

RYU 控制器性能测试报告

全球 SDN 测试认证中心 SDNCTC

2016 . 3 . 8

一、引言

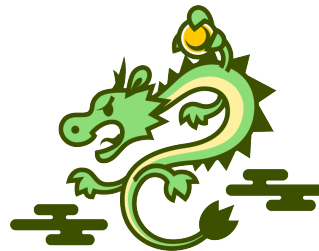
当软件定义网络 (Software Defined Network, SDN) 逐渐成为网络世界新的范式, 转发与控制的分离使得数据平面只作为单纯的数据收发引擎, 而控制平面则承担了全部的逻辑与运算任务。作为控制平面的核心组件, SDN 控制器的性能关乎整个 SDN 网络的性能表现。随着 SDN 商业部署速度地加快, SDN 控制器性能也必将越来越多地成为网络用户关心的焦点。

OFsuite_performance 是全球 SDN 测试认证中心 (SDNCTC) 独立开发的 OFsuite 系列测试工具之一, 此测试工具致力于 OpenFlow 控制器性能测试。能够在通用 Linux 服务器上模拟大量 OpenFlow 1.3 交换机, 并且能够模拟不同的网络拓扑以及全部的 OpenFlow 事件。该测试工具能够在真实的 SDN 网络环境中运行, 从而有效地衡量控制器对 OpenFlow 消息的处理能力。其测试结果能够在网络用户进行 SDN 网络性能评估, 测试及商业部署时提供可靠的数据支撑。除此之外, 还可以提供多控制器连接, TLS 加密通道连接, 测试结果可视化等附加功能。该测试工具简洁、高效、易于使用, 并将持续更新以便为用户提供更丰富的性能测试案例及测试场景。

本报告以开源控制器 RYU 作为被测控制器, 使用 OFsuite_performance 执行测试, 汇总结果得出性能测试报告。全部测试例均为 OFsuite_performance 自动化测试完成, 报告中所展示的结果图表均为测试工具自动生成。

二、测试环境配置

2.1 待测控制器



待测控制器为目下流行的开源控制器 RYU, 版本为 v3.28, 该版本的 RYU 控制器完全支持 OpenFlow v1.3 南向协议。

2.2 服务器配置

待测控制器 RYU 运行于一台单独的服务器上, 其配置如下:

- 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
- 网卡: 1Gbps

2.3 测试工具



测试工具为 OFsuite_performance，执行测试时，OFsuite_performance 运行于一台与 RYU 控制器所在服务器直连的服务器上，服务器配置如下：

- 服务器型号：Dell PowerEdge R720
- 处理器：Intel(R) Xeon(R) E5-2609 v2 @ 2.50GHz 4 核
- 内存：8GB 1600MHz
- 操作系统：Ubuntu server 14.04.1 LTS 64 位
- 网卡：1Gbps

三、测试项目及测试结果

3.1 控制信道容量测试

- **测试目的**

测试控制器 OpenFlow 1.3 控制通道能够维持的最大连接数。

- **测试方法¹**

本测试中，测试工具模拟一定数量的 OpenFlow 1.3 交换机并连接到待测控制器，测试待测控制器所能维持的最大交换机数量，具体测试步骤如下：

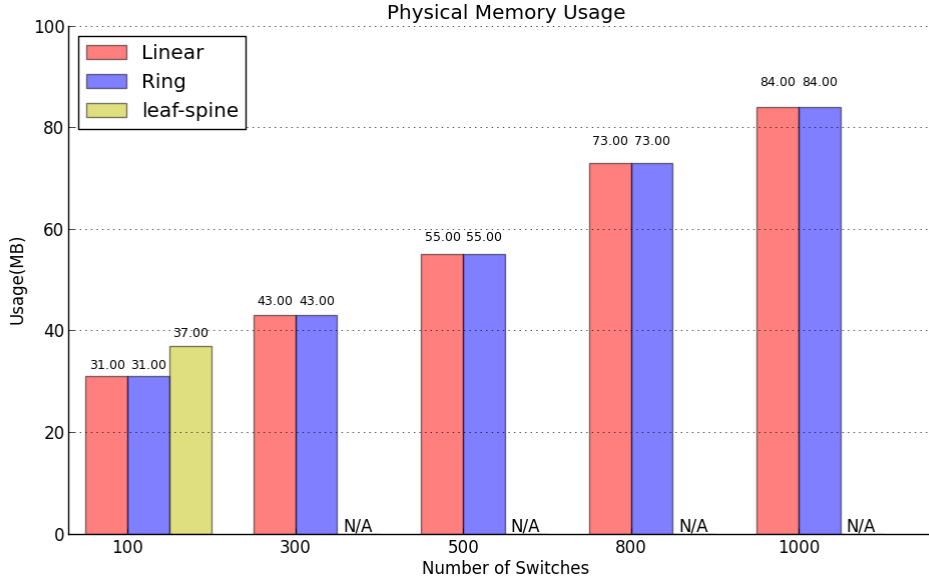
1. 开启待测控制器并同时开启 simple_switch_13 应用；
2. 开启测试工具 OFsuite_performance，使用 “—switches=[N]” 选项设定一定数量的 OpenFlow 1.3 交换机连接到待测控制器；
3. 使用 set-topo 命令设置一种拓扑结构（支持预先定义的 linear/ring/full-mesh/leaf-spine 结构，也可自定义拓扑结构），待连接稳定后记录控制器的内存占用情况；
4. 使用 add-sw 命令增加交换机数量并待连接稳定后记录控制器的内存占用情况；
5. 以不同的交换机数量和拓扑结构迭代执行测试，得到最终结果。

- **测试工具要求**

1. 测试工具可以模拟大量 OpenFlow 1.3 交换机，并与控制器建立连接；
2. 测试工具可以响应全部相关 OpenFlow 1.3 协议消息（Hello，Echo，feature_request, etc）；
3. 测试工具可以随时监测控制通道活性；
4. 测试工具可以灵活调整连接交换机数量；
5. 测试工具具有详尽的 Log 功能。

- **测试结果**

¹ 全部测试例的详细测试拓扑及测试方法请参考全球 SDN 测试认证中心发布的《SDN 控制器性能测试白皮书》



在上述测试结果中可以看到，在相同网络拓扑结构下，随着交换机数量的增加，控制器处理交换机各种请求时所占用的系统内存资源也随之增加；另外，在相同交换机数量下，控制器处理不同复杂程度的网络拓扑结构所占用的系统内存资源也不同，拓扑结构越复杂，内存占用越高。

OFsuite_performance 最多可支持模拟 10K+数量的 OpenFlow 1.3 交换机与控制器建立连接。在本例中，当交换机数量超过 1K 时，被测控制器即出现通道连接断开，Feature Request 消息不发送等问题，故不作为有效结果计入统计。

3.2 拓扑发现时间测试

- **测试目的**

测试控制器对不同交换机数量、不同类型的拓扑结构发现时间。

- **测试方法**

本测试中，测试工具将模拟一定数量的交换机，并且交换机之间互相连接形成一定的网络拓扑结构，测试待测控制器完全发现此拓扑结构所用的时间，具体测试步骤如下：

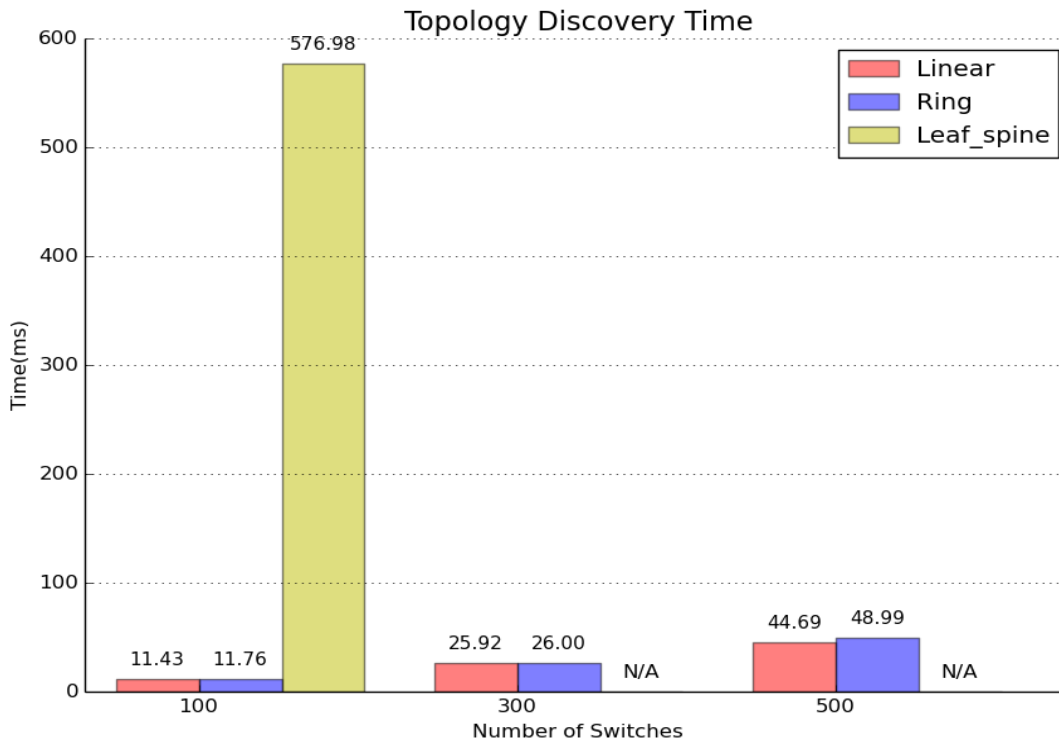
1. 启动待测控制器；
2. 启动测试工具，启动时只设置交换机数量而不设置拓扑结构；
3. 待交换机与控制器的连接稳定后使用 set-topo 命令设定拓扑结构；
4. 测试工具记录控制器下发的第一个 LLDP 消息的时间，并依照流表内容响应该消息；
5. 测试工具记录最后一条 LLDP 消息上送控制器的时间；
6. 使用 show-result 命令查看测试结果；
7. 改变交换机数量并设置相同的拓扑结构进行迭代测试；
8. 不改变交换机数量但改变拓扑结构进行迭代测试；
9. 反复进行测试以得到平均测试结果。

- **测试工具要求**

1. 测试工具可以模拟大量 OpenFlow 1.3 交换机，并与控制器建立连接；

2. 测试工具可以响应全部相关 OpenFlow 1.3 协议消息 (Hello , Echo , feature_request etc) ;
3. 测试工具可以随时监测控制通道活性 ;
4. 测试工具可以创建不同的网络拓扑结构 ;
5. 测试工具可以解析 Packet_out 消息 , 以便提取其中的 LLDP 消息 ;
6. 测试工具可以解析 LLDP 拓扑发现消息 , 以便给出对应的响应 ;
7. 测试工具有详尽的 Log 功能。

• **测试结果**



从上图测试结果中可以看到，相同网络拓扑结构下，交换机数量越多，待测控制器完成拓扑发现所用的时间越长；而相同的交换机数量下，网络拓扑结构复杂程度不同，控制器完成拓扑发现所用的时间也不同，网络拓扑结构越复杂，拓扑发现所用的时间越高。在图中显示的结果看来，Linear 和 Ring 拓扑所需时间相似，而 Leaf-spine 拓扑所需时间则显著增加。同时当交换机数量大于 300 时，RYU 控制器不能有效完成 Leaf-spine 拓扑的发现过程。

3.3 PACKET_OUT 下发速率测试

• **测试目的**

测试待测控制器下发 Packet_out 消息的时延和最大速率。

• **测试方法**

本测试中，测试工具将模拟一个交换机连接到控制器，测试工具将以一定的速率恒速上发 Packet_in 消息，记录上发 Packet_in 消息的时间，Packet_in 消息中包含了 ARP_request 请求，控制器收到此 Packet_in 后会下发 Packet_out 消息，测试工具收到待测控制器下发的 Packet_out 消息之后记录

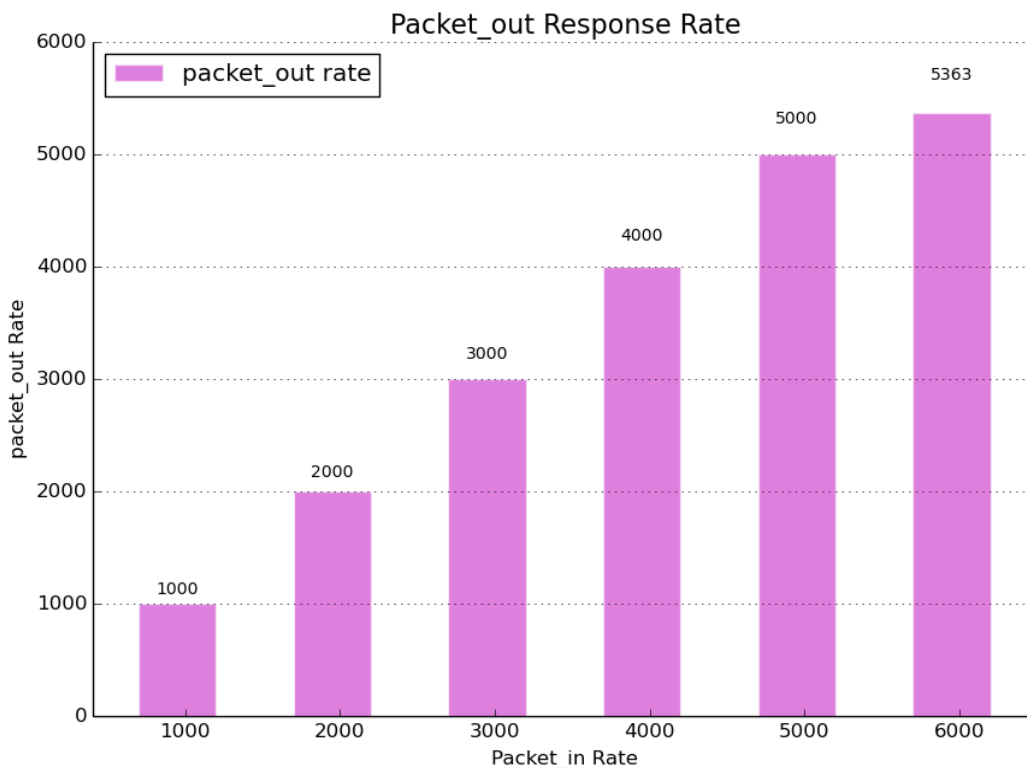
Packet_out 的下发时间，通过所有的 Packet_in 和 Packet_out 记录计算待测控制器对 Packet_in 消息的响应速度(latency)以及 Packet_out 的下发速率(throughput)。Ofsuite_performance 能够在低端服务器上达到每秒 500K 级的恒速 Packet_in 上发速率。具体测试步骤如下：

1. 启动待测控制器；
2. 启动测试工具，设置一个交换机，不设置拓扑结构，等待交换机连接到控制器；
3. 在测试工具的命令行接口使用 set-arp-rate 命令设置交换机上发 Packet_in 的速率；
4. 等待测试工具完成测试；
5. 使用 show-result 命令查看测试结果；
6. 改变 Packet_in 上发的速率再次进行测试；
7. 反复进行迭代测试以得到平均测试结果。

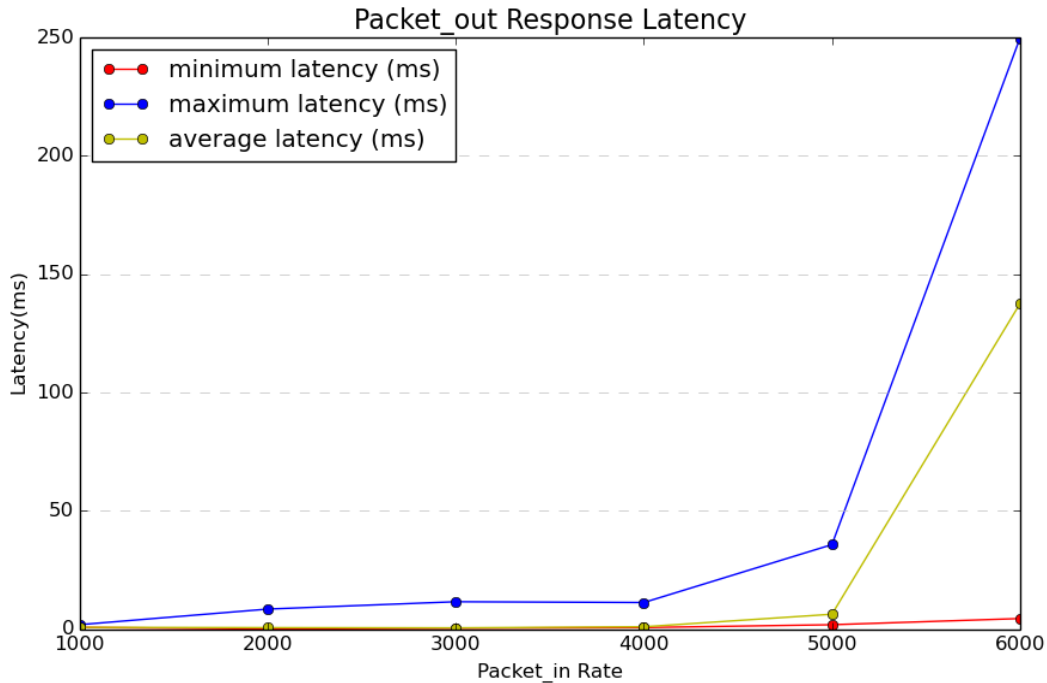
• **测试工具要求**

1. 测试工具可以模拟 OpenFlow 1.3 交换机，并与控制器建立连接；
2. 测试工具可以响应全部相关 OpenFlow1.3 协议消息（Hello，Echo，feature_request etc）；
3. 测试工具可以上发包含 ARP_request 的 Packet_in 消息从而触发待测控制器下发 Packet_out 消息；
4. 测试工具可以自定义 Packet_in 消息 Data field 的内容；
5. 测试工具可以恒速上发 Packet_in 消息；
6. 测试工具可以解析 Packet_out 消息，以便确认该消息为对应 Packet_in 消息所触发；
7. 测试工具有详尽的 Log 功能。

• **测试结果**



在上图的测试结果中可以看到，当测试工具设置的 Packet_in 上发速率低于 5000 packets/s 时，待测控制器能够很好地处理上发的 Packet_in 消息，Packet_out 下发的速度能够与 Packet_in 的速度保持一致，但是当 Packet_in 的速率大于 5000 packets/s，例如 6000 packets/s 时，待测控制器已无法完全处理收到的 Packet_in，下发 Packet_out 的速率也低于 Packet_in 的速率，由此看来，测试已达到被测控制器的最大 Packet_out 下发速率。



上图所示为待测控制器发送 Packet_out 消息的时延，此时间为测试工具上发 Packet_in 消息与控制器回发对应 Packet_out 消息之间的时延，从图上可以看到，随着 Packet_in 消息上发速度得增加，Packet_out 消息的时延也越来越大。详细的测试结果数据如下表所示。

Packet_in Rate	Minimum latency (ms)	Maximum latency (ms)	Average latency (ms)
1000	0.70996	1.85693	0.73175
2000	0.01807	8.448	0.60239
3000	0.35498	11.55786	0.51537
4000	0.63306	11.22705	0.9621
5000	1.83276	35.69214	6.26612
6000	4.44605	249.78613	137.5306

3.4 FLOW MOD 下发速率测试

- 测试目的

测试待测控制器下发流表的时延和最大速率。

- **测试方法**

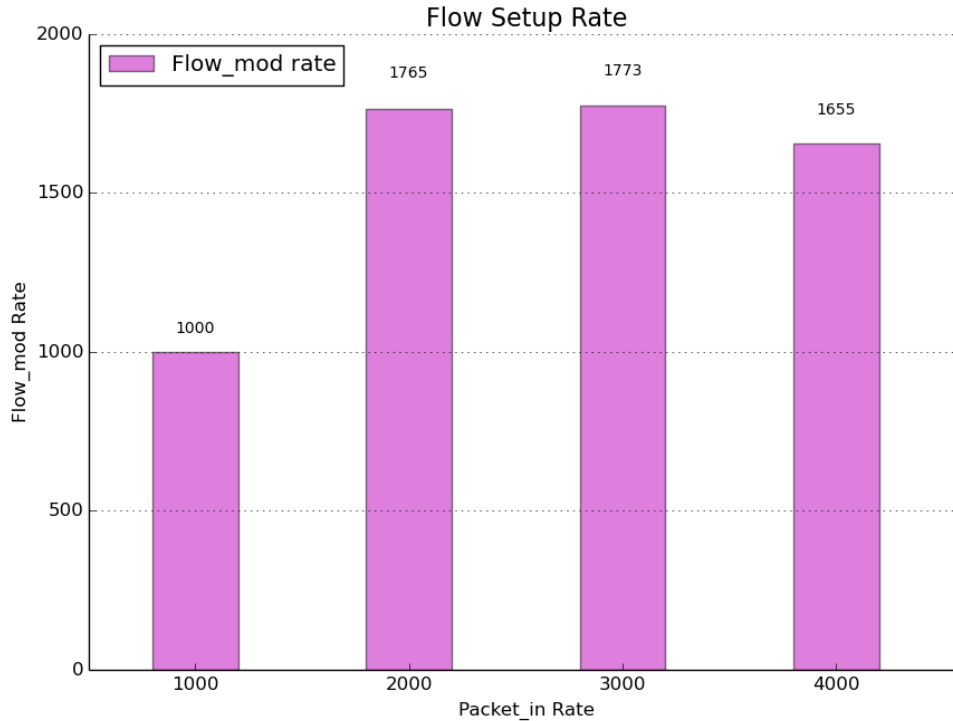
本测试中，测试工具将模拟一个交换机连接到控制器，测试工具将以一定的速率上发包含 ARP_request 的 Packet_in 消息，测试工具在收到待测控制器下发的 Packet_out 之后再上发包含对应 ARP_reply 的 Packet_in 消息，此时待测控制器会下发 flow mod 消息，然后根据 Packet_in 消息和 flow mod 消息的记录，计算待测控制器下发流表的时延和速率，具体测试步骤如下：

1. 启动待测控制器
2. 启动测试工具，设置一个交换机，不设置拓扑结构，等待交换机连接到控制器；
3. 在测试工具上使用 set-arp-rate 命令设置交换机上发 packet in 消息的速率；
4. 等待测试工具完成测试；
5. 使用 show-result 命令查看测试结果；
6. 改变 Packet_in 上发的速率再次进行测试；
7. 反复进行测试以得到平均测试结果。

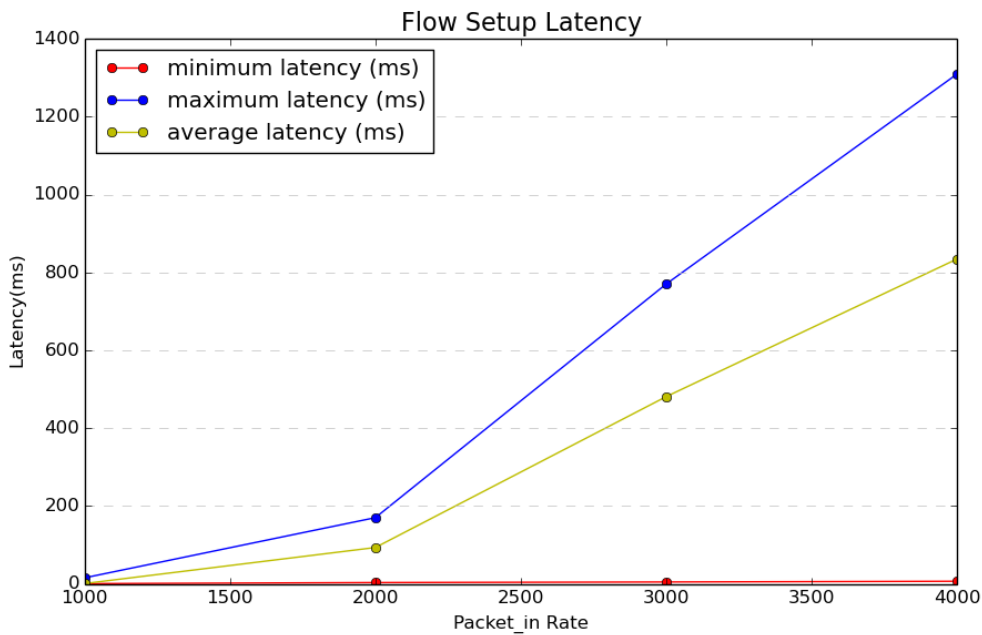
- **测试工具要求**

1. 测试工具可以模拟 OpenFlow 1.3 交换机，并与控制器建立连接；
2. 测试工具可以响应全部相关 OpenFlow 1.3 协议消息（Hello，Echo，feature_request etc）；
3. 测试工具可以上发包含 ARP_request 的 Packet_in 消息从而触发待测控制器下发 Packet_out；
4. 测试工具可以上发包含 ARP_reply 的 Packet_in 消息从而触发待测控制器下发 flow mod 添加流表；
5. 测试工具可以解析 flow mod 消息，确保该消息为测试工具所触发；
6. 测试工具可以自定义恒速上发 Packet_in 消息；
7. 测试工具具有详尽的 Log 功能。

- **测试结果**



在上图的测试结果中可以看到，当测试工具设置的 Packet_in 上发速度较低时，待测控制器能够很好地处理 Packet_in 消息，下发的 flow mod 速率能够与 Packet_in 的速率保持一致，但是当 Packet_in 的速度大于 2000 packets/s 时，待测控制器已无法完全处理收到的 Packet_in，下发 flow mod 的速率也低于 Packet_in 的速率，由此看来，此速率及为在该硬件配置下被测控制器的最大流表安装速率。



上图所示为待测控制器下发流表的时延，从图上可以看出，测试工具发送 Packet_in 消息速率较低

时，待测控制器下发流表的时延相对较低，当测试工具上发 Packet_in 消息速率变高之后，待测控制器下发流表的时延相应增高。详细测试结果列于下表中。

Packet_in Rate	Minimum latency (ms)	Maximum latency (ms)	Average latency (ms)
1000	0.58521	15.56201	0.85142
2000	3.3999	170.104	93.37974
3000	4.82495	769.93799	480.76921
4000	6.71118	1309.08301	834.09681

3.5 端到端链路建立时间测试

- **测试目的**

测试待测控制器在不同的网络拓扑下建立端到端路径的时间。

- **测试方法**

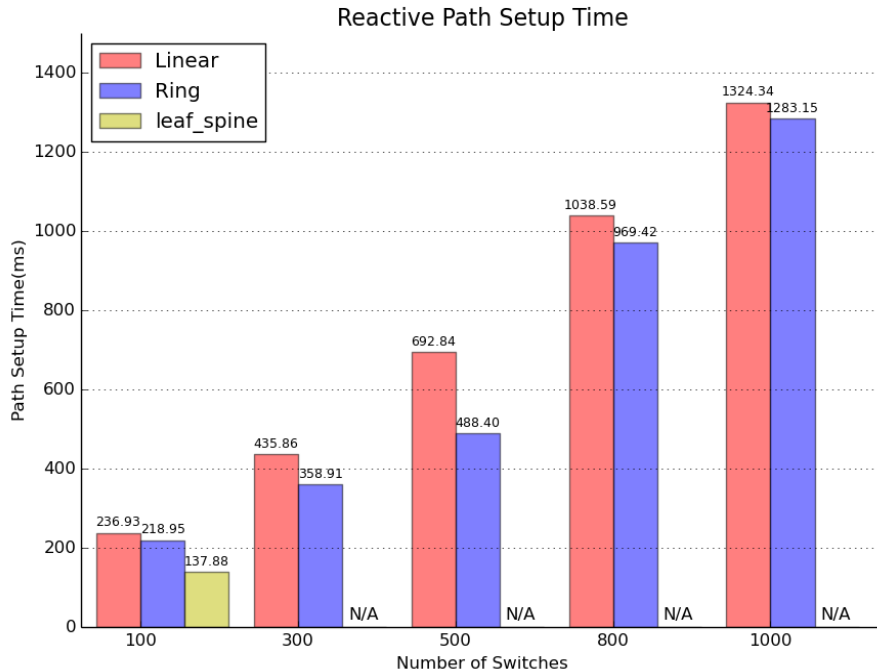
本测试中，测试工具将模拟一定数量的交换机，交换机之间以特定的拓扑结构连接到控制器，此外，测试工具还将模拟两个主机，分别连接到两个交换机上，在改变拓扑结构和交换机数量时，两个主机连接到的交换机不变，通过 ARP_request、reply 以及 PING_request、reply 过程测试待测控制器能否建立两个主机互相连通的端到端路径，以及具体的链路建立时间。具体测试步骤如下：

1. 启动待测控制器；
2. 启动测试工具，设置好交换机数量；
3. 待交换机与控制器的连接稳定之后设置好网络拓扑结构；
4. 等待控制器完成拓扑发现过程；
5. 使用 set-host 命令设置主机连接到一个特定的交换机；
6. 等待测试工具完成测试；
7. 使用 show-result 命令查看测试结果；
8. 改变交换机数量或者拓扑结构再次进行测试；
9. 反复进行迭代测试以得到平均测试结果。

- **测试工具要求**

1. 测试工具可以模拟 OpenFlow 1.3 交换机，并与控制器建立连接；
2. 测试工具可以响应全部相关 OpenFlow 1.3 协议消息（Hello，Echo，feature_request etc）；
3. 测试工具可以上发包含 ARP_request 的 Packet_in 消息从而触发待测控制器下发 Packet_out；
4. 测试工具可以上发包含 ARP_reply 的 Packet_in 消息从而触发待测控制器下发 flow mod 添加流表；
5. 测试工具可以解析 flow mod 消息，以确认链路是否建立成功；
6. 测试工具可以模拟主机连接到交换机上；
7. 测试工具可以模拟主机间的通信。

- **测试结果**



在上图的测试结果中可以看到，在相同的网络拓扑结构下，交换机数量越多，待测控制器建立主机之间的通信路径的时间越长；而相同交换机数量下，网络结构不同，建立主机之间路径的时间也不同，建立此路径的时间与主机之间的跳数有关，当 100 个交换机时，leaf-spine 网络拓扑的主机间跳数要更少，所以建立路径的时间也较短。但当交换机数量大于 300 个时，控制器对 leaf-spine 网络拓扑的发现出现问题（如测试例 3.1，测试例 3.2 所示），故无法建立端到端的连接。

四、全球 SDN 测试认证中心



全球 SDN 测试认证中心是第三方中立的 SDNFV 测试认证实验室，致力于 SDN/NFV 技术研发，测试认证和部署推广。全球 SDN 测试认证中心是 ONF 成员，同时也是 ONF 授权全球除美国外第一家 SDN 测试认证中心。全球 SDN 测试认证中心积极参与 ONF 测试领导委员会和互操作性测试工作组关于测试规范制订，测试认证体系建设，测试工具研发等工作，推动 SDN 技术标准和商业部署。作为独立的第三方中立的评测机构，全球 SND 测试认证中心为各种网络/终端设备、云计算数据中心、运营商、企业等提供第三方测试服务。根据客户实际的需求，制定详细的测试方案和验证计划，并进行严格独立的测试，最终交付详细和权威的测试报告。