

Forwarding and Control Element Separation
Internet Draft
Intended status: Informational
Expires: November 2014

L. Zeng
<Tsinghua University>
May 7, 2014

Consistent Control Mechanism in Software Defined Network
draft-ietf-forces-consistent-control-00.txt

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on Fail 7, 0000.

Copyright Notice

Copyright (c) 2014 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.

Abstract

This document introduces a consistent control mechanism in the framework of Software Defined Network (SDN), which is one method to achieve forwarding and control element separation. In detail, this mechanism uses a centralized control element to control multiple forwarding elements.

Table of Contents

- [1. Introduction](#) [2](#)
- [2. Conventions used in this document](#)..... [2](#)
- [3. Software Defined Network Framework](#)..... [3](#)
- [4. Control Problem in SDN Framework](#)..... [4](#)
- [5. Consistent Flow Control Mechanism](#)..... [5](#)
- [6. Security Considerations](#)..... [6](#)
- [7. IANA Considerations](#) [6](#)
- [8. Conclusions](#) [6](#)
- [9. References](#) [6](#)
 - [9.1. Normative References](#)..... [6](#)
 - [9.2. Informative References](#)..... [6](#)
- [10. Acknowledgments](#) [6](#)

[1. Introduction](#)

Software Defined Network (SDN) is proposed in recent years, and is considered as a promising way to separate forwarding plane and control plane [[FORCES-SDN](#)]. In detail, SDN is an approach to networking in which control is decoupled from hardware and given to a software application.

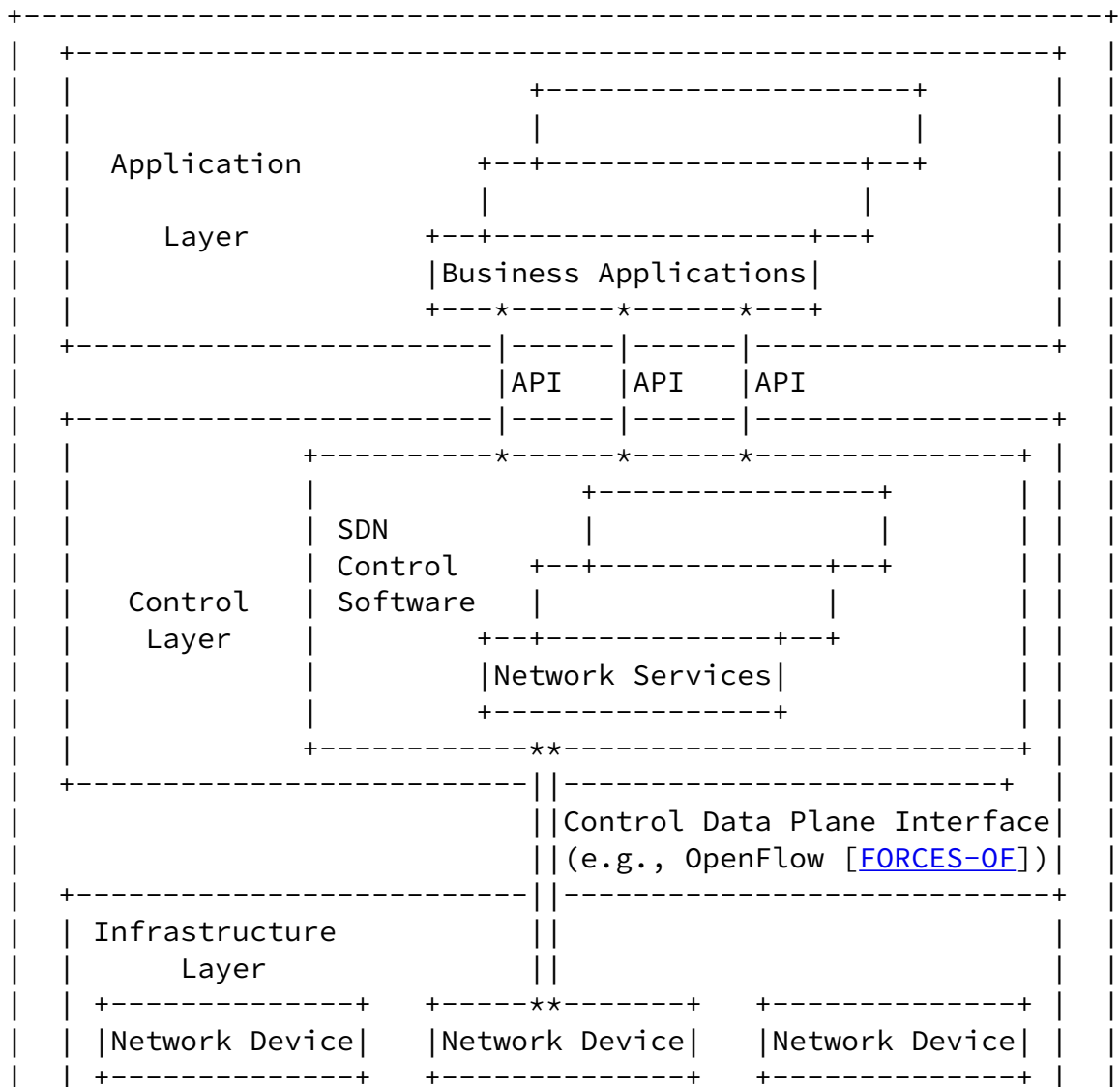
[2. Conventions used in this document](#)

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](#)].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC-2119](#) significance.

3. Software Defined Network Framework

A logical view of the SDN architecture is shown in Figure 1, which consists of three layers: infrastructure layer, control layer and application layer.



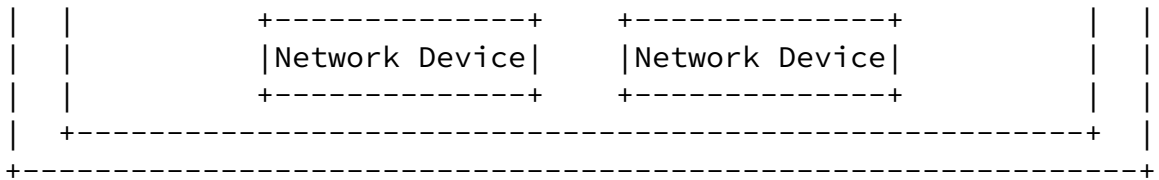


Figure 1 Software-Defined Network Architecture

In infrastructure layer, network devices are only in charge of executing the forwarding functions. Network control intelligence is logically centralized in control layer. In particular, a centralized SDN based controller is in charge of controlling function. In the controller, different network control functions can be developed as customized. As to application layer, different kinds of business applications are deployed. Between application layer and control layer, a set of APIs (Application Programming Interfaces) are designed, which allows business applications to use network control services in control layer. Also, control data plane interface is designed between control layer and infrastructure layer, which is used to interchange control and forwarding information between the controller and network devices.

In the SDN architecture, the controller uses flow entry to control multiple network services, where the forwarding function is executed. In each network service, there exists a flow table to store flow entries sent by the controller. The controller can add/delete/modify flow entries to each network service.

[4. Control Problem in SDN Framework](#)

In SDN framework, the controller uses flow entries to control forwarding behavior of different network devices. In particular, there are special security channel between the controller and network devices to transform flow entry information.

Since multiple network devices make up a distributed system, control problem exists in SDN framework. In detail, it is difficult for the controller to update multiple flow entries simultaneously, due to different latency of different special security channels. If these flow entries are written into network devices at different time, data packets may follow the wrong control instruction and be incorrectly

deal with, leading to system chaos, packets loss, service deteriorate, and etc.

Due to this control problem, it is necessary to study consistent flow control mechanism for SDN framework. The consistent flow control problem is defined as follows: when the controller updates flow table in multiple network devices, each data packet flowing through the network must be processed according to a single network control configuration, either the old control configuration or the new control configuration, but not a mixture of both configurations, or other uncertain rules.

5. Consistent Flow Control Mechanism

A consistent flow control mechanism is proposed in this section. In an update event, the controller updates flow entries from an old flow configuration to a new flow configuration in a set of network devices.

First, the controller divides these network devices into two parts.

- 1) Entry Network Device (END): the first network device in the set of network devices, in which data packet is handled with the now flow configuration.
- 2) Other Network Device (OND): other network devices in the set of network devices, except for the END.

Then, the controller divides flow entries in both configurations into four parts:

- 1) New Flow Entry (NFE): new flow entry only in new configuration.
- 2) Shared Flow Entry (SFE): flow entry existing in both configuration.
- 3) Deleted Flow Entry (DFE): old flow entry only in old configuration.
- 4) Modified Flow Entry (MFE): different control behavior of the same data packet in two configurations.

The important steps of proposed consistent flow control mechanism are

introduced as follows:

Step 1: the controller analyzes network devices and two sets of flow configurations, which are respectively divided into several parts.

Step 2: In END, the controller uploads all data packets influenced by this update event, except for the packets controlled by SFE.

Step 3: The controller writes NFE in OND, and then waits for an end-to-end network latency.

Step 4: The controller finishes all updates in OND, including adding MFE and deleting DFE.

Step 5: The controller finishes all updates in END, and stops uploading data packets from EDN.

6. Security Considerations

This requirements document does not raise in itself any specific security issues.

7. IANA Considerations

IANA does not need to take any action for this draft.

8. Conclusions

This document provides a consistent control mechanism in the framework of Software Defined Network (SDN). In detail, this mechanism uses a centralized control element to control multiple forwarding elements.

9. References

9.1. Normative References

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

9.2. Informative References

- [FORCES-SDN] McKeown N. "Software-defined networking", INFOCOM keynote talk, 2009.
- [FORCES-OF] McKeown N, Anderson T, Balakrishnan H, et al. "OpenFlow: enabling innovation in campus networks", ACM SIGCOMM Computer Communication Review, 2008, 38(2), pp. 69-74.

10. Acknowledgments

This work is supported by Chinese National Major Scientific and Technological Specialized Project (No.~2013ZX03002001), National Basic Research Program of China (973 Program Grant No.~2013CB329105), China's Next Generation Internet (No.~CNGI-12-03-007), and ZTE Corporation.

This document was prepared using 2-Word-v2.0.template.dot.

<Zeng>

Expires November 7, 2014

[Page 6]

Consistent Control Mechanism in Software Defined Network

May 2014

Authors' Addresses

Lieguang Zeng
Department of Electronic Engineering, Tsinghua University
Department of Electronic Engineering, Tsinghua University, Beijing,
China

Email: zenglg@mail.tsinghua.edu.cn

Ye Zhou
Department of Electronic Engineering, Tsinghua University
Department of Electronic Engineering, Tsinghua University, Beijing,
China

Email: yetiero@gmail.com

Mao Yang
Department of Electronic Engineering, Tsinghua University

Department of Electronic Engineering, Tsinghua University, Beijing,
China

Email: yangmao210@163.com

Yong Li

Department of Electronic Engineering, Tsinghua University
Department of Electronic Engineering, Tsinghua University, Beijing,
China

Email: liyong07@tsinghua.edu.cn

Depeng Jin

Department of Electronic Engineering, Tsinghua University
Department of Electronic Engineering, Tsinghua University, Beijing,
China

Email: jindp@mail.tsinghua.edu.cn

Li Su

Department of Electronic Engineering, Tsinghua University
Department of Electronic Engineering, Tsinghua University, Beijing,
China

Email: lisu@tsinghua.edu.cn