Extending SDN Control Through Programmable Mobile Devices

Konstantinos Poularakis[‡], Qiaofeng Qin[‡], Erich Nahum[◇], Miguel Rio[*], Leandros Tassiulas[‡]

[‡]Department of Electrical Engineering and Institute for Network Science, Yale University, USA

[◇]IBM T. J. Watson Research Center, USA

[*]Department of Electrical and Electronic Engineering, University College London, UK

***Abstract*—Today, Software Defined Network (SDN) architectures and applications are revolutionizing the way wired networks are built and operate. However, little is known about the potential of this disruptive technology in wireless mobile networks. In fact, SDN is based on a centralized network control principle, while existing mobile network protocols give emphasis on the distribution of network resources and their management. Therefore, it is challenging to apply SDN ideas in the context of mobile networks. In this paper, we propose methods to overcome these challenges and make SDN more suitable for the mobile environment. Our main idea is to combine centralized SDN and distributed control in a hybrid design that takes the best of the two paradigms; (i) global network view and control programmability of SDN and (ii) robustness of distributed protocols. We discuss the pros and cons of each method and highlight them in an SDN prototype implementation built using off-the-shelf mobile devices.**

I. Introduction

A. Motivation

Software Defined Network (SDN) technology promises to advance communication networks to a whole new level of programmability, which will allow to manage network resources on demand and at a granular level, and offer more flexible services to users. The main concept of SDN is to detach network control functions (control plane) from data forwarding devices (data plane) and shift them to a logically centralized entity with global network view, the controller. Up to date, the vast majority of related studies refer to wired infrastructures, i.e., ISP networks and data centers [1]. Little is known today about the potential of SDN in wireless mobile networks.

Mobile networks can benefit from SDN, in a similar way that this disruptive technology revolutionized data center and wired networks [2]. However, a major challenge that holds back the SDN penetration in mobile networks is their high level of volatility and the difficulty in establishing reliable communication between the control and data planes. For example, if the SDN controller loses connectivity to the mobile nodes, it will be impossible to reconfigure them, resulting in outdated (and clearly suboptimal) routing policies. Even if the controller is reachable, frequent network updates in the mobile network can take a long time and overload the controller.

The aforementioned problems are mainly because of the centralization of all the network control functions into the SDN controller. This centralized approach works well in wired networks which are relatively static and, therefore, the communication between the controller and the data plane nodes is much more stable and reliable than in the mobile counterpart. However, this approach needs to be revisited in the context of highly mobile networks.

At the same time, we note that distributed routing protocols, such as AODV and OLSR [3], have been shown to work well in mobile networks. These protocols can be used for network discovery and routing in presence of network failures and mobility, providing a robust network architecture. Their main limitation compared to SDN is their lack of network view and programmability to realize end-to-end network policies. All the above mean that *it would be beneficial to design a network architecture that combines the benefits of the two paradigms; global network view and programmability of SDN and robustness of distributed routing protocols.*

B. Methodology

Motivated by the above discussion, we revisit the strict separation between control and data planes dictated by SDN. We put forward a *hybrid control design* in which network control is split between the SDN controller and the (data plane) mobile devices. That is, we allow the mobile devices to make their own data forwarding decisions, bypassing the SDN controller.

At this point, we need to emphasize that the co-existence of different forwarding planes in the same network poses risks for fault-free routing, such as forwarding loops and blackholes [4]. Therefore, we need to combine SDN and distributed forwarding planes in a way that ensures that the packets will reach their destinations in reasonable time and without faults.

In this paper, we propose three alternative methods to combine centralized SDN and distributed network control planes.

1) Method 1: Migration to a distributed protocol.: In this method, a part of the mobile network nodes can decide to stop following the SDN forwarding rules and migrate to a distributed routing protocol. This can happen for example when the nodes in a certain region of the network experience rapid network changes and the SDN controller cannot reconfigure them on time. By running a distributed routing protocol, the nodes in that region can adapt to the changes faster than the SDN controller would do. However, deciding which part of the network to migrate and when, as well as the interaction between the two control planes are non-trivial problems.

2) Method 2: Clusters running distributed protocols.: In the second method, the mobile network is partitioned into clusters and a distributed protocol runs in each cluster independently from the others. These distributed protocols can route the traffic within the boundaries of a cluster, but they cannot make any decisions outside the cluster. The centralized SDN controller can be used to alter the behavior of the distributed protocols so as to control the traffic exchanged between nodes belonging to different clusters. However, how to design the clustering is an open question.

3) Method 3: Proactive distribution of backup rules.: In this method, all the nodes run the SDN protocol, and therefore they can be configured by the SDN controller. To realize a distributed control protocol, we proactively distribute backup forwarding rules to the mobile nodes which describe what actions to take and under what conditions. For example, if a certain network link fails the node on one of its endpoints can be advised to route packets using a different link. However, the storage space of the nodes is limited, and therefore it may not be possible to store backup rules for all possible link failure scenarios.

To realize the above hybrid control architectures, we need to install to each device a *local software agent* which can alter the device's forwarding behavior without relying on the SDN protocol. The local agents run mechanisms to detect changes in the network and new flows, and communicate with each other to exchange their observations of the network conditions. This way we can make the node to bypass the SDN controller and run a distributed protocol.

To highlight the benefits of our approach, we implement an SDN prototype of our architecture and execute experiments measuring the performance and limitations. Our prototype implementation consists of common smartphone and laptop devices which are set up with OpenVSwitch (OvS) OpenFlow datapath [5]. We perform the experiments for different network topologies and using multiple wireless interfaces (WiFi and Bluetooth). We find that by pushing a certain level of control logic to the mobile devices can provide a multi-fold reduction in failure reaction time compared to the conventional (fully-centralized) OpenFlow system where the controller responds to all failures. The gains highly depend on the quality of the wireless channel between the controller and the mobile devices.

The contributions of this work are summarized as follows:

- *Hybrid Control Design*. We propose hybrid SDN architectures that can provide the right balance between centralized and distributed network control in mobile edge networks. Our designs equip nodes with software agents that allow them to remain operational even when connectivity to the SDN controller is lost.
- *Summary of Challenges and Comparison*. We discuss the key challenges in realizing these architectures and compare them each other.
- *Proof-of-Concept Prototype Implementation.* We implement a hybrid SDN prototype and execute experiments measuring the performance and limitations. We find that the hybrid SDN system can provide a multi-fold reduction in failure reaction time compared to the conventional (fully-centralized) OpenFlow system for a range of experiments. The gains depend on the quality of the wireless channel between the controller and the mobile devices.

The rest of the paper is organized as follows.

References

[1] D. Kreutz, F. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, S. Uhlig, "Software-Defined Networking: A Comprehensive Survey", *in Proc. IEEE, vol. 103, no. 1, pp. 14-76*, 2015.

[2] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hlzle, S. Stuart, A. Vahdat, "B4: Experience with a Globally-Deployed Software Defined WAN", *in Proc. of ACM Sigcomm*, 2013.

[3] L. Abusalah, A. Khokhar, M. Guizani, "A Survey of Secure Mobile Ad Hoc Routing Protocols", *IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 78-93*, 2008.

[4] S. Vissicchio, L. Cittadini, O. Bonaventure, G. G. Xie, L. Vanberer, "On the Co-Existence of Distributed and Centralized Routing Control-Planes", *in Proc. IEEE Infocom*, 2015.

[5] Open vSwitch, http://openvswitch.org

[6] D. Syrivelis, G. Iosifidis, D. Delimpasis, K. Chounos, T. Korakis, L. Tassiulas, "Bits & Coins: Supporting Collaborative Consumption of Mobile Internet", *in Proc. of IEEE Infocom*, 2015.