

QRMNO: Dynamic Resource Reallocation with Minimal Network Overhead in QoS-aware Software Defined Cloud Data Centers

Mohammad Mahdi Tajiki
Electrical and Computer Engineering
Tarbiat Modares University
Tehran, Iran
Mahdi.Tajiki@modares.ac.ir

Behzad Akbari, Nader Mokari
Electrical and Computer Engineering
Tarbiat Modares University
Tehran, Iran
{b.akbari, mokari}@modares.ac.ir

Abstract—One of the most important aspect of software defined networking (SDN) is resource allocation in which the links are assigned to the flows based on their requirements. Dynamic nature of network traffic along with the existence of burst and big flows make the conventional static approaches inefficient. Therefore, recent researches are dedicated to dynamic resource allocation. To this end, network reprogramming overhead is the most challenging part. In this paper, we exploit SDN features to reallocate the resources with minimum overhead on the network reprogramming phase in a way that guarantees the QoS requirements. Due to the low overhead of the proposed scheme, the resources can be reallocated with a higher frequency. Additionally, it is shown that the delay and packet loss of the proposed scheme is lower than conventional approaches such as the shortest path.

Index Terms—QoS, Resource Reallocation, Traffic Engineering, SDN, Network Reprogramming Overhead.

I. INTRODUCTION

Lots of network applications such as media streaming require minimum packet loss. As a result, QoS-aware resource reallocation plays an important role in traffic engineering. In recent networks, the traffic has an extremely dynamic characteristics. The flow dynamics along with burst nature of network traffic, makes the static resource allocation methods useless. On the other hand, the network side-effect of dynamic resource reallocation becomes a critical issue in software defined networking.

Hilmi et al. [1] present an analytical framework for optimization of forwarding decisions at the control layer to enable Quality of Service (QoS) over OpenFlow networks. Some works such as [2] [3] [4], minimize the delay and packet loss by rerouting video flows. Hence, the end-to-end delay of each flow is guaranteed to be less than a predefined threshold. Civanlar et al. [5] propose an algorithm for video streaming that minimize the packet loss along with the path hops. The focus of [6] is to develop a genetic based routing algorithm that satisfies multiple constraints of the multimedia applications. Similarly, [7] defines the path weight as a QoS measure and minimize it by the aid of ant colony system (ACS).

Links that specify the maximum throughput of each path is defined as critical links by Santosh et al. [8]. They focus on these links and select a path based on its critical link weight to reduce

the blocking rate. The authors of [9] follow a similar approach by minimizing (traffic load)/capacity for each link.

The Ghosh scheme [10] prevents flow from violating a pre-defined threshold by minimizing end-to-end delay and maximizing network throughput. Ongaro [11] exploits OpenFlow features to manage differentiating network services with QoS. They use Integer Linear Programming (ILP) to formulate the network traffic routing in a way that minimizes delay and packet loss. In [12], authors present an optimization-based allocation for inter-DC communication that handles priorities between traffic classes. Finally, it should be mentioned that all of the reviewed algorithms are unaware of the network side-effects (overhead) of resource reallocation.

In this paper, we introduce a dynamic and efficient resource reallocation scheme in which we guarantee flow bandwidth. Moreover, the proposed scheme reallocates the resources in a way that imposes the minimal overhead to the network. The proposed scheme is called QRMNO (QoS-aware Resource reallocation with Minimal Network Overhead). The contributions are as follows:

- To the best of our knowledge, there is no related works that reallocate the resources with QoS constraint along with optimization of the flow table changes.
- In order to make a trade of between the performance and the computational complexity, the granularity of network rescheduling can be variable in the proposed scheme.
- In order to solve the optimization problem two schemes are proposed. The first one finds the optimal solution, however, its computational complexity is high. The second one has a low computational complexity while it is near-optimal.

II. NETWORK MODEL AND ASSUMPTIONS

In this paper, the network is reconfigured in a way that minimizes the flow table changes and simultaneously QoS constraint is met, e.g. the bandwidth of network applications is guaranteed. Consider there are n OpenFlow enabled switches which are connected to each other based on the matrix $B_{n \times n}$ where B_{ij} determines the bandwidth of the link from switch i to switch j . The number of flows in the network is p . The new routing matrix

$A_{n \times n \times p}$ specifies the path selected for each flow, e.g. if $A_{ij}^f \in \{0,1\}$ is equal to 1 then the flow f crosses the link $i \rightarrow j$. The matrix $A0_{n \times n \times p}$ is *current routing matrix* which is going to be reprogrammed. Suppose there are multiple classes of traffic in the network. The *flow requirement matrix* $C_{1 \times p}$ specifies flows guaranteed requirements based on the corresponding class. The i^{th} row of *flow requirement matrix* defines the guaranteed bandwidth for each flow. The source and destination of each flow is specified using *source of flows matrix* ($s_{1 \times p}$) and *destination of flows matrix* ($d_{1 \times p}$), respectively. The parameter μ specifies the maximum link utilization.

III. PROBLEM FORMULATION

The core object of this paper is to efficiently and dynamically allocate resources in a way that 1- guarantees QoS requirements of different applications and 2- proactively prevents the resource waste and congestion 3- minimizes the network overhead in re-scheduling process. The routing matrix should be calculated in a way that satisfies the mentioned constraints. The objective function is as follows:

$$\text{minimize } \left(\sum_{i=1}^n \sum_{j=1}^n \sum_{f=1}^p |A_{ij}^f - A0_{ij}^f| \right), \quad (1)$$

The objective function (1) minimizes the difference of the *new routing matrix* and the *current routing matrix*. The link load is guaranteed to be smaller than the maximum target utilization μ by Constraint (2). The left hand side calculates sum of guaranteed bandwidth of all flows crossing a specified link. The right hand side of it specifies the link allowed bandwidth.

$$\sum_{f=1}^p A_{ij}^f C_f \leq \mu B_{ij}, \quad \forall i, j \in \{1, \dots, n\}, \quad (2)$$

The flows are prevented from returning to the source switches by Equation (3). As it is mentioned earlier, $A_{is_f}^f$ is zero if and only if the flow f crosses the link that connects switch i to the source switch of f . For each flow, the Equation (3) forces the summation of $A_{is_f}^f$ (for all i) to be zero. In other words, none of the flows can cross the link between a switch to the source switch of that flow. On the other hand, (4) makes the flows to stay on the destination switches.

$$\sum_{i=1}^n A_{is_f}^f = 0, \quad \forall f \in \{1, \dots, p\}, \quad (3)$$

$$\sum_{i=1}^n A_{di}^f = 0, \quad \forall f \in \{1, \dots, p\}, \quad (4)$$

Equations (5) and Equation (6) force the flows to leave the origin switches and enter to the destination one, respectively. Equation (5) guarantees the flow to cross exactly one of the source switch's outgoing link. Besides, Equation (6) ensures the same for the incoming links of destination switch.

$$\sum_{i=1}^n A_{s_f i}^f = 1, \quad \forall f \in \{1, \dots, p\}, \quad (5)$$

$$\sum_{i=1}^n A_{i d_f}^f = 1, \quad \forall f \in \{1, \dots, p\}, \quad (6)$$

It should be mentioned that if a switch is neither source nor destination of a flow, the flow must leave that switch after it moves in. This restriction is applied by Equation (7) via balancing the amount of traffic entered to the switch with the amount of traffic left it.

$$\sum_{j=1}^n A_{ij}^f = \sum_{j=1}^n A_{ji}^f, \quad \forall f \in \{1, \dots, p\} \quad \forall i \in \{1, \dots, n\} - \{s_f, d_f\}, \quad (7)$$

Constraint (8) makes sure there is no loop in the new routing matrix. It prevents flows from returning to a switch that is met in the past.

$$\sum_{j=1}^n A_{ij}^f \leq 1, \quad \forall i \in \{1, \dots, n\} \quad \forall f \in \{1, \dots, p\}, \quad (8)$$

$$A_{ij}^f \in \{0, 1\}, \quad \forall f \in \{1, \dots, p\} \quad \forall i, j \in \{1, \dots, n\}. \quad (9)$$

As it can be seen, the optimization problem is in form of Binary Linear Programming which is a branch of Integer Linear Programming. It is notable that the CVX toolbox [14] is used to solve the optimization problem.

IV. PROBLEM RELAXATION

In this subsection, a two-step problem relaxation is proposed to reduce the computational complexity of the proposed scheme. In the first phase, a flow compression technique is described which reduces the number of active flows. Afterwards, since the existence of absolute values in formulation makes it dramatically time consuming, an equivalent objective function is proposed.

A. Flow Compression

For the sake of simplicity, all flows from a specified source to a specified destination are called similar flows (SF). There are lots of small flows in network (flows with size less than 10KB) [13] which can be merged to reduce the time complexity of the solution. To this end, all SF flows that are smaller than a predefined lower band are merged. Note that the outcome must be smaller than an upper bound threshold, otherwise it breaks into two or more flows. Therefore, increment in the lower band reduces the time complexity while it increases the optimality gap. Results analysis show that in large number of flows, the compression reduces the number of active flows dramatically.

B. Low Computational Complexity Approach

Existence of absolute value in the objective function of proposed scheme, intensely increases the optimization time. In order to remove the absolute value from the objective function an equivalent function is proposed which is rewritten as:

$$\sum_{i=1}^n \sum_{j=1}^n \sum_{f=1}^p (A_{ij}^f + A0_{ij}^f - 2 * A_{ij}^f * A0_{ij}^f). \quad (10)$$

V. RESULT ANALYSIS

The proposed analytical model is evaluated on fat-tree topology which is illustrated in Fig. 1 for $k=4$.

A. Overhead Evaluation

For the sake of simplicity, the number of changes in the elements of switches forwarding table is called SFTC (Switches Forwarding Table Changes). As it can be seen in Fig. 4, the SFTC is linearly increased by the increment of flow number. However, the SFTC of conventional approaches is sufficiently greater compared to QRMNO. It should be mentioned that the impact of proposed schemes on the network side-effect of resource reallocation is multiplied when there exist 1) different network traffic requirements 2) big flows and resource partitioning and 3) highly dynamic network.

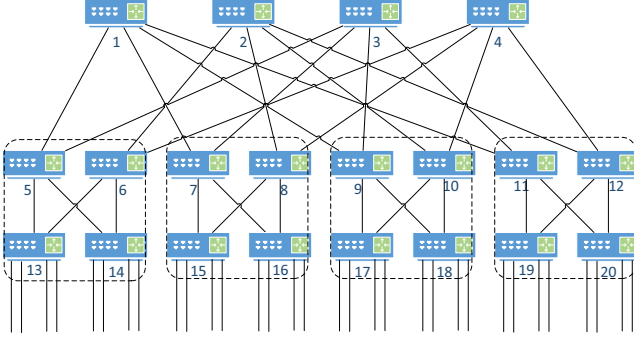


Fig. 1. Fat-Tree topology $K=4$.

1) Impact of SFTC on delay

There are two types of delay which are related to the network reconfiguration overhead: a) end-to-end delay of control messages b) delay of updating the forwarding tables (FT). In the network rescheduling, all flows are considered together, therefore, there is a global knowledge which causes lots of flows to be rerouted. In order to update the FTs, the controller sends a large number of controlling messages to the switches. Suppose there are n' flows that are routed via a new path. Additionally, propagation delay and transmission rate of all links are t_{propag} and t_{trans} , respectively. The processing time of each switches (time of updating an element of FT) is t_{proc} . The end-to-end delay of control messages is computed as follow:

$$t_{propag} + t_{trans} + \max((n' - 1) * t_{trans}, n' * t_{proc}) \quad (11)$$

As it can be seen from Equation (11), by increasing the number of flows that are rerouted, the end-to-end delay of control messages increases. Fig. 2 compares QRMNO with conventional approaches from *number of changed flows* perspective. It is shown that the number of flows with a new route is dramatically decreased in QRMNO. In order to clearly illustrate the impact of considering the *number of rerouted* flows as a parameter, Fig. 2 is depicted.

The required time for updating the *forwarding table (FT)* has two parameters: 1) time of setting an FT element 2) number of elements in a FT that are going to be changed (NFTC). The first one is reliant on the switch, therefore it is independent of rerouting algorithm. However, NFTC is dependent on both the rerouting algorithm and the switch. Since NFTC value of switches are different in resource reallocation, the impact of QRMNO on the switch that has the maximum NFTC value is focused here. Based

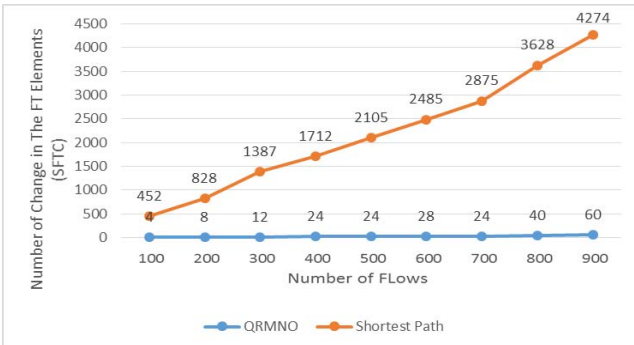


Fig. 4. Overhead comparison of QRMNO and conventional approaches

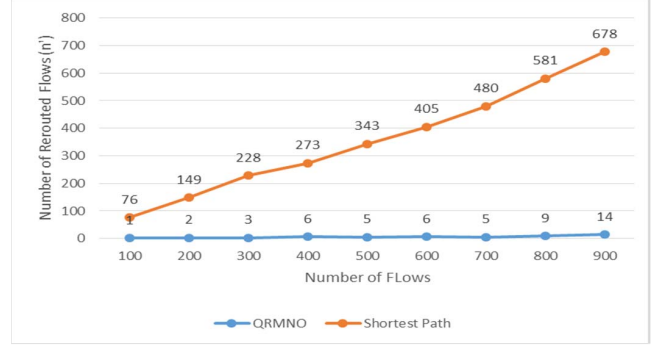


Fig. 2. Number of flows that are rerouted

on the experimental result, the maximum NFTC value of QRMNO is dramatically lower than shortest path selection. Therefore, the delay of updating the FT is lower in QRMNO in compared with conventional approaches like shortest path. Suppose an element takes τ millisecond to be changed in the FT, therefore, Fig. 3 illustrates the time that is needed to completely reconfigure the network with QRMNO and shortest path.

2) Impact of SFTC on packet loss

The impact of SFTC on the packet loss depends on two parameters: 1) routing 2) TCP retransmission time. In order to clarify the effect of network reconfiguration overhead on routing, we consider there are seven OpenFlow enabled switches that are controlled by an SDN controller similar to Fig. 5. All links are x GB/S except the links (S5, S6) and (S4, S5) which are $10x$ GB/S. There are three big flows in the network that are currently routed via $S1 \rightarrow S7 \rightarrow S6 \rightarrow S5 \rightarrow S4$, $S1 \rightarrow S2 \rightarrow S3 \rightarrow S4 \rightarrow S5 \rightarrow S6$, and $S1 \rightarrow S7 \rightarrow S6 \rightarrow S5$. The guaranteed bandwidth for these flows are $F1$ GB/S, $F2$ GB/S, and $F3$ GB/S, respectively. Note that:

$$x < F1 + F2 + F3 < 10x \quad F1 + F3 < x \quad F2 + F3 < x$$

Propagation delay of links are specified by the matrix $D'_{1 \times 7}$, however, by the purpose of simplifying the problem, all link delays are considered to be equal to τ . To purely show the impact of network reconfiguration on the routing, other flows are ignored. At this point, the resources will be reallocated. As it can be seen in Fig. 5, the controller is directly connected to S5, S4, and S3. Therefore, the propagation delay of controller message to S1 is 3τ .

Suppose the resources are reallocated in a time period of T , at the point of resource reallocation, QRMNO reschedules the network similar to Fig. 6 in which the network rescheduling

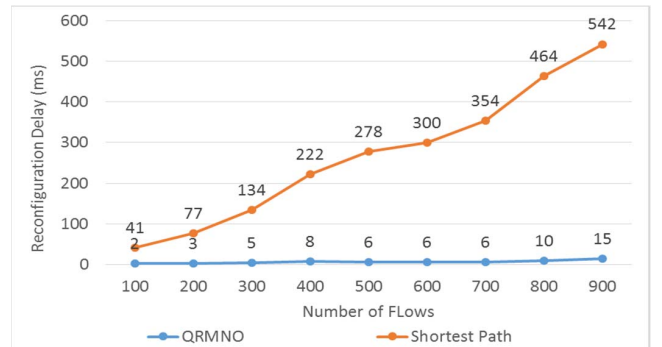


Fig. 3. Delay of network reconfiguration ($\tau = 1$ ms).

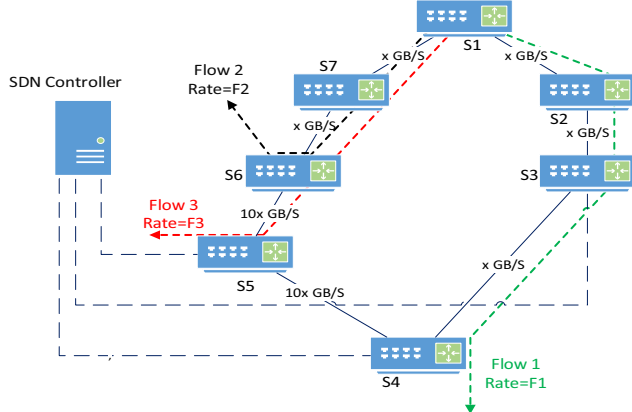


Fig. 5. Resource reallocation based on shortest path.

overhead for these three flows is zero. It should be mentioned that QRMNO changes a flow path if the unused bandwidth of current route is smaller than the guaranteed bandwidth.

On the other hand, conventional approaches such as shortest path, reallocate the resources similar to Fig. 5. The controlling message is delivered to S5 with τ second delay while the delay is 3τ for S1. As a result, the forwarding table of S1 is updated 2τ second after S5 forwarding table is updated. Consequently, the packet loss of Flow 1 and Flow 2 are as follow:

$$2\tau S * \left(F1 \frac{GB}{S} + F2 \frac{GB}{S} \right) = 2 * (F1 + F2) \tau GB$$

Based on the mentioned reasons, the impact of SFTC on routing is a function of 1) number of rerouted flows and 2) network topology. As mentioned earlier, number of rerouted flows is illustrated in Fig. 2 and it is shown that the QRMNO is superior in compare with conventional approaches like shortest path.

VI. CONCLUSION

In this paper, QRMNO a QoS-aware resource reallocation algorithm was introduced. QRMNO reschedules the network in a way that imposes the minimal overhead in forwarding table updates. The impact of QRMNO on the network rescheduling overhead (SFTC) is compared with shortest path in which it dramatically improves the performance. Additionally, the impact of SFTC on delay and packet loss was discussed. Result analysis shows that QRMNO decreases the delay and packet loss. Future

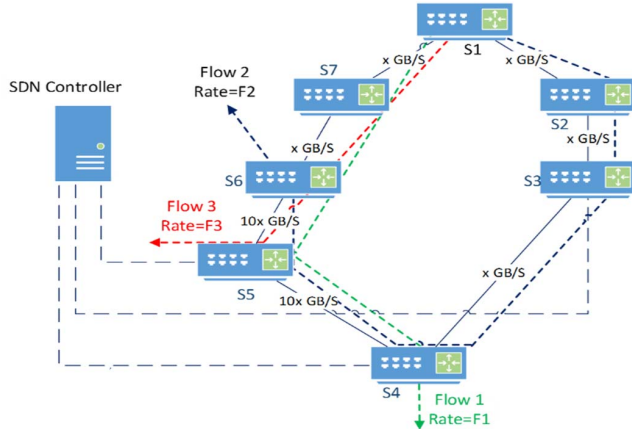


Fig. 6. QRMNO resource reallocation.

works will be dedicated on proposing methods for hybrid network (combination of conventional and OpenFlow switches).

VII. REFERENCES

- [1] E. Hilmi and S. Civanlar, "An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks," *IEEE Transactions on Multimedia*, vol. 3, no. 15, pp. 710-715, 2013.
- [2] E. Hilmi, T. Dane and T. Bagci, "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks," in *IEEE Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Asia-Pacific, 2012, pp. 1-8.
- [3] E. Hilmi and S. Civanlar, "A distributed QoS routing architecture for scalable video streaming over multi-domain OpenFlow networks," in *IEEE Image Processing (ICIP) International Conference*, 2012, pp. 2237-2240.
- [4] E. Hilmi, B. Gorkemli and S. Civanlar, "Scalable video streaming over OpenFlow networks: An optimization framework for QoS routing," in *IEEE Image Processing (ICIP) International Conference*, 2011, pp. 2241-2244.
- [5] C. Seyhan, M. Parlakisik, B. Gorkemli, B. Kaytaz and E. Onem, "A qos-enabled openflow environment for scalable video streaming," in *IEEE GLOBECOM Workshops (GC Wkshps)*, 2010, pp. 351-356.
- [6] R. Leela, N. Thanulekshmi and S. Selvakumar, "Multi-constraint qos unicast routing using genetic algorithm (muruga)," *Applied Soft Computing*, vol. 2, no. 15, pp. 1753-1761, 2011.
- [7] L. Benlai and J. Yu, "One Multi-constraint QoS Routing Algorithm CGEA Based on Ant Colony System," in *IEEE Information Science and Control Engineering (ICISCE) International Conference*, 2015, pp. 848-851.
- [8] K. Santosh, R. Sharma and I. Mishra, "New QoS routing algorithm for MPLS networks using delay and bandwidth constraints," *International Journal of Information*, vol. 3, no. 2, pp. 200-212, 2012.
- [9] Z. Jing and X. Ge, "QoS Multi-path Routing Scheme Based on ACR Algorithm in Industrial Ethernet," in *Springer Conference on Communications, Signal Processing, and Systems*, 2015, pp. 593-601.
- [10] G. Amitabha, S. Ha, E. Carbbe and J. Rexford, "Scalable multi-class traffic management in data center backbone networks," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 31, pp. 2673-2683, 2013.
- [11] F. Ongaro, "Enhancing Quality of Service in Software-Defined Networks," *Diss. Alma Mater Studiorum-University Of Bologna*, 2014, pp. 44-59.
- [12] W. Jason, Y. Wang, X. Dai and B. Bensaou, "SDN-based multi-class QoS-guaranteed inter-data center traffic management," in *IEEE Cloud Networking (CloudNet) International Conference*, 2014, pp. 401-406.
- [13] B. Theophilus, "New Paradigms For Managing The Complexity And Improving The Performance Of Enterprise Networks," *Diss. University Of Wisconsin-Madison*, 2012, pp. 80-112.
- [14] "CVX," 2016. [Online]. Available: <http://cvxr.com/cvx/> [Access 2016]