Poster: Enabling Fast Forwarding in Hybrid Software-Defined Networks

Yijun Sun*, Zehua Guo*, Songshi Dou*, Junjie Zhang†, Changlin Li*, Xiang Ouyang*
*Beijing Institute of Technology, †Fortinet

***Abstract*—Emerging Software-Defined Networking (SDN) technique brings new opportunities to improve network performance. Some SDN-enabled programmable switches are deployed in legacy networks, and thus legacy and programmable switches could coexist, generating hybrid SDNs. In this paper, we study the node upgrade for layer-2 hybrid SDN and propose Shortcutter to accelerate the transmission. Preliminary results show that the proposed Shortcutter can reduce the forwarding path's length 7% on average, compared with baseline solutions.**

I. Introduction

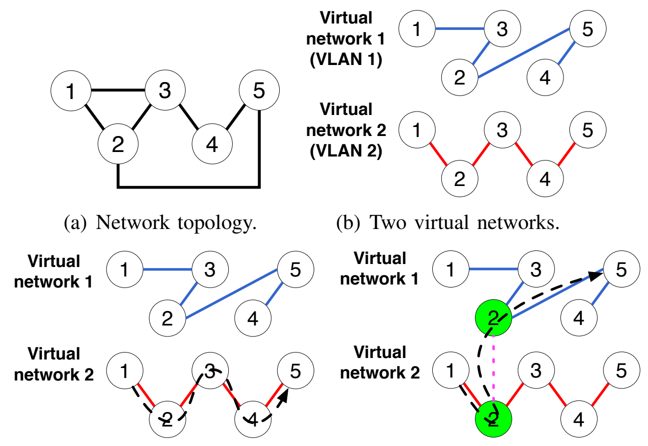
Software-Defined Networking (SDN) introduces network programmability to accelerate network innovation by flexibly routing and rerouting flows. Due to budget and operational considerations, deploying SDN-enabled programmable switches into legacy networks is a gradual process, and only a set of selected legacy devices (e.g., switches/routers) are upgraded to programmable switches at each time of network upgrade. The coexistence of legacy and SDN-enabled programmable switches generates the hybrid SDN.

One key issue for hybrid SDN is to maximize network programmability from SDN by upgrading a given number of legacy devices to SDN devices. Many works study device upgrade for layer-3 networks [1], but few works focus on layer-2 networks. Typically, layer-2 network is used for local area networks (e.g., enterprise and campus) and virtualizes a physical network topology to multiple logically isolated virtual networks. Each virtual network is identified by a unique Virtual Local Area Network (VLAN) ID and connects network nodes via a unique spanning tree. Each flow follows its encapsulated VLAN ID to be forwarded in a virtual network.

Spanning tree coupled with VLAN ID enables to isolate flow forwarding into different virtual networks but increases the length of forwarding path and thus forwarding delay. SDN brings new opportunities to reduce forwarding delay. A programmable switch can enable packets to be forwarded across different virtual networks by changing a packet's VLAN ID. Hence, with reasonable configuration, a flow can be forwarded on a short path which is composed of different path from different virtual networks. Figure 1 uses an example to illustrate the benefit of upgrading a legacy device to a programmable switch. When node 2 is upgraded to a programmable switch, and flows can go across two virtual networks via this node. We use a purple dot line to denote a virtual bridge between virtual networks 1 and 2. Thus, when flow f's packets arrive at node 2, these packets' VLAN ID can be changed from VLAN 1 to VLAN 2 at node 2 and then forwarded to virtual network 1. With VLAN 2, these packets

The corresponding author is Zehua Guo.

978-1-6654-4131-5/21/$31.00 ©2021 IEEE



(a) Network topology. (b) Two virtual networks.
(c) Before node upgrade. (d) After node upgrade.

Figure 1: An example that illustrates the benefit of node upgrade in a layer-2 network. Flow f is from node 1 to node 5 in virtual network 2, and its original forwarding path is $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ in virtual network 2. When node 2 is upgraded to a programmable switch (denoted by green), f's packets are forwarded across virtual networks 1 and 2 via the virtual bridge (denoted by purple dote line) at node 2 by changing their VLAN ID. Thus, f's forwarding path changes to $1 \rightarrow 2 \rightarrow 5$.

can be forwarded to node 5 via the shortest path to node 5 (i.e., $2 \rightarrow 5$) in virtual network 1. Thus, flow f's forwarding path changes from $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ to $1 \rightarrow 2 \rightarrow 5$ and reduces 50% of the original path's length.

In this paper, we study the node upgrade problem for layer-2 hybrid SDN. The main challenge of this problem comes from the difficulty of modeling the benefit of upgrading multiple devices since the sequence of node upgrade matters. We propose Shortcutter to solve the problem. Preliminary results show that the proposed Shortcutter can reduce the forwarding path's length 7% on average, compared with baseline solutions.

II. Problem Analysis and Solution

Given a layer-2 network V with N nodes and L virtual networks, each of which is a unique minimum spanning trees of the N nodes. The i-th virtual network is denoted as V_i ($i \in [1, L]$). In V_i, a flow is denoted by a source-destination node pair and forwarded in its shortest path of V_i. The node upgrade benefit comes from forwarding flows cross virtual networks. Upgrading node n_j ($j \in [1, N]$) in V can be viewed as establishing a virtual link to connect n_j among all virtual networks. As a result, many flows from different virtual networks traverse n_j to reduce their path length, and the number of flows passing through the node increases. Before node upgrade, in V_i, a flow's default path is the shortest path of V_i. After node upgrade, the flow's path is the shortest path among all available forwarding paths across different virtual networks. The upgrade benefit of node n_j is defined as the sum of the reduced length of all flows through n_j.

However, the node upgrade problem is difficult to model since we cannot specifically formulate the impact of consecutively upgrading multiple nodes on reducing the path length of all flows. Additionally, a real layer-2 network can consist of thousands of nodes and hundreds of virtual networks. It is hard to get the exact optimal solution because searching the optimal solution in a large solution space in a brute force fashion is very time-consuming. Thus, a feasible solution is to design a heuristic solution. The naive heuristic solution is to calculate the average hop number using Freud algorithm [2] with the time complexity of O(N^3). In this way, all nodes need to be enumerated every time, and the complexity of the greedy algorithm becomes O(N^4). However, upgrading a node only affects the flows that pass through it, and many unrelated cases are uselessly and repeatedly calculated using Freudian algorithm. Thus, Shortcutter only considers flows passing through the upgraded nodes and uses dynamic programming to get the upgrade benefit under the complexity of O(N^2). After enumerating all nodes, the total complexity of Shortcutter reduces to O(N^3). For the case of thousands of nodes in the layer-2 network, Shortcutter is much faster than existing shortest path algorithms (e.g., Dijkstra and Freud).

III. Preliminary simulation

A. Comparison algorithms

We compare Shortcutter with three centrality-based algorithms.

Degree-centric [3]: it sequentially upgrades nodes following the decreasing order of these nodes' degrees.

Betweenness-centric [4]: it sequentially upgrades nodes following the decreasing order of these nodes' betweenness centrality.

Closeness-centric [5]: it sequentially upgrades nodes following the decreasing order of these nodes' closeness centrality.

Shortcutter: it sequentially upgrades nodes following the decreasing order of these nodes' upgrade benefit.

B. Simulation setup

We follow the powerlaw degree distribution and approximate average clustering [6] to generate the virtual network, each of which consists of 30 nodes. We use five virtual networks to form one physical network by merging nodes that share the same index in different virtual networks together. In each virtual network, each pair of source and destination nodes generates a flow. The node upgrade benefit is measured by the sum of reduced length of all flows from all virtual networks.

C. Simulation results

Figure 2 presents the cumulative probability distribution of node upgrade benefit after upgrading 5 nodes among 50 layer-2 networks. In the figure, Shortcutter reduces the total path length by 7% on average and up to 13% when cumulative probability equals 0.3, compared with the best performance of three comparison algorithms. Figure 3 presents the result when upgrading 1-7 nodes on a layer-2 network, and it shows the relationship between the number of upgraded nodes and the upgrade benefit. As the number of upgraded nodes increases, the node upgrade benefit increases, but the trend tends to be slow. However, Shortcutter always outperforms the three comparison algorithms.

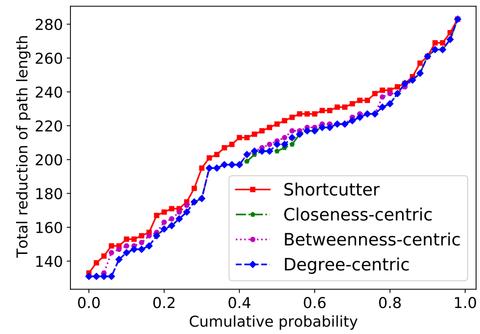


Figure 2: Simulation results of upgrading 5 nodes.

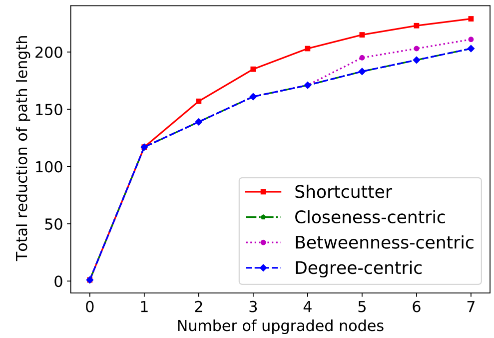


Figure 3: Simulation results of upgrading 1-7 nodes.

IV. Conclusion and future work

In this paper, we propose Shortcutter to upgrade nodes for the layer-2 hybrid SDN. Preliminary results show that Shortcutter outperforms centrality-based algorithms. In the future, we will try to formally formulate the node upgrade problem for the layer-2 hybrid SDN and present efficient solutions.

Acknowledgments

This work was supported by the Key-Area Research and Development Program of Guangdong Province under Grant 2021B0101400001, the National Natural Science Foundation of China under Grant 62002019, and the Beijing Institute of Technology Research Fund Program for Young Scholars.

References

[1] Y. Sinha, K. Haribabu *et al.*, "A survey: Hybrid sdn," *Journal of Network and Computer Applications*, vol. 100, pp. 35–55, 2017.

[2] S. Hougardy, "The floydwarshall algorithm on graphs with negative cycles," *Information Processing Letters*, vol. 110, no. 8, pp. 279–281, 2010.

[3] R. Pastor-Satorras and A. Vespignani, "Epidemic spreading in scale-free networks," *Physical review letters*, vol. 86, no. 14, p. 3200, 2001.

[4] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[5] A. Bavelas, "Communication patterns in task-oriented groups," *JASA*, vol. 22, no. 6, pp. 725–730, 1950.

[6] P. Holme and B. J. Kim, "Growing scale-free networks with tunable clustering," *Physical review E*, vol. 65, no. 2, p. 026107, 2002.