Optimizing Non-Uniform Bandwidth Reservation Based on Meter Table of Openflow*

Liaoruo HUANG†, Qingguo SHEN†(a), Nonmembers, and Zhangkai LUO†, Member

SUMMARY Bandwidth reservation is an important way to guarantee deterministic end-to-end service quality. However, with the traditional bandwidth reservation mechanism, the allocated bandwidth at each link is by default the same without considering the available resource of each link, which may lead to unbalanced resource utilization and limit the number of user connections that network can accommodate. In this paper, we propose a non-uniform bandwidth reservation method, which can further balance the resource utilization of network by optimizing the reserved bandwidth at each link according to its link load. Furthermore, to implement the proposed method, we devise a flexible and automatic bandwidth reservation mechanism based on meter table of Openflow. Through simulations, it is showed that our method can achieve better load balancing performance and make network accommodate more user connections comparing with the traditional methods in most application scenarios.

key words: bandwidth reservation, QoS, Openflow, meter table

1. Introduction

To provide deterministic quality of service (QoS) for real-time streaming, financial transactions, virtual private networks and other critical network applications, bandwidth reservation is usually used in the connection of two remote sites across the wide area network. As these applications have contributed most of the revenue for the Internet Service Providers (ISP), it is of great significance to optimize the load balancing performance of network, thus to accommodate more user connections of these applications. For this reason, a number of bandwidth reservation scheduling methods have been proposed. Some of them try to solve the problem from the spatial way using traffic engineering mechanisms, by which the traffic load can be more evenly distributed to improve the load balancing level [1]. Other methods schedule the reservation requests from temporal way by arranging the appropriate transmission time for some requests to avoid network busy hour, so that the resource at idle time can be utilized to admit more connections [2]. Although these methods can improve the load balancing level to a certain extent, however, we argue that the performance of these methods can still be improved by further optimizing the reserved bandwidth at each link.

In practice, the reserved bandwidth for each connection need to be computed according to the user’s QoS requirement including throughput, delay, jitter and packet loss. Since there is no packet loss in the connection with bandwidth reservation and eliminating jitter is mainly depending on the packet caching rather than bandwidth reservation, in this paper, we mainly concern throughput and end-to-end delay in the determination of reserved bandwidth.

As shown in Fig. 1, in uniform bandwidth reservation, the reserved bandwidth is computed according to the end-to-end delay requirement and the resulted value is usually larger than the required throughput in order to accelerate the transmission of packets. With the traditional uniform bandwidth reservation implemented by Resource Reservation Protocol (RSVP), the over-provisioned bandwidth at each link is by default the same without considering the available resource of each link. Thus unbalanced link resource utilization may occur, resulting in a bottleneck link in the network. In this paper, we argue that the over-provisioned bandwidth at each link is unnecessary the same and should be flexibly adjusted according to the available resource as Fig. 1 shows. In this way, the resource utilization of each link can be further balanced and more user connections can be accommodated. However, implementing flexible resource allocation is always difficult as it may need large overhead or complex mechanisms to configure individual device. To solve this problem, with the overall view and network programmability provided by Software Defined Network (SDN), we devise an automatic bandwidth reservation mechanism based on meter table of Openflow to implement the proposed non-uniform bandwidth reservation (NUBR) mechanism.

The contributions of this paper can be concluded into two parts: (1) we propose a non-uniform bandwidth reservation method which can allocate different bandwidth at each link according to available resource while also fulfilling the end-to-end QoS requirement. Thus, the resource utilization of each link can be further balanced; (2) we devise an automatic and flexible bandwidth reservation mechanism based

---

*The authors are with College of Communications Engineering, PLA Army Engineering University, China.
†This work was supported by the National Natural Science Foundation of China [grant numbers 61271254].
a) E-mail: shennq2016@163.com
DOI: 10.1587/transinf.2017EDL8261
on the meter table of Openflow to implement the proposed non-uniform bandwidth reservation.

2. Non-Uniform Bandwidth Reservation (NUBR)

We denote the network as a directed graph \( G(E, V) \), where \( E \) and \( V \) represent the set of edges and vertexes respectively. In this paper, we mainly study reserving bandwidth for user connections on a given end-to-end path. So we assume that the route for the user connection has been established before bandwidth reservation. The QoS requirement of user connection can be expressed as a two-tuples \( R = (\text{rate} \geq R_{sla}, \text{delay} \leq D_{sla}) \), where \( R_{sla} \) represents the minimum required throughput and \( D_{sla} \) represents the maximum end-to-end delay. We suppose that flow of each connection is shaped by a leaky bucket with size of \( S_{bucket} \) and rate of \( R_{sla} \) at the edge of the network. Therefore, the queuing delay occurs mainly in the network edge and its maximum value can be represented by \( S_{bucket}/R_{sla} \). After shaping, the delay of the packet in the core of the network only consists of transmission delay and propagation delay. So the upper bound of the end-to-end delay \( D_{e2e} \) after bandwidth reservation can be expressed as:

\[
D_{e2e} = \frac{S_{bucket}}{R_{sla}} + \sum_i \left( \frac{L_{max}}{C_i} + \frac{MTU}{T_i} \right)
\]

(1)

where \( L_{max} \) is the maximum packet size in user session, \( MTU \) is the maximum transmission unit of the network, \( C_i \) is the reserved bandwidth at the \( i \)th link, \( T_i \) is the maximum transmission rate of the \( i \)th link. Therefore, the mathematical model of non-uniform bandwidth reservation can be represented as follows:

**Theorem 1 (Non-Uniform Bandwidth Reservation).**

The non-uniform bandwidth reservation method meets:

\[
\begin{align*}
S_{bucket}/R_{sla} &+ \sum_i \left( \frac{L_{max}}{C_i} + \frac{MTU}{T_i} \right) \leq D_{sla} \\
R_{sla} &\leq C_i \leq A_i
\end{align*}
\]

(2)

where \( A_i \) is the available bandwidth of the \( i \)th link in the route. Formula (2) represents that the end-to-end delay should be less than the required maximum delay. The reserved bandwidth at each link should be larger than the minimum required throughput and less than the available bandwidth. Apparently, the traditional uniform and flat bandwidth reservation method is the special case of NUBR, where the reserved bandwidth \( C_i \) at each link is the same (denoted as \( C_f \)). Therefore, the end-to-end delay can be expressed as:

\[
D_{e2e} = \frac{S_{bucket}}{R_{sla}} + \sum_i \frac{MTU}{T_i}
\]

(3)

Thereinto, \( n \) is the number of links that flow passes through. Then, we can get the mathematical model of uniform-bandwidth reservation as follows:

**Theorem 2 (Uniform Bandwidth Reservation).** Let the leaky bucket rate equal \( C_f \), then the reserved bandwidth for flow \( f \) with QoS requirement \( R = (R_{sla}, D_{sla}) \) can be expressed as:

\[
C_f = \max \left\{ R_{sla}, \left( S_{bucket} + \sum_i \frac{L_{max}}{T_i} \right), \left( D_{sla} - \sum_i \frac{MTU}{T_i} \right) \right\}
\]

(4)

By comparison of formula (2) and (4), we can see that with the non-uniform bandwidth reservation model, the bandwidth reserved at each link can be flexibly determined instead of being kept the same, while the end-to-end delay requirement can still be fulfilled.

Based on the proposed model above, the reserved bandwidth at each link has to be calculated carefully to optimize the load balance of the network. First, we define the residual bandwidth ratio \( q_i \) of \( i \)th link as:

\[
q_i = A_i/T_i
\]

(5)

In order to balance the resource utilization of each link, the cost of reserving resources should be inversely proportional to the residual resource ratio after reservation. Then we can get the cost function of reserving \( C_i \) at \( i \)th link as:

\[
\text{cost}(C_i) = C_i/(q_i + \sigma) = C_i T_i/(A_i + T_i \sigma)
\]

(6)

where \( \sigma \) is a very small positive constant used to prevent the denominator from becoming zero, thus allowing full bandwidth utilization. In the mathematical model, we want to minimize the total cost of reserving bandwidth at links of the selected route. So the objective function of the optimization model can be expressed as:

\[
\min \sum_i \text{cost}(C_i)
\]

(7)

Then the delay constraint can be represented as:

\[
D_{e2e} = \frac{S_{bucket}}{R_{sla}} + \sum_i \left( \frac{L_{max}}{C_i} + \frac{MTU}{T_i} \right) \leq D_{sla}
\]

(8)

Besides, the \( C_i \) should conform with:

\[
R_{sla} \leq C_i \leq A_i
\]

(9)

The objective function and the constraints are all convex. Then we can get the Lagrangian function represented as \( L(C_i, \lambda, \mu, \nu) \) with parameters of \( \lambda, \mu, \nu \) and variables of \( C_i \) as:

\[
L(C_i, \lambda, \mu, \nu) = \sum_i [C_i/(q_i + \sigma)] + \\
\lambda \left[ \frac{S_{bucket}}{R_{sla}} + \sum_i \left( \frac{L_{max}}{C_i} + \frac{MTU}{T_i} \right) - D_{sla} \right] + \\
\sum_i \mu_i (R_{sla} - C_i) + \sum_i \nu_i (C_i - A_i)
\]

(10)

Then, according to the Karush–Kuhn–Tucker (KKT)
Algorithm 1 Bandwidth Allocation

Input: \( R = (R_{\text{sla}}, D_{\text{sla}}), \bar{A} = \{A_i\} \)

Output: \( \bar{C} = \{C_i\} \)

1: Compute optimal route for the user connection;
2: find the link with largest \( A_i \), then \( R_{\text{sla}} \rightarrow \bar{C} \);
3: while \( D_{\text{e2e}} \neq D_{\text{sla}} \) and \( \bar{C} \leq \bar{A} \)
4: \( \bar{C} \leftarrow \bar{C} + \Delta C \)
5: Compute every other \( C_i \) according to \( \bar{C} \)
6: end while

Conditions, we can get \( \frac{\partial L(C_i, \lambda, \mu, v)}{\partial C_i} = 0, \frac{\partial L(D_{\text{sla}} - C_i)}{\partial C_i} = 0, v_i(C_i - A_i) = 0, D_{\text{e2e}} = D_{\text{sla}}. \) So for the \( R_{\text{sla}} < C_i \leq A_i \), it follows that \( \mu_i = 0, v_i = 0 \) and

\[
C_i/C_j = \sqrt{q_i + \sigma}/\sqrt{q_j + \sigma} \quad (11-1)
\]

and we can rewrite \( D_{\text{e2e}} \) as:

\[
\frac{S_{\text{bucket}}}{R_{\text{sla}}} + \sum_i \left[ \frac{L_{\text{max}}}{(\sqrt{q_i + \sigma}/\sqrt{q_j + \sigma})C_i} + \frac{MTU}{T_i} \right] = D_{\text{sla}}
\]

(11-2)

where \( q \) denotes the maximum value of residual bandwidth ratio in all links, and \( \bar{C} \) denotes the reserved bandwidth at the link with maximum \( \bar{q} \). According to formula (8) and (11-1), (11-2) each \( C_i \) can be calculated as:

\[
C_i = (\sqrt{q_i + \sigma}/\sqrt{q_j + \sigma})C_i - R_{\text{sla}} + R_{\text{sla}} \quad (12)
\]

Function \( (x)^+ \) equals \( x \) when \( x \geq 0 \), and \( 0 \) otherwise. Therefore, we can have the pseudo codes of the algorithm as Algorithm 1. Firstly, we remove the links which cannot provide sufficient bandwidth and update the network topology. Then we find the optimal route for the user connection using routing algorithm with a given policy. In the resulted route, we find the link with maximum value of residual resource ratio (denoted as \( \bar{q} \)) and set \( \bar{C} \) to \( R_{\text{sla}} \). Thus, the reserved bandwidth of every other link can be computed according to \( \bar{q} \) and \( \bar{C} \) based on formula (12). Thereafter, we let \( \bar{C} \) grow from \( R_{\text{sla}} \) to the maximum available bandwidth (denoted as \( \bar{A} \)) with step size \( \Delta C \), then the vector of \( \bar{C} \) that makes \( D_{\text{e2e}} = D_{\text{sla}} \) is the optimal solution. The complexity to find the link with maximum available bandwidth is \( O(n) \) with \( n \) being the hop number. The complexity of calculating each \( C_i \) according to \( \bar{C} \) and \( \bar{q} \) is \( O(1) \). Then the complexity of the algorithm is \( O(n) \).

3. Implementation

To implement the NUBR, we design an automatic and flexible bandwidth reservation mechanism based on software SDN controller OpenDaylight Litmus and SDN switch Open vSwitch 2.3.0. Through northern interface, SDN controller can provide network status information to the upper applications. Then based on the network status and optimal route computed by the routing algorithm, the NUBR algorithm calculates the optimal bandwidth allocation as introduced in Sect. 2. For each user connection in each Open vSwitch node, we realize the flexible bandwidth reservation based on the mechanisms of multi-layer flow table and meter table provided by Openflow as Fig. 2 shows.

In Openflow, meter table is a special kind of flow table to record and police the flow rate. The structure of meter table is shown in Fig. 2 and described in detail in [3]. The working principle of meter table is similar to that of single rate leaky bucket. For a period of time, if the traffic amount is greater than the amount that can be processed, then the arriving packet will be dropped or remarked.

As shown in Fig. 2, firstly, the incoming packets are classified into \( \text{Class 1}, \ldots, \text{Class n} \) and \( \text{Best Effort} \) through matching the flow entries. Through the meter tables corresponding to each class, the total transmit rate of flows belonging to each class is restricted. Therefore, the packets not conforming with the regulations are either dropped or remarked. And the remaining packets are forwarded to the next hop. So by setting the appropriate band type, rate, burst and other parameters of the meter band, we can realize the bandwidth reservation and policing. For example, for a port with maximum transmit rate of 10Mbps and three classes, we set the rate of \( \text{Best effort} \) as 2Mbps and the band type as drop. Similarly, the rate and band type of both \( \text{Class 1} \) and \( \text{Class 2} \) are set as 5Mbps and drop. So the minimum transmit rates of \( \text{Class 1} \) and \( \text{Class 2} \) both are 3Mbps, and the maximum rates are 5Mbps, which is equivalent to reserve 3Mbps bandwidth for \( \text{Class 1} \) and \( \text{Class 2} \) respectively. In recent years, there have been a lot of research on meter table in Openflow [4], [5]. However, to our best knowledge, there still no research on multi-hop bandwidth reservation method with meter table.

4. Evaluation

The simulation topology is an \( n \times n \) grid network as Fig. 3 shows. The maximum transmission rate of the links connecting CE and PE is set to 100Mbps. Other link rates are set to 1Gbps and we assume that only 33% of the total capacity can be used for bandwidth reservation.

In the simulation, we generate user connections between random source and destination with a certain QoS requirement. Then we evaluate the maximum user connections that network can accommodate under different methods. For comparison, uniform bandwidth reservation used
by RSVP is also evaluated. Moreover, we evaluate three types of QoS requirements with different required throughput and end-to-end delay to simulate typical service levels provided by ISPs as follows. Type 1 \( (R_{sla} = 98\text{kbps}; D_{sla} = 100\text{ms}) \) is to simulate applications such as VoIP, financial transactions, critical production messages, etc., which have low requirements for bandwidth, but high requirements for delay. Type 2 \( (R_{sla} = 350\text{kbps}; D_{sla} = 150\text{ms}) \) is to simulate the applications such as interactive video, video conference, electronic games, etc., which require high bandwidth guarantee and low latency. Type 3 \( (R_{sla} = 1024\text{kbps}; D_{sla} = 500\text{ms}) \) is used to simulate the applications like video on demand, big data transferring etc. which have high requirements for bandwidth, but low requirements for delay. In the evaluation, we analyze the performance of NUBR method in combination with typical routing algorithms. The chosen algorithms are Constrained Shortest Path First (CSPF), Widest Path First (WPF) and Minimum Interference Routing Algorithm (MIRA). \( AC \) is set as 1kbps. The simulation is conducted 50 times and the result is averaged. The maximum user connection numbers under different scenarios are shown in Fig. 4.

From Fig. 4, we can see that the maximum user connection number increases with the network scale for all QoS requirement types in both NUBR and RSVP methods. For QoS requirements of Type 1 and Type 2, the improvements over the RSVP are always growing with \( n \) and can be up to 80.9\% (with CSPF) and 29.7\% (with WPF) respectively when \( n \) reaches 6. However, for QoS requirement of Type 3, the NUBR is almost no better than RSVP no matter of routing algorithms. According to formula (1)−(4), the NUBR method proposed in this paper achieves better load balancing mainly through adjusting the over-reserved bandwidth at each link according to the available resource. Therefore, the larger \( C_f - R_{sla} \), the more bandwidth that NUBR can adjust, and the greater improvements over RSVP can be achieved. The \( C_f \) is mainly determined by the \( D_{sla} \) and the hops of end-to-end path as formula (3) defines. Therefore, for the user request with requirements of smaller \( D_{sla} \) and more hops, the NUBR method can achieve larger improvement comparing with RSVP. At the same time, we can see that the difference of improvements achieved by NUBR using different routing algorithm is small (0.0%−8.9\%). So it can demonstrate that the improvement is mainly achieved by the proposed method rather routing algorithm. Therefore, we can conclude from the simulation result that the performance of the NUBR method proposed in this paper is always no worse than that of RSVP no matter of the service type, routing algorithms and network scale, and it can significantly increase the maximum acceptable user request number comparing with RSVP in most application scenarios, especially for the applications with high delay requirement and the network with large scale.

Moreover, in order to verify the effectiveness of the proposed method in the real scene, we will mix the user connections with different QoS requirements, and examine the performance under different actual ISP network topologies such as NSFNet (14 nodes), ChinaNet (32 nodes), Polska (12 nodes) and FatTree (30 nodes). The QoS requirement of each user connection is randomly chosen from the three types and its source and destination are also randomly determined. We evaluate the improvement of the maximum user connection number when using NUBR comparing with the RSVP under three routing algorithms. The experiment is conducted 50 times and the results are averaged.

From Fig. 5, we can see that although the achieved improvement varies in different scenarios, the proposed method can always achieve better performance than the
RSVP regardless of routing algorithm and network topology type. Therefore, we can conclude that the proposed NUBR method can be applied in the realistic application scenarios.

5. Conclusion

In this paper, we propose a non-uniform bandwidth reservation method based on meter table of Openflow, in which the reserved bandwidth at each link along the path is determined optimally according to its link load and can be different with that of another link. Simulation results demonstrate that the proposed solution can make network accommodate more user connections comparing with the traditional method. However, the NUBR method proposed in this paper still needs to be improved. For example, realizing per flow based bandwidth reservation may introduce additional flow entries to implement the functions such as rate control and forwarding. At the same time, more flow entries in the network equipment also leads to more processing time and memory space consumption. Moreover, in order to reserve different bandwidth at each hop, the meter tables as well as flow tables at each node have to be configured individually, thus more complex configuring mechanisms or upper applications have to be developed. In the next step of our research, we will continue to improve the implementation of the proposed method so that the overhead introduced above can be reduced and the proposed method can be more practical.

References

[1] T. Anjali, C. Bruni, D. Iacoviello, and C. Scoglio, “Dynamic bandwidth reservation for label switched paths: An on-line predictive approach,” Computer Communications, vol.29, no.16, pp.3265–3276, May 2006.
[2] L. Zuo and M.M. Zhu, “Concurrent Bandwidth Reservation Strategies for Big Data Transfers in High-Performance Networks,” IEEE Trans. Netw. Serv. Manage., vol.12, no.2, pp.232–247, June 2015.
[3] OpenFlow Switch Consortium, “Openflow switch specification – version 1.5.1,” https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf, 2015.
[4] P. Heise, F. Geyer, and R. Obermaisser, “Deterministic OpenFlow: Performance evaluation of SDN hardware for avionic networks,” 2015 11th International Conference on Network and Service Management (CNSM), Barcelona, pp.372–377, 2015.
[5] N. Kitsuwan and E. Oki, “Traffic splitting technique using meter table in software-defined network,” 2016 IEEE 17th International Conference on High Performance Switching and Routing (HPSR), Yokohama, pp.108–109, 2016.