

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 16, 2014

L. Li
Bell Labs
Y. Luo
UML
H. Song
Huawei
Y. Yang
Yale
July 15, 2013

**Fine-Grained Control of Control-Plane Performance: Use Cases and
Mechanisms
draft-li-cp-usecases-00**

Abstract

It is commonly assumed that a system controller or network management system has complete knowledge of the data plane, especially in a software-defined network (SDN). That is, the controller knows performance metrics such as the flow table size of each switch, the rate of rule updates between a switch control plane and its data plane, and the maximum latency to install a rule in the flow table of a switch. However, in reality, this is not the case. Measurement studies show that the flow table size depends on the structure of the rules installed. The flow table size is much smaller if there are many wild card rules. The setup latency also depends on the already installed rules. If there are many wild card rules installed, the latency can be much higher. Currently, data centers pre-setup the rules long before the actual associated traffic starts to flow through the network. This puts constraints on the use cases. In this document, we first show that many use cases demand a more predictable control plane. The use cases are applied to networks that require control-plane performance information for dynamic configuration, as well as SDN networks. We then discuss potential mechanisms to enable fine-grained control of control-plane performance.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 16, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	4
2.	Use Cases	4
2.1.	Low-latency Path Setup	4
2.2.	Resource Partition in Networks	5
2.3.	Slice Resource Allocation in Flowvisor	5
3.	Mechanisms For Better Control-Plane Performance Management . .	5
3.1.	Control-Plane Resource Reservation: RSVP-CE	6
3.2.	Control-Plane Congestion Control: cTCP	6
4.	Normative References	6
	Authors' Addresses	6

1. Introduction

It is commonly assumed that the controller of a network knows control-plane performance metrics such as the flow table size of each switch, the maximum rate of rule updates between the switch control plane and its data plane, and the latency to install a rule in the flow table of a switch. However, recent measurement studies show that control-plane performance can be quite complex.

In this document, we list use cases where better understanding and control of control-plane performance can be beneficial. The use cases are applied to the current networks that require control-plane performance information for dynamic configuration, as well as SDN networks.

2. Use Cases

2.1. Low-latency Path Setup

Many applications need fast, predictable path setup to meet application requirements for normal path setup and to recover from link or node failures. Consider SDN in cellular networks, when a user equipment (UE) needs to communicate, the connection needs to be setup with low latency. If the tail latency is high, user experience can be seriously affected.

Similarly, in case of link or node failures, controller needs to react with low latency to reroute traffic. It needs to pick switches with low setup latency. There can be multiple paths from one node to another, and these paths may consist of different vendor equipment. Hence, they may have different reaction time to the same configuration. Even if they are from the same vendor, the configuration delay may be different according to the current state of the equipment (e.g., the number of already installed rules). The path selection can be a reaction to a link failure, which needs fast recovery. Then the controller needs to know the dynamic information about the delay to configure each specific switch or router, and then it can compare the setup latencies of the candidate paths, and select the one with minimal setup latency.

The requirements to setup an emergence communication path in hazard scenarios such as earthquake, flood or storm are similar. Setup latency is a key performance metric.

2.2. Resource Partition in Networks

When a network provider tries to allocate resources for multiple users (e.g., end users or multiple tenants in a data center), it needs to consider not only data-plane resource (e.g., data-plane bandwidth) partition but also control-plane partition.

Specifically, consider that existing routers and switches implement QoS capabilities such as destination/source oriented rate limit, queue management, and peak burst size. A key component to implement these QoS capabilities is ACL rules, and ACLs consume hardware resources. Our measurements show that the resource consumption of ACL rules depends on the structure of the rules. For example, we found that in one vendor's devices, the amount of resources consumed by an ACL rule depends on the range of VLAN ID specified in the rule, where a large range consumes a small amount of resources, while another seemingly small range can consume substantially more resources.

A simple controller will guide its ACL usage according to a device's manual, which specifies a fixed number of rules that a networking device can handle concurrently. However, different ACL policies/templates consume different amounts of resources such as TCAMs, and a controller uses device manual may either under utilize the resources or encounter unexpected resource exhaustion.

One way to address the preceding problem is to introduce measurement capabilities into the control plane of network devices. For example, the controller measures and models control-plane resource consumption, and then computes whether the next group of QoS templates can be fulfilled or not. The controller can also perform load balancing, by assigning different QoS tasks to different network devices. This improves control-plane resource utilization and hence overall system utilization.

2.3. Slice Resource Allocation in Flowvisor

As a special case of the preceding use case, a virtualized network may have a hypervisor such as Flowvisor to allocate resources across slices. The Flowvisor needs to understand the maximum rule update rates and rate limit slice controllers. The Flowvisor also needs to know the table size under different sets of rules to enforce the limit.

3. Mechanisms For Better Control-Plane Performance Management

We now list a few mechanisms that may enable more efficient control-

plane management.

3.1. Control-Plane Resource Reservation: RSVP-CE

Similar to RSVP, which reserves data plane resources, RSVP-CE reserves/preinstalls control-plane resources. It is important that one decouples the resource reservations, for example, by reserving or preinstalling control-plane resources without the data-plane resources.

3.2. Control-Plane Congestion Control: cTCP

Similar to congestion control in the data plane, one can introduce control-plane congestion control. Consider the case without a congestion control protocol. A controller may send a burst of rule updates in response to application demands or link failures. This may lead to large delays and blocking of control-plane updates. As a contrast, control-plane congestion control will adjust to control-plane update rate without building a large queue in switch control planes.

Just as there are black-box based inference (e.g., different versions of TCP based on losses or delays) or more explicit feedback (e.g., ECN) in data-plane congestion control, control-plane congestion control may also allow multiple design flavors, for example "loss" based (i.e., rejection) or better feedback or model based.

4. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

Authors' Addresses

Li Erran Li
Bell Labs

Email: erranlli@gmail.com

Yan Luo
UML

Email: yanluo.uml@gmail.com

Haibin Song
Huawei

Email: haibin.song@huawei.com

Y. Richard Yang
Yale

Email: yry@cs.yale.edu