Isolated In-Band Communication for Distributed SDN Controllers

Rhaban Hark, Amr Rizk, Nils Richerzhagen, Björn Richerzhagen, and Ralf Steinmetz
Multimedia Communications Lab, Technische Universität Darmstadt, Germany
{rhaban.hark|amr.rizk|nils.richerzhagen|bjoern.richerzhagen|ralf.steinmetz}@kom.tu-darmstadt.de

Abstract— **Control planes of Software-defined Networks consist of multiple interconnected controllers to provide scalability and reliability. Existing approaches for distributed control planes add high complexity and neglect in-band connectivity issues. Subsequently, in this work, we propose a low-cost inter-controller communication service, which *(i)* requires only few resources in terms of state, communication overhead, and computational complexity and *(ii)* provides isolation between control and production traffic using shared in-band channels.**

I. Introduction

Software-defined Networking (SDN), which is the de-facto upcoming network core technology, allows simple and flexible network management, where a centralized control entity holds the complete view on the network it manages. Despite the vision of logically centralized control, there is a broad consensus on the need for physically distributed controllers to achieve scalability and reliability. So far, the research community proposed many approaches to distribute control plane [4]. However, despite entailing a high overhead including considerable infrastructure elements such as databases or message brokers (e.g. [5]), related works rarely focus on the design of data-plane inter-controller communication channels. Subsequently, in this work we propose a lightweight inter-controller communication mechanism that is viable when controllers are not connected explicitly as depicted in Figure 1. We target a negligible resource consumption in terms of state, communication overhead and computational complexity. To achieve this goal we overcome multiple challenges that arise due to the lack of a physical out-of-band interconnection of the controllers such as dynamic controller discovery and control traffic isolation. We design the inter-controller communication mechanism as controller service, which is easily added to existing controllers. Operators can built other elaborated distributed mechanisms, e.g. state sharing or distributed monitoring (cf. [2]), simply as an extension on-top of the proposed service.

In particular, the contributions of the work are twofold: *(i)* The service discovers in-band paths to other controllers including a mechanism to select the best path considering the network operator's traffic engineering preferences. *(ii)* The service establishes isolated communication channels between the controllers once it finds a path. Here, the service enables sending bilateral messages between controllers.

Two key conditions define the need for a service as we propose it. Primarily, if there is no physical out-of-band communication possibility for the controllers and, secondly, the controllers conduct only simple communication not including complex state sharing. We show two exemplary use cases for such scenarios.

a) Use Case I: Logically Adjacent - Physically Distributed: Internet networks that belong to the same entity might be logically adjacent to each other. However, real world AS networks may consist of different geolocations of federated AS subnetworks [6]. Networks, which are logically adjacent but separated through foreign networks, require control channels that are physically in-band but logically isolated to provide management functionality.

b) Use Case II: Multi-Tenancy Virtual Network Infrastructures: Virtual Infrastructures can be used to set up virtual SDNs [1]. Although a tenant has the full control of the virtual entities, no dedicated physical control network is available. In such scenarios, an in-band control channel is of choice.

II. In-Band Communication Service

In this section, we show how controllers discover other controllers, how the service establishes and isolates the discovered control channel, and how the service leverages *One-Hot-Coding* to reduce the allocated TCAM space.

a) Controller Discovery: The proposed discovery mechanism relies on *Port-Changed* events generated in the data plane switches and forwarded to the individual controllers to inform about connected (or disconnected) links. When a new port is connected, the service dispatches a discovery message through this port containing *(i)* the controller's ID, *(ii)* an initial cost value and *(iii)* a digital signature of the source controller as payload. The cost value may include information such as the number of hops, used to optimize the paths between controllers. Discovery packets are marked with a reserved Ethernet VLAN ID, say `1001`. SDN rules on every switch in the network direct messages with this VLAN ID to the controller. Thus, the switch forwards the dispatched discovery packet to the corresponding controller, which manage the linked switch when the message traverses a newly connected link. If the same controllers controls the switch on the other side of the link, it discards the message. If another controllers (target controller) controls the switch, the target controller processes the discovery request as follows. We assume that every controller can determine whether a

This work is funded within the BMBF SENDATE-PLANETS Project (ID 16KIS0471) and the DFG CRC 1053 – MAKI B1, B4, C2, and C3.

ISBN 978-3-901882-94-4 © 2017 IFIP

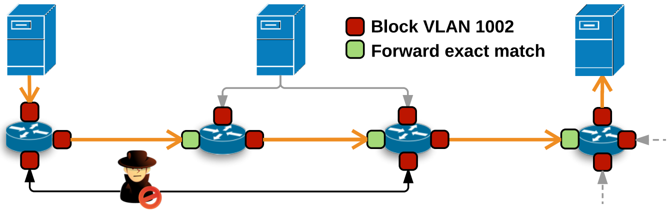

Fig. 1: Distributed control plane using in-band communication channels between controllers. Thick orange lines indicate an established control path.

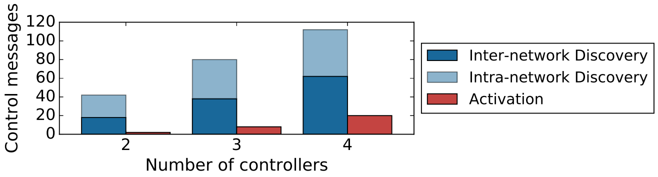

Fig. 2: The number of control messages for control path discovery and activation is linearly dependent on the number of controllers (tree topology).

discovery's origin is trustworthy based on the signature using public key mechanisms not detailed here. The target controller stores the peering-point as *(switch, port, costs)* tuple in a list for discovered controllers. The new connection is compared to other connections to the source controller (if any) such that in case the new connection has the lowest cost, the target controller induces two further steps: *(i)* forwarding the discovery with cumulated costs to other, already discovered, controllers and *(ii)* activating the path. Cumulative cost are intuitive for additive metrics such as latency or hop counts as well as min metrics such as link bandwidths. All controllers on the discovery's path handle the message as described leading to a recursive forwarding of discoveries as long as this is the best or first connection.

A controller activates a new path by sending an activation message using the reserved VLAN as reply to a discovery message. The next controller on the reverse path receives the activation message and is either an intermediate network controller or the targeted controller (source of the preceding discovery). In the first case, the intermediate controller installs a rule in every switch on the path between the ingress switch and egress switch in its network matching on a reserved VLAN for inter-controller communication, here `1002`, and the source and target controllers coded in the packet header as described later. In the other case, the destination controller simply installs a forwarding rule for inter-controller packets in the ingress switch. If a controller finds an activation packet as answer to a dispatched discovery it stores the established connection in its connection list and, if best or first, activates this path as well. Thereafter, the service established bidirectional communication channels to all controllers reachable over the newly connected link. Figure 2 depicts that the number of messages for discovery and path activation grows linearly with the number of controllers involved in the system. For this preliminary evaluation, we set up a tree-based controller overlay topology with a fan-out of {1,2,3}.

Whenever a connection to another controller is lost, the module activates the next best peering-point from the list of discovered connections or, if empty, dispatches discovery messages through all ports.

b) Isolated Communication Channels: Next, we describe how the control traffic is isolated from production traffic. Before the service discovers other controllers, it installs in each switch a high-priority rule to drop all packets with VLAN `1002`. When a controller installs a rule for control traffic, it uses, in addition to the destination and the VLAN ID `1002`, the input ports among the path in every switch to match on. As depicted in Figure 1 only the ingress ports on the control path from the leftmost controller (marked in green) allow control messages destined to the rightmost controller. The first switch will discard all control messages coming from any unexpected source, so that they cannot enter the network. This way, no unauthorized entity can reach a controller with a control message.

c) State Cost Reduction Using One-Hot-Coding: To save expensive TCAM space we combine multiple rules matching on the same input port and forwarding to the same output port as common in intermediate networks. For this, each controller is identified with exactly one bit of a bit string using *One-Hot-Coding*, which we store in the respective source and destination MAC address fields, respectively. A switch can easily match multiple IDs in one rule using wildcards.

III. Conclusion

In this work, we introduce a lightweight communication system for controllers of a distributed SDN control plane without a dedicated control network. It includes an approach to discover paths to other controllers while carrying compound link costs in terms of the operators' preferred metric to enable path optimization. Furthermore, we attain a strict isolation of the control traffic from the production traffic using VLAN tags and port-based matching. We finally introduce an additional mechanism to save rule costs using *One-Hot-Coding* rule aggregation. In a concise evaluation, not shown in detail in this extended abstract, we provide a proof-of-concept and show the linear dependence of the number of messages for discovery on the number of involved controllers. A prototypical implementation of this service is available on [3].

References

[1] D. Dietrich, A. Rizk, and P. Papadimitriou, "Multi-Provider Virtual Network Embedding With Limited Information Disclosure," *IEEE TNSM*, vol. 12, no. 2, pp. 188–201, 2015.

[2] R. Hark, D. Stingl, N. Richerzhagen, K. Nahrstedt, and R. Steinmetz, "DistTM: Collaborative Traffic Matrix Estimation in Distributed SDN Control Planes," in *IFIP Networking*, 2016, pp. 82–90.

[3] R. Hark. GitHub: Floodlight In-band SDN Control Communication Module. Latest access: March 2017. [Online]. Available: https://github.com/rhaban/In-band-SDN-Control-Communication

[4] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[5] K. Phemius, M. Bouet, and J. Leguay, "DISCO: Distributed Multi-domain SDN Controllers," in *IEEE NOMS*, 2014, pp. 1–4.

[6] M. Roughan, W. Willinger, O. Maennel, D. Perouli, and R. Bush, "10 Lessons from 10 Years of Measuring and Modeling the Internet's Autonomous Systems," *IEEE J-SAC*, vol. 29, no. 9, pp. 1810–1821, 2011.