applied sciences

MDPI

Article

Joint QoS and Congestion Control Based on Traffic Prediction in SDN

Mohammad Mahdi Tajiki [1], Behzad Akbari [1,*], Mohammad Shojafar [2] and Nader Mokari [1]

1 Department of Electrical and Computer Engineering, Tarbiat Modares University Tehran, Tehran 14115-111, Iran; mahdi.tajiki@modares.ac.ir (M.M.T.); nader.mokdari@modares.ac.ir (N.M.)
2 Department of Information Engineering, Electronics, and Telecommunication, Sapienza University of Rome, via Eudossiana 18, 00184 Rome, Italy; mohammad.shojafar@uniroma1.it
* Correspondence: b.akbari@modares.ac.ir

Received: 3 November 2017; Accepted: 29 November 2017; Published: 5 December 2017

Abstract: Due to the various network requirements of applications, quality of service (QoS)-aware routing plays an important role in the networks. Recently proposed resource allocation algorithms focus on the current traffic matrix, which is not applicable for dynamic networks. In this paper, we exploit an estimation of flow matrix that gives our scheme the ability to sufficiently reduce the total packet loss and simultaneously raise the network throughput. In this way, we mathematically formulate the QoS-aware resource reallocation in software-defined networking (SDN) networks based on the traffic prediction. To solve this optimization problem, two schemes are proposed: (i) exact solution; and (ii) fast suboptimal one. The proposed schemes are compared with the accuracy perspective. Moreover, the impact of prediction on resource reallocation is discussed. In this regard, it is shown that, compared with the conventional scheme, the proposed scheme decreases the packet loss and increases the throughput significantly.

Keywords: quality of service (QoS); resource reallocation; software-defined networking (SDN); network reconfiguration

1. Introduction

In the last several years, computer networks have envisaged an evolution by the diffusion of smartphones, over the top (OTT) services, and cloud computing. This enforces network practitioners and researchers to a basic transformation from traditional networks to programmable networks. Software-defined networking (SDN) is a radical new idea in networking that enables innovation through network programmability. The main idea of SDN is the separation of network intelligence (control *plane*) from the forwarding device (data plane). The network intelligence is logically centralized (called *controller*), which enables the programmers to exploit global knowledge of the controller. Hence, it makes SDN proper for emerging technology like 5G, cloud data centers, and Internet exchange points (IXPs). One of the critical issues of SDN is resource reallocation in a way that meets users' quality of service requirements.

In quality of service (QoS)-aware resource reallocation (QRA), resources (links) are assigned to each network traffic flow based on service level agreement (SLA). The selected path for each flow should satisfy the flow requirements such as maximum tolerable delay or minimum required bandwidth. There are some important issues in QRA as follows: (1) existence of burst and dynamic traffic; (2) different traffic classes; (3) big data and resource partitioning; and (4) heavy demands and limited resources. In recent networks, the traffic characteristics obey an extremely dynamic model; as an example, it can be referred to OTT services as a source of dynamics. Additionally, there are lots of burst traffic in the network, such as virtual machine (VM) migration traffic flows.

The flow dynamics along with burst nature of network traffic makes the static resource allocation methods useless. Moreover, there are several traffic classes in the network that have different requirements, e.g., video conferences are delay sensitive while file transfer protocol (FTP) connection is bandwidth sensitive.

In this context, several questions are arising, such as: Is it possible to practically tune the network resources with the dynamic behavior of network? How to exploit prior information of SDN controller to assign resources to the current flows while guaranteeing QoS? The goal of this paper is to shed light on these issues.

1.1. Overview of Our Contributions

In this paper, we introduce a dynamic and efficient QRA scheme called *QRTP* (QoS-aware resource Reallocation based on Traffic Prediction in SDN), in which we guarantee delay and bandwidth. It proactively decreases congestion by minimizing total packet loss. The proposed approach not only uses the current flow matrix but also adopts a predicted flow matrix to conquer the dynamic and burst nature of network traffic. Our contributions are as follows:

(i) To the best of our knowledge, there is no related work that considers traffic prediction in resource reallocation. This is the first work that exploits predicted flow matrix to dynamically reschedule the network. In addition, this is the first work that studies the impact of flow traffic estimation on network resource reallocation. The simulation results show significant reduction in the total packet loss along with improvement in the network throughput.
(ii) The proposed approach can support different traffic classes with various QoS requirements in flow level granularity. To do this, delay and bandwidth are aimed to be fulfilled while packet loss is minimized for current and predicted traffic load.
(iii) Two schemes are proposed for solving the introduced QRTP problem: (I) an optimal solution which is time consuming; and (II) a relaxed but fast one.
(iv) The proposed schemes are compared with the performance perspective. In this way, we use a real network traffic and topology for experiments.
(v) Our formulation makes a trade-off between performance and computational complexity. In this way, it reschedules the network with flow granularity which can be the size of the exchanged information of "a special application" or "all communications from one data center to another one".

1.2. Paper Organization

The rest of this paper is organized as follows: a background of routing and rerouting solutions in SDNs are given in the next section. Section 3 presents the system architecture and outline of our work. Section 4 presents the system model, problem formulation in our work. In Section 5, we detail the proposed method. Section 6 includes the simulation and metric descriptions. In Section 7, the performance of the proposed method is analyzed and validated in terms of the presented metrics. Finally, Section 8 presents conclusions and future outlook.

2. Related Work

Finding efficient routes to achieve the desired SDN performance leads to facilitating the processing and decreasing the end-to-end delay. SDN traffic is measured and analyzed in order to enhance the performance of an operational network at both the traffic and resource levels. Recently, several works are done in routing over the SDNs [1]. Several recent works have exploited the global visibility offered by SDN controllers to distribute traffic across pre-computed paths in optimal and suboptimal manners such as Software-driven wide area network (SWAN) [2] and B4 [3] that the former one uses k-shortest paths for routing across the SDN switches while the latter one presents a greedy heuristic to ensure fairness; however, both of them lack sufficient flexibility and path diversity to handle unexpected situations, i.e., traffic bursts, link failures. Recently, authors in [4,5] tackled models (e.g., oblivious

routing for wide-area network traffic engineering [6,7]) and estimate traffic tools in order to cope with these problems. In detail, authors in [5] as a power routing toolbox presents several failure scenarios, path budgets, to simulate a variety of realistic workloads and compare the performance of the existing routing on the "failed" network, to the best possible routing on the failed network. From the SDN load balancing point of view, the literature presents load balancing technologies for data and control layer traffic in order to centralize multiple optional link utilization rates and flow characteristics. For example, Authors in [8] present a routing algorithm that manage and optimize traffic in the SDN using an additional backend server. Moreover, another similar solution is [9] that comprises routing optimization that deals with various shapes of incoming flows using a heuristic equal-cost multi-path routing (ECMP) algorithm to monitor, manage and route the aggregated flows. Although these methods are interesting, they did not cover the adaptive rerouting in such a manner that recovers the current situation of the SDN when faced with link/network congestion. In [10], authors present user aspect overview functions through the SDN architecture by experiencing the desired performances of requests such as suitability and applicability. This holistic architecture has the flavor of useful practical concepts, but it suffers from reliability and availability when faced with link and network congestion.

Some other routing methods target the quality of services (QoSs) over such networks. The SDN provides an open control interface to support QoS requirements and preserves flexible network traffic strategies to satisfy different network applications. A QoS-aware network reconfiguration for software defined networks is described in [11]. They reallocate the resources in a way that minimizes the network reconfiguration overhead. Similarly, The papers [12,13] propose two rerouting algorithms to guarantee the QoS constraints while the energy consumption of the network is minimized. In this way, they provide a mathematical formulation of the problem. In addition, authors in [14] present a novel network function virtualization (NFV)/SDN orchestration between Internet of things (IoT) gateways and the deployed VMs that are allocated to the edge node. The presented paper tested in terms of packet delivery for VMs in various cases. The paper and the results are interesting but after deep analysis, we realized that the work did not address the traffic prediction and various QoS requirements that matter in QRTP. The work [15], proposes a traffic engineering algorithm that focuses on improving data center network utilization. In [16–19], the authors reroute video flows in order to minimize delay and packet loss of total traffic. They guarantee that the end-to-end delay of each flow is less than a predefined threshold. Civanlar et al. [20] propose a video streaming resource allocation algorithm, with two different classes, which minimize both selected path hobs and packet loss. The mentioned approaches support at most three different traffic classes. Liang et al. [21] define path weight as a function of QoS parameters, and they use an ant colony system (ACS) to minimize it. Similarly, The paper [22] exploits a genetic algorithm (GA) for minimization of delay and packet loss. The authors of [21,22], instead of defining a mathematical formulation, propose a function that is minimized by GA heuristic methods. In [23], authors focus on critical links that are the links specifying the maximum throughput of each path. In this regard, they select a path based on its critical link weight. Kulkarni's approach not only guarantees end-to-end delay but also end-to-end bandwidth. The work [24] follows a similar approach by minimizing (traffic load)/capacity of each link. A resource allocation approach with three classes that focus on the delay of high priority traffic is introduced in [25]. The Ghosh scheme [26] prevents flow from violating a predefined threshold by minimizing end-to-end delay and maximizing network throughput. Ongaro [27] uses Integer Linear Programming (ILP) to formulate the network traffic routing in a way that minimizes delay and packet loss. All the reviewed QRA algorithms just focus on the current network traffic while they are unaware of future pattern of demands.

3. System Architecture

In this section, the proposed architecture and its components are presented. The architecture is conceptually aligned with the SDN layering discussed in [28]. With reference to Figure 1, we consider

three different layers. The *Infrastructure* layer consists of networking devices and corresponds to the Forwarding Plane and Operational Plane presented in [28]. The *Control* layer interacts with the networking devices in order to program their behavior from a logically centralized perspective. It corresponds to the control plane and management plane in [28]. Finally, the *Application* layer includes the applications and services that define the overall network behavior.

Looking at Figure 1, the Infrastructure Layer includes the networking devices. We refer to the networking devices as *switches*, following an SDN-based terminology, but these devices could also act as routers.

We assume that there is a logically centralized SDN controller in the Control layer, connected to the set of SDN switches via the Southbound protocols (examples of Southbound protocols are shown in Figure 1). The main role of the SDN controller is to setup the forwarding tables of the single networking devices in order to properly configure the packet forwarding. The SDN controller interacts with the networking devices and gathers information on the topology and on the network traffic. The SDN controller can include additional functionality, that we represent as additional modules in Figure 1. In particular, in our architecture, we consider two monitoring modules. These modules are considered as a part of the controller to speed up the process and reduce the overhead of gathering information from the switches.

The Control layer offers a set of functionality to the Application layer, through the Northbound application programming interface (API). The application and services in the Application layer use this API in order to implement the desired behavior. In our architecture, the proposed resource allocation/reallocation modules are included in the Application layer. These modules take the decisions about the path selection and request the Control layer to enforce them through the Northbound API. In the other direction, the Control layer provides the Application layer with information about topology and network traffic.

As can be seen in Figure 1, the proposed architecture has the following modules:

- *Resource allocator (routing module)*: this module belongs to the application layer. When a new flow enters the network, the resource allocator assigns the required resources to the flow. This module does not reroute existing flows and only focuses on newly arrived ones.
- *Resource re-allocator (re-routing module)*: this module belongs to the application layer. It is capable of performing a reallocation of some flows in order to react to some event (like congestion) or periodically to optimize the use of resources or the user perceived QoS.

We can exploit any of the existing routing algorithms as the resource allocator. Therefore, we focus on the resource re-allocator, which represents the more general case. In addition, it is possible to also cover the resource allocation process with some adaptations of the re-allocator (e.g., only one flow is considered in the optimization).

In this paper, we introduce QRTP, a resource reallocation algorithm (rerouting module) that satisfies QoS requirements by means of rescheduling individual flows using a software defined networking paradigm. The switches are OpenFlow enabled; therefore, there is a central controller communicating with each switch via OpenFlow protocol. Each switch has a forwarding table used for routing the packets of the flow. In order to configure the network, the controller updates the switch's forwarding table. The network topology and current flow matrix can be obtained via querying the switches. We suppose that there is an estimation of traffic based on the current flow matrix and the history of network flows.

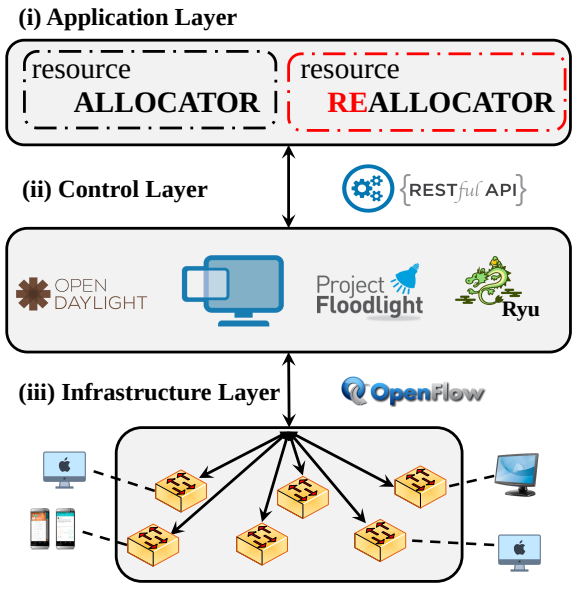

Figure 1. The architecture of the proposed scheme. The architecture is composed of three main layers (i) Application Layer (the upper layer); (ii) Control Layer (the middle layer); and (iii) Infrastructure Layer (the down layer). Our main contributions are in the Application Layer.

An outline of the resource allocation in our scheme is presented in Algorithm 1. Consider the network is in a steady state and a new flow is arrived (line 2 of the algorithm). The controller allocates the resources to the mentioned flow based on typical routing protocols such as ECMP [29] (line 3 of the algorithm). The controller calculates the utilization rate of the links periodically. If the utilization rate of the network links passes a predefined threshold, the controller reallocates the network resources using the proposed scheme (lines 4 and 5 of the algorithm). To this end, QRTP uses the current network traffic and the estimated one, to reschedule the traffic flows in a way that minimizes the packet loss. Figure 2 illustrates a visual process of the proposed scheme. As can be seen, in this architecture, a typical routing algorithm routes one flow in each step while the proposed rerouting algorithm reroutes several flows to reduce the packet loss and provides a better QoS to the users.

Algorithm 1 : Outline

```
1:  while true do
2:      if new flow arrives then
3:          Traditional Routing Algorithm
4:      else if congestion is detected or predicted then
5:          Proposed Rerouting Algorithm
6:          Reset timer
7:      else if timer elapses then
8:          Proposed Rerouting Algorithm
9:          Reset timer
10:     end if
11: end while
```

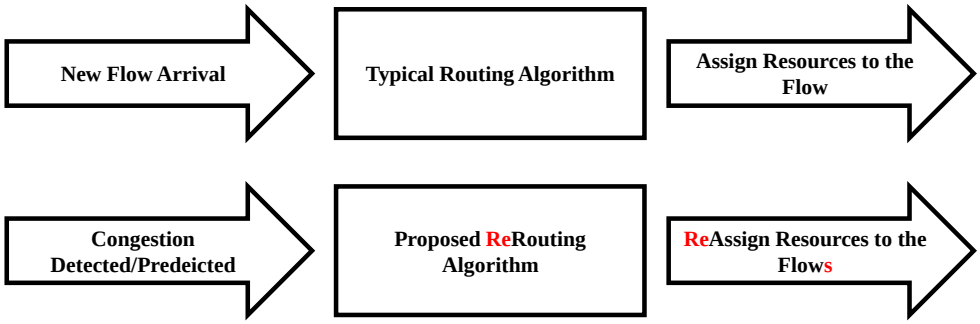

Figure 2. Outline of the proposed schemes. The proposed scheme can work with any of existing routing algorithm. In this architecture, a typical routing algorithm routes one flow in each step while the proposed rerouting algorithm reroutes several flows to reduce the packet loss and provides a better QoS (quality of service) to the users.

4. System Model

There are n OpenFlow enabled switches that are connected to each other based on the matrix $\mathbf{B}_{(n\times n)}$, where $\mathbf{B}_{ij}$ determines the bandwidth of the link from switch i to switch j. The number of the flows in the network is p. The routing matrix $\mathbf{A}_{(n\times n\times p)}$ specifies the path selected for each flow, e.g., if $\mathbf{A}_{ij}^{f} \in \{0,1\}$ is equal to 1, then the flow f crosses the link $i \rightarrow j$. Suppose that there are multiple classes of traffic in the network. The flow requirement matrix $\mathbf{C}_{(p\times 2)}$ indicates flows' quarantined requirements based on the corresponding class. The i-th row of flow requirement matrix contains $Delay$, $Rate$, where $Delay$ clarifies the maximum tolerable propagation delay of flow i and $Rate$ defines the guaranteed bandwidth for the flows. Current flow matrix is represented by FM_p, where FM_f (or $FM[f]$) is recent bandwidth consumed by flow f. Similarly, the parameter PF_p is the predicted flow matrix that states the forecasted bandwidth for each flow. Table 1 reports the main notations of the paper.

Table 1. Main Notation.

Symbol	Definition
Mathematical Parameters	
n	Number of switches
p	Number of flows
$FM_{1\times p}$	Current flow matrix
$PF_{1\times p}$	Predicted flow matrix
$MS_{1\times p}$	Maximum size of flow
$\mathbf{B}_{n\times n}$	Link bandwidth matrix
$\mathbf{C}_{p\times 2}$	Flow requirement matrix
$\mathbf{D}_{n\times n}$	Link delay matrix
$s_{1\times p}$	Source switch of flows
$d_{1\times p}$	Destination switch of flows
$\mathbf{B}_{n\times n}$	Link bandwidth matrix
μ	Maximum link utilization
Decision Variable	
$\mathbf{A}_{n\times n\times n}$	Routing matrix

Problem Formulation

For the sake of simplicity, we set the following equation:

$$MS_f = \max\left(PF_f,\, FM_f\right). \tag{1}$$

We recall that the main aim of this paper is to dynamically and efficiently allocate resources in a way that (1) guarantees QoS requirements of different applications; and (2) proactively prevent congestion and resource waste. To this end, the routing matrix can be obtained such that it minimizes packet loss subject to QoS (i.e., Equations (3) and (4)) and flow conservation constraints (i.e., Equations (5)–(8)). The formulation is as follows:

$$\min \max_{\forall i,j \in \{1,\ldots,n\}} \frac{\sum_{f=1}^{p} \mathbf{A}_{ij}^{f} \cdot MS_f}{\tau \cdot \mathbf{B}_{ij}}, \tag{2}$$

subjected to:

$$\sum_{i=1}^{n} \mathbf{A}_{i\,s_f}^{f} + \sum_{i=1}^{n} \mathbf{A}_{d_f\,i}^{f} = 1 - \sum_{i=1}^{n} \mathbf{A}_{s_f i}^{f} = 1 - \sum_{i=1}^{n} \mathbf{A}_{d_f\,i}^{f} = 0, \quad \forall f \in \{1,\ldots,p\}, \tag{3}$$

$$\sum_{f=1}^{p} \mathbf{A}_{ij}^{f} \mathbf{C}_{f2} \leq \mu \mathbf{B}_{ij}, \quad \forall i,j \in \{1,\ldots,n\}, \tag{4}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{A}_{ij}^{f} \mathbf{D}_{ij} \leq \mathbf{C}_{f1}, \quad \forall f \in \{1,\ldots,p\}, \tag{5}$$

$$\sum_{j=1}^{n} \mathbf{A}_{ij}^{f} = \sum_{j=1}^{f} \mathbf{A}_{ji}^{f}, \quad \forall f \in \{1,\ldots,p\}, \forall i \in \{1,\ldots,n\} - \{s_f, d_f\}, \tag{6}$$

$$\sum_{j=1}^{n} \mathbf{A}_{ij}^{f} \leq 1, \quad \forall i \in \{1,\ldots,n\}, \forall f \in \{1,\ldots,p\}, \tag{7}$$

$$\mathbf{A}_{ij}^{f} \in \{0,1\}, \quad \forall f \in \{1,\ldots,p\}, \forall i,j \in \{1,\ldots,n\}, \tag{8}$$

where Equation (3) prevents flows from returning to the source switches while it makes the flows stay on the destination switches. Furthermore, it forces the flows to leave the origin switches and enter the destination ones. In other words, Equation (3) is the flow conservation constraint. Inequality (4) guarantees the link load to be smaller than the maximum target utilization μ. Equation (5) states delay requirement of flows. To this end, it calculates the delay of the selected path for each flow. This calculated value must be less than the maximum tolerable delay of the flow. When a switch is not the source or destination of a specified flow, the flow must leave that switch if it moves in. This limitation is met by Equation (6). Equation (7) makes sure that there is no loop in the routing matrix. Since the optimization problem is in form of Binary Linear Programming, it is a branch of integer linear programming that can be easily solved by the MATLAB (MATLAB version: R2016b (9.1.0.441655) 64-bit (win64) 7 September 2016, MathWorks Company, USA, 1984–2016, Professional License) convex programming language (CVX) toolbox.

5. Proposed Solution

In this section, a problem relaxation is proposed via forwarding table entries compression and proposing an upper bound function for the objective function.

5.1. Forwarding Table Entries Compression

In order to speed up our scheme, all flows that are under a specified threshold in size are merged into one flow until the new flow rate is below the threshold. To this end, we set a predetermined threshold and merge all flows with a similar destination, source and flow size less than the threshold into a new flow. The flows requirement is not involved i.e., the tolerable delay of the new flow is a minimum of the original flow's tolerable delay. An increment in the threshold decreases the optimization time while increasing the probability of sub-optimality. Most flows in the data centers are small in size ($\leq$10 KB) [30]; therefore, they can significantly reduce the number of flows in the context.

Considering the upper bound of the compression as 120 Kb/s, in the forwarding Table 2, flows 1, 2, and 4 are combined into a new flow with 112 Kb/s transmission rate (first element of Table 3). Similarly, flows 3 and 7 from Table 2 are combined into a new flow (second element of Table 3). Due to the fact that none of the new flows can exceed 112 Kb/s rate, flows 7 and 8 cannot be combined with each other.

Table 2. Sample flow specification.

Flow	Source Switch	Destination Switch	Rate
1	A	B	32 Kb/s
2	A	B	48 Kb/s
3	B	D	64 Kb/s
4	A	B	32 Kb/s
5	X	Y	640 Kb/s
6	A	B	80 Mb/s
7	B	D	12 Kb/s
8	A	B	45 Kb/s

Table 3. Sample flow specification.

Flow	Source Switch	Destination Switch	Rate
Aggregate 1	A	B	112 Kb/s
Aggregate 2	B	D	76 Kb/s
5	X	Y	640 Kb/s
6	A	B	80 Mb/s
8	A	B	45 Kb/s

5.2. Relaxing Objective Function

To cut the optimization time of the proposed scheme, an upper bound function of the objective function is offered. The upper bound function is to minimize the sum of all link utilization instead of minimization of maximum link utilization. The function is as follows:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\sum_{f=1}^{p}\mathbf{A}_{ij}^{f}\cdot\max(PF_f,TM_f)}{\tau\cdot\mathbf{B}_{ij}}. \tag{9}$$

To prevent violation of the links' bandwidth constraints, Equation (10) is subjoined. Although it decreased the optimization time, the γ must be selected wisely. It should grow up to more than 1, down to the fact that packet loss is unavoidable as the user's traffic exceeds the network capacity in peak times:

$$\sum_{f=1}^{p}\mathbf{A}_{ij}^{f}\cdot MS_f\leq\gamma\mathbf{B}_{ij}. \tag{10}$$

6. Description Scenario

In the following subsections, the experimental setup and experimental evaluation are detailed.

6.1. Experimental Setup

In this subsection, we describe the simulation setup. The proposed analytic model is evaluated on the real network topology shown in Figure 3. We use real network traffic that belongs to a university data center traffic that can be found in [17]. In this way, the size of flows are taken from the dataset; however, we assign IP addresses to the switches randomly due to lack of information about the IP layout. Link delays are weighed equally. It should be mentioned that the link's bandwidth is reduced to create a critical condition for the resource allocator. In order to generate flows with a predefined rate, we consider all flows using user datagram protocol (UDP) and we used the CVX toolbox to solve the optimization problem.

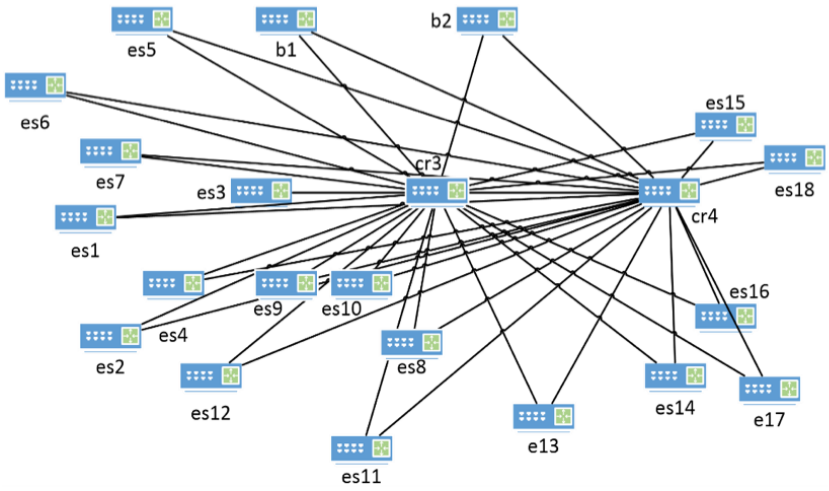

Figure 3. Real data center topology.

6.2. Experimental Metrics

We consider five different metrics (packet loss, reconfiguration overhead, link utilization, maximum link utilization, and computational complexity) in the simulation to measure the proposed scheme. These metrics are:

1. *Packet loss*: the ratio of the number of lost packets to the total number of sent packets;
2. *Reconfiguration network effect*: the number of elements that should be set into the switches forwarding tables; and
3. *Link utilization*: the percentage of a network's bandwidth that is currently being consumed by the SDN traffic;
4. *Maximum link utilization*: the utilization of the link that has the maximum amount of link utilization.
5. *Computational complexity*: the time in seconds that it takes to run each set of flow in QRTP and relaxed QRTP (RQRTP).

7. Simulation Results

In this section, the proposed scheme is evaluated via different measurement metrics.

7.1. Packet Loss

Before the evaluation of the proposed schemes, a simple topology is considered to show the impacts of flow matrix prediction on the QRA algorithms. Figure 4 illustrates a small topology with four switches in order to clarify prediction impact on total packet loss. As it can be seen, there are three flows with sizes of 160, 170 and 20 MB/s, respectively. For the sake of simplicity, we assume that there is no change in the size of flows 2 and 3, e.g., consider flows 2 and 3 are constant rate video streaming.

On the other hand, flow 1 is a Hadoop connection and increases the rate of the flow during the time. Consider there is an estimation algorithm that forecasts b MB/s increment in the size of the first flow.

The impact of this increment (b) on the total packet loss is depicted in Figure 5. As can be seen, the total packet loss is susceptible to the amount of flow increments. Conventional approaches incur lots of packet loss due to lack of information about the future behavior of flows. As an example, in order to minimize the maximum link utilization, conventional approaches routes flows 1 and 3 via the path $1 \rightarrow 2 \rightarrow 4$; similarly, the path $1 \rightarrow 3 \rightarrow 4$ will be selected for flow 2. Therefore, although the maximum link utilization is 95% for current traffic, the packet loss ratio goes up by increasing the first flow size. This happens because the conventional approaches' routes flow based on their current state without considering their future patterns and do not change the network configuration during the different time slots. On the other hand, our scheme (which is a prediction based approach) selects $1 \rightarrow 2 \rightarrow 4$ and $1 \rightarrow 3 \rightarrow 4$ for flows {1} and {2, 3}, respectively.

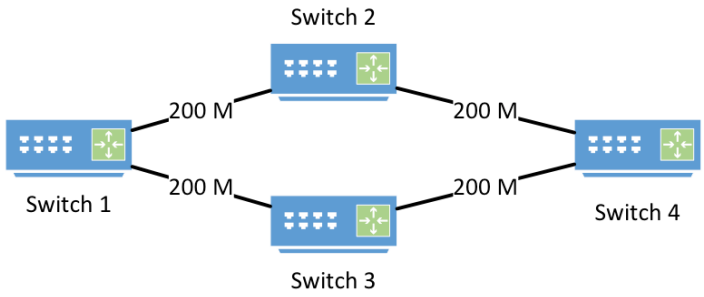

Figure 4. A sample four nodes topology.

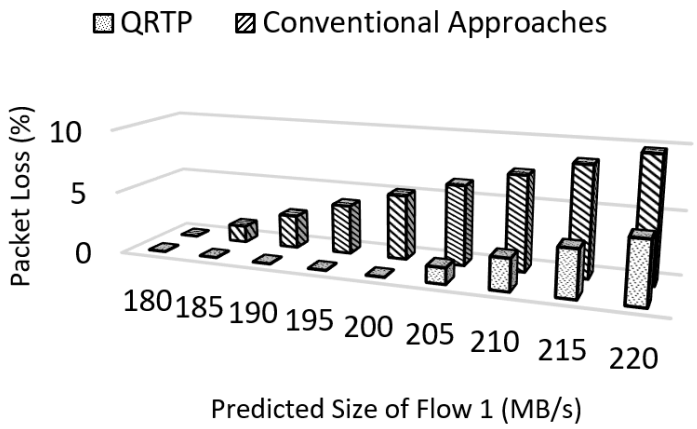

Figure 5. Prediction impact on packet loss. The impact of the proposed algorithm is more clear when the size of the traffic flows increases during the time (transmission control protocol (TCP) connections have similar patterns).

It can be shown that, although the maximum link utilization for current flow matrix is 100%, the total packet loss decreases dramatically. In this case, the total packet loss is decreased significantly, in comparison with conventional approaches. It should be mentioned that the network capacity for flows from switch 1 to 4 is 400 MB/s, which means that the packet loss is unavoidable when the total flows size goes far from this threshold, i.e., the packet loss is unavoidable when the size of flow 1 goes up from 200 MB/s, which is illustrated in Figure 5.

7.2. Link Utilization and Throughput

Figure 6 shows the link utilization in the aforementioned scenario. Proper resource allocation makes all the links highly utilized while improper ones overload some links and under-utilize others. Therefore, the network throughput decreases sufficiently. The effect of comprehensive resource allocation on the network throughput is depicted in Figure 7.

Considering flow 1 = 100 MB/s, flow 2 = 110 MB/s, and flow 3 = 90 MB/s, the network throughput in both approaches is the same. Increasing the size of flow 1 to 110 MB/s does not have a negative impact on the approaches, since, in both approaches, the resources are assigned to the flows in a way that they can tolerate this amount of traffic rate increment. However, when the size of flow 1 increases more than 110 MB/s, in the configuration which is done by the conventional approaches, the link between S1 to S3 becomes a hot-spot and could not handle this much traffic load. On the other hand, the proposed scheme can handle up to 200 MB/s flow rate increment. Therefore, the throughput increases up to 400 MB/s in our scheme, while it cannot exceed 310 MB/s in conventional approaches.

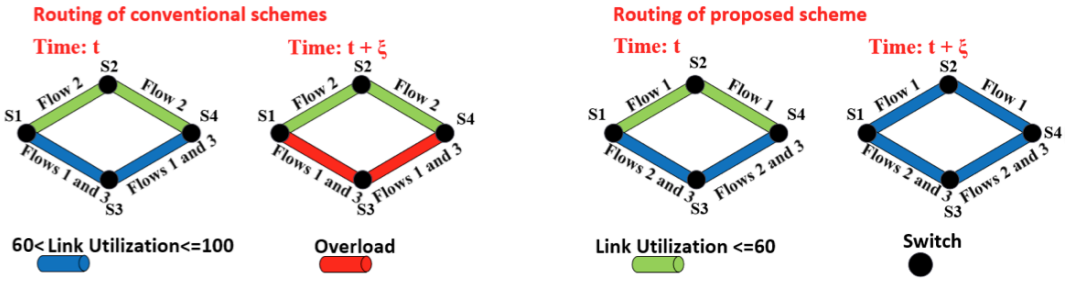

Figure 6. Prediction effect on the link utilization.

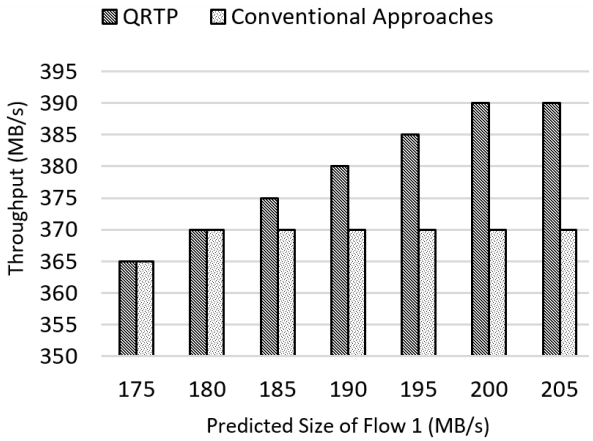

Figure 7. Prediction impact on throughput. Conventional algorithms incur congestion when there are big flows while the proposed algorithm exploits the prediction algorithms along with the resource re-allocation ability of (Software defined networks) SDNs to increase the network throughput.

7.3. Network Reconfiguration Effect

In order to evaluate the performance and complexity of the proposed schemes on the real networks, we consider the topology shown in Figure 3. Three parameters are evaluated: (1) number of routing matrix elements; (2) maximum link utilization and (3) optimization time. To investigate the proposed schemes from the computational complexity perspective, the impact of forwarding table entries compression on the number of routing matrix elements is examined.

Figure 8 depicts the *number of routing matrix element* versus the *number of active flows* in the network. Based on this figure, it is shown that the number of routing matrix elements dramatically decreases due to forwarding table entries compression. As it can be seen, the effect of this compression method is multiplied by increasing the number of flows. This happens because increasing the number of active flows increases the probability of flow aggregation and consequently decreases the number of routing matrix elements.This behavior makes the proposed compression technique proper for practical environments.

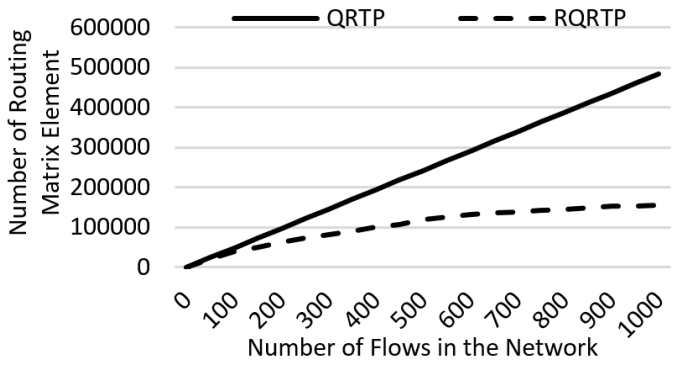

Figure 8. Comparison of QRTP (QoS-aware resource reallocation traffic prediction) and relaxed QRTP (RQRTP) based on the number of routing matrix element. Increasing the number of active flows in the network makes the superiority of RQRTP more clear. This happens because RQRTP exploits flow aggregation technique to reduce the number of routing matrix elements.

7.4. Computational Complexity and Maximum Link Utilization

In this subsection, the execution time of the QRTP algorithm and RQRTP are compared. Section 6 presents the configuration of the system used to execute the algorithms.

The results of computational complexity analysis are stated in Table 4. The optimization time of QRTP increases exponentially by increasing the number of active flows; however, since RQRTP decreases the number of active flows by aggregating the number of inputs, it has a dramatically lower optimization time. Additionally, QRTP has an absolute value in its objective function, which increases the optimization time of QRTP compared with RQRTP. As shown, the execution time of RQRTP is very low; hence, it can be used as a solution to reconfigure the network for resource reallocation.

Table 4. Computational complexity.

Flow	QRTP (s)	RQRTP (s)
10	2.16	2.12
50	2.32	2.31
100	3.16	2.66
200	27.89	2.98
300	40.89	4.22
400	106.70	27.53
500	178.13	41.45
1000	1227.26	46.07
2000	>1 h	88.77

Note that we ignore problem transformation time, i.e., all time comparisons are based on the optimization time. With the purpose of exploring the effect of relaxation on the accuracy of the solution, the maximum link utilization is calculated. In Figure 9, the maximum link utilization of the QRTP and Relaxed QRTP (RQRTP) is compared. As can be seen, the results of these schemes are the same while the optimization time of the relaxed version is sufficiently lower in a large number of flows.

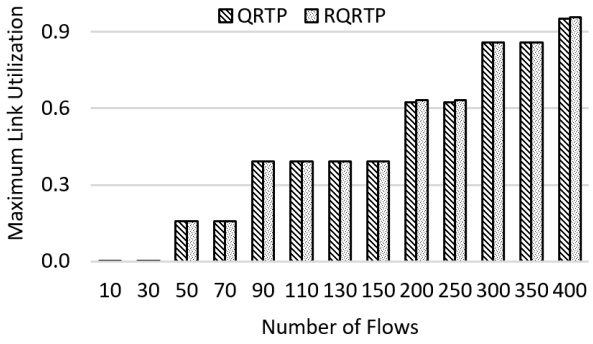

Figure 9. Comparison of link utilization.

8. Conclusions

In this paper, a QoS-aware resource reallocation algorithm was introduced that exploits software defined networking advantages to proactively prevent congestion by using traffic forecasting techniques. In this way, an optimization problem in the form of binary linear programming was proposed. The scheme that solves the proposed problem (i.e., QRTP) was compared with the conventional approaches from throughput and packet loss perspective. Based on the experimental result, QRTP can reduce the total packet loss more than 22%, and simultaneously improves the network throughput more than 1.2 times. In addition, we compared the two proposed schemes from different aspects. Three metrics were considered: (i) maximum link utilization; (ii) the size of routing matrix; and (iii) computational complexity (based on optimization time). Simulation results showed that, although the result of RQRTP is comparable with the QRTP, the optimization time of relaxed solution is dramatically lower than the original solution. In order to prove this claim, it was shown that RQRTP improved the optimization time more than 50 times with 400 flows in the network. Moreover, the size of routing matrix decreased more than 3 times while the maximum link utilization was approximately the same.

Future work will be dedicated to applying the proposed scheme on service function chaining. Another field of future interest would be considering the queuing delay of the switches and the processing delay of the server to improve the QoSs. In addition, considering the energy consumption of the network devices as a new part of objective function would be of interest.

Acknowledgments: Authors want to thank the editor and the reviewers for their valuable comments and suggestions that helped us to improve the quality of our paper. Also, they would like to thank Zahra Pooranian Research Associate at the University of Padova, Italy for her kind comments and advices.

Author Contributions: All authors together proposed, discussed and finalized the main idea of this work. Mohammad Mahdi Tajiki and Mohammad Shojafar proposed the idea, calculated and plotted the feasible regions, and implemented the algorithm and their comparisons. Behzad Akbari and Nader Mokari helped in the paper preparations, English corrections and submission.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

ACS	Ant colony system
API	Application programming interface
ECMP	Equal-cost multi-path routing
FTP	File transfer protocol
GA	Genetic algorithm
ILP	Integer linear programming
OF	OpenFlow
OTT	Over the top

QoS	Quality of service
QRA	QoS-aware resource allocation
QRTP	QoS-aware resource reallocation traffic prediction algorithm
RQRTP	Relatexed QoS-aware resource reallocation traffic prediction algorithm
SDN	Software defined network
SLA	Service level agreement
VM	Virtual machine

References

1. Wang, N.; Ho, K.; Pavlou, G.; Howarth, M. An overview of routing optimization for internet traffic engineering. *IEEE Commun. Surv. Tutor.* **2008**, *10*, 36–56.
2. Hong, C.Y.; Kandula, S.; Mahajan, R.; Zhang, M.; Gill, V.; Nanduri, M.; Wattenhofer, R. Achieving high utilization with software-driven WAN. *ACM SIGCOMM Comput. Commun. Rev.* **2013**, *43*, 15–26.
3. Jain, S.; Kumar, A.; Mandal, S.; Ong, J.; Poutievski, L.; Singh, A.; Venkata, S.; Wanderer, J.; Zhou, J.; Zhu, M.; et al. B4: Experience with a globally-deployed software defined WAN. *ACM SIGCOMM Comput. Commun. Rev.* **2013**, *43*, 3–14.
4. Medina, A.; Taft, N.; Salamatian, K.; Bhattacharyya, S.; Diot, C. Traffic matrix estimation: Existing techniques and new directions. *ACM SIGCOMM Comput. Commun. Rev.* **2002**, *32*, 161–174.
5. Kumar, P.; Yuan, Y.; Yu, C.; Foster, N.; Kleinberg, R.; Soulé, R. Kulfi: Robust traffic engineering using semi-oblivious routing. *arXiv* **2016**, arXiv:1603.01203.
6. Applegate, D.; Cohen, E. Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs. In Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Karlsruhe, Germany, 25–29 August 2003; ACM: New York, NY, USA, 2003; pp. 313–324.
7. Shojafar, M.; Cordeschi, N.; Baccarelli, E. Energy-efficient adaptive resource management for real-time vehicular cloud services. *IEEE Trans. Cloud Comput.* **2016**, *99*, 1–14.
8. Curtis, A.R.; Kim, W.; Yalagandula, P. Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection. In Proceedings of the 2011 Proceedings IEEE INFOCOM, Shanghai, China, 10–15 April 2011; pp. 1629–1637.
9. Benson, T.; Anand, A.; Akella, A.; Zhang, M. MicroTE: Fine grained traffic engineering for data centers. In Proceedings of the Seventh Conference on Emerging Networking EXperiments and Technologies, Tokyo, Japan, 6–9 December 2011; ACM: New York, NY, USA, 2011; p. 8.
10. Liotou, E.; Tseliou, G.; Samdanis, K.; Tsolkas, D.; Adelantado, F.; Verikoukis, C. An SDN QoE-Service for dynamically enhancing the performance of OTT applications. In Proceedings of the 2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX), Pylos-Nestoras, Greece, 26–29 May 2015; pp. 1–2.
11. Tajiki, M.M.; Akbari, B.; Mokari, N. Optimal Qos-aware network reconfiguration in software defined cloud data centers. *Comput. Netw.* **2017**, *120*, 71–86.
12. Tajiki, M.M.; Salsano, S.; Shojafar, M.; Chiaraviglio, L.; Akbari, B. Energy-efficient Path Allocation Heuristic for Service Function Chaining. In Proceedings of the 2018 21th Conference on Innovations in Clouds, Internet and Networks (ICIN), Paris, France, 20–22 February 2018; pp. 1–8.
13. Tajiki, M.M.; Salsano, S.; Shojafar, M.; Chiaraviglio, L.; Akbari, B. Joint Energy Efficient and QoS-aware Path Allocation and VNF Placement for Service Function Chaining. *arXiv* **2017**, arXiv:1710.02611.
14. Vilalta, R.; Mayoral, A.; Pubill, D.; Casellas, R.; Martínez, R.; Serra, J.; Verikoukis, C.; Muñoz, R. End-to-End SDN orchestration of IoT services using an SDN/NFV-enabled edge node. In Proceedings of the 2016 Optical Fiber Communications Conference and Exhibition (OFC), Anaheim, CA, USA, 20–24 March 2016; pp. 1–3.
15. Khoshbakht, M.; Tajiki, M.M.; Akbari, B. SDTE: Software Defined Traffic Engineering for Improving Data Center Network Utilization. *Int. J. Inf. Commun. Technol. Res.* **2016**, *8*, 15–24.
16. Egilmez, H.E.; Civanlar, S.; Tekalp, A.M. An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks. *IEEE Trans. Multimed.* **2013**, *15*, 710–715.
17. Egilmez, H.E.; Gorkemli, B.; Tekalp, A.M.; Civanlar, S. Scalable video streaming over OpenFlow networks: An optimization framework for QoS routing. In Proceedings of the 2011 18th IEEE International Conference on Image Processing (ICIP), Brussels, Belgium, 11–14 September 2011; pp. 2241–2244.

18. Egilmez, H.E.; Dane, S.T.; Bagci, K.T.; Tekalp, A.M. OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks. In Proceedings of the 2011 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPAASC), Hollywood, CA, USA, 3–6 December 2012; pp. 1–8.
19. Egilmez, H.E.; Civanlar, S.; Tekalp, A.M. A distributed QoS routing architecture for scalable video streaming over multi-domain OpenFlow networks. In Proceedings of the 2012 19th IEEE International Conference on Image Processing (ICIP), Orlando, FL, USA, 30 September–3 October 2012; pp. 2237–2240.
20. Civanlar, S.; Parlakisik, M.; Tekalp, A.M.; Gorkemli, B.; Kaytaz, B.; Onem, E. A qos-enabled openflow environment for scalable video streaming. In Proceedings of the 2010 IEEE Globecom Workshops (GC Wkshps), Miami, FL, USA, 6–10 December 2010; pp. 351–356.
21. Liang, B.; Yu, J. One multi-constraint QoS routing algorithm CGEA based on ant colony system. In Proceedings of the 2015 2nd International Conference on Information Science and Control Engineering (ICISCE), Shanghai, China, 24–26 April 2015; pp. 848–851.
22. Leela, R.; Thanulekshmi, N.; Selvakumar, S. Multi-constraint qos unicast routing using genetic algorithm (muruga). *Appl. Soft Comput.* **2011**, *11*, 1753–1761.
23. Kulkarni, S.; Sharma, R.; Mishra, I. New QoS routing algorithm for MPLS networks using delay and bandwidth constraints. *Int. J. Inf.* **2012**, *2*, 285–293.
24. Zhao, J.; Ge, X. QoS multi-path routing scheme based on ACR algorithm in industrial ethernet. In *Proceedings of the Third International Conference on Communications, Signal Processing, and Systems*; Springer: Cham, Switzerland, 2015; pp. 593–601.
25. Wang, J.M.; Wang, Y.; Dai, X.; Bensaou, B. SDN-based multi-class QoS-guaranteed inter-data center traffic management. In Proceedings of the 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), Luxembourg, 8–10 October 2014; pp. 401–406.
26. Ghosh, A.; Ha, S.; Crabbe, E.; Rexford, J. Scalable multi-class traffic management in data center backbone networks. *IEEE J. Sel. Areas Commun.* **2013**, *31*, 2673–2684.
27. Ongaro, F. Enhancing Quality of Service in Software-Defined Networks. Ph.D. Thesis, Computer Engineering Department, University of Bologna, Bologna, Italy, 2014.
28. Haleplidis, E.; Pentikousis, K.; Denazis, S.; Salim, J.H.; Meyer, D.; Koufopavlou, O. *Software-Defined Networking (SDN): Layers and Architecture Terminology*; Technical Report; Internet Research Task Force (IRTF): Vancouver, BC, Canada, 2015.
29. Hopps, C. *RFC 2992: Analysis of an Equal-Cost Multi-Path Algorithm [OL]*; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2000.
30. Theophilus, B. New Paradigms for Managing the Complexity and Improving the Performance of Enterprise Networks. Ph.D. Thesis, University of Wisconsin-Madison, Madison, WI, USA, 2012.