Ryuo

Using High Level Northbound API for Control Messages in SDN

Shaoyu Zhang[1], Yao Shen[1], Matthias Herlich[2], Kien Nguyen[3], Yusheng Ji[2], Shigeki Yamada[2]

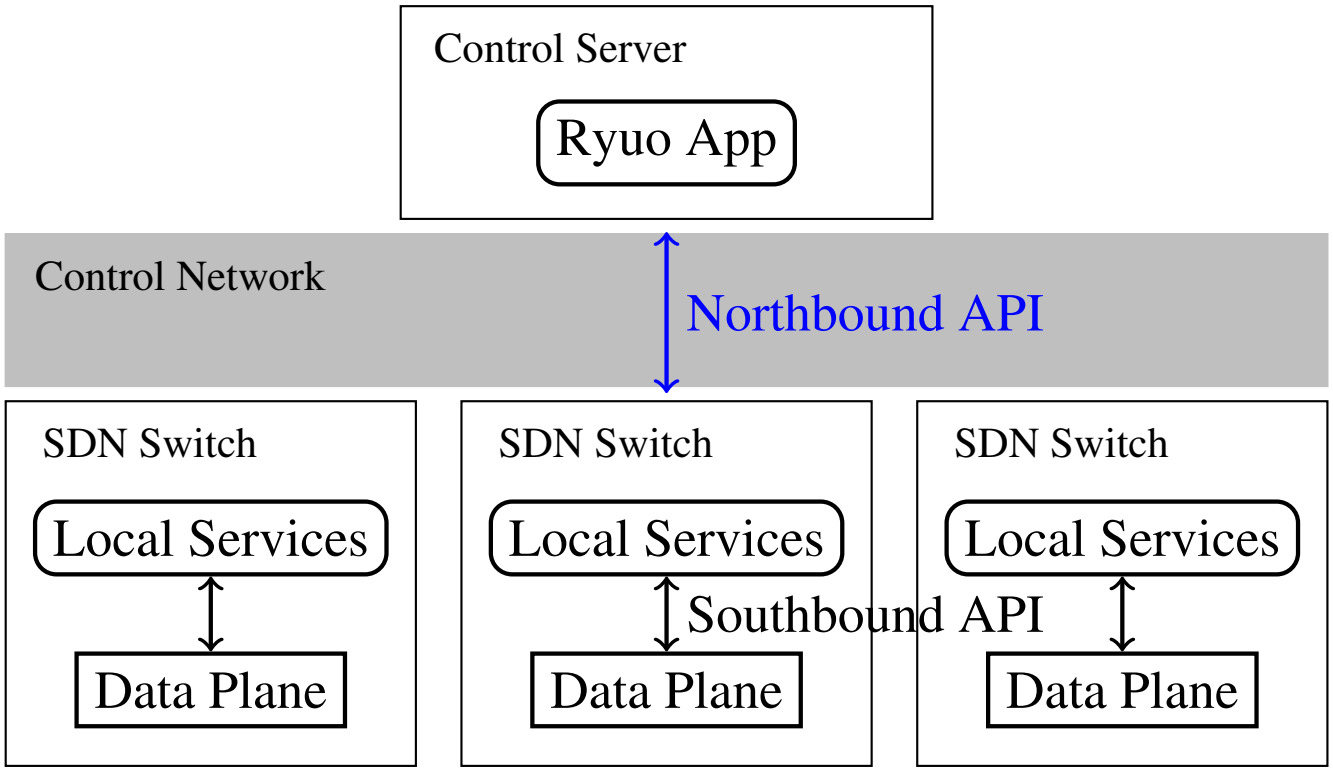
[1]Shanghai Jiao Tong University, China
[2]National Institute of Informatics, Japan
[3]National Institute of Information and Communications Technology, Japan

August 18, 2015

Issues of OpenFlow

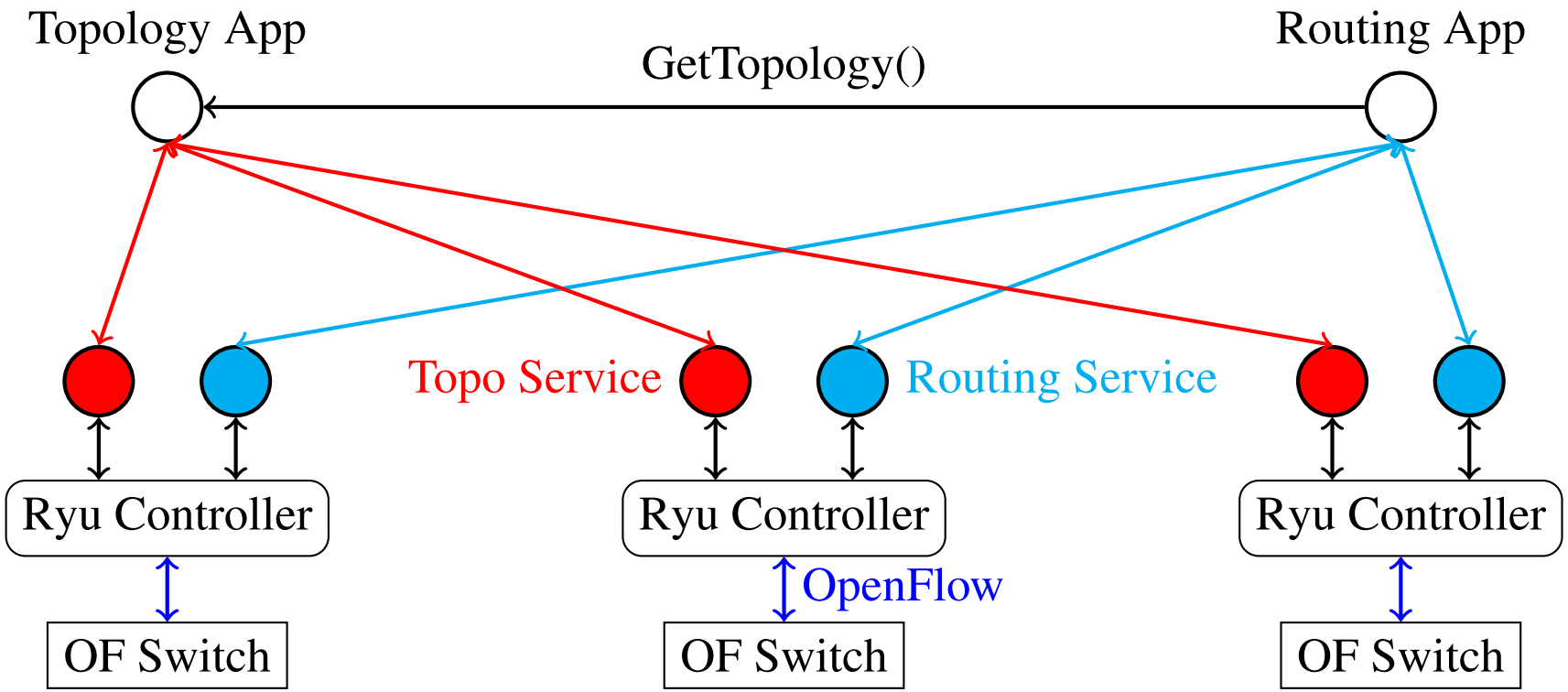
- Control latency
- Applications couple with OpenFlow at some degree
- All control logic centralized

Ryuo: Architecture



- Ryuo App: Focus on business logic
- Local Service: Provides high level API, ensures compatibility
- Domain specific control message

Example: Topology Discovery and Routing



Evaluations

- Control Traffic
- Control Latency
- Throughput of handling local events

Evaluation Enviornment

- Pica8 P-3295 OpenFlow switch
  - CPU: 825MHz PowerPC
  - Memory: 512MB
- PC: For host, Ryu controller, Ryuo Application
  - CPU: Intel Core i5-3470, 3.2 GHz
  - Memory: 4GB

Control Traffic Evalution

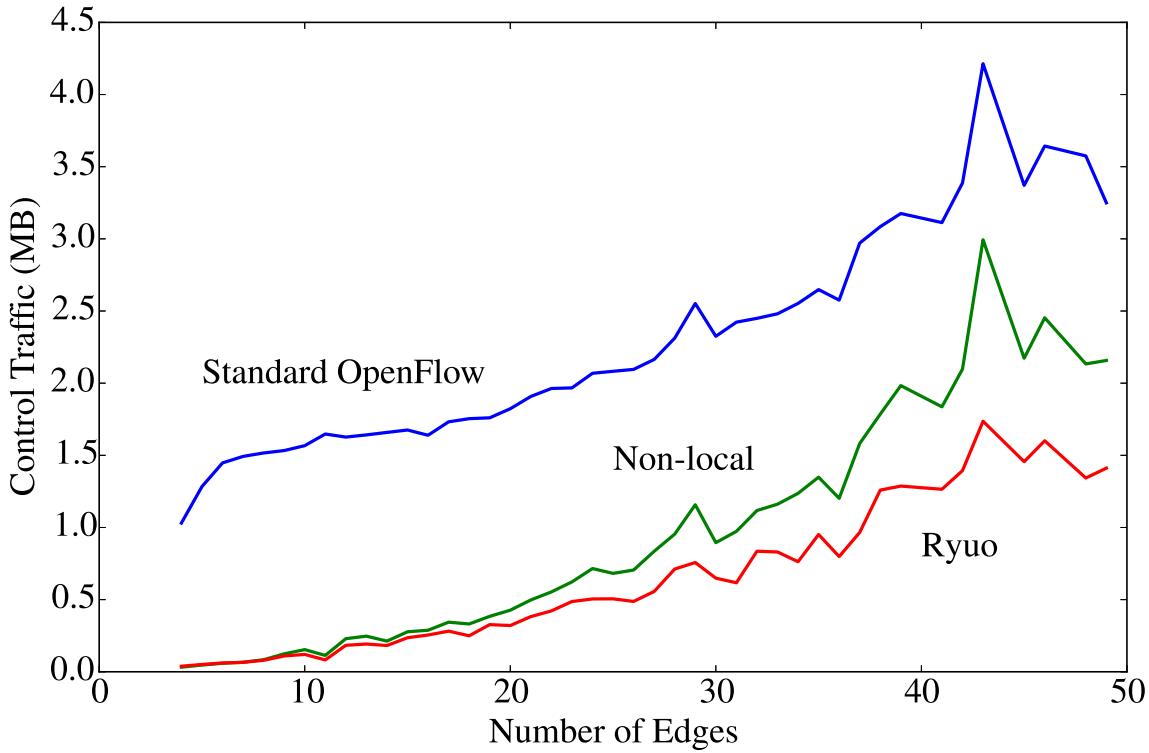
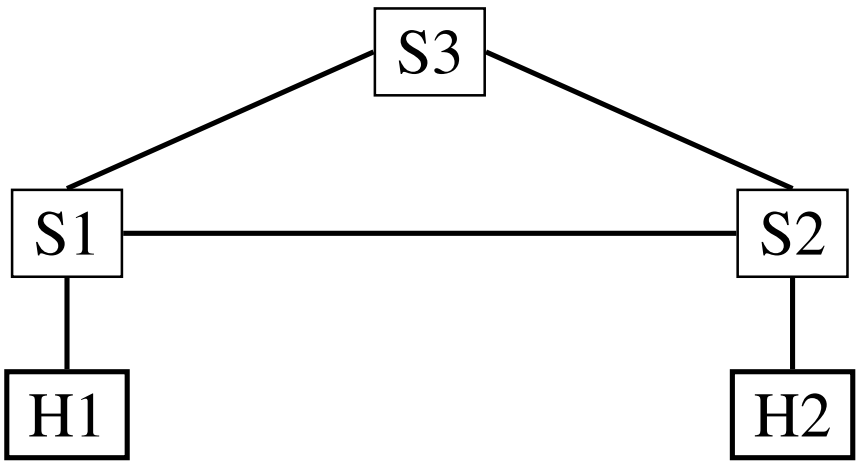


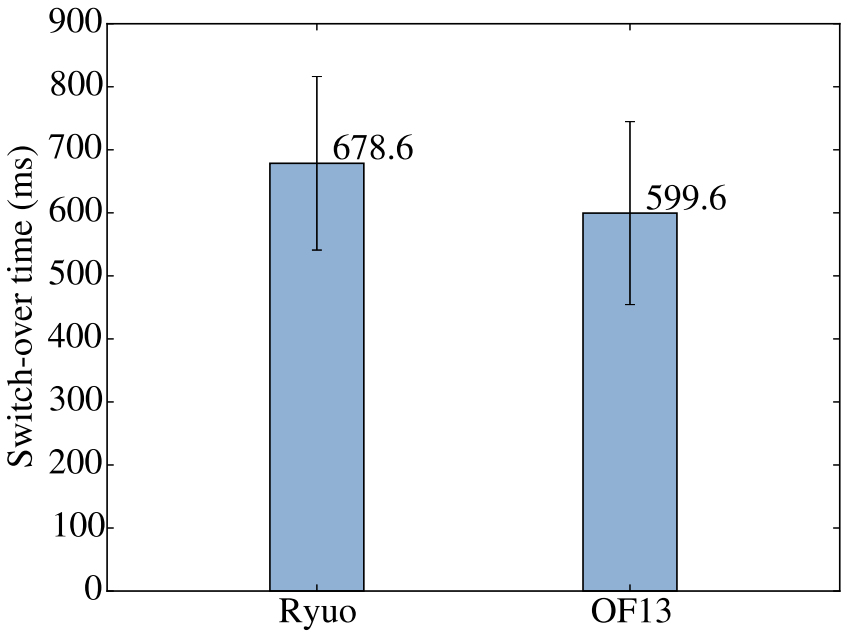
Figure: OpenFlow vs Local Controller Approach (approx.) vs Ryuo

Evaluation: Fast Failover with OpenFlow 1.0



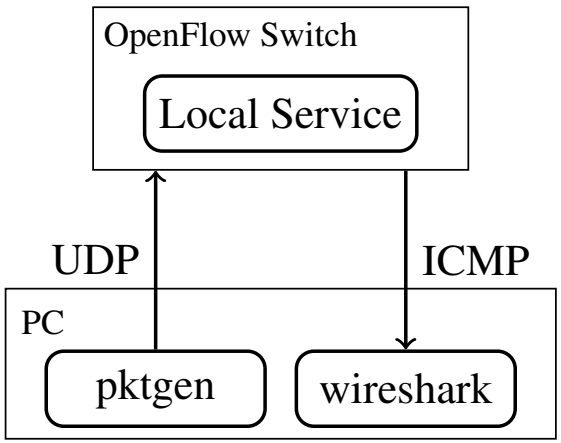
- H1: 1K packets/s.
- H2: Wireshark.
- S1-S2 $\rightarrow$ S1-S3-S2
- Count pakcets lost

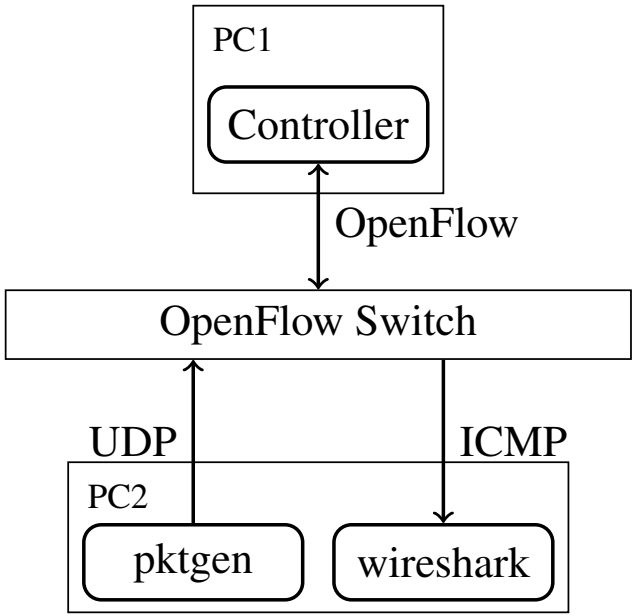
Evaluation: Fast Failover with OpenFlow 1.0



- Local Service: ~0.1s slower.

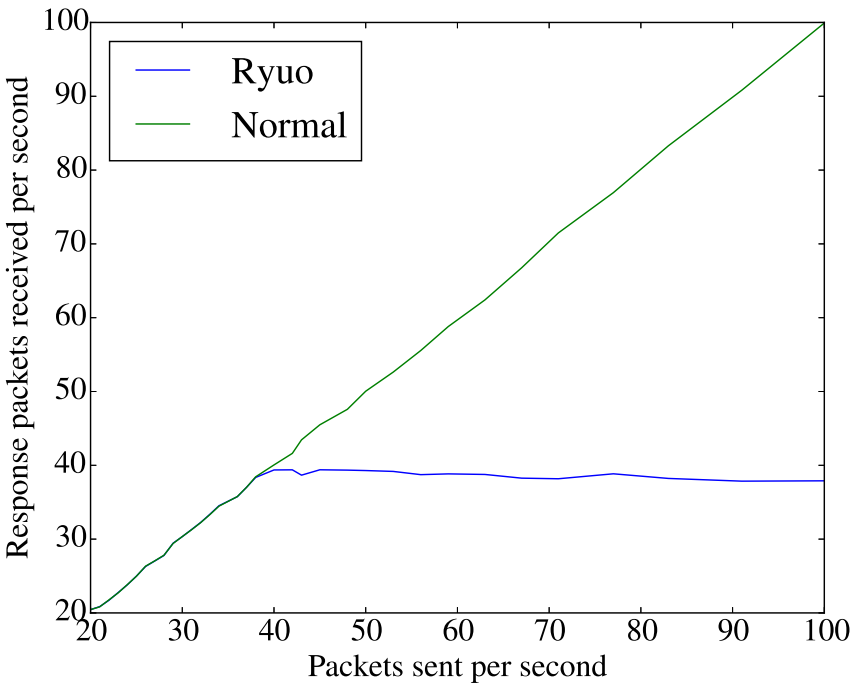
Throughput Evaluation: Setup





- Throughput: Number of events handled in one second.
- Send UDP packets, record responses.

Throughput Evaluation: Result

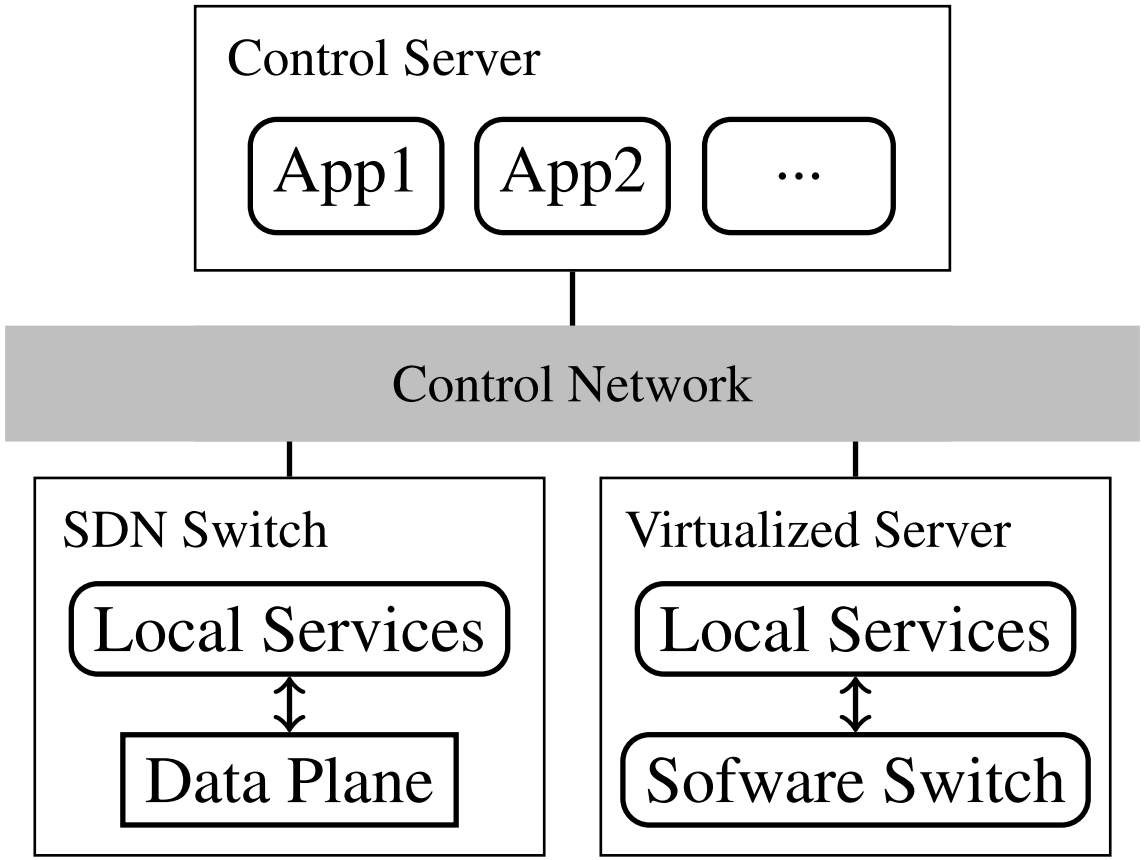


- Switch can only handle about 40 packets per seconds.
- Not a problem on a virtualized server.
- New switches with more powerful CPUs.[1]

Ryuo: Implementation

- Ryuo Apps and Local Services based on Ryu.
- Global Local Communication: Pyro4[2]
- Available Online: https://github.com/epcc-networking/ryuo

Ryuo: Deployment



- Two ways to deploy:
  - Directly on OpenFlow switches.
  - On servers with multiple VMs and Open vSwitch.

Related Work

- Kandoo[3]: Local and Root Controller
- Beehive Netctrl[4]: Distributed Network Controller
- Orion[5]: Hierarchical Control Plane

Conclusion

- Advantages:
  - Reusable Local Services provide high level API.
  - Easy to extend southbound API.
  - Less control traffic.
- Disadvantanges:
  - Limited by resources on the switch.
  - Local Service deployment.

Questions?

Thank you!

References I

[1] AS5712-54X with ONIE / 10GbE Data Center Switch. [Online]. Available: http://www.edge-core.com/ProdDtl.asp?sno=457&AS5712-54X

[2] Pyro - Python Remote Objects. [Online]. Available: http://pythonhosted.org/Pyro4/

[3] S. Hassas Yeganeh and Y. Ganjali, "Kandoo: A framework for efficient and scalable offloading of control applications," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*. ACM, 2012, pp. 19–24.

References II

[4] ——, "Beehive: Towards a simple abstraction for scalable software-defined networking," in *Proceedings of the 13th Workshop on Hot Topics in Networks*. ACM, 2014, pp. 13:1–13:7.

[5] Y. Fu, J. Bi, K. Gao, Z. Chen, J. Wu, and B. Hao, "Orion: A hybrid hierarchical control plane of software-defined networking for large-scale networks," in *IEEE 22nd International Conference on Network Protocols (ICNP)*, 2014, pp. 569–576.