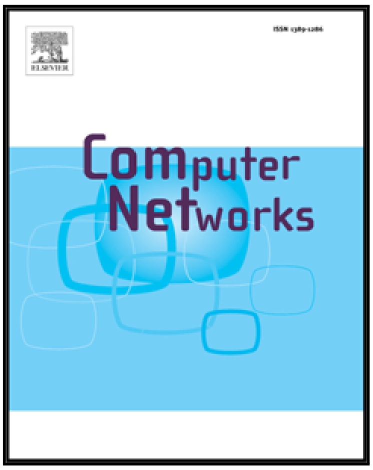
Accepted Manuscript



An Intelligent Rule Management Scheme for Software Defined Networking

Lei Wang, Qing Li, Richard Sinnott, Yong Jiang, Jianping Wu

PII: S1389-1286(18)30624-8
DOI: 10.1016/j.comnet.2018.07.027
Reference: COMPNW 6557

To appear in: *Computer Networks*

Received date: 9 December 2017
Revised date: 11 June 2018
Accepted date: 29 July 2018

Please cite this article as: Lei Wang, Qing Li, Richard Sinnott, Yong Jiang, Jianping Wu, An Intelligent Rule Management Scheme for Software Defined Networking, *Computer Networks* (2018), doi: 10.1016/j.comnet.2018.07.027

An Intelligent Rule Management Scheme for Software Defined Networking

Lei Wang[a], Qing Li[b], Richard Sinnott[c], Yong Jiang[a], Jianping Wu[d]

[a] *Graduate School at Shenzhen, Tsinghua University, Shenzhen, China*
[b] *Network Technology Laboratory, Huawei Technologies co.ltd, Hong Kong, China*
[c] *Computing and Information Systems Department, University of Melbourne, Australia*
[d] *Department of Computer Science and Technology, Tsinghua University, Beijing, China*

Abstract

Software Defined Networking (SDN) enables network innovation and brings flexibility by separation of the control and data planes and logically centralized control. However, this network paradigm complicates flow rule management. Current approaches generally install rules reactively after table misses or pre-installs them by flow prediction. Such approaches consume nontrivial network resources during interactions between the controller and switches (especially for maintaining consistency). In this paper, we explore an intelligent rule management scheme (IRMS), which extends the one-big-switch model and employs a hybrid rule management approach. To achieve this, we first transform all rules into path-based and node-based rules. Path-based rules are pre-installed whilst the paths for flows are selected at the edge switches of the network. To maintain consistency of forwarding paths, we update path-based rules as a whole and employ a lazy update policy. Node-based rules are optimally partitioned into disjoint chunks by an intelligent partition algorithm and organized hierarchically in the flow table. In this way, we significantly reduce the interaction cost between the control and data planes. This scheme enforces an efficient sliding window policy to enhance the hit rate for the installed chunks. We evaluate our scheme by comprehensive experiments. The results show that IRMS reduces the total flow entries by more than 59.9% on average and the update time by over 56%. IRMS also reduces the flow setup requests by more than one order of magnitude.

Keywords: `Software-Defined Networking, Rule Management, Rule Update, Cache`

1. Introduction

As an emerging networking paradigm, Software Defined Networking (SDN) [1] is widely influencing the evolution of network architectures. Separating the control plane from the data plane and centralizing the intelligence of the network into the controller(s) provides essential conveniences for network management and allows acceleration of network innovations [2, 3, 4], whereas this centralization introduces obstacles to flow rule management. Considering flexibility, the controller typically installs rules reactively when a new flow incurs a table miss. However, this flexibility sacrifices forwarding performance because frequent interactions between the control and data planes cause nontrivial resource consumption and communication latency to increase.

The state-of-the-art rule management schemes focus on caching more rules in the data plane to reduce the performance penalties for the table misses. For instance, CAB [5] splits the rule space into many non-overlapping buckets and treats the rules in a bucket as a whole for installation and updates. A big challenge for these approaches is the consistency of rules along with a forwarding path, since any inconsistency of the cached rules may require rule reinstallation or even cause the wrong packet behavior.

A more radical approach is installing the rules before flows occur. DIFANE [6] and CacheFlow [7] are representative solutions of these proactive schemes. They firstly divide the rule set into several subsets according to rule dependencies and switch capacity and then distribute them on the certain selected switches. However, such proactive schemes lose the ability to generate rules dynamically according to the evolving network states. The high cost for updating is also an obstacle in these schemes, since any modification of an individual match field or change of rule placement is likely to break the existing dependencies, and it will cause rule redistributing. Furthermore, installing all possible rules in advance imposes a heavy pressure on the flow tables of switches, since SDN switches usually store the rules in the ternary content addressable memory (TCAM), which is a scarce and expensive resource. Additionally, abundant match fields and fine-grained rules in SDN aggravate memory pressure.

In our paper, we propose an **I**ntelligent **R**ule **M**anagement **S**cheme (**IRMS**) that aims at providing a novel trade-off between flexibility and forwarding performance. We maintain intelligence at the network edge where interactions with the controller occur. All the core switches concentrate on forwarding tasks to achieve higher performance.

To achieve this, we classify flow rules as two types: *path-*