Universität Paderborn — Fakultät EIM
Fachgebiet Rechnernetze
http://www.upb.de/cs/cn/

Bachelor / Master Thesis

Fast Consistent Network Updates in SDN

Background

Networks continue to change because of node failures, new network policies or changing traffic demands. One way to deal with these changes is the idea to extract control functionality out of network nodes and centralize it at a single point of control – this concept is commonly called Software-Defined Networking (SDN).

In SDN, adapting to changes happens by propagating configuration updates from the central control to individual network elements. Such network updates in SDN should be fast and must be consistent, least data flows are damaged. Consistency requires four properties: i) packet coherence: every packet is processed with single network state; not mix of old and new rules, ii) loop freedom: packet flow is loop-free, iii) blackhole freedom: no packet is unintentionally dropped, and iv) congestion freedom: no link is loaded beyond its capacity. When distributing new network state from the SDN controller to multiple switches, the correctness and performance updates typically depend on: i) the number of rules to be installed in switches ii) the ordering of the update operations and iii) the load on links during an update.

SDN is, in principle, a very versatile tool; a typical application area are data centers. In such a center, distributed applications are a common workload, resulting in traffic patterns like shuffles. To support such applications, the concept of *coflows* [1] has been developed. However, current network control systems do not support coflows explicitly. One possible approach to do so could be the Smallest-Effective-Bottleneck-First (SEBF) heuristic.

Thesis Goals

During a network update, a network controller distributes the network state to multiple switches and the new network configuration is complete once the last update operation is complete. We can either treat update operations as a single transfer or schedule groups of updates. The goal of this thesis is to speed up the flow update using better heuristics. This requires understanding of both the coflow [1] and the network update [4] abstraction. The candidate is required to design and implement the coflow property in an SDN network controller by either simulation or prototype. The result will then be evaluated and compared with existing (coflow-unaware) approaches [3, 2] for network updates in SDN.

Milestones

- Choose tool to simulate or prototype the network updates in SDN, for instance, ez-Segway [3].
- Design and implement coflow property in the SDN network controller.
- Evaluate and compare the results with existing approaches [3, 2] for network updates in SDN.

This thesis topic is scalable and the workload can be adapted to fit bachelor or master thesis requirements.

Required knowledge

- Basic networking knowledge (and willing to learn)
- Recommended: Future Internet lecture
- Programming skills

Literatur

[1] M. Chowdhury and I. Stoica. Coflow: A networking abstraction for cluster applications. In *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, HotNets-XI, pages 31–36, New York, NY, USA, 2012. ACM.

[2] X. Jin, H. H. Liu, R. Gandhi, S. Kandula, R. Mahajan, M. Zhang, J. Rexford, and R. Wattenhofer. Dynamic scheduling of network updates. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 539–550, New York, NY, USA, 2014. ACM.

[3] T. D. Nguyen, M. Chiesa, and M. Canini. Decentralized consistent updates in sdn. In *Proceedings of the Symposium on SDN Research*, SOSR '17, pages 21–33, New York, NY, USA, 2017. ACM.

[4] M. Reitblatt, N. Foster, J. Rexford, C. Schlesinger, and D. Walker. Abstractions for network update. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '12, pages 323–334, New York, NY, USA, 2012. ACM.

Contact: Asif Hasnain, asif.hasnain@upb.de, Phone: 05251 / 60 1757, Room: O3.164