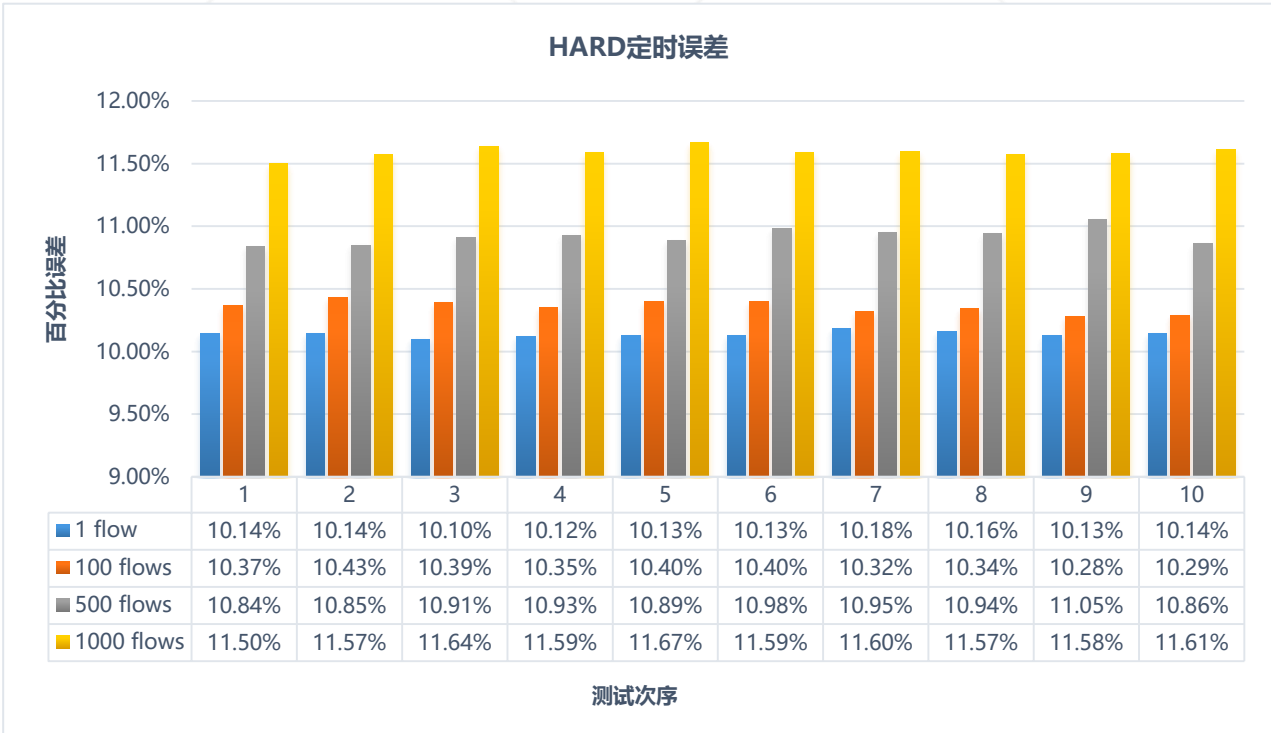
- **测试前提条件**

待测交换机兼容 OpenFlow 1.3 协议

- **测试步骤**

1. 设置待测交换机连接到测试工具
2. 删除待测交换机上的所有流表
3. 向待测交换机添加一条流表，其中 hard 定时时间为 5 秒，动作为 OUTPUT 到一个端口，记录此时时间为 T_s
4. 发送匹配步骤 3 流表的数据流到待测交换机的一个端口，此时该流被转发到相应的出端口，记录转发出去的最后一个数据包的时间为 T_e
5. 测试结果为 $T_{result} = T_e - T_s$
6. 计算百分比误差为 $E = (T_{result} - 5)/5 * 100\%$
7. 反复测试安装有不同数量的流表时，HARD 定时的准确性

- **测试结果**



本测试例中，HARD 定时测试的是从下发流表开始，到流表超时失效，数据包不再被转发为止的时间，并由此计算出定时误差。从上图测试结果中可以看到，随着流表数量的增加，其中一条流表定时误差也在增大。

4.5.2 IDLE 定时误差

- 测试目的

测试流表 IDLE 定时的准确性

- 测试前提条件

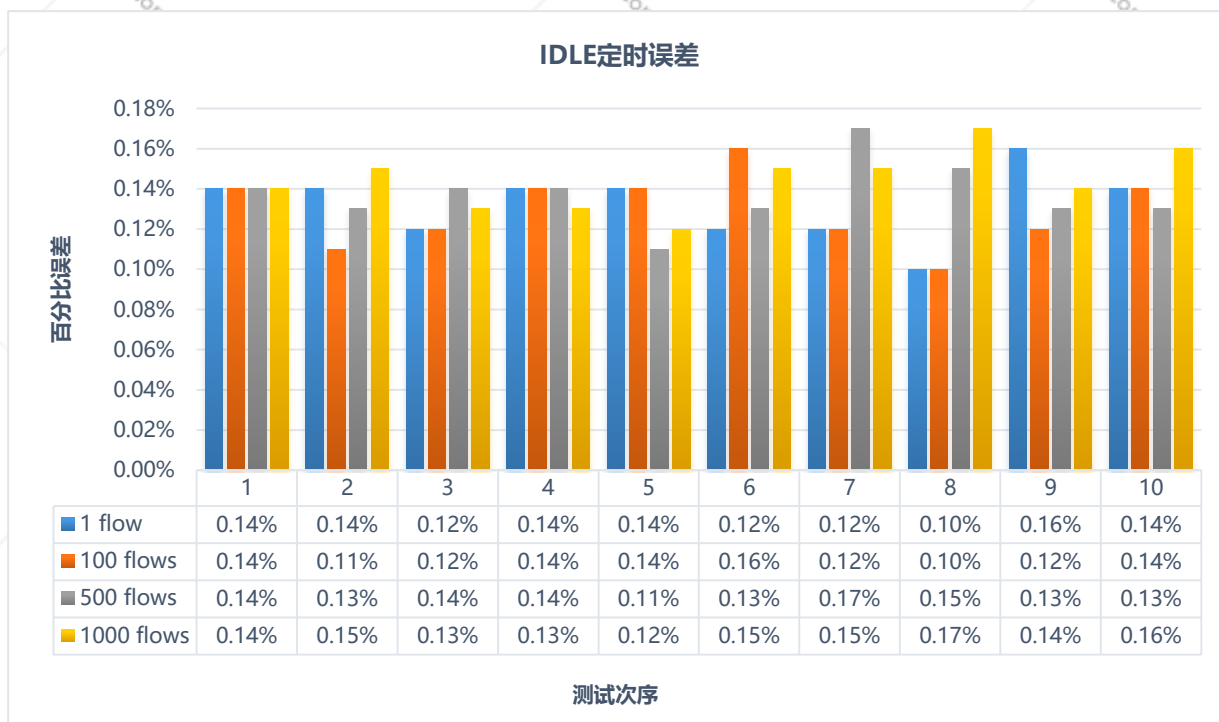
待测交换机兼容 OpenFlow 1.3 协议

- 测试步骤

1. 设置待测交换机连接到测试工具
2. 删除待测交换机上的所有流表

3. 向待测交换机添加一条流表，其中 IDLE 定时时间为 5 秒，并设置 OFFPF_SEND_FLOW_REM 标志
4. 在数据通道发送一个匹配步骤 3 流表的数据包到交换机的一个端口，记录此时间为 T_s
5. 等待待测交换机发送 OFF_FLOW_REMOVED 消息，记录收到此消息的时间为 T_e
6. 测试结果为 $T_{result} = T_e - T_s$
7. 计算 IDLE 定时的百分比误差为 $E = (T_{result} - 5)/5 * 100\%$
8. 反复测试安装不同数量的流表时，IDLE 定时的准确性

测试结果



本测试例中，IDLE 定时测试的是从下发流表开始，到待测交换机上报 OFF_FLOW_REMOVED 消息为止的时间。从上图测试结果中可以看到，IDLE 定时误差并不会随着流表数量的增加而增大。

注：对比 4.5.1，IDLE 定时误差与 HARD 定时误差测试结果相差较大，我们认为这是由于两个测试例使用的测试方法在原理上存在本质区别而导致的，直接对比这两个测试结果是没有意义的。