

SDN Traffic Engineering using Segment Routing and DNS

Improving Quality of Experience in SANREN

Luqmaan Salie

Supervisor: Prof. Josiah Chavula

Department of Computer Science
University of Cape Town



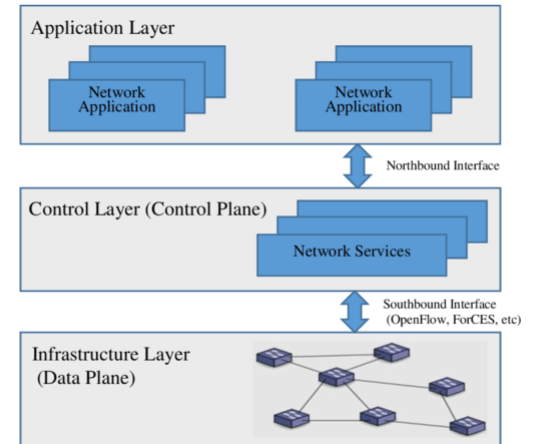
Introduction

SDN – Software Defined Networking

Separates control and forwarding planes, and uses a controller to carry out forwarding.

Automatically changes routes when congestion, failures, or delays occur.

- All routers can be programmed, from a single location
- Avoids router misconfigurations
- OpenFlow protocol configures flow tables on all routers

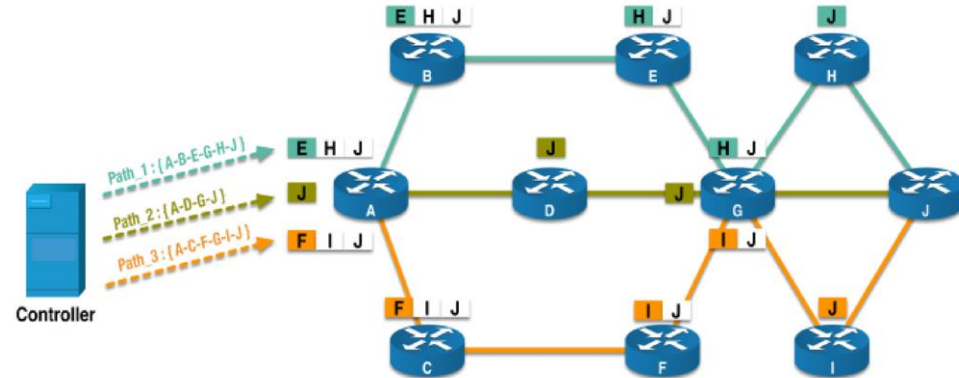


SR – Segment Routing

Source-based tunneling technique that allows hosts to transport packets with a list of segments.

Segments specify a route to traverse (topological instruction), or a service for the packet to reach (service instruction).

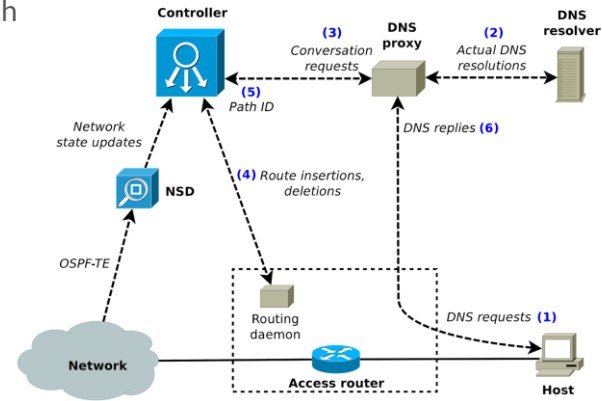
- Reduces forwarding rule overhead
- Improves network flexibility and scalability



SRN – Software Resolved Network

Allows network operators to control the network paths used by applications with specific policies.

- Applications interact with DNS resolver to specify path requirements (delay or bandwidth)
- Controller and DNS resolver work together to forward packets along optimal path
- Controller only interacts with edge routers





Research Questions

To what extent can we improve end-to-end performance in SANReN using SDN, SR, and DNS?

1. Would application-level traffic segmentation using SDN, SR, and DNS result in better overall performance?
2. Would an SDN + SR solution result in better resource utilisation?



Methodology



Methodology – Experiment and evaluation tools

Performance of application-level traffic segmentation (Q1) will be measured using 2 different tools:

- Perfsonar [11] – Internal SANReN performance testing
- Speedchecker [12] – to test performance when accessing SANReN externally

Performance of our solution will be evaluated with the following tools:

- Ryu [13] – python SDN controller framework
- Mininet [14] – tool for creating virtual networks
- SRv6 - uses a Linux Kernel [15] – open source SRN implementation



Methodology – Experiments

Performance of application-level traffic segmentation (Q1) will be measured using 3 types of tests:

- Delay (ping) – ping zero-rated websites in South Africa which reside inside SANReN
- Page load time – total time for page to load from when URL is entered until page is done loading
- Traceroute – conduct traceroutes to each website to find reasons for bad performance



Methodology – Proposed solution

A software resolved network using SDN, SR, and DNS.

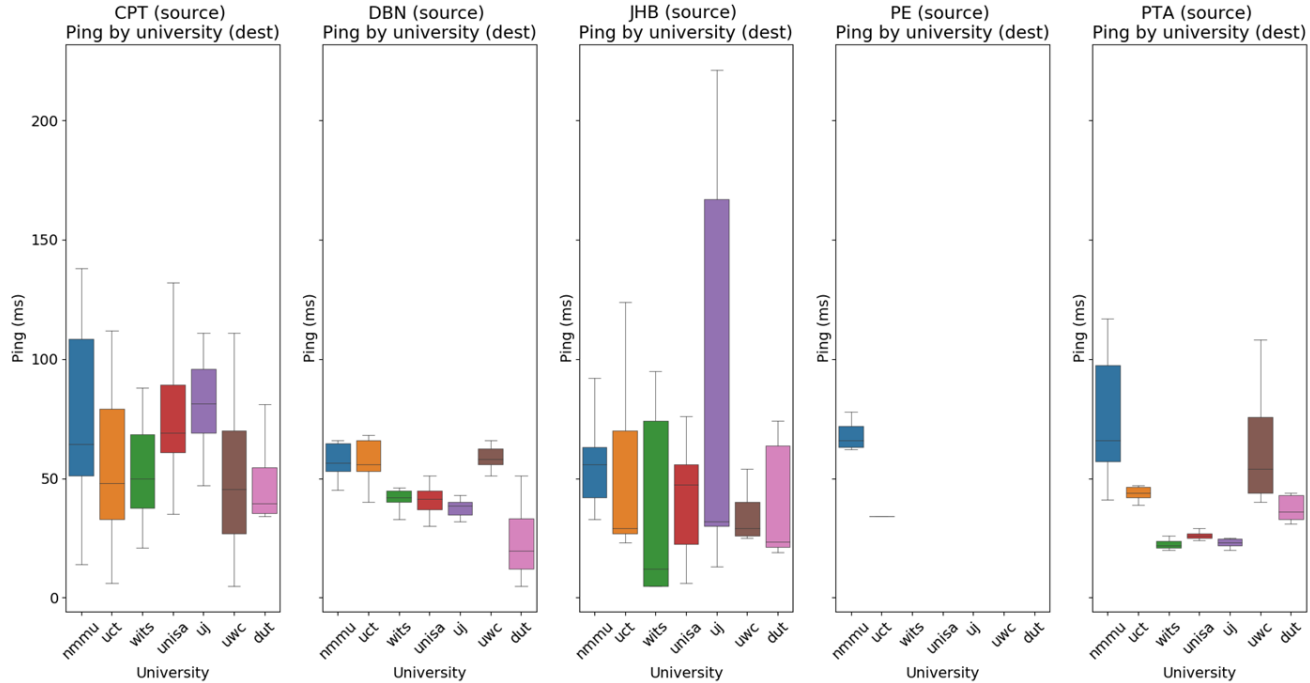
SRN controller components:

- Multipath TCP calculator – calculates shortest paths and allocates paths to flow rules
- Traffic Engineering and Segment Routing component – maps flow rules to SR paths, determines type of traffic
- DNS resolver – resolves target domain names, and can select path for specific traffic
- Quality of Experience management component – keeps track of network state (latency, throughput, packet loss, etc.)

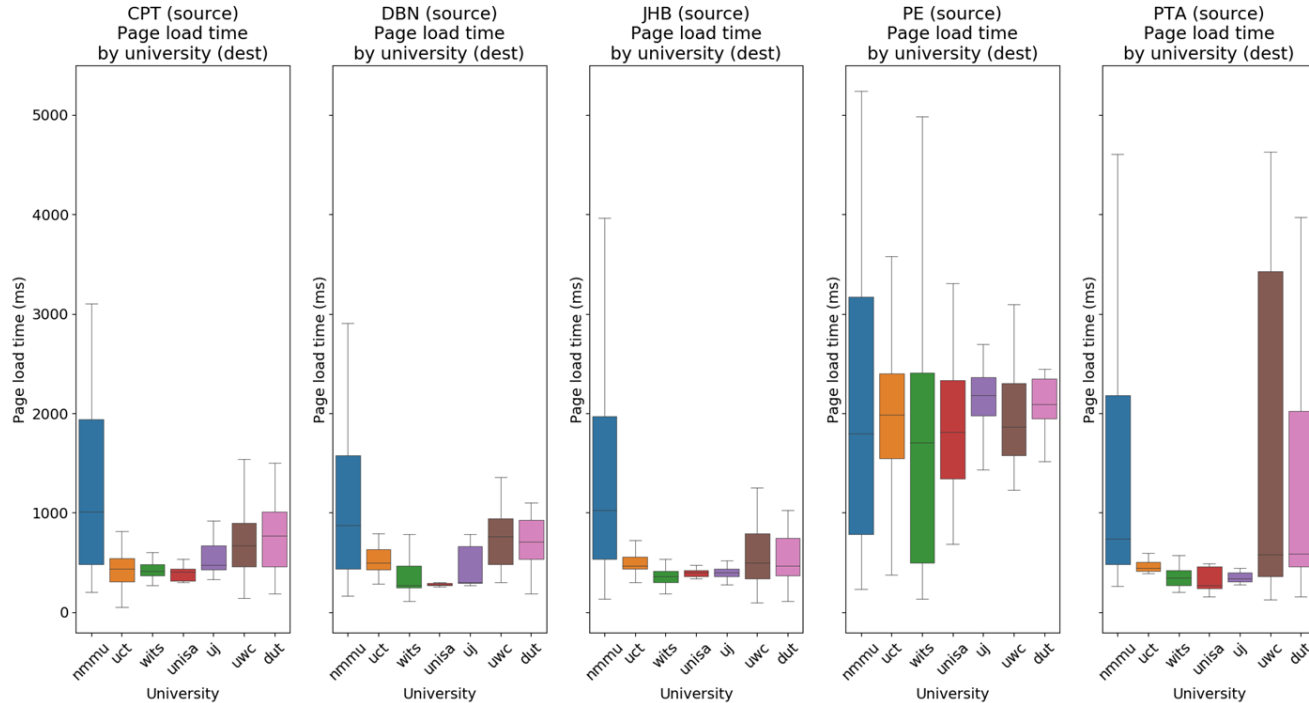


Network Benchmark Tests

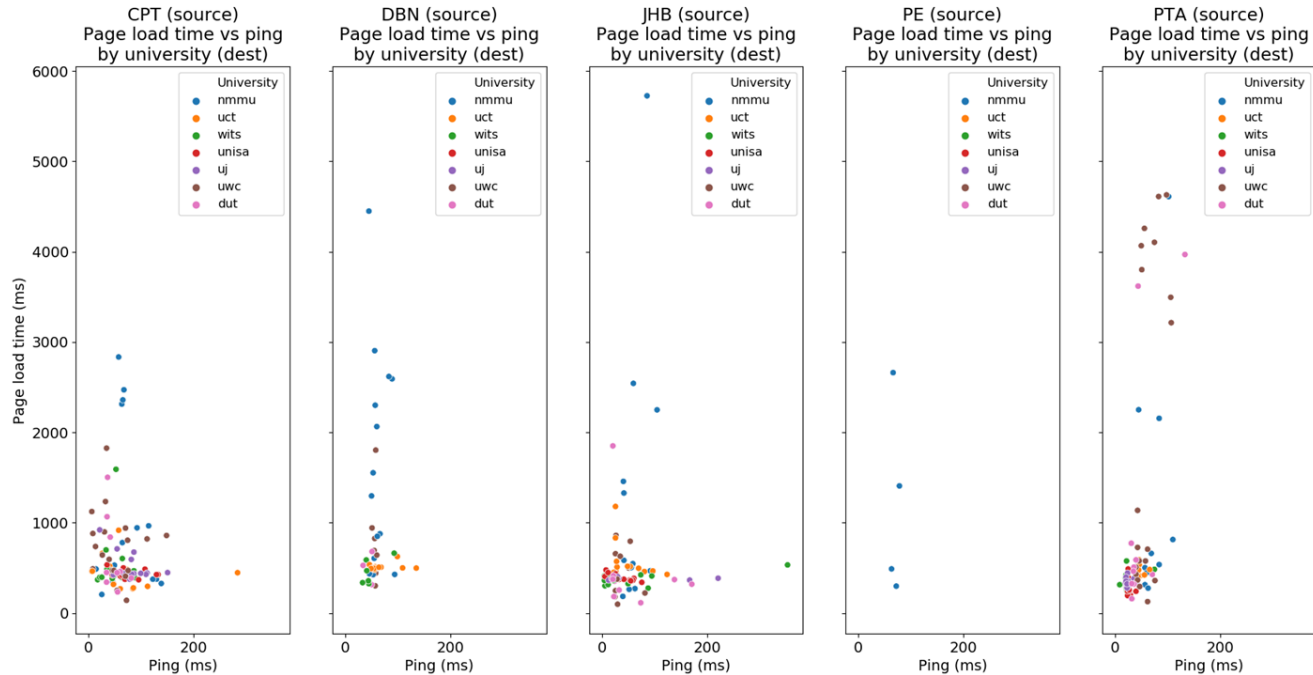
Results – External: Delay



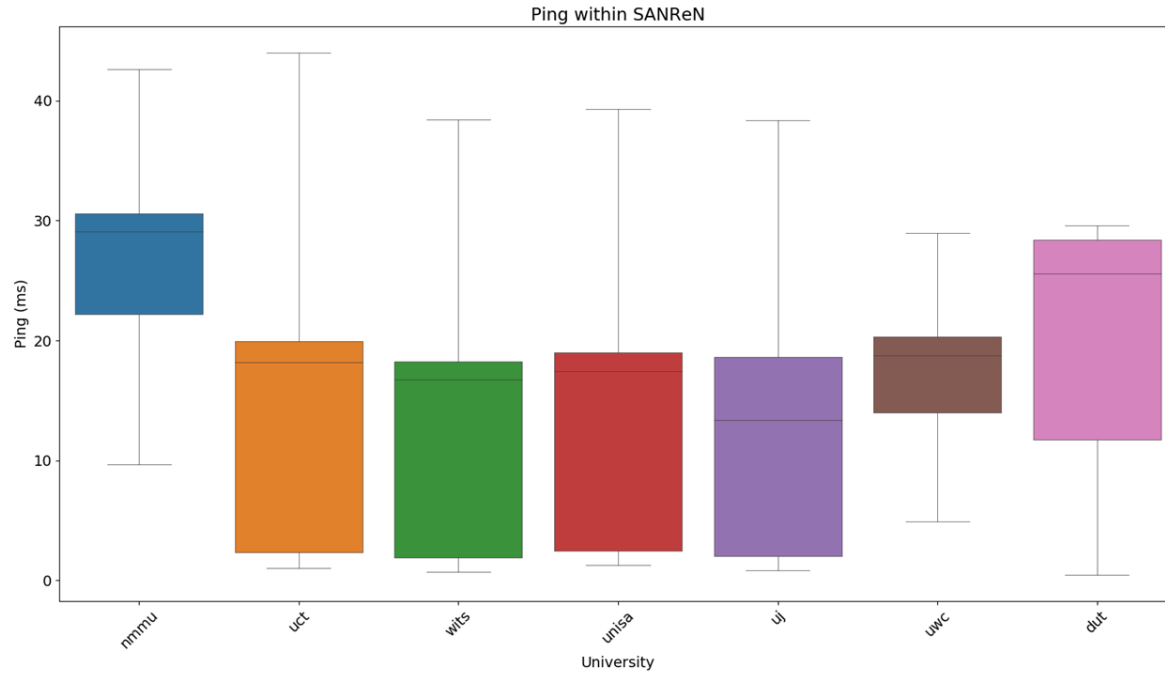
Results – External: Page load time



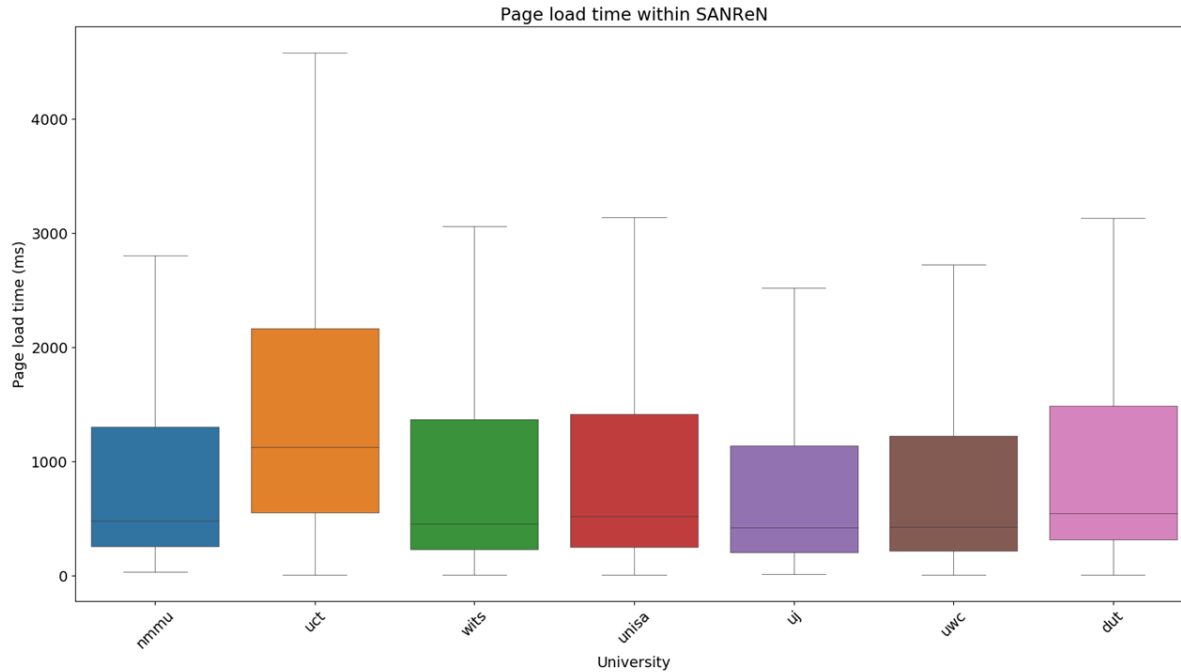
Results – External: Delay v Page load time



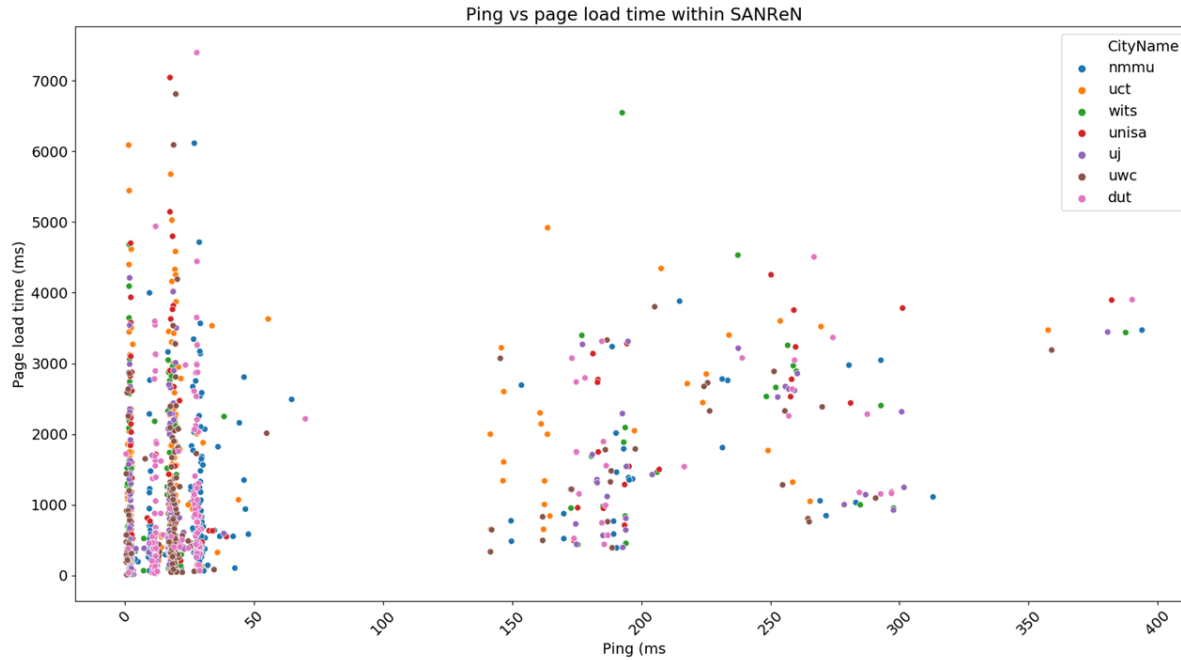
Results – Internal: Delay



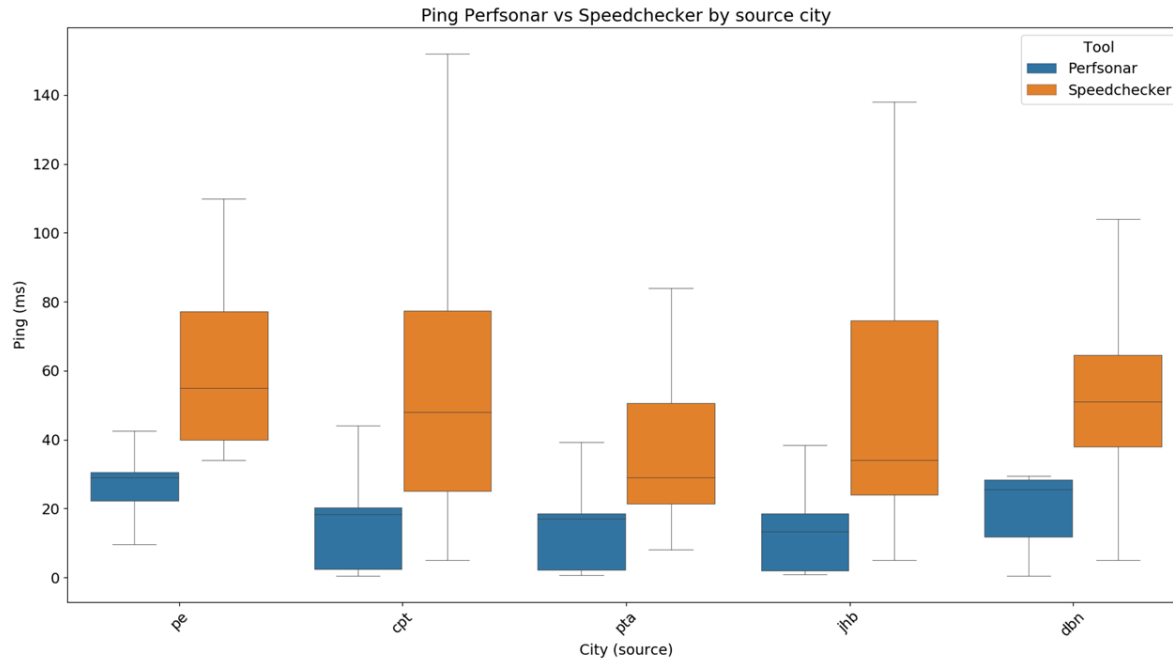
Results – Internal: Page load time



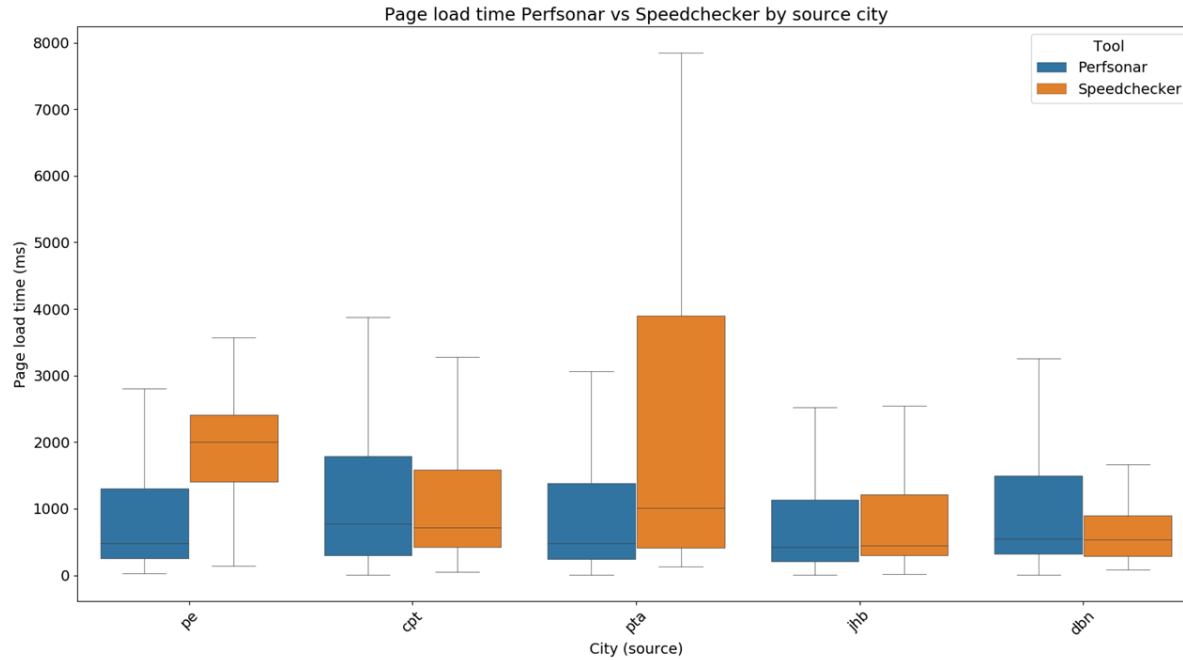
Results – Internal: Delay v Page load time



Internal v External: Delay



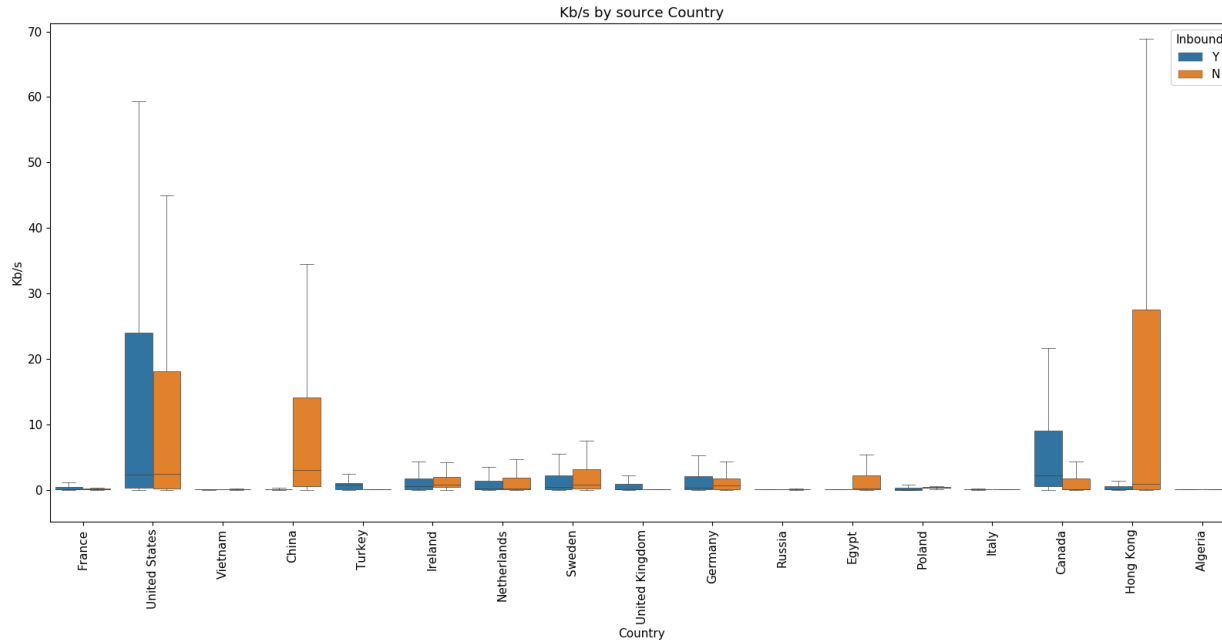
Internal v External: Page load time



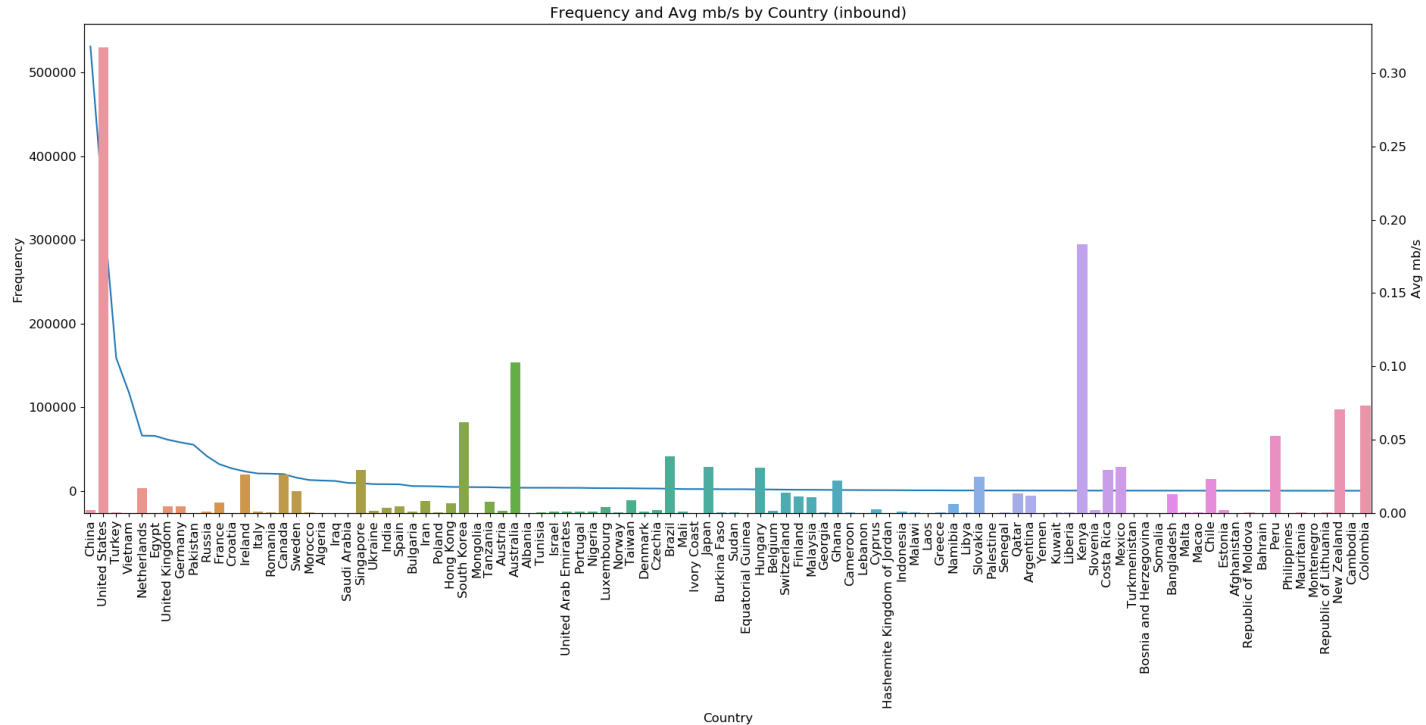
SANReN Data



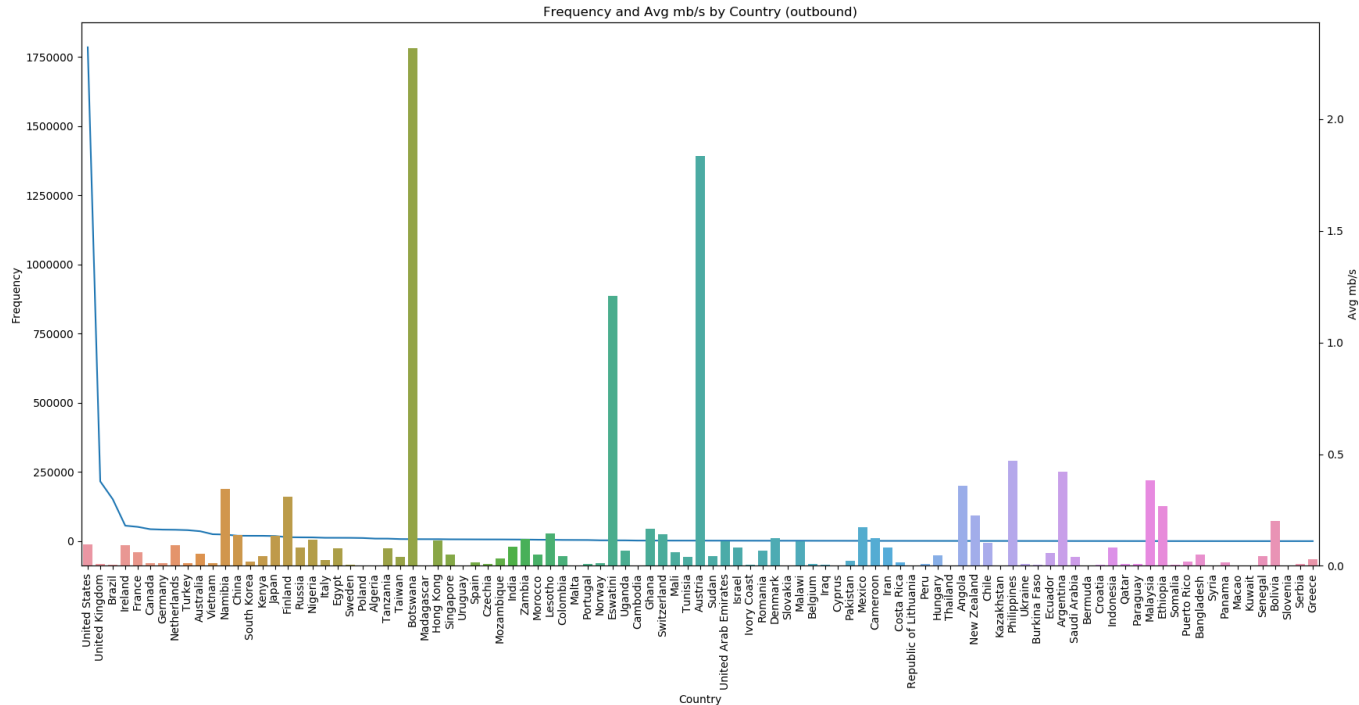
Kb/s by country – inbound vs outbound



Frequency & Average mb/s by Country (inbound)



Frequency & Average mb/s by Country (outbound)





Conclusion



Conclusion

- Software Defined Networking and Segment Routing can improve network performance
- DNS Resolver provides additional performance improvements and flexibility
- To what extent can we improve end-to-end performance in SANReN using SDN, SR, and DNS?
- Solution will be evaluated using a virtual scaled-down version of SANReN
- Results of proposed solution will be compared to results gathered from existing architecture



References

1. Chetty, M., Sundaresan, S., Muckaden, S., Feamster, N., and Calandro, E. Measuring broadband performance in South Africa. In Proceedings of the 4th Annual Symposium on Computing for Development (New York, NY, USA, 2013), ACM DEV-4 '13, ACM, pp. 1:1–1:10.
2. Chandra, B., Koizumi, T., Hayashi, M., and Yamamoto, H. A study on network reliability evaluation for developing countries. In 2012 18th Asia-Pacific Conference on Communications (APCC) (Oct 2012), pp. 749–753.
3. Chavula, J., Feamster, N., Bagula, A., and Suleman, H. Quantifying the effects of circuitous routes on the latency of intra-Africa internet traffic: A study of research and education networks. In e-Infrastructure and e-Services for Developing Countries (Cham, 2015), A. Nungu, B. Pehrson, and J. Sansa-Otim, Eds., Springer International Publishing, pp. 64–73.
4. South African National Research and Education Network (SANReN). <https://www.sanren.ac.za/>.
5. Sezer, S., Scott-Hayward, S., Chouhan, P. K., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M., and Rao, N. Are we ready for SDN? implementation challenges for software-defined networks. IEEE Communications Magazine 51, 7 (July 2013), 36–43.
6. Kim, H., and Feamster, N. Improving network management with software defined networking. IEEE Communications Magazine 51, 2 (February 2013), 114–119.
7. Neghabi, A., Navimipour, N., Hosseinzadeh, M., and Rezaee, A. Load balancing mechanisms in the software defined networks: a systematic and comprehensive review of the literature. IEEE Access PP (03 2018), 1–1.
8. Filsfils, C., Nainar, N. K., Pignataro, C., Cardona, J. C., and Francois, P. The segment routing architecture. In 2015 IEEE Global Communications Conference (GLOBECOM) (Dec 2015), pp. 1–6.
9. Pang, J., Xu, G., and Fu, X. SDN-based data center networking with collaboration of multipath TCP and segment routing. IEEE Access 5 (2017), 9764–9773.
10. Lebrun, D., Jadin, M., Clad, F., Filsfils, C., and Bonaventure, O. Software resolved networks: Rethinking enterprise networks with ipv6 segment routing. In Proceedings of the Symposium on SDN Research (New York, NY, USA, 2018), SOSR '18, ACM, pp. 6:1–6:14.
11. "Speedchecker," <https://www.speedchecker.com/>, 2020.
12. A. Hanemann, J. W. Boote, E. L. Boyd, J. Durand, L. Kudarimoti, R. Łapacz, D. M. Swany, S. Trocha, and J. Zurawski, "Perfsonar: A service oriented architecture for multi-domain network monitoring," in International conference on service-oriented computing. Springer, 2005, pp. 241–254.
13. Ryu SDN Framework. <https://osrg.github.io/ryu-book/en/Ryubook.pdf>.
14. Fontes, R. R., Afzal, S., Brito, S. H. B., Santos, M. A. S., and Rothenberg, C. E. Mininet-Wi-Fi: Emulating software-defined wireless networks. In 2015 11th International Conference on Network and Service Management (CNSM) (Nov 2015), pp. 384–389.
15. SRv6 - Linux Kernel Implementation. <https://segment-routing.org/index.php/SRN/SRN>

Questions?