QoS-Aware Flow Routing with Minimizing Active Links and Rule Capacity Constraints in SDN Networks

Priyanka Kamboj, Sujata Pal

Indian Institute of Technology, Ropar

2018csz0003@iitrpr.ac.in, sujata@iitrpr.ac.in

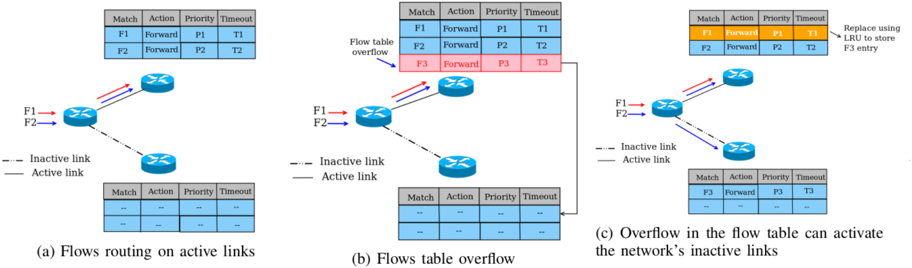
Outline

- Introduction
- Contribution
- System Architecture
- Proposed Approach
- Results and Discussion

Introduction

- SDN provides abstraction by separation of control plane from the data plane.
- Each networking device saves flow rules (or entries) in its ternary content addressable memory (TCAM) to forward the packets.
- The TCAM memory is expensive hardware with limited size and high power consumption.
- Limited size of TCAM memory inspires us to think about the flow rule replacement in SDN switches.

Illustration example



Match	Action	Priority	Timeout
F1	Forward	P1	T1
F2	Forward	P2	T2

Match	Action	Priority	Timeout
--	--	--	--
--	--	--	--

(a) Flows routing on active links

Match	Action	Priority	Timeout
F1	Forward	P1	T1
F2	Forward	P2	T2
F3	Forward	P3	T3

Match	Action	Priority	Timeout
--	--	--	--
--	--	--	--

(b) Flows table overflow

Match	Action	Priority	Timeout
F1	Forward	P1	T1
F2	Forward	P2	T2

Match	Action	Priority	Timeout
F3	Forward	P3	T3
--	--	--	--

(c) Overflow in the flow table can activate the network's inactive links

Figure: Illustration example: a) Traffic flows based on priority are routed on active links denoted by dark lines; b) Flow table overflow occurs if incoming flows exceed the switch's rule capacity; and c) Flow table overflow can activate the inactive links

Contributions

- We design a minimum cost routing problem to considers the QoS requirements of traffic flows. Further, we propose a greedy-based heuristic solution to minimize the network's active links as an ILP in SDN networks.
- We introduce a flow rule replacement scheme to prevent the flow table overflow in the SDN switches.
- We also analyze the impact of idle timeout on flow table entries on the proposed approach and the benchmark methods.

Problem Formulation

- Consider a network in the form of a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ where $\mathcal{V}$ is the set of switches and $\mathcal{L}$ denotes the set of links in the network.
- The set of paths P_f for a flow $f \in \mathcal{F}$ is described as:

$$P_f = \left\{p_f^1, p_f^2, ..., p_f^n\right\}, \ \ \forall f \in \mathcal{F}. \tag{1}$$

$$\alpha_{m,n}^{f} := \begin{cases} 1, & \textit{if flow } f \textit{ is routed on link } (m,n) \in \mathcal{L} \\ 0, & \textit{otherwise} \end{cases} \tag{2}$$

$$R_f(m) = \sum_{n \in \mathcal{N}(m)} \alpha_{m,n}^{f}, \ \ \forall \ m \ \in \mathcal{V}, \tag{3a}$$

$$R(m) = \sum_{f \in \mathcal{F}} R_f(m), \ \ \forall \ m \ \in \mathcal{V}, \tag{3b}$$

$$R(m) \leq R^{max}, \ \ \forall \ m \ \in \mathcal{V}. \tag{3c}$$

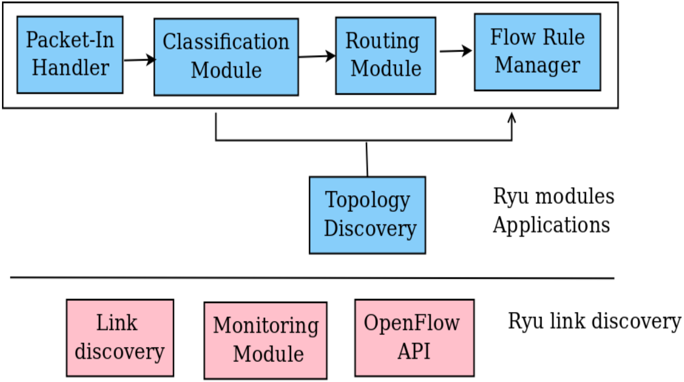
Problem Formulation [contd.]

The capacity of a path is computed by routing a flow $f_i \in \mathcal{F}$ over an edge $(m, n) \in \mathcal{L}$ and is mathematically defined as

$$C(f_i) = \min_{(m,n)\in\mathcal{L}} \; C_{m,n} \; \alpha_{m,n}^{f} \tag{4a}$$

$$C_{m,n}^{res} = C_{m,n} - \sum_{f\in\mathcal{F}} q_f^{min} \cdot \alpha_{m,n}^{f} \tag{4b}$$

System Architecture



Proposed Rule Replacement Approach
A. Routing Path Selection

- We model a cost function to determine the routing cost of link (m, n) as follows:

$$\phi_{m,n} = \alpha^f_{m,n} + \beta \; max \frac{R^{util}_m}{R^{max}_m} + \gamma \frac{C^{util}_{m,n}}{C_{m,n}} \tag{5}$$

- The optimization problem is stated as below:

$$\min \sum_{f \in \mathcal{F}} \sum_{(m,n) \in \mathcal{L}} \phi_{m,n} \alpha^f_{m,n} \tag{6a}$$

$$\textbf{s.t.} \sum_{(m,n) \in \mathcal{L}} \alpha^f_{m,n} - \sum_{(m,n) \in \mathcal{L}} \alpha^f_{m,n} = \begin{cases} 1, \ \text{if } m = s, \\ -1, \ \text{if } m = t, \\ 0, \ \forall m \ \neq s, t \end{cases} \tag{6b}$$

$$\sum_{f \in \mathcal{F}} \alpha^f_{m,n} b_f \leqslant C_{m,n}, \quad \forall (m, n) \in \mathcal{L} \tag{6c}$$

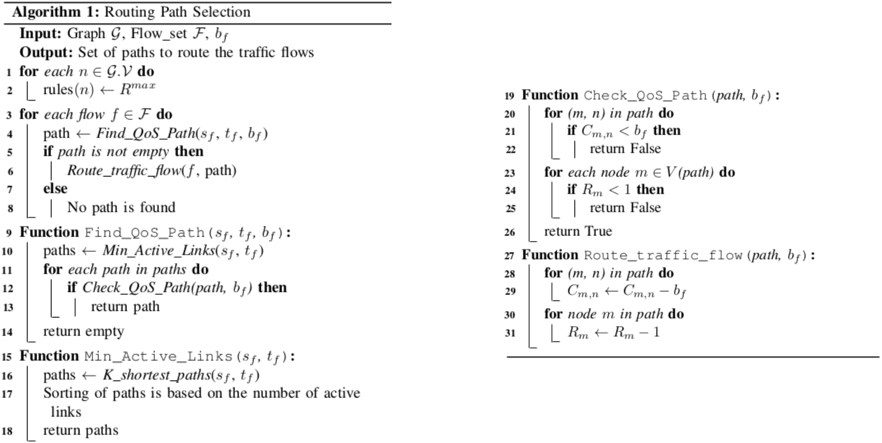
Proposed Rule Replacement Approach
A. Routing Path Selection [contd.]

$$R_m^{util} \leq R_m^{max}, \ \forall \ m \ \in \mathcal{V} \tag{7a}$$

$$C_{m,n}^{res} \geq 0, \ \forall (m, n) \in \mathcal{L} \tag{7b}$$

$$\alpha_{m,n}^{f} \in \{0, 1\} \, . \tag{7c}$$

A. Routing Path Selection [contd.]



```
Algorithm 1: Routing Path Selection
Input: Graph $\mathcal{G}$, Flow_set $\mathcal{F}$, $b_f$
Output: Set of paths to route the traffic flows
for each $n \in \mathcal{G}.\mathcal{V}$ do
    rules($n$) $\leftarrow R^{max}$
for each flow $f \in \mathcal{F}$ do
    path $\leftarrow$ Find_QoS_Path($s_f$, $t_f$, $b_f$)
    if path is not empty then
        Route_traffic_flow($f$, path)
    else
        No path is found
Function Find_QoS_Path($s_f$, $t_f$, $b_f$):
    paths $\leftarrow$ Min_Active_Links($s_f$, $t_f$)
    for each path in paths do
        if Check_QoS_Path(path, $b_f$) then
            return path
    return empty
Function Min_Active_Links($s_f$, $t_f$):
    paths $\leftarrow$ K_shortest_paths($s_f$, $t_f$)
    Sorting of paths is based on the number of active
     links
    return paths
Function Check_QoS_Path(path, $b_f$):
    for (m, n) in path do
        if $C_{m,n} < b_f$ then
            return False
    for each node $m \in V$(path) do
        if $R_m < 1$ then
            return False
    return True
Function Route_traffic_flow(path, $b_f$):
    for (m, n) in path do
        $C_{m,n} \leftarrow C_{m,n} - b_f$
    for node $m$ in path do
        $R_m \leftarrow R_m - 1$
```

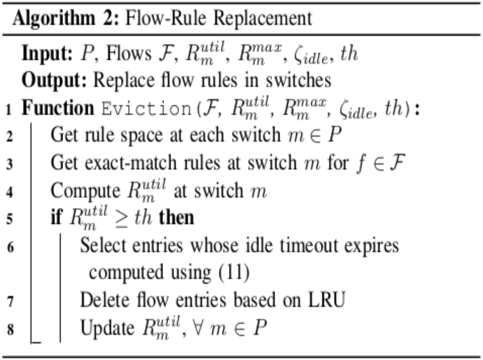
B. Flow Rule Placement

- The rule utilization at switch m is represented as R_m^{util} and is computed as

$$R_m^{util} = R^{max} \cdot \frac{E_m}{R_m} \tag{8}$$

where E_m is the exact-match entries in the m^{th} switch flow table and R_m is the total rules present at switch m.

B. Flow Rule Placement



```
Algorithm 2: Flow-Rule Replacement
Input: $P$, Flows $\mathcal{F}$, $R_m^{util}$, $R_m^{max}$, $\zeta_{idle}$, $th$
Output: Replace flow rules in switches
Function Eviction($\mathcal{F}$, $R_m^{util}$, $R_m^{max}$, $\zeta_{idle}$, $th$):
    Get rule space at each switch $m \in P$
    Get exact-match rules at switch $m$ for $f \in \mathcal{F}$
    Compute $R_m^{util}$ at switch $m$
    if $R_m^{util} \geq th$ then
        Select entries whose idle timeout expires
         computed using (11)
        Delete flow entries based on LRU
        Update $R_m^{util}$, $\forall\ m \in P$
```

- The Idle timeout (ζ_{idle}) is computed as follows:

$$\zeta_{idle} = \zeta_{idle}^{max} - \frac{R_m^{used}}{R_m^{total}} * (\zeta_{idle}^{max} - \zeta_{idle}^{min}) \tag{9}$$

where R_m^{used} and R_m^{total} represent the number of entries used and the total flow table's size.

Performance Evaluation

- The proposed approach is simulated over the Mininet network emulator, Ryu SDN controller.
- We used the Distributed Internet Traffic Generator (D-ITG) to model the traffic flows to generate real traces in the network.
- We have performed the comparison of our proposed approach with benchmark schemes: ROSA [3], Lagrangian Relaxation based algorithm (LARAC) [4], and shortest path with minimum delay (SPD)[6].
- We used two network topologies – Goodnet and Sprint from Internet topology zoo.

Results and Discussions

- End-to-End delay
- Packet loss
- Active links
- Effect of ζ_{idle}^{max} and ζ_{idle}^{min} on table-full events

End-to-End delay

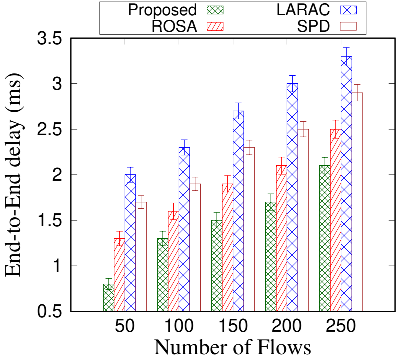


Figure: Goodnet topology

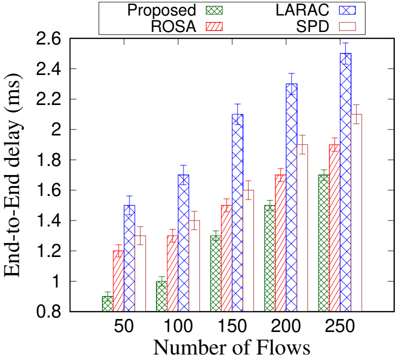


Figure: Sprint topology

Packet Loss

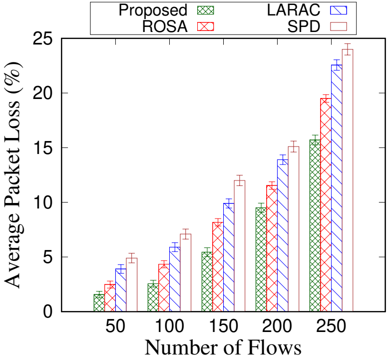


Figure: Goodnet topology

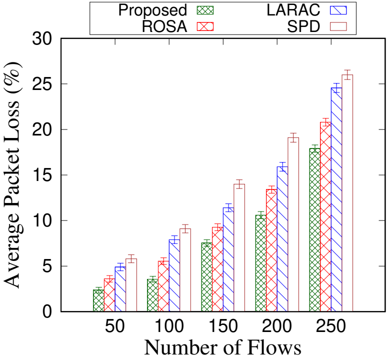


Figure: Sprint topology

Active Links

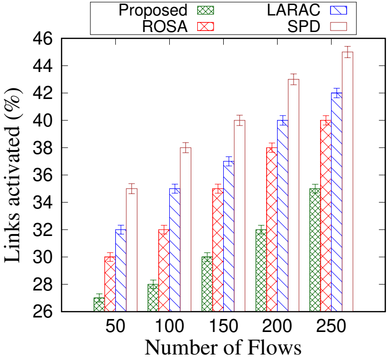


Figure: Goodnet topology

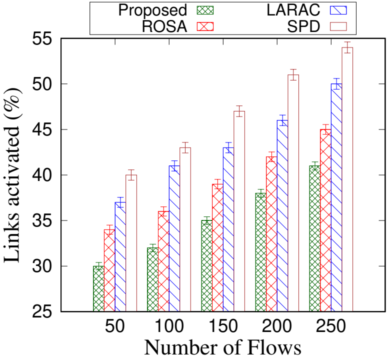


Figure: Sprint topology

Effect of ζ_{idle}^{max} and ζ_{idle}^{min} on table-full events

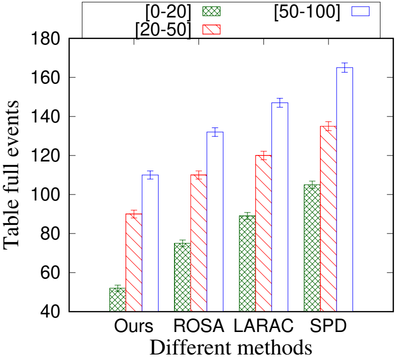


Figure: Goodnet topology

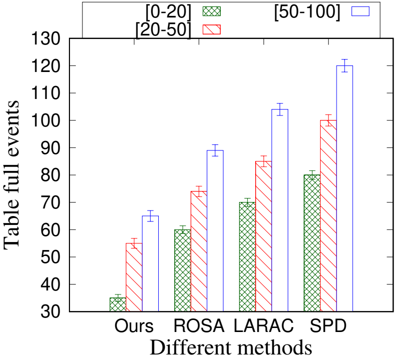


Figure: Sprint topology

Conclusions

- Formulated an optimization problem to minimize the active links in the network.
- Presented a flow rule replacement scheme to avoid the overflow in the flow table of SDN switches.
- Our proposed approach uses the idle timeout to delete the stale flow entries to vacate space for new incoming flows in the flow table.
- Our proposed approach reduces active links, end-to-end delay compared to the benchmark schemes.

References

N. Kang, Z. Liu, J. Rexford, and D. Walker, "Optimizing the "One Big Switch" Abstraction in Software-Defined Networks," in Proc. of the 9th ACM Conf. on Emerg. Netw. Experiments and Technol., (Santa Barbara, California, USA), pp. 13–24, Dec. 2013.

Y. Kanizo, D. Hay, and I. Keslassy, "Palette: Distributing Tables in Software-Defined Networks," in Proc. IEEE INFOCOM, (Turin, Italy), pp. 545–549, April 2013

Y. Njah and M. Cheriet, "Parallel Route Optimization and Service As- surance in Energy-Efficient Software-Defined Industrial IoT Networks," IEEE Access, vol. 9, pp. 24682–24696, 2021

H. E. Egilmez, S. Civanlar, and A. M. Tekalp, "An Optimization Framework for QoS-Enabled Adaptive Video Streaming Over OpenFlow Networks," IEEE Trans. on Multimedia, vol. 15, no. 3, pp. 710–715, 2013

P. Kamboj, S. Pal, and A. Mehra, "A QoS-aware Routing based on Bandwidth Management in Software-Defined IoT Network," *in 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, Denver, USA: IEEE, Oct. 2021, pp. 579–584.

J. M. Llopis, J. Pieczerak, and T. Janaszka, "Minimizing Latency of Critical Traffic through SDN," in IEEE Int. Conf. on Netw., Architecture and Storage (NAS), (Long Beach, CA, USA), pp. 1–6, IEEE, Aug. 2016.

SY.-C. Wang and S.-Y. You, "An Efficient Route Management Framework for Load Balance and Overhead Reduction in SDN-Based Data Center Networks," IEEE Trans. on Netw. and Service Manage., vol. 15, no. 4, pp. 1422–1434, 2018.

A. Botta, A. Dainotti, and A. Pescapé, "A tool for the generation of realistic network workload for emerging networking scenarios," Comput. Netw., vol. 56, no. 15, pp. 3531–3547, 2012.

Y. Zhao, X. Wang, Q. He, C. Zhang, and M. Huang, "PLOFR: An online flow route framework for power saving and load balance in SDN," *IEEE Systems Journal*, vol. 15, no. 1, pp. 526–537, 2020.

Thank You !