Grandet: Cost-aware Traffic Scheduling without Prior Knowledge in SD-WAN

Yuchao Zhang[*✉], Huahai Zhang[*], Peizhuang Cong[*], Wendong Wang[*], Ke Xu[†]
[*]School of Computer Science (National Pilot Software Engineering School),
Beijing University of Posts and Telecommunications, Beijing, China
[†]Department of Computer Science and Technology, Tsinghua University, Beijing, China
{yczhang, zhanghuahai, congpeizhuang, wdwang}@bupt.edu.cn, xuke@tsinghua.edu.cn

***Abstract*—The rapid growth of traffic demands on wide-area networks (WANs) has resulted in escalated transmission costs for cross-national enterprises. Many researchers have proposed traffic scheduling methods that can effectively reduce transmission costs and improve network performance. However, the majority of research in this field assumes that traffic demands and network link quality are known in advance, disregarding the impact of information agnostic. While some works try to obtain this knowledge through prediction, they lack awareness of prediction errors, which makes it difficult for their scheduling strategies to achieve theoretical results. In this paper, we propose a novel scheduler *Grandet* that aims to reduce transmission costs without any prior knowledge. First, instead of requiring prior knowledge or accurate prediction, *Grandet* determines the intervals of flow sizes and link quality parameters through confidence-based Bootstrap method combined with neural network model, thus quantifying the uncertainty of these information. Then, we design a cost-aware online traffic scheduling framework using the uncertainty intervals from interval determination to optimize the cost minimization problem. Through rigorous theoretical analysis, we prove the approximate optimality of *Grandet* in minimizing transmission costs. Trace-driven and large-scale simulations show that *Grandet* successfully reduces transmission costs by over 23%, reduces deadline miss rate by over 31%, and reduces Service Level Agreement (SLA) dissatisfaction rate by over 37%.**

***Index Terms*—Traffic Engineering, SD-WAN, Bootstrap Method, Lyapunov Optimization**

I. Introduction

In recent years, as the explosive growth of traffic demands, the network is constantly being expanded and upgraded. To reduce transmission costs and improve network availability, an increasing number of enterprises have been moving from earlier MPLS-based WAN solutions to SD-WAN solutions [1]. For example, Google's B4 [2] and Microsoft's SWAN [3] improve bandwidth utilization through software-defined and centralized traffic engineering systems. To establish WANs covering global enterprise sites around the world, cross-national enterprises lease multiple types of heterogeneous links (e.g., private lines based on MPLS, LTE/5G, broadband Internet, etc.) [4] from Internet Service Providers (ISPs), as shown in Figure 1. Enterprises typically purchase bandwidth and choose charging models (e.g., 95^{th} percentile charging model [5]) from ISPs for each link. WAN links with varying unit prices have different network quality (e.g., jitter, delay, and packet loss rate) [6], leading to a strong correlation between bandwidth costs and the performance of traffic scheduling.

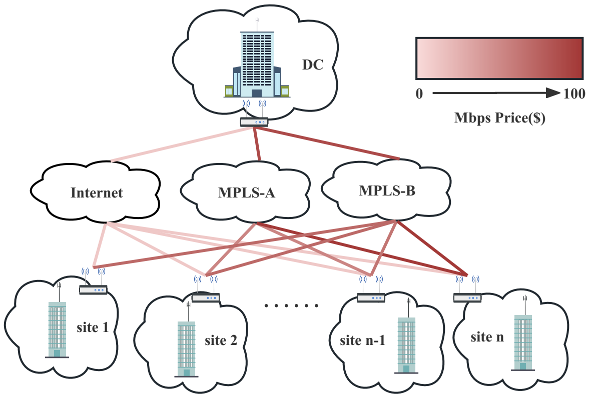


Fig. 1. SD-WAN in large-scale hub-spoke networking mode, where the marginal cost of MPLS is several times higher than that of the Internet.

The SD-WAN solution allows dynamical selections of transmission paths to efficiently schedule mix-flows based on link conditions and time-varying traffic demands [1], which provides an opportunity to significantly reduce transmission costs [7]. In recent years, many researchers have explored novel traffic scheduling methods, which can be roughly divided into two categories:

i) Clairvoyant schedulers: These methods assume that flow sizes are known in advance [8]. Some focus on path selection and traffic allocation for immediate traffic [9], [10], and some make full use of the charging model by shaping [7] or delaying [11] deferrable traffic. Clairvoyant schedulers offer excellent performance in terms of minimizing transmission costs and guaranteeing the quality of service (QoS), but the feasibility is limited due to the need for prior knowledge like traffic demands in each scheduling period. Because it is impractical to ascertain accurate traffic demands due to the dynamic of flows. As a result, these methods often fall short of expectations due to the agnosticity of prior knowledge in actual deployment. Therefore, they rarely achieve deployment.

ii) Non-clairvoyant schedulers: In order to design a practical method that can address the information-agnostic problem, some methods [12] [13] try to predict traffic demands to

The work was supported in part by the Key Project of Beijing Natural Science Foundation under M21030, the National Natural Science Foundation of China(NSFC) under Grant 62172054 and 62072047, the National Key R&D Program of China under Grant 2019YFB1802603.

eliminate the need of prior knowledge through experience-driven heuristics or machine learning. But such knowledge is difficult to obtain with high confidence through prediction [8]. In fact, most of these methods lack awareness of the errors between the predicted value and the actual traffic demands. This disparity leads to a gap between scheduling and practical operation. In addition, there are also some studies exploring algorithms that do not require flow size information, e.g., Aalo [14] and PIAS [15], which leverage relative priority for scheduling. However, compared with clairvoyant algorithms, they perform worse due to the lack of precise flow scheduling.

In this context, *is it possible to design a non-clairvoyant algorithm that does not rely on accurate predictions but approaches the effectiveness of clairvoyant schedulers in terms of reducing costs and guaranteeing QoS?*

In light of the uncertainty of arriving flow sizes in traffic scheduling, designing such an online algorithm involves two challenges: i) It is difficult to predict flow sizes precisely with high confidence. If absolute flow sizes are employed for scheduling, the scheduling results are strongly dependent on the accurate prediction. Consequently, inaccurate predictions will lead to suboptimal or even poor scheduling results. ii) Due to the complexity of the network and the long-term charging model, online traffic scheduling algorithms are likely to fall into local optimum. Besides, some traffic demands, especially immediate traffic such as real-time video transmission, have strict SLA requirements (e.g., delay, jitter, packet loss rate, etc.) [16]. Since the link's SLA quality is also dynamic and unknowable [16], it is necessary to take the link conditions into account to guarantee service quality, which increases the complexity of scheduling.

In this paper, we introduce *Grandet*, an online traffic scheduler that aims to minimize bandwidth costs and guarantee the QoS without prior knowledge. Since the accurate prediction is difficult to be obtained with high confidence, we advocate using estimation intervals to replace the need for precise prediction of traffic demands or link states. To this end, *Grandet* combines the Bootstrap method with a neural network model to determine the intervals of flow sizes and link SLA quality with a pre-specified confidence probability. Based on the fuzzy and uncertain intervals, we propose a cost-aware online traffic scheduling framework. We first formulate a stochastic optimization problem with the objective of cost minimization and then decompose the problem into a solvable online scheduling optimization problem through Lyapunov optimization techniques [17]. We conduct extensive trace-driven simulations to evaluate *Grandet*. Simulation results demonstrate that *Grandet* reduces transmission costs by more than 23% in the large-scale workload. In terms of service guarantee, *Grandet* reduces deadline miss rate by more than 31% and reduces SLA dissatisfaction rate by more than 37%.

In summary, our main contributions include:

- We reveal that existing traffic scheduling algorithms either require prior knowledge or highly depend on the accuracy of flow size predictions. Moreover, a brief example of the potential performance deviations resulting from inaccurate predictions is carried out.
- We propose an online traffic scheduler, *Grandet*, which can minimize transmission costs and improve service quality without any prior knowledge.
- We conduct extensive trace-driven simulations to evaluate the performance of *Grandet*, and the results show that *Grandet* is cost-effective and QoS-guaranteeing.

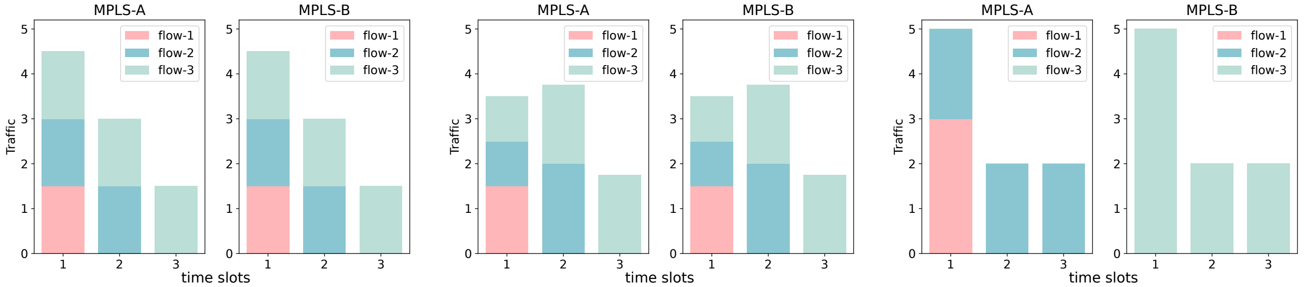
II. RELATED WORK AND MOTIVATION

In this section, we summarize the related work and expose the problems of existing works. Then, we motivate the design of *Grandet* through an illustrative and simple example.

A. Related work

Network cost optimization: Regarding the transmission costs, there has been a lot of excellent work. Since the percentile charging model is widely used, we mainly focus on the works based on it. DTM [9] and Cascara [10] schedule immediate traffic by pricing-aware algorithm to reduce bandwidth costs. TrafficShaper [7] designs a pricing-aware online control framework to control deferrable traffic's transmission rate for reducing transmission costs while maintaining a low deadline miss rate. In order to manage costs and provide service guarantees, Pretium [18], a framework integrating dynamic pricing into traffic engineering, model percentile charging as a compact set of linear inequalities. While these methods are effective in cost reduction, they are clairvoyant algorithms, which require prior knowledge of flow sizes, hence difficult to be practically applied.

Traffic scheduling under information-agnostic: Information-agnostic solutions typically improve network performance through information estimation or priority-based scheduling. Regarding information estimation, researchers have explored heuristics and machine learning to predict flow sizes [8], [12], [13], [19]. But in fact, it is difficult to achieve stable high precision in predicting these knowledge [8]. In addition, these prediction algorithms lack awareness of prediction errors, which may lead to unpredictable results. DarkTE [20], which realizes that prediction errors can be problematic, uses random forests to classify flows and then allocates the rate and path to flows based on confidence to mitigate occasional classification errors. As for priority-based scheduling, e.g., Aalo [14] and PIAS [15], although they outperform the baselines, their performance is always much worse than the clairvoyant ones. To solve the minimum cost problem, Homa [1] proposes a randomized greedy algorithm which iteratively find the cheapest link for each unit of demand and performs well with sufficient capacity. Different from aforementioned works, we leverage interval estimation based on confidence instead of point prediction for traffic scheduling, which can significantly reduce the impact introduced by prediction errors. Meanwhile, compared to priority-based scheduling, we provide more precise scheduling solutions.



(a) Load balance with correct information, total billed bandwidth=6 (b) Load balance with incorrect information, total billed bandwidth=7 (c) Optimal scheduling, total billed bandwidth=4

Fig. 2. A small-scale motivating example with two links (taking MPLS-A and MPLS-B of site n in Figure 1 as an example) with identical unit cost and bandwidth upper limit of 5 for both. As for traffic demands, there are three flows with size {3,6,9}, while their deadlines are {1,2,3}. For simplicity, we consider the 50^{th} percentile charging model over three time slots.

B. A motivating example

Percentile charging model. ISPs typically charge with the widely embraced percentage charging model [5]. In each charging period (e.g., a month), the ISP records the bandwidth usage in every time slot (e.g., 5 minutes) and takes the maximum θ^{th} percentile as the billed bandwidth. Therefore, the $(1-\theta)\%$ time slots with the highest bandwidth usage are not included in charging, i.e., 'free' slots. In this context, the traffic scheduler operates in a discrete-time mode, that is, traffic scheduling is executed at the beginning of each time slot. Thus, we can arrange more traffic in $(1-\theta)\%$ free slots to reduce traffic in other time slots, which provides the opportunity for cost reduction.

Potential problems of existing algorithms. For better illustration, we use a small-scale example to demonstrate the problem, as shown in Figure 2. Due to the identical configuration of two links, the heuristics (Load Balance [4]) that network operators have historically used will simply allocate traffic fairly between two links. Regarding the deferrable traffic, we take ES (Equal Splitting) [21] as an example, which can evenly distribute traffic to each time slot within the deadline. Under such a scheduling mechanism, when flow sizes are estimated correctly, the total billed bandwidth is $P_{50}(\{4.5,3,1.5\}) + P_{50}(\{4.5,3,1.5\}) = 6$ (P_{50} is a function that calculates the maximum 50^{th} percentile in the sequence.), as shown in Figure 2(a).

Incorrect estimation of flow size can result in unexpected additional costs. In this example, we assume that the flow sizes of flow-2 and flow-3 are incorrectly estimated as 4 and 6 at the beginning of time slot 1. Consequently, the flows would be scheduled with incorrect information until the specific flow sizes are detected at the beginning of the second time slot. In this case, the total billed bandwidth is $P_{50}(\{3.5,3.75,1.75\}) + P_{50}(\{3.5,3.75,1.75\}) = 7$, as shown in Figure 2(b). Compared with having a correct estimate of flow sizes, the total cost increases by 1/6. Because of the challenge in accurately determining flow sizes [8], schedulers that rely on accurate flow size prediction are difficult to achieve desired goals.

Feasibility of optimal scheduling. Nevertheless, it is possible to achieve optimal results even without prior knowledge. In this small-scale example, if a scheduling algorithm has a mechanism to actively utilize free slots, optimal scheduling result can be achieved irrespective of correct or incorrect information. Assume that free slots are actively used in the first time slot, that is, the scheduler tries to maximize the traffic on two links in time slot 1. For the correct traffic estimation $\{3,6,9\}$ or the incorrect ones $\{3,4,6\}$, there will be the same optimal scheduling results, with total billed bandwidth equal to $P_{50}(\{5,2,2\})+P_{50}(\{5,2,2\}) = 4$, as shown in Figure 2(c). Therefore, even if accurate flow sizes information is difficult to obtain with high confidence, optimal scheduling can be achieved or approached.

Observations. From the motivating example, it can be observed that: i) Due to the lack of prior knowledge in the implementation, the performance of clairvoyant algorithms likely deviates from the expectation. ii) Even without prior knowledge, a good scheduling algorithm may approach optimal scheduling. This requires full consideration of how to mitigate the impact of information-agnostic, how to leverage the rules of the charging model, and how to spread traffic in each time slot.

III. SYSTEM OVERVIEW

Grandet is a novel traffic scheduler that realizes cost-aware and service-guaranteed traffic scheduling without requiring any prior knowledge. Its two primary goals include: (1) determining the intervals of flow sizes and link quality parameters with high confidence. (2) reducing transmission costs while guaranteeing QoS through precise path selection and rate control of flows. As depicted in Figure 3, *Grandet* is mainly composed of two modules: interval determination agents and a traffic scheduling controller. Next, we will present more details about the two modules.

A. Distributed interval determination agent

Distributed interval determination agents run on edge devices, continuously monitoring SLA quality of each link and traffic demands of each application. At the beginning of each time slot, the interval determination agents estimate traffic demands and SLA quality intervals with historical traffic data and then update these interval estimates to the centralized traffic scheduling controller. To achieve cost-effective and

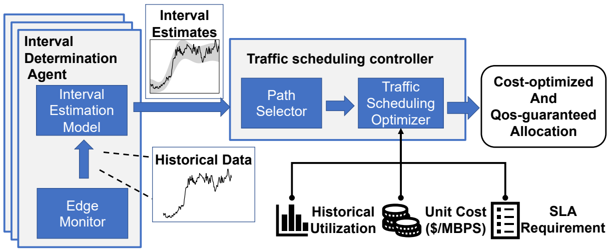

Interval Determination Agent
Interval Estimation Model
Edge Monitor
Interval Estimates
Historical Data
Traffic scheduling controller
Path Selector
Traffic Scheduling Optimizer
Cost-optimized And Qos-guaranteed Allocation
Historical Utilization
Unit Cost ($/MBPS)
SLA Requirement

Fig. 3. An overview of *Grandet*

service-guaranteed traffic scheduling, it is necessary to obtain sufficient knowledge about traffic demands and link SLA quality. However, obtaining such knowledge accurately is difficult due to its time-varying characteristic. Most time series prediction algorithms can only produce point predictions with little awareness of their accuracy or errors, which make them hard to provide credible information for optimal traffic scheduling. Therefore, we propose two design principles on how to provide credible knowledge for improving traffic scheduling: 1) estimating flow sizes and SLA quality with error awareness. 2) providing information with high confidence instead of point prediction with poor confidence. On account of the stability and accuracy of the Bootstrap method in time series prediction, we use it to generate interval estimation for flow sizes and SLA quality, which will be elaborated in Section IV.

B. Centralized traffic scheduling controller

The centralized controller collects information from distributed interval determination agents and executes the traffic scheduling optimizer. Then, it deploys the updates on transmission paths and rate allocation for each application to edge devices. To design a traffic scheduler that effectively reduces transmission costs and guarantees QoS, it is necessary to analyze the factors that affect costs and QoS. According to the observations of the motivating example, the impact of information agnostic, the rules of the charging model, and traffic distribution among time slots dominate the effects of traffic scheduling. Therefore, we propose three design principles on how to optimize traffic scheduling: 1) using flow sizes and link SLA quality interval estimation thus reducing the impact of information-agnostic. 2) transferring more traffic in free slots and minimizing the maximum traffic in other time slots thus taking full advantage of percentage charging model's rules. 3) leveraging interval estimations of traffic demands to achieve rate control thus distributing traffic within deadlines reasonably. Taking all factors into account, we formulate a stochastic optimization problem to minimize the long-term average cost. By decomposing the long-term stochastic optimization problem into sub-problems of each slot with the Lyapunov optimization techniques, we design an online traffic scheduling framework, which will be elaborated in Section V.

IV. Flow sizes and SLA intervals determination

In this section, we introduce how to determine the intervals of flow sizes and SLA quality with high confidence. Figure 4 presents the interval determination framework, which combines the Bootstrap method with NeuralProphet [22].

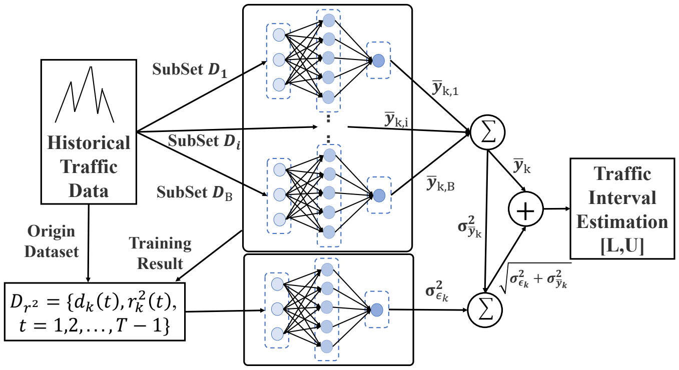


Fig. 4. Interval determination framework, combining bootstrap method and neural network prediction model to generate interval estimation

By bootstrapping predictions on an ensemble of neural networks [23], interval estimation can achieve higher accurate and stable compared to other uncertainty quantification methods like MVE or Bayesian. In addition, NeuralProphet, a powerful time series forecasting tool, is a combination of Prophet and neural network, which retains original characteristics and all advantages of Prophet. At the same time, by introducing AR-Net to model time series auto-regression, it can combines the scalability of neural network with the interpretability of AR model to improve its accuracy and scalability. Therefore, we can enhance the stability and accuracy of interval determination to quantify uncertainty by integrating the benefits of the Bootstrap method and NeuralProphet. The details of interval determination framework are as follows.

First, we need to make it clear that the prediction target (e.g.., traffic demand d_k, where time subscript t is eliminated for brevity) logically consists of true regression y_k and noise ϵ_k following a normal distribution with zero mean, which can be expressed by

$$d_k = y_k + \epsilon_k. \tag{1}$$

The core idea of Bootstrap is to train multiple models by resampling, and then generate less biased estimates of true regression and noise through these models to construct interval estimations. Using B subsets $\{D_i\}_{i=1}^{B}$ resampled from the original dataset, B NeuralProphet models are trained. Then, B^* models, whose loss function value is less than the mean value, are selected to construct interval estimations. The less biased estimated value $\overline{y}_k$ of the true regression y_k is obtained by averaging the point predictions from the selected B^* model, which is given by

$$\overline{y}_k = \frac{1}{B^*}\sum_{i=1}^{B^*}\overline{y}_{k,i}, \tag{2}$$

where $\overline{y}_{k,i}$ is the point prediction of i^{th} model. In essence, the interval estimation represents the probability distribution of target values by quantifying the prediction error $(d_k - \overline{y}_k)$. With the knowledge that $(y_k - \overline{y}_k)$ and ϵ_k are statistically independent [24], we can get the variance of $(d_k - \overline{y}_k)$ deduced from Equation (1), which is given by

$$(d_k - \overline{y}_k)^2 = \sigma^2_{\overline{y}_k} + \sigma^2_{\epsilon_k}, \quad (3)$$

where $\sigma^2_{\overline{y}_k} = (y_k - \overline{y}_k)^2$ is the variance of the model uncertainty and $\sigma^2_{\epsilon_k}$ is the variance of the random noise.

Thus, the upper and lower bounds of the estimation interval are obtained by adding or subtracting the uncertainty of the difference between d_k and $\overline{y}_k$ to the estimated value $\overline{y}_k$ of the true regression, which is given by

$$U_k^{(\alpha)} = \overline{y}_k + z_{\frac{\alpha}{2}} \cdot \sqrt{\sigma^2_{\overline{y}_k} + \sigma^2_{\epsilon_k}}, \quad (4)$$

$$L_k^{(\alpha)} = \overline{y}_k - z_{\frac{\alpha}{2}} \cdot \sqrt{\sigma^2_{\overline{y}_k} + \sigma^2_{\epsilon_k}}, \quad (5)$$

where $z_{\frac{\alpha}{2}}$ is the $\frac{\alpha}{2}$ quantile of the standard normal distribution. The prediction target will be bracketed in $[L_k^{(\alpha)}, U_k^{(\alpha)}]$ with a confidence level $(1-\alpha)\%$.

The variance of model uncertainty $\sigma^2_{\overline{y}_k}$ is approximated as

$$\sigma^2_{\overline{y}_k} = \frac{1}{B^* - 1} \sum_{i=1}^{B^*} \left(\overline{y}_{k,i} - \overline{y}_k\right)^2. \quad (6)$$

From Equation (3), the variance of the noise $\sigma^2_{\epsilon_k}$ is estimated as follow:

$$\sigma^2_{\epsilon_k} \simeq E\left[(d_k - \overline{y}_k)^2\right] - \sigma^2_{\overline{y}_k}. \quad (7)$$

To get an estimation of the noise's variance $\sigma^2_{\epsilon_k}$, we train a model with the same structure to predict the residual r_k^2, which is given by

$$r_k^2 = max\left\{(d_k - \overline{y}_k)^2 - \sigma^2_{\overline{y}_k}, 0\right\}. \quad (8)$$

Using the dataset $D_{r^2} = \{d_k(t), r_k^2(t) | t = 1, 2, ..., T-1\}$, we train the prediction model for the variance of noise $\sigma^2_{\epsilon_k}$ according to the following loss function:

$$L = \frac{1}{2} \sum_{t=1}^{T-1} \left[ln\left(\sigma^2_{\epsilon_k}(t)\right) + \frac{r_k^2(t)}{\sigma^2_{\epsilon_k}(t)} \right]. \quad (9)$$

In order to keep the variance positive, the activation function of the output layer is configured as exponential function. With the approximate estimation of $\sigma^2_{y_k}$ and $\sigma^2_{\epsilon_k}$, the upper bound and lower bound can be obtained by (4) and (5). Thus, we can determine the intervals of flow sizes and SLA quality with confidence level $(1-\alpha)\%$.

$$I_k = \left[\overline{y}_k - z_{\frac{\alpha}{2}} \sqrt{\sigma^2_{\overline{y}_k} + \sigma^2_{\epsilon_k}}, \overline{y}_k + z_{\frac{\alpha}{2}} \sqrt{\sigma^2_{\overline{y}_k} + \sigma^2_{\epsilon_k}}\right]. \quad (10)$$

By determining the interval, we quantify the uncertainty of traffic demands and SLA quality, thus providing confidence-based information to optimize traffic scheduling. Note that different parameter configurations should be adopted for flow sizes and SLA quality interval estimations, such as setting the periodicity of flow sizes prediction and the aperiodicity of SLA quality prediction. In the following section, we exploit interval estimation to realize cost-aware traffic scheduling.

V. ONLINE TRAFFIC SCHEDULING OPTIMIZATION

In this section, we present an online traffic scheduling framework that aims to minimize transmission costs while improving the satisfaction rate of deadline and SLA requirements. Firstly, we present the formulation of the long-term problem with the objective of minimizing average cost. In our formulation, we restrict the transmission paths and relax traffic satisfying constraints using flow sizes and SLA quality intervals with a pre-specified confidence level. Since the long-term problem is unrealistic to solve directly without future information, we leverage the Lyapunov optimization techniques to decompose it into slot-by-slot cost minimization problems.

A. Problem formulation

We consider a typically hub-spoke network as shown in Figure 1, where there are multiple applications at each site. Each application transmits immediate traffic like real-time video transmission or deferrable traffic like database geo-backup with a fixed deadline requirement. Since the percentile charging model records the average traffic of each time slot and then charges with maximum θ^{th} bandwidth usage for the entire charging period, time can be regarded as discrete. Suppose that time is divided into T time slots, with each slot spanning 5 minutes.

At time t, the unfinished deferrable traffic of application k is denoted as $d^t_{k,i}$, where i indicates its deadline. The estimation interval of arriving deferrable traffic size is expressed as $[d^{t,L}_{k,n}, d^{t,U}_{k,n}]$. Therefore, for deferrable traffic, the expected minimum transmission rate of application k is $d^{t,L}_k = \sum_{i=1}^{n-1} d^t_{k,i}/i + d^{t,L}_{k,n}/n$, while the expected maximum transmission rate is $d^{t,U}_k = \sum_{i=1}^{n-1} d^t_{k,i}/i + d^{t,U}_{k,n}/n$. As for immediate traffic, its traffic demand interval is directly expressed by the interval estimation $[d^{t,L}_k, d^{t,U}_k]$. Let $x^t_{e,k}$ denote the allocated bandwidth for application k in link e, $x^t_k = \sum_{e \in E} x^t_{e,k}$ denote the aggregate allocated bandwidth for application k, and $f^t_e = \sum_{k \in K} x^t_{e,k}$ denote the bandwidth usage of link e at time t. In each charging period with N time slots, let $Z_e = P_\theta(\{f^t_e | t \in 1, 2, ..., N\})$, where P_θ is the function that calculate the maximum θ^{th} bandwidth usage, denote the maximum θ^{th} percentile billed bandwidth of link e. There is no difference between upload and download traffic in formulation, but when calculating the billed bandwidth, their percentile billed bandwidths are calculated respectively and the maximum value will be taken as the actual billed bandwidth. For simplicity, the distinction between upload traffic and download traffic is omitted here, similarly hereinafter.

To minimize transmission costs and achieve better service quality, we formulate the long-term average cost minimization problem with QoS Constraints. For a long-term T, there are $P = T/N$ charging periods. So, the objective function of the cost minimization problem **P1** is expressed as follows:

$$\lim_{T \to \infty} \frac{1}{P} \sum_{p=1}^{P} \sum_{e \in E} vc_e \cdot \max\{Z_e(p) - C^b_e, 0\}, \quad (11)$$

where vc_e is the marginal cost of elastic bandwidth, C_e^b is the basic bandwidth, i.e. committed bandwidth. Since the cost of committed bandwidth is fixed and how to set the committed bandwidth is not considered in this paper, it is omitted in the cost function.

For each application, in order to satisfy traffic requirements with high confidence, $d_k^{t,U} \leq x_k^t$ should be made. In our formulation, we relax this inequality with long-term constraints:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} (d_k^{t,U} - x_k^t) \leq 0, \forall k \in K. \tag{12}$$

This relaxation guarantees that x_k^t is greater than the expected maximum transmission rate on average, which effectively prompts flow to be completed within its deadline [7].

However, such a relaxed constraint may result in an overly reduced transmission rate for cost reduction, leading to an unexpected increase in deadline misses. So, in order to ensure the completion time of flows, we limit the aggregate transmission rate of each application to be greater than the expected minimum transmission rate in each time slot:

$$x_k^t \geq d_k^{t,L}, \forall k \in K. \tag{13}$$

This constraint limits the lower bound of the transmission rate for each application. But this lower bound incurs little cost increases because the committed bandwidth is usually sufficient to satisfy the lower bound of traffic demand.

For all time slots t in each charging period p, the following constraints need to be satisfied:

$$x_{e,k}^t = 0, (\tau_e^t, \delta_e^t, \varphi_e^t) \geq (\tau_k, \delta_k, \varphi_k), \forall k \in K, e \in E_k, \tag{14a}$$

$$f_e^t \leq C_e^m \cdot (1 - u_e^t) + C_e^M \cdot u_e^t, \forall e \in E, \tag{14b}$$

$$\sum_{t=0}^{N} u_e^t \leq N \cdot (1-\theta)\%, \forall e \in E, \tag{14c}$$

$$Z_e(p) \geq f_e^t - \mathcal{M} \cdot u_e^t, \forall e \in E. \tag{14d}$$

Among them, $\tau_k, \delta_k, \varphi_k$ are jitter, delay, and packet loss rate requirement of application k respectively, and $\tau_e^t, \delta_e^t, \varphi_e^t$ are the upper bound of SLA quality estimation interval. So, (14a) means that the application traffic will be allocated to the links that strictly satisfy the SLA quality requirements of the application with high confidence, thus guaranteeing the transmission quality. (14b) is the link capacity constraint, where C_e^m is the upper bound of elastic bandwidth, e.g., the specified maximum value of percentile billed bandwidth, and C_e^M is the upper bound of physical bandwidth. This constraint ensures that the link's maximum θ^{th} bandwidth usage does not exceed the elastic bandwidth limitation. u_e^t indicates whether time slot t is a free slot, and (14c) indicates the quantity limitation of free slots. (14d) is the charging rules of the percentile charging model [10], which is used to obtain the percentile billed bandwidth.

Given the above objective function and constraints, P1 is a long-term average cost minimization optimization problem that is impractical to solve directly. On the one hand, the traffic arrival pattern and link fluctuation are impractical to predict accurately for the entire charging period. On the other hand, percentile charging relies on the usage of free slots throughout the charging period, but optimizing the usage of free slots is complicated. These challenges make it impossible to solve P1 without future information. Therefore, it's necessary to simplify P1 by decoupling and decomposing it.

B. Decomposition with Lyapunov optimization

Since it is impractical to solve problem P1, we take advantage of Lyapunov optimization techniques [17] to decompose P1 into a solvable online optimization problem. In particular, the approximate optimality of this online optimization problem is proven by rigorous theoretical analysis in Section VI.

Since obtaining the overall optimal free slots allocation scheme is difficult, it is necessary to transform P1 into a decomposable relaxed problem before decomposing the problem into an online optimization problem. Due to the tight coupling between free slots utilization and percentile billed bandwidth in P1, Eqs. (14c)(14d) need to be transformed as follows:

$$u_e^t + u_e^{t'} \leq N \cdot (1-\theta)\% - \mu_e + 1, \forall e \in E, \tag{15a}$$

$$(1 - u_e^{t'}) \cdot (z_e^t - f_e^{t'}) + u_e^{t'} \cdot (f_e^{t'} - z_e^t) \geq 0, \forall e \in E, \tag{15b}$$

where t' is the time slot with the smallest bandwidth usage in previous free slots, z_e^t is the current maximum θ^{th} percentile billed bandwidth, μ_e represents the number of slots whose used bandwidth exceeds z_e^t, i.e., the number of free slots already used. Through these two inequalities, we relax the cost minimization problem from period-by-period to slot-by-slot, thus having a relaxed problem **P2**:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{e\in E} vc_e \cdot \max\{z_e^t - C_e^b, 0\}. \tag{16}$$

Although P1 and P2 are different in terms of objectives and constraints, the optimal solution of P2 can achieve the optimality close to that of P1, as P2 for each slot can be regarded as a sub-problem of P1.

Then, we utilize Lyapunov optimization techniques to decompose P2 into sub-problems for each slot. We first construct a set of virtual queues to control traffic backlog. For each application k, there is a virtual queue $Q_k(t)$ and $Q_k(0) = 0$. In each time slot, the queues are updated as follows:

$$Q_k(t+1) = max\{Q_k(t) + d_k^{t,U} - x_k^t, 0\}. \tag{17}$$

These virtual queues record the deviation between the expected transmission rate and the allocated ones, which represents the historical traffic backlog. In this context, the long-term constraints in Eq. (12) are transformed into the requirement of satisfying queues stability conditions $\lim_{T\to\infty} Q_k(T)/T = 0$.

For each time slot t, we define the Lyapunov function L(t), which is a scalar measure of the traffic backlog:

$$L(t) = \frac{1}{2} \sum_{k=1}^{K} Q_k(t)^2. \tag{18}$$

$L(t)$ reflects the traffic backlog level of the entire system. The smaller $L(t)$ is, the smaller the total traffic backlog will be, leading to a shorter average flow completion time, i.e., a lower deadline miss rate. To achieve queues stability conditions, $L(t)$ should be kept low. Then, we introduce Lyapunov drift as the change of the traffic backlog from t to $t+1$:

$$\triangle L(t) = L(t+1) - L(t). \tag{19}$$

It is easy to check that to keep a low traffic backlog level, we should make $\triangle L(t)$ as small as possible. Therefore, to stabilize the queue backlog while minimizing the time average of transmission costs, we construct *drift-plus-penalty* to decompose P2 into sub-problems for each time slot. In each time slot, combining problem P2 with the Lyapunov drift, the problem **P3** for reducing costs while guaranteeing deadlines is expressed as follows:

$$\min \triangle L(t) + V\mathbb{E}\Big\{ \sum_{e \in E} vc_e \cdot \max\{z_e^t - C_e^b, 0\} | Q(t) \Big\}, \tag{20}$$

where V is a control weight for the trade-off between cost minimization and deadline guarantee. With suitable V, the average expected transmission costs can be significantly reduced while guaranteeing service quality.

To transfer more traffic in free slots and avoid falling into local optimum, there should be fewer free slots being used in each time slot. Therefore, we direct traffic scheduling optimization by adding an objective with lower priority:

$$\min \sum_{e \in E} u_e^t. \tag{21}$$

Given the secondary priority objective above, the use of free slots can be minimized under the premise of ensuring the same cost. Furthermore, according to traffic or link characteristics, more objective functions with different priorities can be added to optimize traffic scheduling.

Through solving P3, transmission paths and rate allocation for the traffic of each application can be determined to schedule traffic. Using the most advanced solver, e.g. GUROBI [25], the sub-problems for each time slot can be solved within a limited time. So far, we have developed an online traffic scheduling optimization framework that can minimize transmission costs while guaranteeing QoS.

VI. Theoretical Analysis

Theorem 1: For any $V > 0$, the online traffic scheduling optimization framework can achieve the following performance guarantee: *The average expected cost approximates the optimal scheduling plus the addition with a constant factor $1/V$. Similarly, the average queue size of the traffic backlog is bounded with a constant factor of V.*

Proof 1: Putting $Q_k(t+1)^2 =\leq (Q_k(t) + d_k^{t,U} - x_k^t)^2$ into $\triangle L(t)$ and rearranging it, the bound of Lyapunov drift is:

$$\triangle L(t) \leq B(t) + \sum_{k=1}^{K} Q_k(t)(d_k^{t,U} - x_k^t), \tag{22}$$

where $B(t) = \frac{1}{2}\sum_{k=1}^{K}(d_k^{t,U} - x_k^t)^2$. Assuming that the arrival traffic demands are bounded, there is a constant $B \geq \mathbb{E}[B(t)|Q(t)] > 0$. Deduce from Equation (22), the bound on the conditional expected Lyapunov drift is given by

$$\mathbb{E}[\triangle L(t)|Q(t)] \leq B + \sum_{k=1}^{K} Q_k(t)\mathbb{E}[(d_k^{t,U} - x_k^t)]. \tag{23}$$

In many cases, the traffic can be ultimately completed before deadlines, so that the difference between desired bandwidth and the actual allocated bandwidth satisfies the following inequality for real numbers $\varepsilon > 0$:

$$\mathbb{E}[(d_k^{t,U} - x_k^t)|Q(t)] \leq -\varepsilon. \tag{24}$$

Therefore, substituting Equation (24) into Equation (23), the bound on conditional expected Lyapunov drift is given by

$$\mathbb{E}[\triangle L(t)|Q(t)] \leq B - \varepsilon \sum_{k=1}^{K} Q_k(t). \tag{25}$$

Taking the conditional expectation of Equation (25) and summing it over previous time slots with telescoping sum method, we have:

$$\mathbb{E}[L(t)] - \mathbb{E}[L(0)] \leq Bt - \varepsilon \sum_{t'=0}^{t-1}\sum_{k=1}^{K} \mathbb{E}[Q_k(t')]. \tag{26}$$

Then extending the above formula to P3, the conditional expectation bound of *drift-plus-penalty* is given by

$$\mathbb{E}[\triangle L(t) + Vp(t)|Q(t)] \leq Vp^* + B - \varepsilon \sum_{k=1}^{K} \mathbb{E}[Q_k(t)], \tag{27}$$

where $p(t)$ is the cost function in P3, and $p*$ is the desired target for the time average of $p(t)$. Assume that $p(t)$ has a lower bound p_{min}. Summing this inequality (27) over previous time slots with the telescoping sum method, we have:

$$\begin{gathered} \mathbb{E}[L(t)] - \mathbb{E}[L(0)] + V\sum_{t'=0}^{t-1}\mathbb{E}[p(t')] \leq \\ Vp^*t + Bt - \varepsilon \sum_{t'=0}^{t-1}\sum_{k=1}^{K} \mathbb{E}[Q_k(t')], \end{gathered} \tag{28}$$

$$V\sum_{t'=0}^{t-1}\mathbb{E}[p(t')] \leq Vp^*t + Bt + \mathbb{E}[L(0)], \tag{29}$$

$$\varepsilon \sum_{t'=0}^{t-1}\sum_{k=1}^{K}\mathbb{E}[Q_k(t')] \leq V(p^* - p_{min})t + Bt + \mathbb{E}[L(0)]. \tag{30}$$

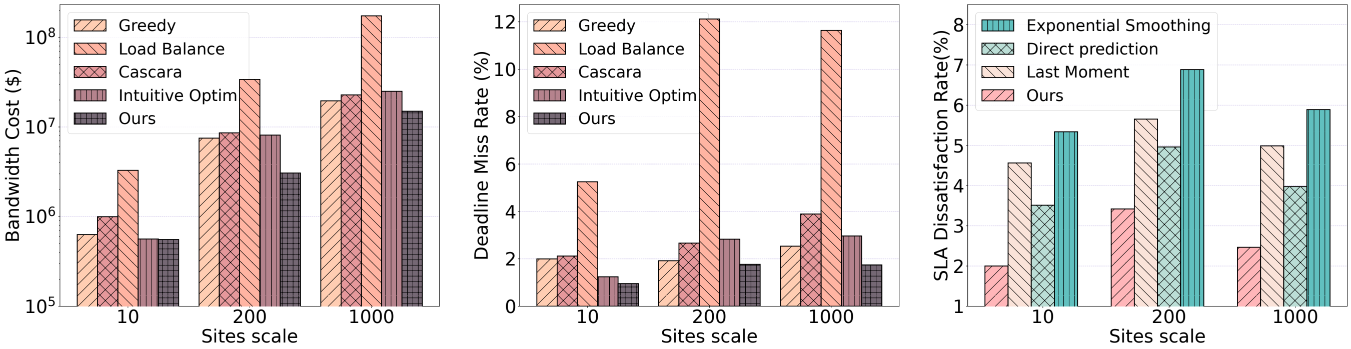
Therefore, the average expected cost is higher than the optimal scheduling with a constant factor of $1/V$ and the average queue size is bounded with a constant factor of V, which is given by

$$\frac{1}{t}\sum_{t'=0}^{t-1}\mathbb{E}[p(t')] \leq p^* + \frac{Bt + \mathbb{E}[L(0)]}{Vt}, \tag{31}$$

$$\frac{1}{t}\sum_{t'=0}^{t-1}\sum_{k=1}^{K}\mathbb{E}[Q_k(t')] \leq \frac{V(p^* - p_{min})t + Bt + \mathbb{E}[L(0)]}{\varepsilon t}. \tag{32}$$

TABLE I
ALGORITHMS COMPARISON IN THE TOPOLOGY WITH 1000 SITES

Metric / Method	Performance		Characteristic		
	Bandwidth cost($)	Deadline miss rate(%)	Non-clairvoyant	Cost-aware	Free slots aware
Load Balance [4]	173,837,122	11.63	✓	×	×
Homa:Greedy [1]	19,584,862	2.53	✓	✓	×
Cascara [10]	22,807,430	3.89	×	✓	✓
Intuitive optimization	24,993,579	2.96	×	✓	✓
Grandet	**14,985,313**	**1.74**	✓	✓	✓



(a) Comparison of bandwidth cost

(b) Comparison of deadline miss rate

(c) Comparison of SLA miss rate

Fig. 5. The performance of different algorithms in different workloads.

VII. EVALUATION

In this section, we conduct extensive simulations to verify the effectiveness of *Grandet*.

A. Setup

Dataset: We use a one-month workload from an Internet Service Provider, which contains three different applications across 1000 sites. Moreover, two small-scale workloads, with 10 and 200 sites respectively, are used to verify the performance of our algorithm under different network topologies.

Applications: One application transmits immediate traffic like real-time video transmission, while the other two transmit deferrable traffic like database geo-backup or service migration with a fixed deadline requirement (4 time slots).

Network topology: The network topology is a typical hub-spoke pattern (as shown in Figure 1). Each site is connected to the data center via MPLS and Internet, with a total of 3215, 631, 34 links for 1000, 200, 10 sites scale, respectively.

Comparison of Different Algorithms: We compare *Grandet* with the following algorithms in various aspects.

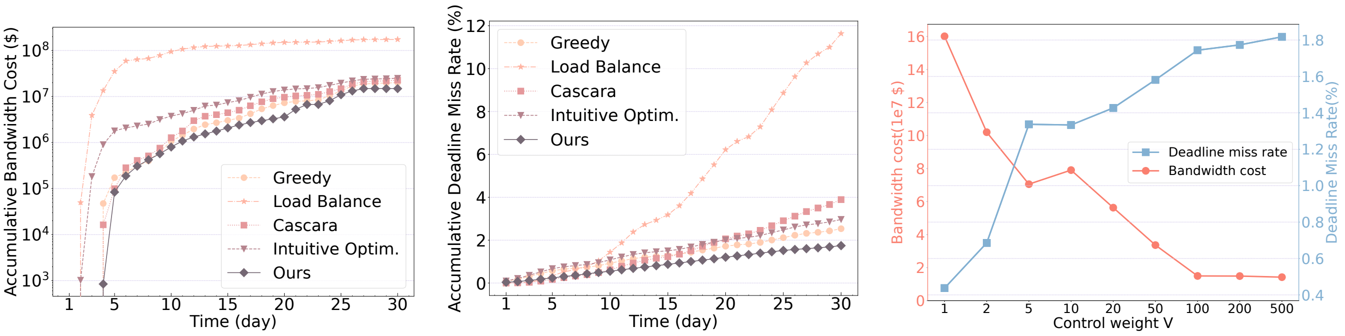
- **Load Balance:** The load balance [4] method distributes arriving traffic across overlay links according to the proportion of the committed bandwidth.
- **Homa (Greedy):** The greedy method in Homa [1] allocates traffic with a random permutation of demands. For each unit of demand, it iteratively finds the cheapest link subject to link capacity and QoS constraints.
- **Cascara:** Cascara [10] records the number of free slots and the maximum θ^{th} percentile for each link, and distributes traffic based on a multi-priority scheme. The first priority is the remaining free slots of the link, that is, traffic is preferentially allocated to the links with more free slots. The second priority is link capacity, thus saving the free slots of links with higher capacity for the remaining billing cycle.
- **Intuitive optimization:** The intuitive online optimization method refers to an optimization algorithm constructed with cost minimization function as objective and Eqs. (14a)(14b)(15a)(15b) as constraints. And the allocated traffic for applications is constrained to be equal to the predicted traffic demands.

Since these schemes lack the mechanism for scheduling deferrable traffic, we integrate ES (Equal Splitting) [21] method into these algorithms to control the rate of deferrable traffic. The ES method evenly distributes the arriving flow into the next K time slots (K is set as the fixed deadline), which is a simple and cost-effective solution. Although ES is affected by information agnostic, the actual average effect is not substantial due to a long deadline. Regarding the input of these algorithms, we predict the flow sizes and SLA quality parameters by a NeuralProphet model, which performs best among the prediction methods we have implemented. Regarding the charging model, we adopt 95^{th} percentile charging model, which means that $(30 \times 24 \times 60/5) \times 5\% = 432$ free slots are available, i.e., 36 hours.

B. Results

We evaluate the effectiveness of the algorithm in terms of bandwidth costs, deadline miss rate and SLA satisfaction rate.

Bandwidth costs: As shown in Figure 5(a), *Grandet* is more cost-effective than other algorithms. On the one hand, *Grandet* mitigates the impact of information agnostic by quantifying the uncertainty through interval determination. On the other hand, under the percentile charging model, *Grandet* coordinates



(a) Accumulative bandwidth cost in a month

(b) Accumulative deadline miss rate in a month

(c) Impact of control weight V

Fig. 6. The cost and deadline miss rate growth curves of different algorithms, and the impact of control weight V in the workload with 1000 sites.

free slots usage of each link from a global perspective, thus effectively transmitting more traffic with fewer free slots. In addition, when scheduling deferrable traffic, it can reasonably spread traffic to free slots and other slots before the deadline. As depicted in Table I, we can further observe that compared with Load Balance, Greedy, Cascara, and Intuitive optimization methods, *Grandet* reduces the bandwidth costs by more than 91%, 23%, 34%, and 40% respectively in the large-scale workload with 1000 sites.

To have a comprehensive understanding of the reasons for cost increase, we record the accumulative costs in days, as shown in Figure 6(a). The cost of the Load Balance method is exceedingly high, because the marginal cost of different links in the network topology varies greatly and the Load Balance method will exhaust the free slots of high-quality links within the first two days, which leads to significant cost increase. It is worth noting that the cost of Cascara is higher than greedy ones. This is because Cascara allocates traffic with the free slot as the first priority, leading to the situation that links with high marginal cost may prematurely use up free slots and billed bandwidth is forced to increase after the ninth day. As for intuitive optimization, due to the lack of free slots saving mechanism like Eq. (21), it may result in arbitrary traffic assignment until free slots are exhausted, that is, it may fall into local optimum. Therefore, compared with other pricing-aware methods, the cost of intuitive optimization increases steeply from the second day. Such results demonstrate that our algorithm is effective in reducing transmission costs.

Deadline miss rate: Generally speaking, if the basic bandwidth and the maximum elastic bandwidth are set reasonably, most deadlines can be satisfied. However, manually setting basic bandwidth and maximum elastic bandwidth cannot guarantee fault tolerance. At the same time, the SLA quality of links often fluctuates, leading to traffic backlog. These conditions cause deferrable traffic to queue up, resulting in deadline miss. As shown in Figure 5(b) and Figure 6(b), in terms of deadline miss rate, our algorithm has a slower growth rate than other algorithms. This is partly because our algorithm is more flexible than ES in rate control. Using free slots reasonably, a large amount of backlogged traffic can be transmitted in free slots, which can ultimately improve the deadline satisfaction rate. Besides, the interval determination method has higher accuracy in determining whether the link meets the SLA requirement, thus reducing allocation errors to immediate traffic that results in squeezing the transmission of deferrable traffic. As depicted in Table I, compared with Load Balance, Greedy, Cascara, and Intuitive optimization methods, *Grandet* can reduce the deadline miss rate by over 85%, 31%, 55%, and 41% respectively in the workload with 1000 sites.

SLA dissatisfaction rate: Link quality parameters (like jitter, delay, and packet loss rate) often fluctuate abruptly, making accurate prediction challenging. In fact, determining whether a link satisfies the SLA quality requirements of an application does not necessitate accurate prediction of quality, but requires a credible upper bound of quality parameters. Therefore, compared with other time series prediction algorithms, the confidence-based interval determination algorithm has more advantages. As long as the confidence probability parameter is set properly, the actual SLA quality can be bracketed in the estimation interval with a high probability, thus selecting the available links with high confidence. As shown in Figure 5(c), compared with exponential smoothing, using the monitoring value of the last moment, and direct prediction with a NeuralProphet model, our algorithm has a lower SLA dissatisfaction rate.

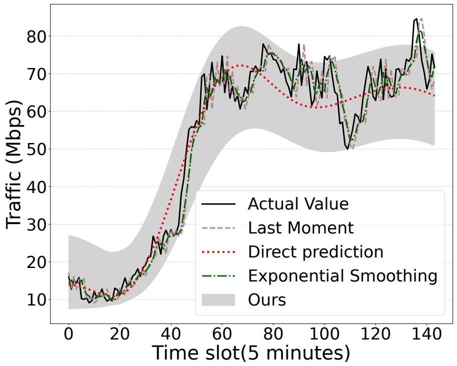


Fig. 7. The prediction result of different prediction algorithms in half a day

Traffic prediction accuracy: To show the difference between point prediction and interval determination, we intercept half-a-day traffic data of an application as an example, as shown in Figure 7. In order to obtain a more stable traffic interval estimation, daily seasonality and weekly seasonality parameters are set to be true, so the direct prediction result is relatively smooth. At first glance, exponential smoothing and using monitoring value of the last moment has a curve close to the real value. However, through careful observation, the difference between prediction values and the actual demands in each slot is 10% on average, which can make a significant performance deviation for algorithms using point prediction. In the case of interval estimation, traffic demands mostly fall within the estimation interval, thus providing credible knowledge to our traffic scheduling framework. Such results demonstrate that interval estimation can provide more reliable knowledge than point prediction methods.

Impact of control weight V: Control weight V has a significant impact on the performance of *Grandet* since it serves as a trade-off between transmission costs and task completion time. To evaluate the effect of weight V on transmission costs and deadline miss rate, we conduct a series of simulations with V ranging from 1 to 500. As shown in Figure 6(c), as V increases, the transmission cost decreases. On the contrary, as V increases, the deadline miss rate also increases. This is because the queue backlogs dominate P3 when V is small, while transmission costs dominate when V is large. Therefore, by choosing an appropriate value of V, such as $V \in [5, 100]$, our algorithm can significantly reduce transmission costs while guaranteeing an acceptable deadline miss rate.

VIII. CONCLUSIONS

In this paper, we reveal that existing traffic scheduling algorithms are highly dependent on the accuracy of flow size prediction. When the predicted value deviates from the actual value, existing algorithms may struggle to reach theoretical results. To this end, we present a novel traffic scheduler *Grandet*, which can effectively schedule traffic without prior knowledge and does not strongly rely on accurate prediction. In the design of *Grandet*, we use the Bootstrap method combined with the neural network to determine the range of flow sizes and link SLA quality parameters with high confidence. Then, we design an online traffic scheduling framework that can achieve approximate optimality in reducing transmission costs while guaranteeing QoS. Extensive trace-driven simulation results show that *Grandet* is cost-effective and QoS-guaranteeing.

REFERENCES

[1] D. Zad Tootaghaj, F. Ahmed, P. Sharma, and M. Yannakakis, "Homa: An efficient topology and route management approach in sd-wan overlays," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 2351–2360.

[2] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: Experience with a globally-deployed software defined wan," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, p. 3–14, aug 2013.

[3] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer, "Achieving high utilization with software-driven wan," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, p. 15–26, aug 2013.

[4] "Huawei sd-wan solution," 2022. [Online]. Available: https://support.huawei.com/enterprise/zh/doc/EDOC1100212003

[5] R. Stanojevic, N. Laoutaris, and P. Rodriguez, "On economic heavy hitters: Shapley value analysis of 95th-percentile pricing," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 75–80.

[6] "Sd-wan or mpls: A pricing analysis," 2022. [Online]. Available: https://www.sd-wan-experts.com/wp-content/uploads/2017/03/Pricing-Paper.pdf

[7] W. Li, X. Zhou, K. Li, H. Qi, and D. Guo, "Trafficshaper: Shaping inter-datacenter traffic to reduce the transmission cost," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1193–1206, 2018.

[8] V. Dukic, S. A. Jyothi, B. Karlas, M. Owaida, C. Zhang, and A. Singla, "Is advance knowledge of flow sizes a plausible assumption?" in *16th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2019, Boston, MA, February 26-28, 2019*, J. R. Lorch and M. Yu, Eds. USENIX Association, 2019, pp. 565–580.

[9] Z. Duliński, R. Stankiewicz, G. Rzym, and P. Wydrych, "Dynamic traffic management for sd-wan inter-cloud communication," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1335–1351, 2020.

[10] R. Singh, S. Agarwal, M. Calder, and P. Bahl, "Cost-effective cloud edge traffic engineering with cascara," in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*. USENIX Association, Apr. 2021, pp. 201–216.

[11] H. Shen and C. Qiu, "Scheduling inter-datacenter video flows for cost efficiency," *IEEE Transactions on Services Computing*, vol. 14, no. 3, pp. 834–849, 2021.

[12] P. Poupart, Z. Chen, P. Jaini, F. Fung, H. Susanto, Y. Geng, L. Chen, K. Chen, and H. Jin, "Online flow size prediction for improved network routing," in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, 2016, pp. 1–6.

[13] S. Wang, S. Wang, D. Zhou, Y. Yang, W. Zhang, T. Huang, R. Huo, and Y. Liu, "Large-scale and rapid flow size estimation for improving flow scheduling," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops*, 2020, pp. 1141–1146.

[14] M. Chowdhury and I. Stoica, "Efficient coflow scheduling without prior knowledge," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, ser. SIGCOMM '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 393–406.

[15] W. Bai, L. Chen, K. Chen, D. Han, C. Tian, and H. Wang, "PIAS: practical information-agnostic flow scheduling for commodity data centers," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 1954–1967, 2017.

[16] P. T. Anh Quang, S. Martin, J. Leguay, X. Gong, and F. Zeng, "Intent-based policy optimization in sd-wan," in *Proceedings of the SIGCOMM '21 Poster and Demo Sessions*, ser. SIGCOMM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 74–75.

[17] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[18] V. Jalaparti, I. Bliznets, S. Kandula, B. Lucier, and I. Menache, "Dynamic pricing and traffic engineering for timely inter-datacenter transfers," in *Proceedings of the 2016 ACM SIGCOMM Conference*, ser. SIGCOMM '16. New York, NY, USA: ACM, 2016, p. 73–86.

[19] Y. Zhang, X. Nie, J. Jiang, W. Wang, K. Xu, Y. Zhao, M. J. Reed, K. Chen, H. Wang, and G. Yao, "Bds+: An inter-datacenter data replication system with dynamic bandwidth separation," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 918–934, 2021.

[20] R. Xu, W. Li, K. Li, X. Zhou, and H. Qi, "Darkte: Towards dark traffic engineering in data center networks with ensemble learning," in *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, 2021, pp. 1–10.

[21] L. Golubchik, S. Khuller, K. Mukherjee, and Y. Yao, "To send or not to send: Reducing the cost of data transmission," in *2013 Proceedings IEEE INFOCOM*, 2013, pp. 2472–2478.

[22] O. Triebe, H. Hewamalage, P. Pilyugina, N. Laptev, C. Bergmeir, and R. Rajagopal, "Neuralprophet: Explainable forecasting at scale," *CoRR*, vol. abs/2111.15397, 2021.

[23] H. Du, E. Barut, and F. Jin, "Uncertainty quantification in cnn through the bootstrap of convex neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 13, pp. 12 078–12 085, May 2021.

[24] A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances," *IEEE Transactions on Neural Networks*, vol. 22, no. 9, pp. 1341–1356, 2011.

[25] "Gurobi." [Online]. Available: https://www.gurobi.com/