論文の内容の要旨

論文題目：Exploring Benefits of Deep Dataplane Programmability Through In-Network Processing Use Cases

(ネットワーク処理ユースケースに基づくディープデータプレーンプログラマビリティの利点の探求)

氏名：ハミド　ファルハディイ

Software-Defined Networking (SDN) is a new networking paradigm that is getting more and more popular because it is considered as a prominent method to facilitate networking practices. SDN defines two core features, i) decomposition of control and data planes and ii) a centralized control plane (i.e., controller) based on a network-wide view. Those SDN characteristics provide control plane programmability, sup- port networking applications, and enable research and innovation in networking. While SDN has a strong proposal in control plane, it may have some shortcomings in data plane.

In this thesis, we recognize and address two major problems in the current SDN data plane:

1. Inefficient packet classification in the switch: SDN data plane uses a limited set of predefined packet headers (e.g., source and destination IP addresses) to classify packets for switching and repeats checking all header fields for classification on every switch in the network. Moreover, there are potentially many header fields than can be matched together to classify every packet that can make complex combinations. So, there are three subproblems in this problem: i) SDN classification is redundant ii) SDN classification is predefined and iii) SDN classification is complex.

2. Switch actions are hardwired in the switch hardware: After classifying packets SDN data plane applies a set of predefined actions on every packet (e.g., forward and drop). All these actions are hardwired directly on the switch hardware and network operators can not make any change or adapt new requirements. So there are two subproblems in this problem: i) SDN switch actions are predefined and ii) there are limited number of actions.

We posit that solving aforementioned problems in the SDN data plane is a key to the successful migration of current hardware-centric networking practices to SDN. The general benefits of such a migration lie in two aspects:

. For academia: We propose technologies that foster Data plane programmability along with an open data plane architecture to unlock innovation in the networking.

. For industry: We relax the constraints of traditional predefined, hardwired data planes which lets new and small firms enter the networking software development market and conduct the industry to a more competitive and innovative atmosphere.

Building on top of Deeply Programmable Networking (DPN) concept, which is already proposed in the community, we developed the following technologies as mitigation solutions to SDN problems. To address Problem 1 we propose TagFlow. TagFlow is a classification and forwarding architecture for SDN. The main contributions of TagFlow are two folds: first, using lightweight classification at the core it offloads the main classification load to the network edge. Second, we show TagFlow releases the classification load on core devices and using that opportunity it is possible to leverage from heavier classification mechanisms (e.g., application layer

classification) than common methods (i.e., header classification). As for Problem 2 we propose User-Defined Actions (UDA), a flexible architecture to accommodate user generated packet processing mechanisms within the switch data plane. UDAs let SDN programmers to freely build arbitrary use cases specific to their own needs without the limitation of traditional predefined actions in SDN. The main contributions of UDAs are two folds: first, through extensive experiments we indicate that our UDAs can elevate millisecond-scale running time of current proposals to nanosecond-scale (including proposals from northbound applications of SDN community and virtual appliances of Network Function Virtualization or NFV community). Second, to raise the importance of the ease of programmability when dealing with network programmability, we show that our proposal decrease the lines of code compared to implementing the same functionality as a northbound application and as a standalone middlebox.

The technical benefits of our proposals lie in two aspects:

• Primary benefit: We can adapt new architectures or protocols and basic packet processing without hardware replacement

• Secondary benefit: Reduced cost for networking devices and reduced latency in packet transmission over the network.

Aforementioned proposals are examples of application programming interfaces of the DPN. These interfaces are the mean to expose programmability deeply and comprehensively in data in addition to control plane (in the form of APIs) to realize deeply programmable networks.