On-demand network bandwidth reservation combining machine learning and linear programming

Kouichi Genda
College of Engineering, Nihon University
genda.kouichi@nihon-u.ac.jp

***Abstract*— Network bandwidth reservation is a representative service, where users directly reserve network resources on an on-demand basis. It utilizes the advantages of software defined networks, such as flexibility. To provide bandwidth reservation services extensively, an instantaneous response to user requests (e.g., less than 1 s) and a high user request acceptance ratio (e.g., over 90%) are required. In this study, we propose a bandwidth reservation method to meet these two requirements by combining machine learning (ML) and linear programming (LP), particularly for unpredictable bandwidth demands in which the usage time is strictly indicated. In the proposed method, a user request is instantaneously judged through ML, and network resource allocation, including traffic routing, is optimally determined through LP. We demonstrate that the proposed method provides a suboptimal acceptance ratio with a difference of less than 1% compared with the optimal solution and an instantaneous response of less than 0.1 ms under a general computation environment.**

Keywords—Bandwidth reservation, Bandwidth calendaring, Machine learning, Linear programming, Software defined network

I. INTRODUCTION

Bandwidth reservation is a representative service utilizing the attractive characteristics of software defined networks (SDNs), such as flexibility, where users can directly reserve network resources on an on-demand basis. Bandwidth reservation, often called bandwidth calendaring, is already commercially available for enterprises and over-the-top (OTT) operators. For instance, enterprises can dynamically adjust the bandwidth and traffic routes of their leased lines established over an SDN [1].

Bandwidth reservation can be roughly classified into two categories [2]. The first category is bandwidth reservation for predictable demands such as the release of a new service for enterprises and data backups among datacenters (DC's) for OTT operators. The network operators typically know the demands a priori. In addition, such demands are delay tolerant and can be scheduled at any time within a specified time window. The bandwidth reservation with regard to the traffic among DC's has attracted considerable research interest [2, 3]. The second category is bandwidth reservation for unpredictable demands. The demands occur when an arbitrary user needs an infrequent or unexpected network bandwidth. There are some studies on bandwidth reservation for unpredictable demands [4-6].

In this paper, we focus on bandwidth reservation for unpredictable bandwidth demands without delay tolerance in which the usage time is strictly indicated. We propose that, under a mature SDN environment, both personal and business users can use bandwidth reservation services on a daily basis.

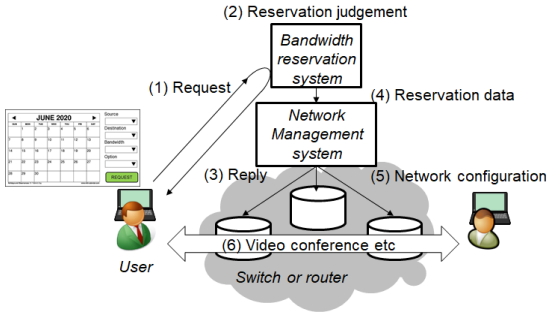


Fig. 1 Illustration of the bandwidth reservation service flow for demands in which the use time is indicated strictly.

Fig. 1 illustrates our target. A user sends a request to a bandwidth reservation system in a network; in the request, information such as the strict usage time, source and destination locations (nodes), and required bandwidth, are indicated. In a bandwidth reservation system, a request is judged as accepted or rejected, and the result is sent back to the user. Information on the accepted requests is sent to the network management system. This network management system establishes traffic routes with the required bandwidth for the accepted requests during their usage time.

For providing bandwidth reservation services extensively, instantaneous responses (e.g., less than 1 s) to user requests are mandatory because users make reservations using an intuitive calendar interface similar to general hotel booking portal sites. In addition, the user request acceptance ratio should be maximized (e.g., more than 90%) not only to satisfy user demands but also to increase network utilization.

However, it is difficult to meet these two service requirements simultaneously. In [6], a mathematical scheduling algorithm to judge a reservation request without delay tolerance is proposed, wherein Dijkstra's algorithm is applied to find a traffic route for a request. In [5], a method based on the K-shortest path (KSP) algorithm [7] is used to determine network resources, including the traffic route for user requests, to achieve instantaneous response. With the Dijkstra and KSP algorithms, as network resources and traffic routes are allocated preferentially to requests arriving earlier and are fixed permanently, the acceptance ratio of user requests is less than optimal. By contrast, the problem of effective bandwidth

978-3-903176-36-2 ©2021 IFIP

allocation to maximize the acceptance ratio of user demands can be solved using a method based on a linear programming (LP) approach. However, it may be difficult to achieve a fast response because of the considerable computation time. Consequently, it is difficult to achieve our target with conventional methods. Thus, a breakthrough is required to allocate network resources instantaneously and optimally for unpredictable bandwidth demands with strict usage time.

We focus on the attractive characteristics of machine learning (ML) technology: its prediction speed, and the suboptimal accuracy. ML has been extensively applied in many application domains such as computation vision and voice recognition [8]. Herein, we tackle the ML technology in the network domain, especially for bandwidth reservation.

The main contributions of this study are as follows:

- A bandwidth reservation method combining the ML and LP technologies is proposed for unpredictable bandwidth demands without delay tolerance.
- It is demonstrated that the proposed method can satisfy the aforementioned two service requirements: instantaneous response and high user request acceptance ratio.

This paper extends the existing work [14] especially in terms of evaluating the overall performance of the proposed method.

The remainder of this paper is organized as follows. Section II discusses the related work on bandwidth reservation methods and the application of ML in the network domain. Section III describes the bandwidth reservation method using LP. Section IV presents the algorithm combining ML and LP, and Section V evaluates its performance. Finally, Section VI concludes the study.

II. Related work

A. Network bandwidth reservation

Bandwidth reservation for predictable demands has attracted considerable research interest, particularly with regard to the traffic between DC's [2, 3]. Kandula *et al*. [3] proposed a bandwidth calendaring solution for large and long-running traffic transfer between DC's, where traffic flows are allocated to maximize the requests served before the deadline. In [2], Gkatzikis *et al*. discussed time-varying bandwidth reservation in an inter-DC environment, when network reconfiguration is not supported.

Bandwidth reservation for unpredictable demands with delay tolerance has been previously examined [4,5]. Tsujino [4] introduced traffic engineering technologies for bandwidth demands with delay tolerance. Considering link load balance over a network, unpredictable demands arriving after bandwidth assignment can be accommodated as much as possible. Kamamura *et al*. [5] proposed a recommendation-based bandwidth reservation for unpredictable bandwidth demands. By recommending certain reservation plans (e.g., future multiple time slots) with multi-grades (e.g., multiple prices), network resources can be efficiently utilized while satisfying user demands. Wu *et al*. [6] proposed bandwidth reservation scheduling of a future time slot for minimizing the impact on the immediate reservation of the next time slot.

B. Application of ML in the network domain

Among the ML applications that have been discussed in the network domain, traffic routing and resource allocation are related to our target. Although we do not discuss all the relevant studies here, some state that reinforcement learning is an effective approach for solving traffic routing and resource allocation problems because it is highly capable of solving decision-making problems [9, 10]. In addition, graph neural network models [12] have been discussed recently for network optimization and traffic management because of their high tolerance for network topology changes. However, effective applications of ML in the network domain have not yet been reported. This is because it is difficult to characterize the appropriate features or appropriate input/output patterns to reflect the highly dynamic and uncertain nature of the network behavior [10,13].

In addition, because ML technology tends to work in the best effort way, it finds it difficult to provide precise and guaranteed solutions [9–11], i.e., ML technology provides suboptimal solutions. Thus, these suboptimal ML solutions need to be adopted carefully. Furthermore, to utilize the characteristics of ML effectively, our approach, which involves ML-based applications in combination with other technologies such as LP, has not yet been studied for network bandwidth reservation.

III. Bandwidth reservation method using LP

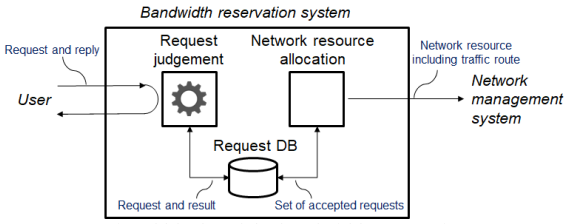
Here, we discuss a bandwidth reservation method using LP to maximize the acceptance ratio of unpredictable bandwidth demands in which the usage time is strictly indicated. This method is premised on the use of SDN with fine-tuned bandwidth management. We define the unit of usage time as a timeslot (TS) (e.g., 1 h), and refer to the strict usage time indicated in a user request as the requested timeslot (Requested TS). Using TS as the unit of the usage time is reasonable because we generally make a reservation, such as for a rental room, in units of one hour.

The bandwidth reservation method using LP is presented in Algorithm 1, when a new request arrives. The request information includes the Requested TS (*cal*), source and destination nodes (s, t), and required bandwidth (b^{st}). Here, we assume that the holding time is fixed as the TS, to simplify the

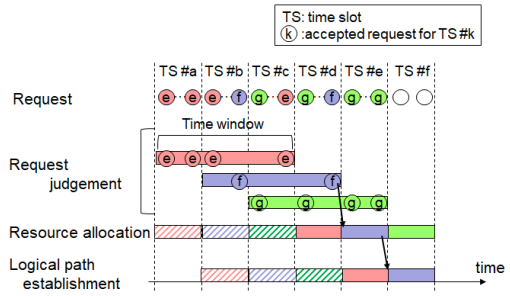
Algorithm 1 Bandwidth reservation method using LP

Input: Request(cal, s, t, b^{st})
Output: Reply

1: Determine the reserved bandwidth of all logical paths at *cal*
2: Temporarily increase the bandwidth of logical path (s, t), F^{st}, at *cal*
$F^{st} \leftarrow F^{st} + b^{st}$
3: Allocate network resource for all logical paths by using LP
4: **If** network resource for all logical paths can be allocated **then**
5: Update the set of requests accepted at *cal*
6: Reply = *accept*
7: **else**
8: Reply = *reject*



(a) Functional blocks of the bandwidth reservation system



(b) Time sequence of the bandwidth reservation process

Fig. 2 Overview of the proposed bandwidth reservation method.

discussion. In this method, when a request arrives, the network resources for all the logical paths at *cal* are reoptimized using LP to solve a multi-commodity maximum flow problem. A request is accepted when the bandwidths for all the logical paths can be allocated.

With bandwidth reservation using LP formulations, the acceptance ratio of user requests at the Requested TS is maximized; however, it is difficult to achieve an instantaneous response time.

IV. Proposed Bandwidth Reservation Method

We propose a bandwidth reservation method combining ML and LP to achieve two service requirements. In the proposed method, request judgment and network resource allocation, including traffic route determination, are performed independently.

Fig. 2 depicts an overview of the proposed bandwidth reservation method; Fig. 2(a) shows the bandwidth reservation system comprising three functional blocks. User requests are immediately judged by the ML-based request judgment function. All the requests and results are stored in the request database (request DB). The network resources in each TS are determined based on the set of accepted requests by the LP-based network resource allocation function.

Fig. 2(b) displays the time sequence of the proposed method. Three phases are executed in a pipeline manner. In the request judgment phase, a new request is instantaneously judged as accepted or rejected through ML. The requests in each TS are received within the designated time window. The network resource allocation for each request is not determined at the request judgment time. The network resources for the accepted requests are collectively determined in the resource allocation phase. Accepted requests with the same source and destination are bundled into a single request per logical path. The network resources for the logical paths are optimally determined immediately before the bandwidth is provided in the logical path establishment phase.

Algorithm 2 Request judgement using ML

Input: Request(cal, s, t, b^{st})
Output: Reply

1: Determine the network state at *cal*
2: Predict whether the Request(cal, s, t, b^{st}) is accepted under the network state at *cal* by using the ML classifier
3: **If** the result is classified into '*accept*' **then**
4: Update the set of requests accepted at *cal*
5: Reply = *accept*
6: **else**
7: Reply = *reject*

A. Request judgment using ML

An ML classifier is used to instantaneously request judgment. The classifier is devised using the supervised learning approach, wherein labeled datasets generated from the LP solutions discussed in Section III are trained. The datasets include new request information (cal, s, t, b^{st}), the network state at *cal*, and the label of the request (accepted or rejected). For the network state at *cal*, we apply the reserved bandwidth for all the logical paths at *cal*. This is because we avoid using the concrete information on the physical network, including network resource allocation, at the judging time. The label of the request is the result generated through LP. The concrete ML classifier used here is discussed in Section V. The request judgment using ML is summarized in Algorithm 2. When a new request arrives, the trained classifier predicts whether a new request is accepted or not under the network state at *cal*. A request is accepted when the predicted result is classified as "accept."

B. Resource allocation using LP

The network resources for the accepted requests are collectively and optimally allocated through LP at the designated time. Here, the objective function of LP minimizes the maximum link utilization across the network, where the link capacity is designed for the bandwidth reservation service in advance. In request judgment using an ML classifier, the bandwidth required for certain links may exceed the designed capacity due to misclassification. Thus, we adopt an objective function to minimize excess of the designed capacity. Misclassification is discussed in Section V.

The resource allocation problem is formulated as follows:

$$\text{minimize} \quad \rho, \tag{1a}$$

subject to

$$\sum_{j:(i,j)\in E} y_{ij}^{uv} - \sum_{j:(i,j)\in E} y_{ji}^{uv} = F^{uv}, i = u, \tag{1b}$$

$$\sum_{j:(i,j)\in E} y_{ij}^{uv} - \sum_{j:(i,j)\in E} y_{ji}^{uv} = 0, i \neq u, v, \tag{1c}$$

$$\sum_{(u,v)\in P} y_{ij}^{uv} \leq \rho \cdot C_{ij}. \tag{1d}$$

Eq. (1a) is an objective function that minimizes link utilization ρ across the network. Eqs. (1b) and (1c) state the traffic flow constraints. y_{ij}^{uv}, a real variable, is a fraction of the traffic of logical path (u, v), where u and v are the edge nodes of the logical path. F^{uv} is the bandwidth reserved for (u, v). Eq. (1d) is the constraint for the link capacity. C_{ij} is the capacity of (i, j) designed for the bandwidth reservation service in advance. The traffic route of (u, v) is a set of y_{ij}^{u}, where $y_{ij}^{uv} > 0$.

Because the network resources of the accepted requests are collectively determined, the requirement of the computation time for resource allocation through LP is relaxed to be a TS, (e.g., 1 h) at most. This is one of the significant features of the proposed method.

V. Performance Evaluation

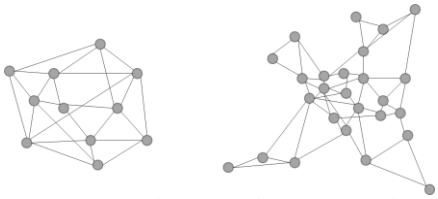
Because the bandwidth reservation service has two service requirements, we evaluate the performance of the proposed method with respect to the request blocking ratio and request judgment time. Request blocking ratio = 1.0 − request acceptance ratio. The performance of the proposed method is compared with those of the methods using the KSP algorithm [5,7] and the LP-based method discussed in Section III. The method, in which a new request is judged based on the residual link capacity using the KSP algorithm, is the benchmark for the minimum request judgment time. The LP-based method is the benchmark for the minimum request blocking ratio.

The proposed method can be adopted under an arbitrary number of requests received in a TS; however, to clearly understand the performance of the proposed method, let us assume that a TS receives 30 requests constantly. This value corresponds to the strong peak distribution where user requests concentrate at a specific time, as defined in [5]. The number of evaluated TSs, and the user requests are 1000 and 30 000, respectively. The source and destination node pair of a request is randomly distributed. The request bandwidth follows an exponential distribution with an average bandwidth of eight (arb. unit). The holding time of a request is constantly one TS. As shown in Fig. 3, two network model are evaluated to determine the dependency of the network size. The link capacity designed for the bandwidth reservation service is 20 (arb. unit). The problems formulated through LP are solved using the GNU Linear Programming Kit solver. The ML classifier is developed using the scikit-learn library.

A. ML classifier

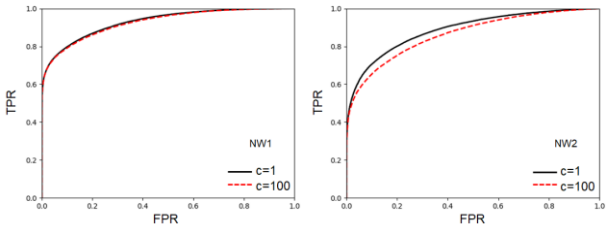
To take the first step toward resolving the challenge of judging bandwidth requests through ML, we adopt a general linear support vector machine (SVM) as the ML classifier model, which is a representative supervised learning model.

To develop the ML classifier, 500 000 labeled training datasets in each network are prepared using LP as discussed in Section III, among which 75% of the datasets are randomly selected for training the classifier, and the remainder are used for testing the trained classifier. The number of datasets belonging to "*accept*" is equal to that belonging to "*reject*."



(a) NW1 (11 nodes) (b) NW2 (26 nodes)

Fig. 3 Network models.



(a) NW1 (b) NW2

Fig. 4 ROC curve of the adopted ML classifier, where c is a hyperparameter of the linear SVM classifier.

When a classifier is trained and tested using the datasets, the order of the prepared data is shuffled.

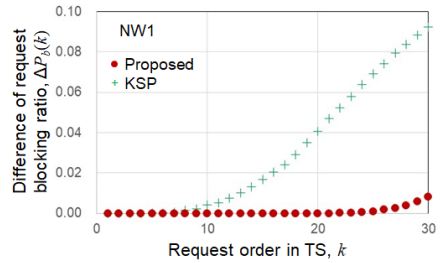
Fig. 4 shows the receiver operator characteristic (ROC) curves of the ML classifier adopted in this study, where the ROC curve is an evaluation metric for binary classification problems, and is evaluated using the datasets for testing. The horizontal and vertical axes denote the false positive rate (FPR) and true positive rate (TPR), respectively. The FPR indicates the incorrect positive results, i.e., false positives (FP), that occur in negative sample spaces (true negative (TN) + FP). The TPR indicates the correct positive results, i.e., true positives (TP), that occur in the positive sample space (TP + false negative (FN)). The area under the curve (AUC) score indicated by the ROC curve is approximately 0.9; the AUC score is a measure of a classifier's ability to distinguish between two classes. The score is sufficiently large to evaluate the effectiveness of the proposed method; thus, we adopt this classifier with c=1 for predicting the request judgment, where c is a general hyperparameter of the linear SVM classifier.

B. Request blocking ratio

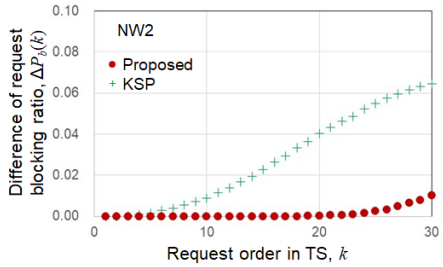
Fig. 5 depicts the request blocking ratio for the 30 000 new user requests. The horizontal axis denotes the request order k ($1<=k<=30$) in a TS. The vertical axis denotes the difference in request blocking ratio from the blocking ratio of the LP-based method $P_{b,LP}(k)$, defined in Eq. (2), which is the average cumulative blocking ratio during a TS in 1000 TSs. The value of $n_j(i)$ is unity when the i-th request is accepted in TS #j. N is the number of evaluated TSs; N=1000 in this evaluation. The results of the proposed method indicate the request blocking ratio for the requests accepted for TP.

$$\Delta P_b(k) = \left| P_b(k) - P_{b,LP}(k) \right|, \tag{2a}$$

$$P_b(k) = 1 - \sum_{j=1}^{N} \sum_{i=1}^{k} n_j(i) \Big/ N \cdot k \,. \tag{2b}$$



(a) NW1



(b) NW2

Fig. 5 Request blocking ratio in a TS.

In both network models, the difference between the LP-based method and the proposed method is less than 1% (0.01). This difference is attributed to the misclassification of the ML classifier. By contrast, the difference between the LP-based method and the method using the KSP algorithm is considerably larger. This is because a single route is allocated for each logical path in the method using the KSP algorithm. These results indicate that the ML classifier provides suboptimal solutions for judging user requests.

When applying an ML classifier, misclassification needs to be carefully considered. The performance evaluations confirm that misclassifications occur for approximately 2% and 7% of the evaluated requests in NW1 and NW2, respectively; this value includes both FN and FP. In particular, FP requests are accepted when the network resource is insufficient. Therefore, the network resource (link capacity, in this study) may exceed the designed one for the bandwidth reservation service. To reduce the influence on other network services to the maximum extent possible, it is necessary to allocate network resources considering the occurrence of misclassifications and to reduce the number of misclassifications as much as possible. In this paper, the network resource of the accepted requests is allocated through LP discussed in Section IV.B. During performance evaluation, we found a TS in which a maximum of 1.2 times the link capacity designed in advance was required. This means that, when the network resource margin is not sufficiently large, unexpected additional network resources need to be reduced by reducing the number of misclassifications. We intend to study this further in the future.

C. *Request judging time*

Although we omit a detailed discussion, the computation time of the proposed method, as well as that of the method using the KSP algorithm, is less than 0.1 ms using a computer with Intel® Core™ i9 and 32 GB of memory. Therefore, the computation time of the proposed method is adequate to meet the service requirements. By contrast, because the computation time of the LP-based method requires more than several seconds in NW2, it is difficult to adopt the LP-based method.

VI. Conclusion

A bandwidth reservation method combining ML and LP was proposed to evolve the network bandwidth reservation service under a mature SDN environment, where the service requires both instantaneous response to a user request and a high user demand acceptance ratio. To meet these two requirements, in the proposed method, a user request is judged instantaneously on arrival by an ML classifier, and the network resource for the accepted request is optimally allocated through LP immediately before the bandwidth is provided. Simulation results indicated that the proposed method provides suboptimal acceptance ratio and instantaneous request judgment.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP20K11798.

References

[1] "Arcstar universal service," https://www.ntt.com/business/services/network/private-network/arcleasedline.html, (in Japanese), accessed on Feb. 2021.
[2] L. Gkatzikis, S. Paris, I. Steiakogiannakis, and S. Chouvardas, "Bandwidth calendaring: Dynamic services scheduling over Software Defined Networks," Proc. IEEE International Conference on Communications (ICC), pp. 22–27, May 2016.
[3] S. Kandula, I. Menache, R. Schwartz, and S. R. Babbula, "Calendaring for wide area networks," Proc. ACM SIGCOMM, pp. 515–526, 2014.
[4] M. Tsujino, "Load balanced model for bandwidth calendaring problem," Proc. Global Information Infrastructure and Networking Symposium (GIIS), pp. 23–25, 2018.
[5] S. Kamamura, R. Hayashi, H. Date, H. Yamamoto, T. Miyamura, Y. Uematsu, and K. Genda, "Recommendation-based bandwidth calendaring for packet transport network," IEICE Trans. Commun., vol. E100-B, no.01, pp.122–130, Jan. 2017.
[6] Q. Wu and P. Dharam, "Advance bandwidth scheduling with minimal impact on immediate reservations in high-performance networks," Proc. IEEE Network Operations and Management Symposium (NOMS), pp.679–682, 2012.
[7] J.Y. Yen, "Finding the K shortest loopless paths in a network," Manage. Sci., vol.17, no.11, pp.712–716, 1971.
[8] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
[9] J. Xie et al., "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges", IEEE Commun. Surveys Tuts., vol. 21, no. 1, pp. 393–430, 1st Quart. 2019.
[10] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, "Machine learning for networking: Workflow advances and opportunities," IEEE Netw., vol. 32, no. 2, pp. 92–99, March/April 2018.
[11] S. Ayoubi et al., "Machine learning for cognitive network management," IEEE Commun. Mag., vol. 56, no. 1, pp. 158–165, Jan. 2018.
[12] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61–80, 2009.
[13] N. Kato et al., "The deep learning vision for heterogeneous network traffic control: proposal, challenges, and future perspective," IEEE Wireless Commun., vol. 24, no. 3, pp. 146–153, June 2017.
[14] K. Genda, "Network bandwidth reservation method combining machine learning and linear programming," IEICE Communications Express, vol. 10, no. 6, pp. 331-336, 2021.