Ryuo: 可伸展、低控制延迟的两层 SDN 控制器

Ryuo: A Scalable 2-layer SDN Controller with Low Control Latency

张绍宇
1130339066

导师：沈耀
计算机系
电子信息与电气工程学院
上海交通大学

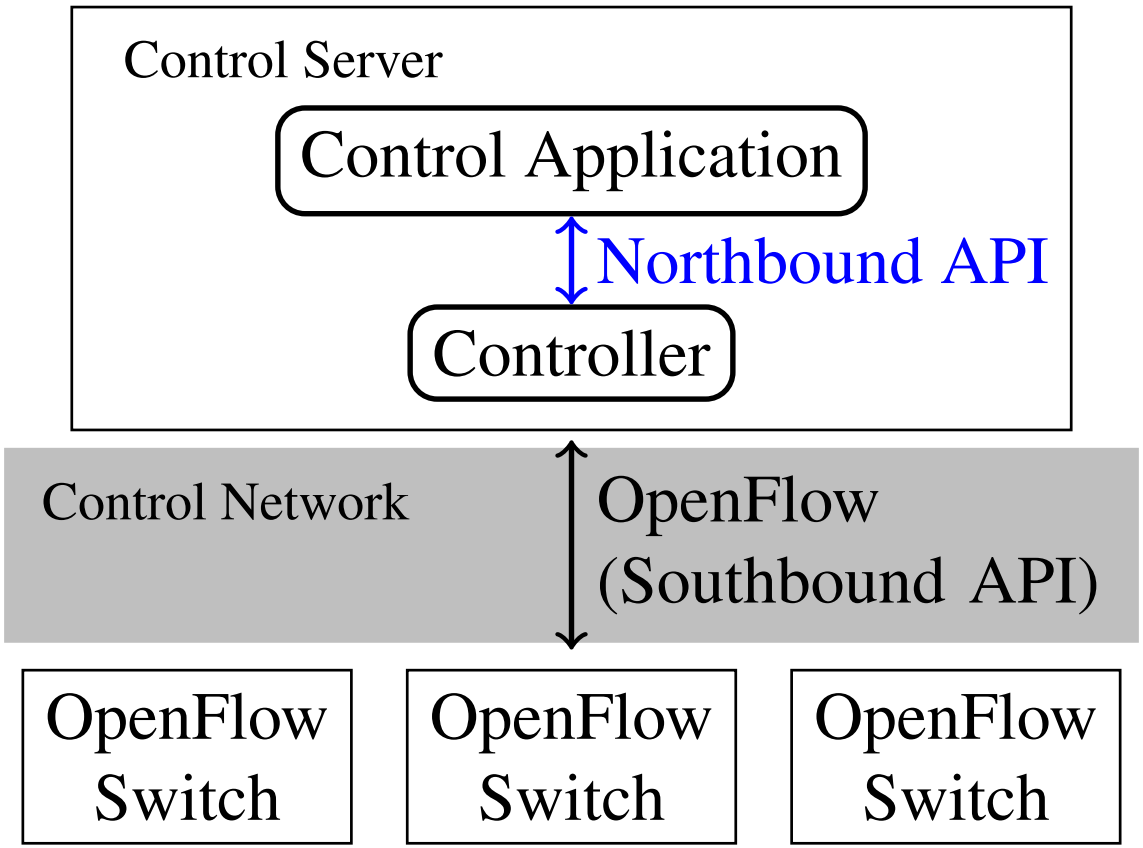
2016.1.18

内容提要

软件定义网络

- 控制平面与数据平面的分离
- 网络设备：运行专有软件 $\rightarrow$ 实现统一开放的接口
- 一般采用逻辑上集中式的控制平面
- 更好的可编程性

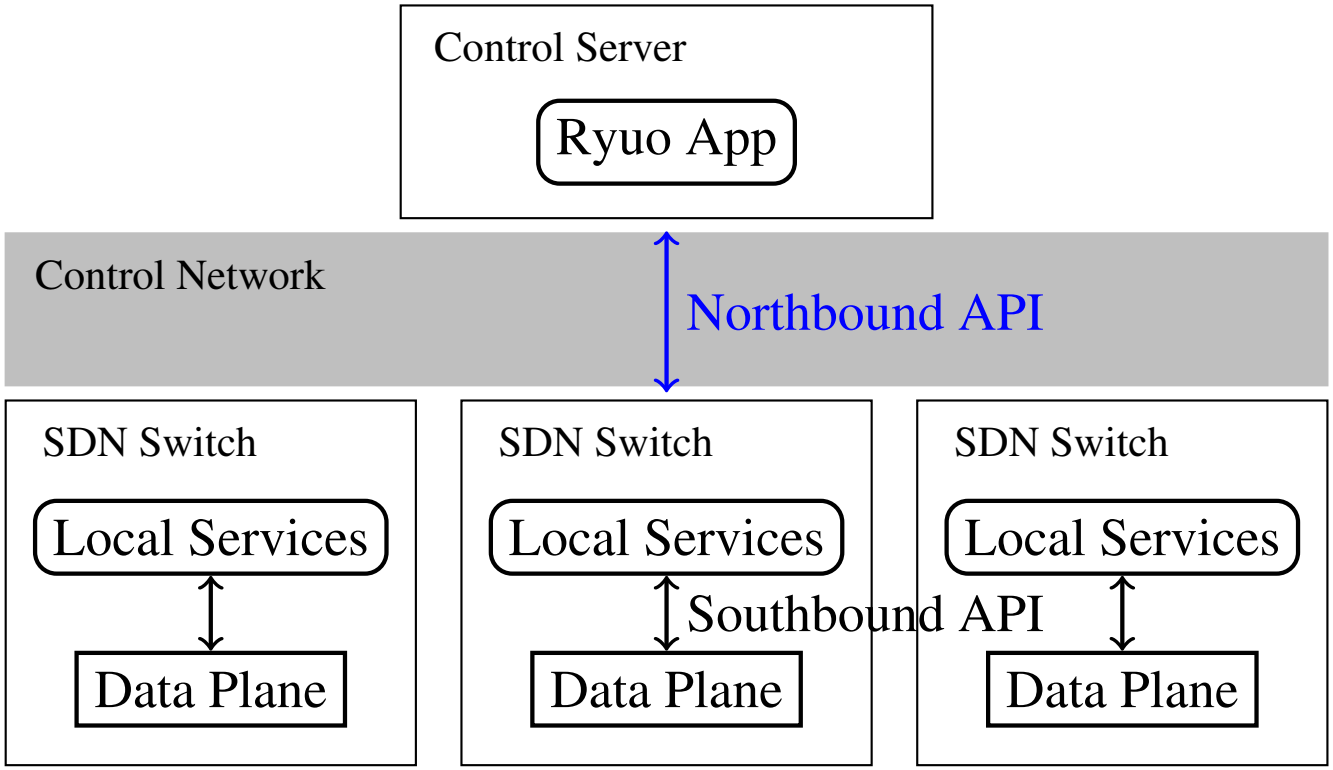
OpenFlow


Control Server
Control Application
Northbound API
Controller
Control Network
OpenFlow
(Southbound API)
OpenFlow
Switch
OpenFlow
Switch
OpenFlow
Switch

OpenFlow 所存在的问题

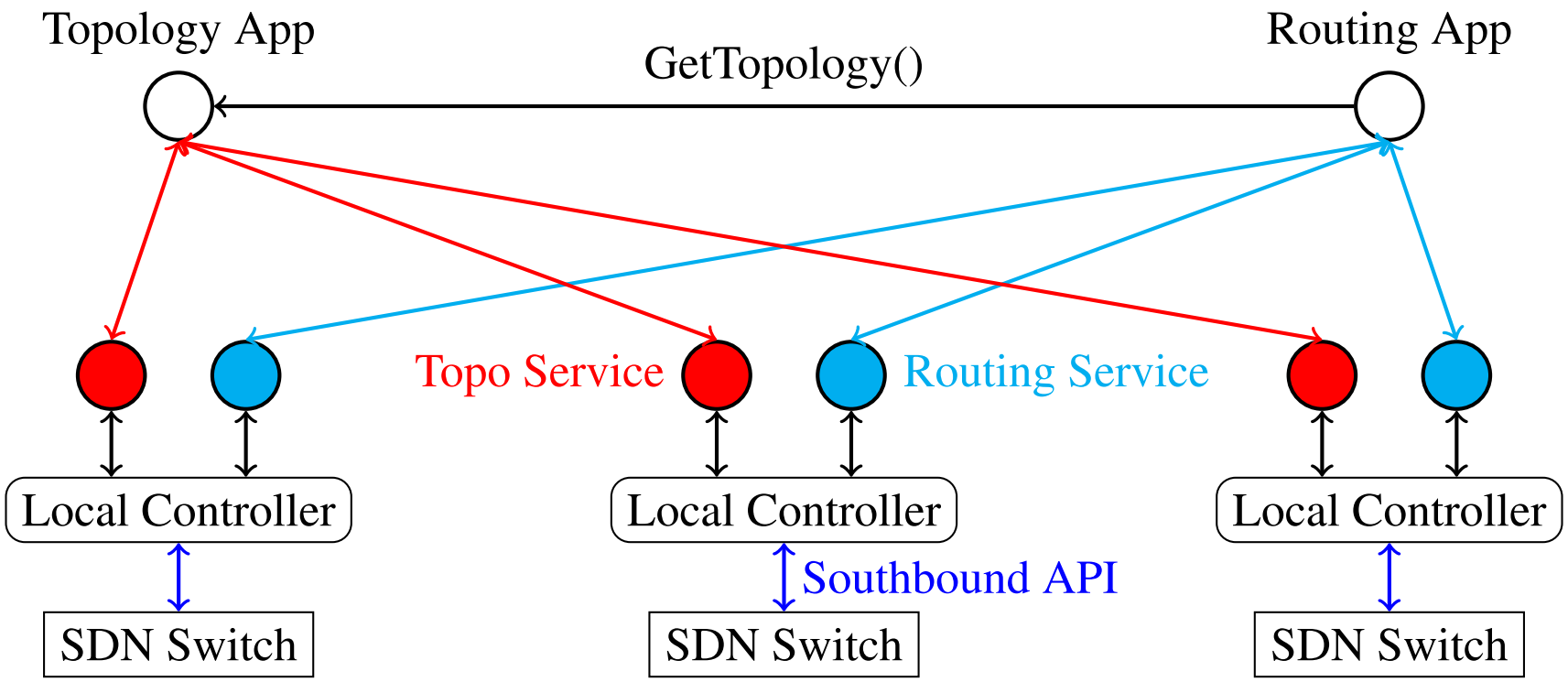
- 所有的控制逻辑在全局控制平面完成
  - 中央控制器负载
  - 控制网络负载
  - 控制延迟
- 低抽象层次（面向流）的控制消息
  - 控制网络负载
  - 应用与 OpenFlow 协议有一定耦合

Ryuo 的架构



- Ryuo 应用: 拥有网络全局信息
- 本地服务: 处理本地事件，向 Ryuo 应用提供高抽象级接口
- 在控制网络上传递领域特定的控制消息

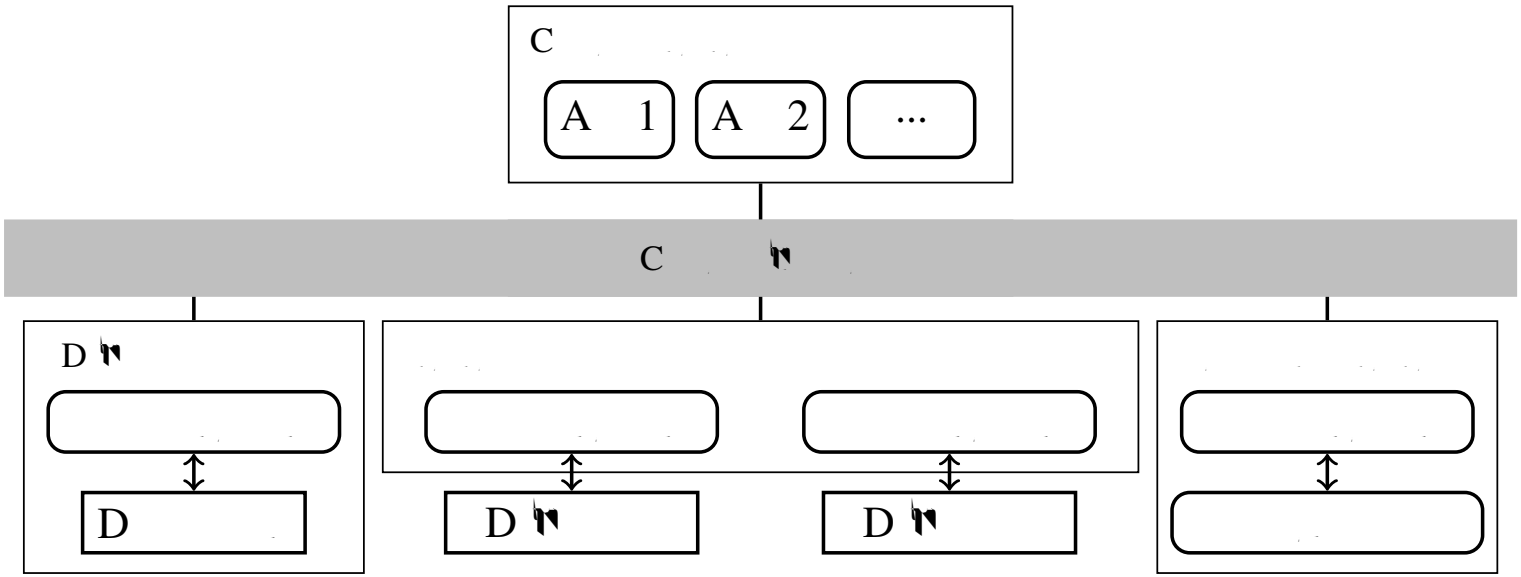
示例：拓扑发现以及 IP 路由



Ryuo 的实现

- Ryuo 应用以及本地服务基于 Ryu 控制器开发
- Ryuo 应用与本地服务间通信采用 RPC 框架 Pyro4
- Ryuo 以及实验用代码见
  https://github.com/epcc-networking/ryuo

部署 Ryuo



- 本地控制平面
  - 直接部署在硬件 SDN 交换机上
  - 运行在服务器上，直接控制本地的软件 SDN 交换机
  - 运行在服务器上，管理一台或几台邻近的硬件 SDN 交换机

性能测试

- 控制流量
- 控制延迟
- 处理网络事件的吞吐量

测试环境

- Pica8 P-3295 OpenFlow 交换机
  - CPU: 825MHz PowerPC
  - 内存: 512MB
- PC: 用做主机、控制器
  - CPU: Intel Core i5-3470, 3.2 GHz
  - 内存: 4GB
- Mininet
- Open vSwitch

控制流量测试

- 部署了拓扑发现以及 IP 路由两个 Ryuo 应用
- Mininet
- 拓扑结构来自 Topology Zoo
- 采用 wireshark 记录控制网络上的所有数据包

$$d_{overall} = \frac{ControlTraffic_{Ryuo}}{ControlTraffic_{Ryu}} \tag{1}$$

$$d_{abstraction} = \frac{ControlTraffic_{Ryuo}}{ControlTraffic_{Ryu} - Traffic_{LLDP+ICMP+ARP}} \tag{2}$$

控制流量

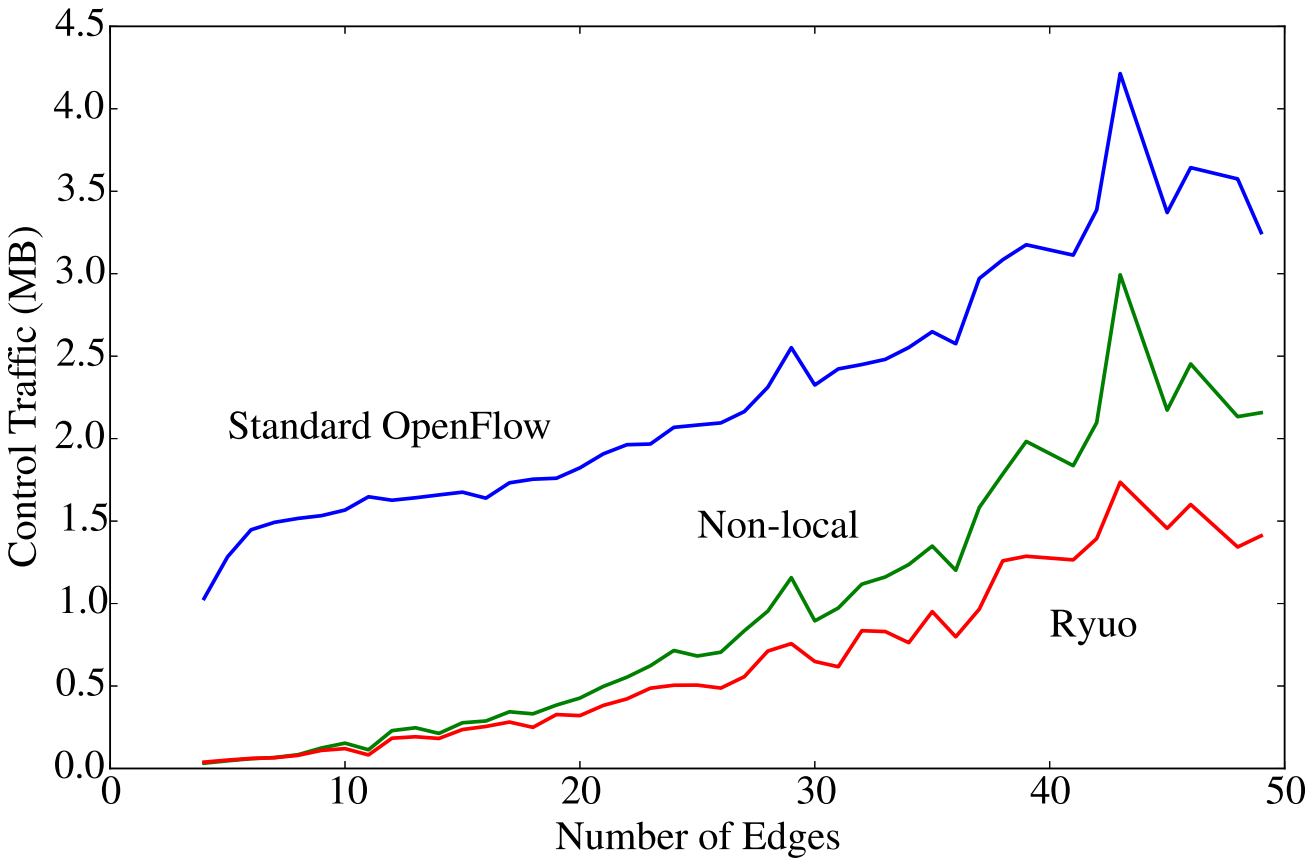
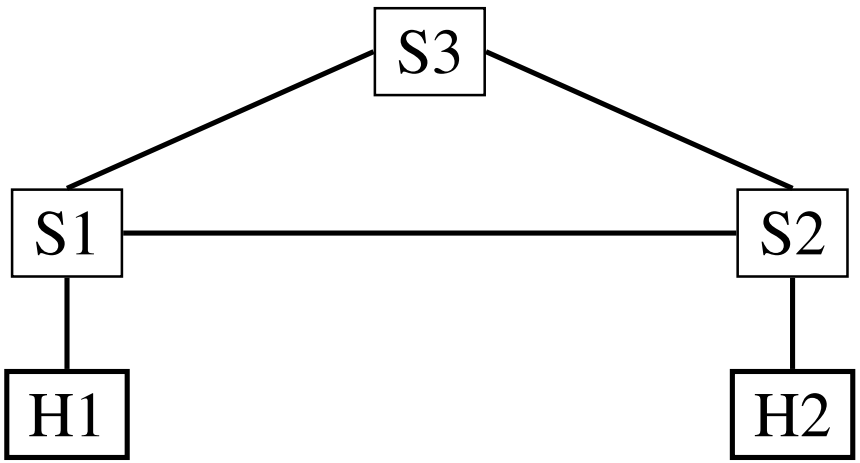


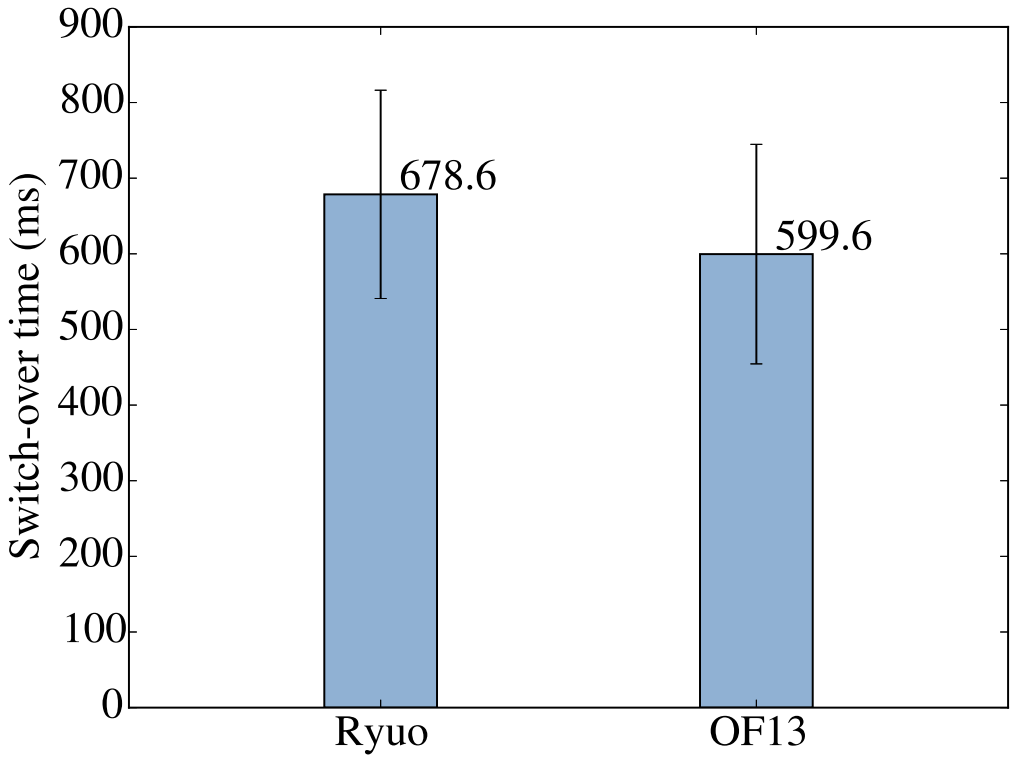
Figure: OpenFlow vs 类似于 Kandoo 的本地控制器 (approx.) vs Ryuo

使用 OpenFlow 1.0 实现 Fast Failover Group



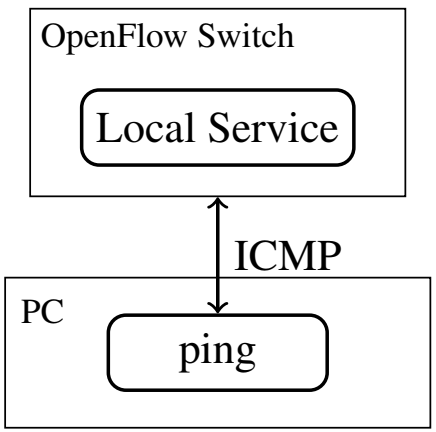
- H1: 1K packets/s.
- H2: Wireshark.
- S1-S2 $\rightarrow$ S1-S3-S2
- 统计丢失的数据包数量

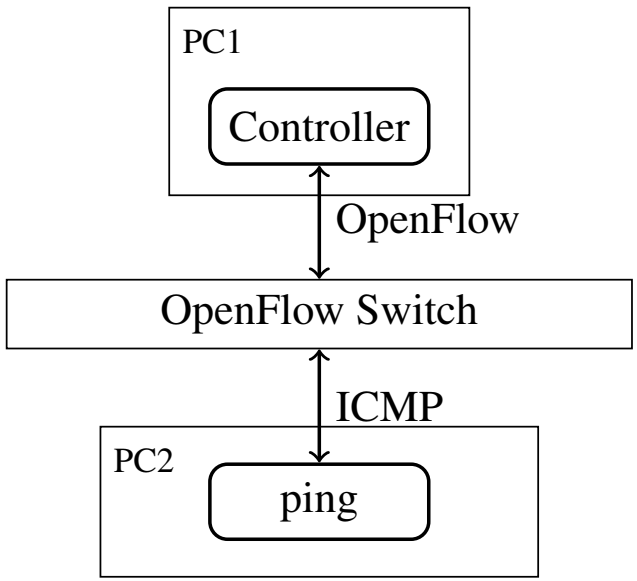
使用 OpenFlow 1.0 实现 Fast Failover Group



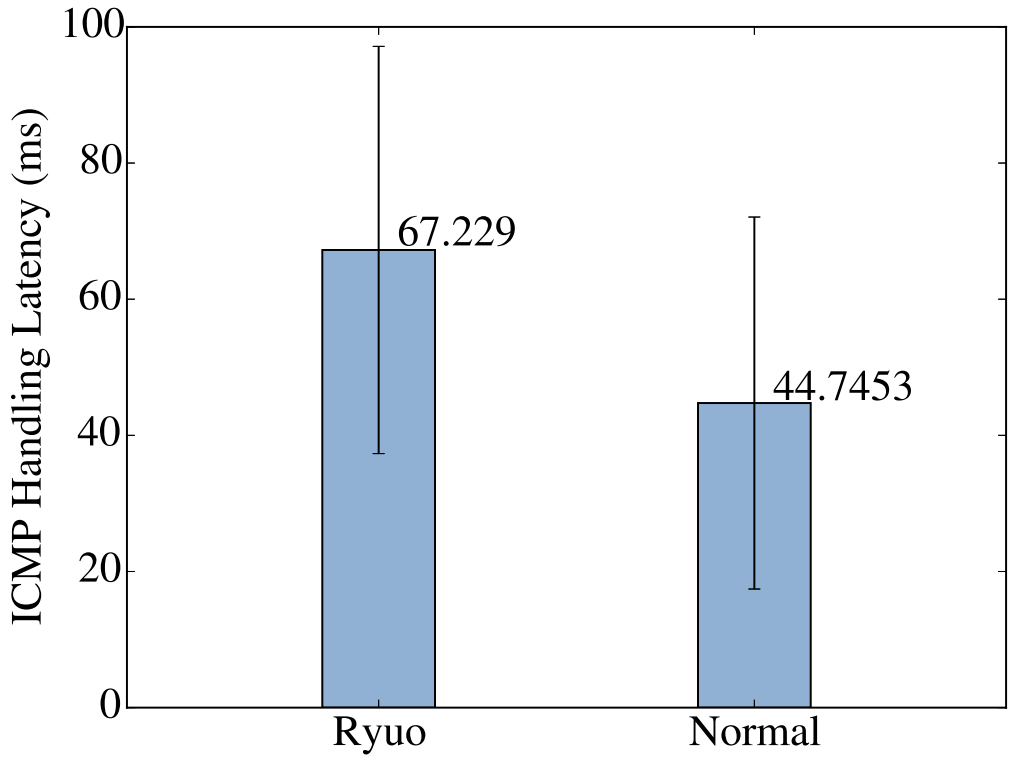
- 本地服务慢约 80ms

简单任务的控制延迟测试设置



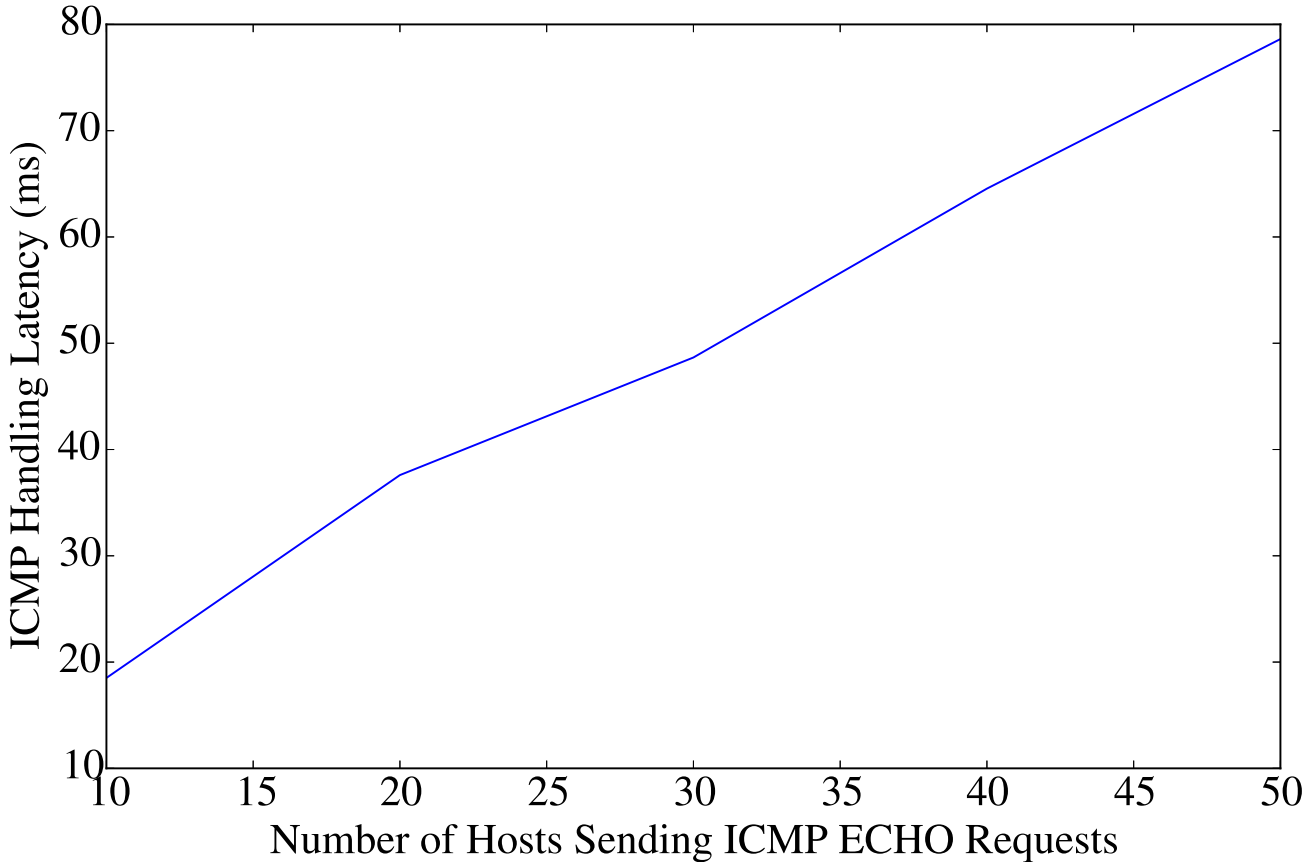


简单任务的控制延迟



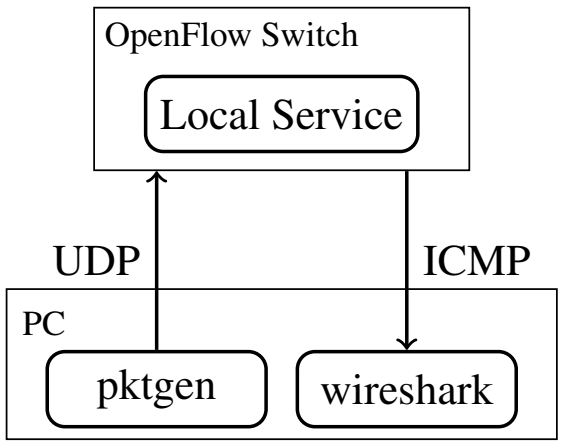
- 本地服务慢约 20ms

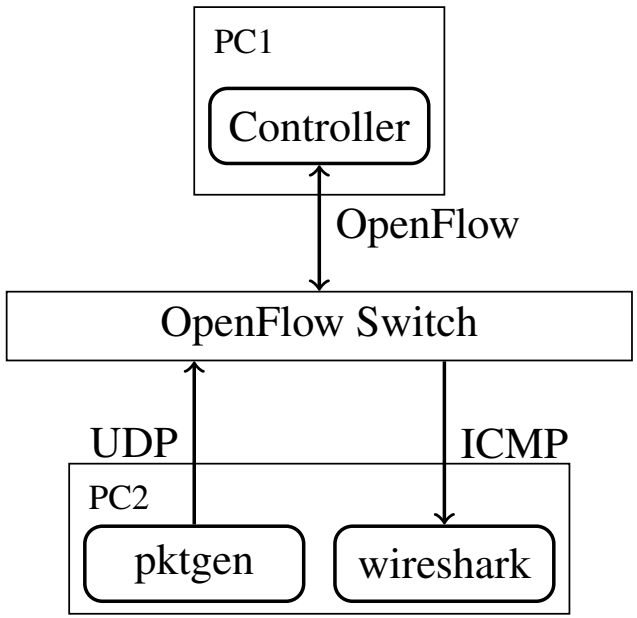
控制延迟与并发请求数的关系



- 控制延迟与并发请求数正相关

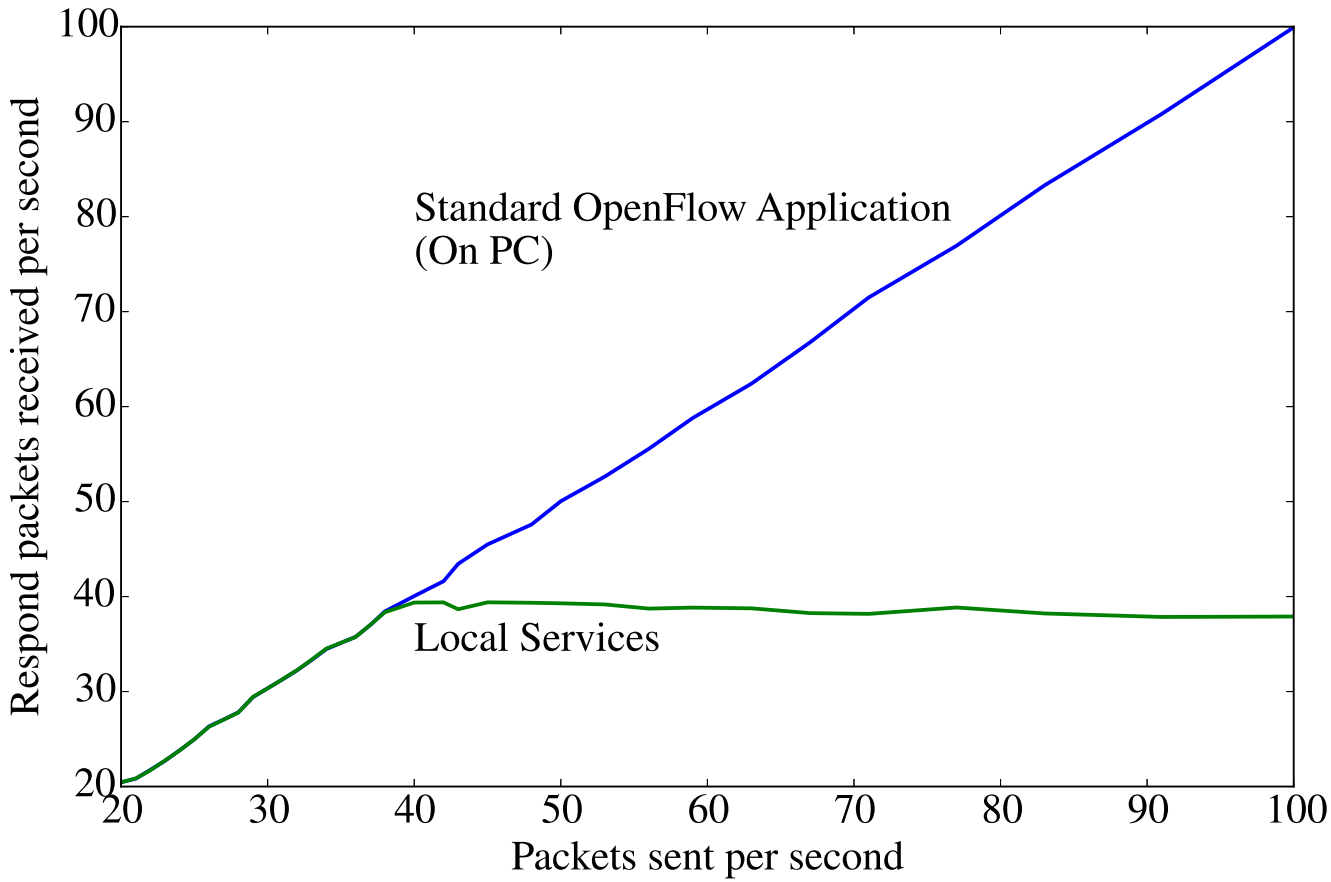
吞吐量测试设置





- 吞吐量：每秒钟处理的网络事件数量

吞吐量测试结果



- 本地服务：约 40 packets/s

总结

- Ryuo 的优势:
  - 降低全局控制平面、控制网络的负载
  - 可重用的本地服务模块
  - 领域特定的高级接口
  - 更低的控制延迟
- 不足:
  - 本地控制平面的部署和运维的复杂性
  - 受限于硬件交换机的计算能力

Q&A

谢谢！