Computer Communications xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

USA: Faster update for SDN-based internet of things sensory environments

Tao Liu[a], Chi Harold Liu[*,b,c], Wendong Wang[a], Xiangyang Gong[a], Xirong Que[a], Shiduan Cheng[a]

[a] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[b] School of Software, Beijing Institute of Technology, Beijing 100081, China
[c] Department of Computer Information and Security, Sejong University, 209 Neungdong-ro, Gunja-dong, Gwangjin-gu, Seoul, South Korea

ARTICLE INFO

Keywords:
Software defined network
Network management
Update
Forwarding policy
Internet of things

ABSTRACT

Internet of Things (IoT) require ubiquitous and seamless network connectivity. Meanwhile, it also asks for effective service management, data transmission and analysis. Towards this end, software defined networks (SDN) technology is introduced as a key solution to enable IoT network management. When IoT requires an efficient forwarding policy reconfiguration as the response to the alteration of system requirement (e.g., network environment or user demand), SDN is able to adjust the configurations easily to meet its demand. Previous research efforts tried to complete the reconfiguration as quickly as possible, since the update speed is critical to the performance of network management. However, update time can be further reduced. In this paper, we propose a novel update mechanism, which is referred to as rule co-existence update. It is able to enlarge the solution space and obtain more individual solutions, without any negative effect to the packet headers or throughput of flows. Then, we propose a scheme called USA (i.e., Update Speed Accelerator), as a novel algorithm for network updates based on the above-mentioned rule co-existence, that accelerates the update of forwarding policy with spare rule space on current network switches for general IoT environments. Based on the obtained suboptimal greedy solution, USA shows a novel way for rule co-existence to accelerate the update of forwarding policies. It also proposes a simplified algorithm to ensure the forwarding correctness during the update process. Extensive simulation results show that USA can clearly reduce the update time for nearly half of policies in our experiments with few rule space overhead, which is less than 8% of all policies, and maximum update time is less than previous works when spare rule space is limited.

1. Introduction

Internet of things (IoT), have attracted lots of interests from both academia and industry during the past a few years. As a networked system that is composed of massive amount of small things or objects, IoT can potentially generate significant impact on both the private and business applications, e.g., e-health, assisted living, industrial automation, logistic, and many more. Meanwhile, nearly all IoT applications ask for ubiquitous and seamless network connectivity to ensure their user performance, based on the rapid development of existing communications technologies, e.g., Zigbee, Bluetooth, Lora, etc. [1,2] Furthermore, cloud computing is also considered as a key enabler for IoT [3,4], that offers virtually managed resources and capabilities to the end users, which could supply effective service management and data analysis functionalities. Meanwhile, it can also expand the scope of IoT services into a number of real life scenarios with more detailed big data [5–9].

From this perspective, in order to satisfy the requirement for efficient IoT network management, software defined networks (SDN) is then introduced in as a key network architecture to supply flexible, automated and dynamic management. It could naturally handle the heterogeneous devices, supply differentiated services for different applications, provide an agile development of network functions and quickly respond to the demands of the applications or the alteration of underlying networks. However, in the SDN-based IoT scenario, there are still problems that need to be solved for energy saving [10,11], privacy preserving [12], management improving [13] and so on.

In this paper, we specifically focus on the problem of how to reconfigure the forwarding policy of the network devices for SDN-based IoT, while the network communications will not be interrupted during the reconfiguration process. The procedure of reconfiguration is referred to as the *update* in this paper. Given dynamic conditions of access networks and various requirements of IoT services, IoT must provide a rapid response to the changes of environment and demands of

* Corresponding author at: School of Software, Beijing Institute of Technology, Beijing 100081, China; and Department of Computer Information and Security, Sejong University, South Korea.
E-mail addresses: tony.leu777@gmail.com (T. Liu), chiliu@bit.edu.cn, liuchi02@gmail.com (C.H. Liu), wdwang@bupt.edu.cn (W. Wang), xygong@bupt.edu.cn (X. Gong), xrque@bupt.edu.cn (X. Que), chsd@bupt.edu.cn (S. Cheng).

https://doi.org/10.1016/j.comcom.2018.02.015
Received 15 May 2017; Received in revised form 24 January 2018; Accepted 18 February 2018

applications. Meanwhile, to adapt the trendy technique and unpredictable incident, IoT networks may be changed frequently. However, it is impossible for IoT to halt the data transmission process and reconfigure the devices along the path. Therefore, it is crucial to obtain a mechanism to keep the network functional when migrating the network configuration from an initial one to a target one.

The issue of updating forwarding policy in SDN was raised before, and there have been some related research efforts. However, previous works are based on static networks in a constant environment, e.g., campus network; but IoT networks are dynamic, and thus the update time become very critical. This is because that if the update is too slow, the network may be completely changed, and then the correctness of update procedure cannot be guaranteed, and temporary abnormalities may lead to large-scale service anomalies due to chain reaction. Besides, the speed of the forwarding policy update is also important, because it determines the agility of the control loop. For the above reasons, we focus on the update time for SDN enabled IoT in this paper.

Furthermore, reliability issues of IoT is also important, in addition to the update speed. Some IoT devices may be energy limited or in the hostile environment, and it is not recommended to retransmit the data. Meanwhile, due to the changing network environment, unnecessary data retransmissions would add a lot of extra burdens to the network controllers, and thus it is expected that packet loss should be zero during the update procedure.

In SDN-based IoT, packet forwarding is determined by per-flow rules which are applied by SDN-based IoT forwarding devices (also referred to as switch in this paper). The controller has to issue an update message to switches, to add, delete or modify some rules. To avoid the service disruption during the update, the forwarding correctness (e.g, black-hole-free and loop-free) have to be preserved. However, because of unpredictable factors of the network during the update, like the delay between controller and switches, and non-deterministic processing time for switch to install new rules, the update strategy must be carefully considered. The naive strategy, as issuing new rules from the controller to switches in a single round, may lead to an incorrect packet forwarding path.

To ensure the correctness of the update of the forwarding policy, part of previous research efforts [14] tried to install both initial rules and target rules on all corresponding switches based on the different tags, and forwarded the traffic exclusively according to initial rules or target rules. We refer to this approach as *two-phase update*. Its disadvantage is that it need double rule space (i.e., TCAM memory) to accommodate the update, while rule space is precious. Thus, update time would be prolonged [15] if remainder available rule space is insufficient for the flows to be updated, Furthermore, modifications to the packet header may lead to other problems: as [16] mentioned, extra tagging operation may be problematic in the presence of middleboxes which change headers, and, as [17] mentioned, this mechanism may experience temporary yet significant drops of throughput during the update. The other mechanisms, referred to as *rule replacement*, generate a carefully-computed update sequence of switches, while for each switch the initial rule is replaced by the target rule as an atomic operation, to preserve forwarding correctness during the update. With the approach, network update would not consume extra rule space, but delicate update sequence may also extend network update time, which may be several times more than two-phase update [18,19].

However, previous works do not take full advantage of the spare rule space of flow-table on network switches. Rule space is not always fully occupied in order to response to the burst of flows, the increasing delicacy management for special flows or other reasons [20]. Meanwhile, the size of the flow tables is growing at a pace aiming to meet the needs of future SDN deployment because of these efforts [21,22]. Therefore, spare rule space could be available, but it may be not abundant when considering the update of forwarding policy. In [23], two phase update is mixed with rule replacement update to obtain a feasible solution; however, extra header modifications and throughput reduction may be also introduced.

In this paper, we propose a mechanism that initial rule and target rule in some switches can co-existence with the specified ingress port and priority, which is referred to as *rule co-existence* in this paper. This mechanism would not modify the packet headers, thus it would not add complexity to the management plane, nor affect the throughput of the network. However, based on the rule co-existence in switches, more feasible solutions may arise than previous update mechanism for the correct update of a forwarding policy, while these solutions may be discarded by previous work, e.g., packets could traverse the same switch during update without any trouble. Meanwhile, considering that to meet the need of the changes, in most scenarios forwarding policies of multiple flows should be updated. Thus, the update time of the forwarding policies of multiple flows should be the time for all flows to complete the update. Although spare rule space on each switch could be available for the co-existence of rules, they are also limited and should be used delicately for the acceleration of forwarding policies. Therefore, to accelerate the update time of forwarding policies based on the rule co-existence, we also propose the algorithm to reduce the longest update time of forwarding policies within the limit of spare rule space, while keeping the forwarding correctness during the update procedure.

The contribution of this paper is three-fold:

- We propose a new update mechanism, the rule co-existence, which allows the initial rule and target rule to coexist in the flow-table on same switches in SDN-IIoT. This mechanism could supply more solutions than the rule replacement update mechanism, but would not introduce the negative effect of the two phase update mechanism.
- We propose an algorithm to accelerate the update of forwarding policies based on the rule co-existence, while keeping the update procedure correct. The algorithm could reduce the longest update time of forwarding policies based on current spare rule space.
- We perform simulations on real topology data sets and boot configurations as previous work does. Our simulation shows the benefits of our update mechanism, compared with previous update mechanisms.

The rest of this paper is organized as follows. Related work is reviewed in Section 2. In the following Section 3 we describe what is the rule co-existence update mechanism. The update problem is formulated in Section 4. The algorithm to accelerate the update time is discussed in Section 5. Last, we present the simulation results in Section 6. Finally, we discuss a practical issue in Section 7 and conclude the paper in Section 8.

2. Related work

Previous works achieve the update based on either (a) the two-phase update [14,15], or (b) the rule replacement [16–18]. The two-phase update mechanism installs both the tagged initial rules and tagged target rules on all forwarding devices (referred as switches below), and tags the packets on the ingress switch to ensure that packets are forwarded exclusively in either the initial policy or the target policy based on the disparate tags. The rule replacement schemes replace the initial rule with the target rule in each switch as an atomic operation, thus a proper update sequence for switches is essential. It is first proposed for IGP migration, and then used in SDN update; and thus it is plausible that it is also suitable for IoT scenarios. However, these two mechanisms are not efficient or flexible, because the spare rule space in flow tables are ignored.

The two-phase update [14] asks for double rule space at all corresponding switches in forwarding path because of the coexistence of the initial rule and the target rule, so that the update time depends on the minimal available rule space on all corresponding switches. When the minimal available rule space is more than flows which need to be updated, the update time is short, i.e., delivering all tagged initial rules