Joint Resource Provisioning for Internet Datacenters with Diverse and Dynamic Traffic

Dan Xu, Xin Liu, *Member, IEEE*, and Zhisheng Niu, *Fellow, IEEE*

Abstract—Demand proportional resource provisioning schemes have been proposed to achieve datacenter energy efficiency, where servers are turned on/off according to the load of requests. Most existing schemes focus on delay sensitive jobs (SENs) only. However, in datacenters, there exist a vast amount of delay-tolerant jobs (TOLs), such as background/maintenance jobs. Thus, we study joint SEN and TOL resource provisioning in this paper, with a focus on TOLs. We consider traffic dynamics of SENs and TOLs in different time scales, and electricity price temporal dynamics and location diversity. Our goal is to minimize total costs, while guaranteeing QoS for SENs and achieving a desirable delay performance for TOLs. Specifically, we propose a joint server provisioning, SEN load dispatching, TOL load shifting, and SEN/TOL capacity allocation scheme, which leverages TOL queue information and does not assume any system statistical information. We also design other benchmark schemes that leverage different system information. Both analytical results and extensive simulation results show the efficiency of the proposed scheme, named OrgQ, in reducing total costs and TOL queue delay.

Index Terms—Datacenters, energy efficiency, cost-effectiveness, delay-sensitive jobs, delay-tolerant jobs, queue, traffic diversity, traffic dynamics, stochastic optimization, convex optimization

1 Introduction

Cloud computing based Internet applications have been increasingly popular in recent years. Meanwhile, cloud service providers such as Google and Microsoft have to budget many millions of dollars for their Internet datacenters (IDCs) annually, in particular, for energy costs. Thus, how to provide desirable cloud services at a low cost is an important issue to be addressed.

Researchers have proposed various schemes to reduce IDC energy consumption. Among them, the so-called "capacity right-sizing" is a promising direction, e.g., in [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. The key idea is to provision servers dynamically based on the load of requests. Extra servers are proposed to be shut down or scheduled in a sleeping mode to save energy. In this paradigm, to determine a proper number of active servers, it is important to know the volume of load. For example, in [3], sophisticated statistical models are used to predict the load of a Microsoft datacenter that provides Live messenger services.

Just obtaining the load size information, however, is still far from a fine-grained load-awareness. In datacenters, there exist various jobs that have different traffic patterns and service requirements. The existing capacity right-sizing schemes mentioned above often focus on request-response interactive applications (e.g., search), which require a small service latency. In datacenters, besides those delay-sensitive jobs (SENs), there are also a large amount of delay-tolerant batch jobs, e.g., scientific computing jobs. Giving a higher priority to SENs, the "extra" servers can be utilized to process those delay-tolerant jobs (TOLs) rather than shut down, which is often referred to as *trough/valley filling*. Some existing work has considered resource provisioning for TOLs jobs only, e.g., in [12], [13], [14], [15], [16]. There are also some literatures considering both SENs and TOLs, e.g., [18], [19], however, joint resource provisioning for SENs and TOLs has not been studied in-depth yet. For example, in [18], the authors consider capacity for interactive workloads as a given variable, and optimize capacity for batch jobs only. In our paper, we fully consider energy costs and service requirements by SENs and TOLs, respectively, as well as their interactions. We design joint SEN and TOL provisioning schemes, where capacity for SENs and capacity for TOLs are both control variables. This is our first key contribution.

In addition to the prioritized service requirements of datacenter jobs, there are many other challenges in datacenter resource provisioning. On one hand, capacity demand of SENs and TOLs is time varying. Short-term SEN traffic dynamics cannot be avoided since SENs need to be served promptly (TOL traffic burstiness can be reduced by a buffer due to the relatively large service latency requirement.). On the other hand, turning on/off servers incurs a large time latency, i.e., up to several minutes [34]. Thus, one cannot tune the number of active servers based on the instantaneous capacity demand of SENs. More importantly, given a higher priority to serving SENs and the relatively static total server resource, available capacity for TOLs is random and usually difficult to predict or learn in statistics. Thus, joint capacity allocation for SENs and TOLs is challenging. Joint

- *D. Xu is with AT &T Labs, San Ramon, CA 94583. E-mail: danxu@research.att.com.*
- *X. Liu is with the Department of Computer Science, University of California, Davis, CA 95616. E-mail: xinliu@ucdavis.edu.*
- *Z. Niu is with Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing, P.R. China. E-mail: niuzhs@tsinghua.edu.cn.*

Manuscript received 29 July 2014; revised 9 Oct. 2014; accepted 14 Nov. 2014. Date of publication 18 Dec. 2014; date of current version 8 Mar. 2017. Recommended for acceptance by A. Vasilakos. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TCC.2014.2382118

resource provisioning in datacenters with both SEN and TOL traffic dynamics is our second key contribution.

In this paper, we consider a set of geo-distributed IDCs. For distributed IDCs, load shifting brings both opportunities and constraints. First, due to service agility, different classes of SENs or TOLs may require different sets of IDCs. Moreover, IDCs may be heterogenous in service rates and energy consumption for each class of SENs or TOLs. Thus, a wise load shifting scheme can improve service efficiency and reduce energy consumption. Second, electricity prices exhibit diversity in both location and time. As studied in [2], [5], [21], [23], price-aware load shifting can reduce energy costs significantly. In this paper, we leverage both location and temporal price diversity to reduce IDC energy costs. Different from server provisioning, load shifting can be performed in a small time scale, e.g., on the order of hundreds of milliseconds [22]. How to jointly and efficiently use server provisioning, load shifting, and SEN/TOL capacity allocation, which have different time granularities, to provision SENs and TOLs for distributed IDCs is a challenging problem, which is our another key contribution.

We study joint resource provisioning for SENs and TOLs. Our goal is to guarantee QoS of SENs, i.e., by constraining SEN overloading probability, and achieve a good delay performance for TOLs at a low cost. To achieve this goal, we design joint server provisioning, SEN load dispatching (from front-end portals to IDCs), TOL load shifting (among IDCs), and SEN/TOL capacity allocation. The joint schemes are configured and optimized by a decision maker based on an integrated convex optimization model. The decision-maker determines the number of active servers, SEN load dispatching ratios, TOL load shifting amount, and TOL capacity sharing ratios (as discussed in details later) in a large time scale, e.g., on the order of tens of minutes. Then the joint schemes are executed at different time granularities. Server provisioning is performed with a large time interval, i.e., the same as that of the decision-maker computing system parameters. In a smaller time scale, e.g., hundreds of milliseconds, instantaneous SEN load dispatching is performed based on the current dispatching ratios computed by the decision-maker. When SENs arrive an IDC, capacity allocation is performed instantaneously to serve the SENs. TOL load shifting is also performed in a small time scale following the optimal configurations. Then, capacity allocation is performed instantaneously to provision TOLs based on both the remaining instantaneous capacity for TOLs and TOL capacity sharing ratios at each IDC. Our main contributions are summarized as follows:

- We explicitly differentiate SENs and TOLs in IDCs. We consider joint SEN and TOL resource provisioning, with traffic dynamics of both SENs and TOLs considered. Both the large-time-scale, i.e., hourly, and small-time-scale traffic dynamics, i.e., hundreds of milliseconds, of SENs are considered to capture the real-world traffic models.
- We design joint server provisioning, SEN load dispatching, TOL load shifting, and SEN/TOL capacity allocation schemes for geo-distributed IDCs with different time granularities. Our schemes minimize the total energy costs, assure the QoS for SENs, and guarantee TOL queue stability. Note that we focus on TOL provisioning in this paper. Specifically, we propose a queue-based trough-filling scheme, named OrgQ. We also consider a back-pressure routing based TOL provisioning scheme, named SubQ, and find its disadvantages in the scenario of geo-distributed IDCs with electricity price diversity. Moreover, to show the advantages of OrgQ, we also design benchmark schemes which do not leverage any TOL queue information, in both a stationary ergodic setting and a non stationary ergodic setting.
- We perform extensive simulations to compare the performance of OrgQ to other schemes based on simulated traffic trace and real traffic trace. Our results show that OrgQ outperforms both the benchmarks and SubQ, since it can achieve a better tradeoff between costs and queue delay. We also show various properties of our proposed schemes which help people better understand datacenter resource provisioning.

The rest of paper is organized as follows. In Section 2, we discuss related work. In Section 3, we describe the system model. In Section 4, we design benchmark resource provisioning schemes. We propose queue based joint resource provisioning schemes, i.e., SubQ and OrgQ in Section 5. We discuss the implementation and other issues in Section 6. We evaluate our proposed schemes in Section 7, followed by conclusions in Section 8.

2 Related Work

Our work is closely related to capacity right-sizing or power-proportional design [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. For example, in [1], the authors proposed server provisioning and dynamic speed/voltge scaling (DVS) schemes for a data center, through load prediction and feedback control. Load prediction-based server provisioning and load dispatch were proposed in [3] for connection-intensive Microsoft datacenter. In [2], the authors minimized total energy costs of geo-distributed datacenters with a delay constraint for interactive jobs. In [4], the authors considered a relatively large time interval such that current load of requests can be estimated. The authors implicitly considered interactive jobs only. The authors studied the impact of trough filling on energy saving by the proposed scheme through simulations. In [5], the authors designed distributed load shifting and resource provisioning scheme for geo-distributed IDCs. However, they didn't differentiate delay tolerant jobs and delay sensitive jobs either. The authors of [6], [7] designed power-proportional resource provisioning schemes but didn't differentiate delay sensitive jobs and delay tolerant jobs either.

The above work mainly focuses on the load of interactive jobs, with service level agreement (SLA) or other QoS metrics assured. Delay tolerant jobs, e.g., batch jobs were not explicitly considered. In our earlier conference paper [13], we proposed a Lyapunov optimization technique based online dynamic speed scaling algorithm to provision delay tolerant jobs for geo-distributed IDCs, which exploits electricity price diversity to save energy costs. In the same conference, the authors of [14], proposed similar server

provisioning method for delay-tolerant load in one datacenter. In a recent work of [15], the authors studied delay tolerant batch jobs scheduling problem among geo-distributed datacenters. They also proposed a Lyapunov optimization technique based online algorithm to exploit electricity price diversity and also considered server inlet temperature constraint. In another recent work [16], the authors proposed scheduling algorithms to provision virtual machine resource for delay tolerant scientific jobs. Their optimization objective was to minimize total execution costs with the constraints of completion deadline of scientific workflow. In addition, job queue based power management schemes for a datacenter or multi-servers were also studied in [17].

Some work considered both interactive jobs and batch jobs but did not jointly optimize resource provisioning for them. For example, in [18], the authors considered capacity for interactive workloads as a given variable. They optimized capacity allocation for batch jobs and renewable energy usage for one datacenter. In [19], the authors considered capacity allocation the interactive jobs and batch jobs. However, in their optimization model, they only decided the number of servers in each IDC that should be allocated to batch jobs. They did not model the performance of batch jobs and did not consider different classes of batch jobs.

Our work is significantly different from all the above work in the following aspects. First, we consider both delay sensitive jobs and delay tolerant jobs and design joint IDC resource provisioning for them. In our scheme, capacity for SENs and TOLs are both control variables. We also design different QoS mechanisms for SENs and TOLs, respectively, which achieve a good tradeoff between cost effectiveness and service performance. Second, we consider traffic dynamics of delay sensitive jobs in different time granularities. Considering small-time-scale traffic variation leads to an interesting and challenging problem of joint SEN and TOL resource provisioning. Third, from an algorithmic perspective, we explore three different schemes with different system information leveraged. Our schemes include joint server provisioning, SEN load dispatching, TOL load shifting, and SEN/TOL capacity allocation. Each of the joint scheme has different time granularity requirement. We jointly optimize all of them by an integrated optimization model.

Overloading is an important problem in datacenters. In our paper, we use overloading probability to guarantee SEN performance. There is a work studying overloading detection problem in virtualized clouds [20], where the authors design offline algorithm to detect host overloading for stationary workload and extended it to non stationary workload based on a slide window method. There are many papers on other related topics, such as inter-datacenter load shifting [21], [22], [23], [24], dynamic speed scaling [25], [26], datacenter traffic engineering, virtual switch assignment for cost effectiveness [27], [28], [29], and auction or game theory based datacenter resource provisioning [32], [33].

3 System Models

3.1 The IDC and Server Model

We consider one service provider with N IDCs in different locations. For each IDC, we determine the number of active servers, i.e., by server provisioning, in a relatively large time scale. We define the time interval between two adjacent server provisioning as one time slot. Let t denote a generic time slot. For simplicity, we assume different time slots have the same duration, denoted by T_p (in sec), which is about tens of minutes, e.g., as in [3], [4]. Let C_i^t denote the number of active servers in time slot t, which is a control variable. C_i^t is bounded by C_i^m, i.e., total number of servers in IDC i.

An active server operates at a CPU speed of s (Hz). Following the models in [22], [25], [26], we normalize s, i.e., $0 \leq s \leq 1$, where 0 represents the idle state of an active server, and 1 represents the maximum frequency. For simplicity, we assume servers are homogenous in the maximum speeds. We define the capacity of an IDC i as the sum speed of all active servers. If each server runs at the same speed s, the total capacity in time slot t is $C_i^t s$. Clearly, the maximum capacity with C_i^t active servers is C_i^t. In this paper, we consider CPU resource as the the main bottleneck and focus on CPU capacity scheduling. The impact of other equipments, i.e., memory and I/O, will be considered in heterogenous service rates, as specified later. Note that scaling up/down the speed s of an active server only takes several microseconds [26], which is negligible.

3.2 Workload Model

We consider two categories of workloads: delay sensitive jobs (SENs) and delay tolerant jobs (TOLs). SENs tolerate a small service latency and have a higher service priority. The remaining capacity can be utilized by the TOLs, which can be served with a large delay, e.g., from minutes to hours.

We consider different classes of SENs. First, different types of service requests, e.g., search, web browsering, and email login, are considered as different classes of SENs, because they may have different traffic patterns, service requirements, and resource usages. Further, if the same types of SENs originate (first arrive) at different front-end portals, we treat them as different classes, because they may need to be served by different sets of IDCs. For example, it is desirable to let search requests from San Francisco be served by west coast IDCs and let those from New York be served by east coast ones in the U.S. In this paper, we consider J classes of SENs, indexed by j, $j \in \{1, 2, \ldots, J\}$.

Traffic of SEN j (SEN j refers to a class of jobs instead of a single job) varies over time, in both large time scales and small time scales. Let T_s denote the small time interval at which traffic of SEN j is measured. T_s can be from tens of milliseconds to seconds. Thus a time slot t can be further divided into $n_s = \frac{T_p}{T_s}$ small time slots, named sub-slot. Let $D_j^{t\tau}$ (in bit) denote the workload of SEN j of a sub-slot τ in time slot t. $D_j^{t\tau}$ varies randomly over different sub-slots in time slot t, which is considered as the small-time-scale traffic variation. We assume $D_j^{t\tau}$ is identically (but NOT independently) distributed for different sub-slots in time slot t. The mean and standard deviation of $D_j^{t\tau}$ are denoted by λ_j^t and σ_j^t, respectively. λ_j^t and σ_j^t vary over different time slot t, which is considered as the large time scale traffic variation. In the beginning of each time slot t, λ_j^t and σ_j^t can be estimated, as in [4].

TOLs mainly include background analytical and maintenance jobs, which can also be divided into different classes to capture different resource requirements and locations of

sources. TOLs originate at an IDC, instead of being dispatched from a front-end portal. That is, TOLs usually do not come from Internet users. We consider K different classes of TOLs in the N IDCs. Let k index a class of TOLs, $k \in \{1, 2, \ldots, K\}$. Let D_k^t (in bit) denote the arrival traffic size of TOL k in time slot t. D_k^t varies over different time slot t. Let λ_k denote the average traffic arrival rate of TOL k. There is $\lambda_k = E(D_k^t)/T_p$. We introduce $\vec{\lambda^l} = (\lambda_k | k = 1, \ldots, K)$ as the TOL traffic arrival rate vector.[1] The small-time-scale traffic variations of TOLs can be ignored since their traffic can be smoothed via a large buffer.

3.3 Service Model

3.3.1 SENs

A front-end portal, such as a DNS server, dispatches SENs to IDCs. Due to distance constraints, a front-end portal may connect to a subset of the N IDCs. Thus, a class of SENs receive service from a subset of IDCs. Let Γ_j denote the set of IDCs that receive and serve SEN j, which is different for different classes of SENs. For simplicity, we consider a dispatching model where SEN j is shifted to IDC i, $i \in \Gamma_j$, according to a fixed ratio r_{ij}^t in time slot t. We have $\sum_{i \in \Gamma_j} r_{ij}^t = 1$ as the SEN load dispatching constraint.

SENs need to be served with a small time latency. For simplicity, we assume that to satisfy the total time latency requirement (taking load dispatching delay into account), a single unit of job of SEN j requires $\frac{1}{\mu_{ij}}$ units of (normalized) capacity at IDC i. Or alternatively, one unit of capacity can serve μ_{ij} units of SEN jobs with the service latency requirement satisfied. Thus, capacity demand of SEN j at an IDC i, $i \in \Gamma_j$, in sub-slot τ of time slot t, denoted by $S_{ij}^{t\tau}$ (per sec), is equal to $\frac{r_{ij}^t D_j^{t\tau}}{\mu_{ij} T_s}$. More generally, a convex function can be used to model SEN traffic capacity demand for our schemes. The unit capacity requirement μ_{ij} is heterogenous for different pairs of SEN j and IDC i. This is because, different SENs may require different memory, I.O. resource, and etc. Consider an IDC i. Let Π_i^h denote the set of classes of SENs that are dispatched to it. The total capacity demand by SENs at IDC i is thus $\sum_{j \in \Pi_i^h} S_{ij}^{t\tau} = \sum_{j \in \Pi_i^h} \frac{r_{ij}^t D_j^{t\tau}}{\mu_{ij} T_s}$.

SEN capacity demand varies in time slot t, while the maximum available capacity C_i^t is fixed in time slot t. Thus, it is likely that at an IDC, capacity demand by SENs is larger than the total available capacity in some sub-slots, i.e., overloading occurs. In this case, SENs may have a large service latency. Thus, overloading needs to be constrained. We require a QoS metric, named overloading probability, to be constrained by a threshold δ_i at each IDC i. That is

$$\Pr\left(\sum_{j \in \Pi_i^h} S_{ij}^{t\tau} > C_i^t\right) \leq \delta_i. \tag{1}$$

Clearly, the overloading probability in (1) is temporal, which indicates the time fraction of SENs' receiving a sub-optimal service. We use it to control SEN QoS in our optimization model. In the simulations, we study SEN queue delay. Note that overloading probability is defined for aggregated SENs in an IDC. An overloading may be incurred just by a single or some classes of SENs. Thus overloading probability for each specific class of SENs is smaller than δ_i.

The SEN capacity demand at IDC i, i.e.,

$$\sum_{j \in \Pi_i^h} S_{ij}^{t\tau} = \sum_{j \in \Pi_i^h} \frac{r_{ij}^t D_j^{t\tau}}{\mu_{ij} T_s},$$

may follow a certain distribution. In this paper, we model the distribution of capacity demand by SENs at each IDC i by Gaussian distribution based on Central Limit Theorem (CLT). To apply CLT, we implicitly assume traffic of each class of SENs is independent. The intuition of this assumption is as follows: traffic dependency among different classes of SENs is typically exhibited in a large time scale by traffic statistics, e.g., each class of SENs has more traffic during daytime than during night. In our schemes, traffic statistics are given inputs to the overloading probability model. We model traffic of each class of SENs as a random variable in a small time scale, i.e., every sub-slot. It is reasonable to assume they are independent in a small time scale, since it is unlikely that different classes of SENs have the same traffic pattern every tens of milliseconds. The approximation is close when Π_i^h is a large set. In our setting, there can be over tens of different classes of SENs which come from different locations. Thus it is reasonable to apply CLT. Gaussian distribution is widely used in the existing literature to approximate the distribution of the aggregated load or bandwidth demand, e.g., in [35], [36]. Since $\sum_{j \in \Pi_i^h} \frac{r_{ij}^t D_j^{t\tau}}{\mu_{ij} T_s}$ has a mean of $\sum_{j \in \Pi_i^h} \frac{r_{ij}^t \lambda_j^t}{\mu_{ij} T_s}$ and a standard deviation of $\sqrt{\sum_{j \in \Pi_i^h} \frac{{r_{ij}^t}^2 \sigma_j^{t\,2}}{{\mu_{ij}}^2 {T_s}^2}}$. Let's introduce a random variable

$$X = \left(\sum_{j \in \Pi_i^h} \frac{r_{ij}^t D_j^{t\tau}}{\mu_{ij} T_s} - \sum_{j \in \Pi_i^h} \frac{r_{ij}^t \lambda_j^t}{\mu_{ij} T_s}\right) \Bigg/ \sqrt{\sum_{j \in \Pi_i^h} \frac{{r_{ij}^t}^2 \sigma_j^{t\,2}}{{\mu_{ij}}^2 {T_s}^2}}.$$

Thus X follows standard Gaussian distribution, i.e., with mean of 0 and deviation 1. Further, consider the Q-function, i.e., the tail probability of the standard Gaussian distribution, $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-\frac{\mu^2}{2}) d\mu$. Clearly, $Q(x)$ is a decreasing function. So does its reverse function, $Q^{-1}()$. We can write (1) as

$$-C_i^t + \sum_{j \in \Pi_i^h} \frac{r_{ij}^t \lambda_j^t}{\mu_{ij} T_s} + Q^{-1}(\delta_i) \sqrt{\sum_{j \in \Pi_i^h} \frac{{r_{ij}^t}^2 \sigma_j^{t\,2}}{{\mu_{ij}}^2 {T_s}^2}} \leq 0, \tag{2}$$

which is the overloading probability constraint under Gaussian distribution. We later use $f(\delta_i)$ for $Q^{-1}(\delta_i)$ to avoid notation confusion with TOL queues. We also use λ_{ij}^t and σ_{ij}^t to denote $\frac{\lambda_j^t}{\mu_{ij} T_s}$ and $\frac{\sigma_j^t}{\mu_{ij} T_s}$, respectively. Since the real aggregated SEN traffic at each IDC may not follow a Gaussian distribution, we will evaluate other distributions of SEN traffic in the simulations, and discuss how to address the distribution discrepancy.

1. In this paper, the superscript h means higher priority, which is used for SENs. The superscript l, i.e., lower priority, is used for TOLs.

3.3.2 TOLs

Traffic of TOLs can be shifted from the IDC it originates to other IDCs to exploit their available capacity or lower prices. Let $i^{'}$ denote the IDC a TOL k originates at. Due to distance or other constraints, a class of TOLs can only be shifted to a subset of IDCs. Similarly, let Γ_k denote the set of IDCs that can receive and serve TOL k, which is different for different classes of TOLs.

TOL load shifting is constrained by the link bandwidth between two IDCs. Let $B_{i^{'}i}$ denote the total link bandwidth between IDC $i^{'}$ and i. When $i = i^{'}$, we can set $B_{i^{'}i} = \infty$. Further, let $B^t_{i^{'}ik}$ denote the bandwidth assigned to TOL k between IDC $i^{'}$ and i in time slot t. $B^t_{i^{'}ik}$ is a control variable. Let $\Upsilon_{i^{'}i}$ denote the set of TOLs that first arrive at IDC $i^{'}$ and can be served by IDC i. Thus, we have $\sum_{k\in\Upsilon_{i^{'}i}} B^t_{i^{'}ik} \le B_{i^{'}i}$ as the TOL load shifting or link bandwidth constraint. Let $D^t_{i^{'}ik}$ denote the amount of traffic (in bit) of TOL k shifted from IDC $i^{'}$ to i in time slot t. We have $D^t_{i^{'}ik} = B^t_{i^{'}ik}T_p$.

TOL k is served by the remaining capacity when capacity SENs is guaranteed. Clearly, the overall available capacity for TOLs varies randomly over different sub-slots in time slot t. Thus, one cannot directly determine the capacity of each sub-slot for each class of TOLs in the beginning of a time slot t, which is not necessary either. We consider a capacity allocation model where different classes of TOLs share the available capacity according to fixed ratios during time slot t. Let r^t_{ik} denote the capacity sharing ratio of TOL k at IDC i in time slot t. Thus, in a sub-slot τ, TOL k receives a capacity of

$$S^{t\tau}_{ik} = \max\left[0, r^t_{ik}\left(C^t_i - \textstyle\sum_{j\in\prod_i^h} S^{t\tau}_{ij}\right)\right].$$

Let Π^l_i denote the set of TOLs served by IDC i. We have $\sum_{k\in\Pi^l_i} r^t_{ik} \le 1$ as the TOL capacity allocation constraint. r^t_{ik} is a control variable jointly determined with C^t_i and r^t_{ij}. Determining r^t_{ik} is considered as a part of configuring trough-filling (in a large time scale). With r^t_{ik}, instantaneous trough-filling (in a small time scale) is performed in each sub-slot τ after observing $\sum_{j\in\Pi^h_i} S^{t\tau}_{ij}$, i.e., allocating capacity of $S^{t\tau}_{ik}$ to TOL k.

However, it is difficult to optimize r^t_{ik} directly. Let's first make an approximation of the expectation of $S^{t\tau}_{ik}$ as

$$\begin{aligned}\hat{S^t_{ik}} = E\left(S^{t\tau}_{ik}\right) &= E\left\{\max\left[0, r^t_{ik}\left(C^t_i - \sum_{j\in\prod_i^h} S^{t\tau}_{ij}\right)\right]\right\} \\ &\approx \max\left[0, r^t_{ik}\left(C^t_i - \sum_{j\in\Pi^h_i} r^t_{ij}\lambda^t_{ij}\right)\right].\end{aligned} \tag{3}$$

The above approximation is reasonable since we consider a small overloading probability, i.e., a small chance of the case $C^t_i \le \sum_{j\in\Pi^h_i} S^{t\tau}_{ij}$. Further, in our optimization model, let's consider a constraint of $\sum_{k\in\Pi^l_i} \hat{S^t_{ik}} + \sum_{j\in\Pi^h_i} r^t_{ij}\lambda^t_{ij} \le C^t_i$, i.e., the expected capacity of SENs and TOLs being smaller than C^t_i. Thus we have $\hat{S^t_{ik}} \approx r^t_{ik}(C^t_i - \sum_{j\in\Pi^h_i} r^t_{ij}\lambda^t_{ij})$ hold. We optimize $\hat{S^t_{ik}}$ instead of r^t_{ik}. Then $\hat{S^t_{ik}}$, C^t_i, and r^t_{ij} are control variables of our formulated convex optimization problems, each of which has a unique solution. Thus, the capacity sharing ratios r^t_{ik} can be uniquely approximated by $\hat{S^t_{ik}}/(C^t_i - \sum_{j\in\Pi^h_i} r^t_{ij}\lambda^t_{ij})$. Further, the constraint $\sum_{k\in\Pi^l_i} \hat{S^t_{ik}} + \sum_{j\in\Pi^h_i} r^t_{ij}\lambda^t_{ij} \le C^t_i$ makes the constraint $\sum_{k\in\Pi^l_i} r^t_{ik} \le 1$ satisfied. Since max function is convex, by Jensen's inequality (i.e., $f(E(X)) <= E(f(X))$ whenever f is a convex function), the actual $\hat{S^t_{ik}}$ is larger than the approximated version. Thus the approximation of $\hat{S^t_{ik}}$ is conservative. Note that the expected capacity $\hat{S^t_{ik}}$ is different from the average capacity of TOL k at IDC i given a time slot t, denoted by S^t_{ik}, which is calculated by $\frac{1}{n_s}\sum_{\tau=1}^{n_s} S^{t\tau}_{ik}$. The latter one is still a random variable since $S^{t\tau}_{ik}$ is random in each sub-slot.

As mentioned, TOLs are not served prompted. They wait for server resource in a buffer. In the IDC $i^{'}$ where TOL k originates, there is a queue for the unfinished jobs of TOL k. Let $Q_k(t)$ denote the queue length in the beginning of time slot t. $Q_k(t)$ depends on traffic served by IDC $i^{'}$ and that shifted to other IDCs, and the traffic arrival size D^t_k, in each time slot t. We have

$$Q_k(t+1) = \max\left[Q_k(t) - \sum_{i\in\Gamma_k} D^t_{i^{'}ik}, 0\right] + D^t_k. \tag{4}$$

$Q_k(t)$ can be considered as the length of the original queue, abbreviated as o-queue, of TOL k. Moreover, TOLs shifted to an IDC i are also buffered in a queue, named sub-queue, abbreviated as s-queue. Note that we can consider there is a s-queue in IDC $i^{'}$ where TOL k originates for notation consistency. Let $Q_{ik}(t)$ denote the s-queue length of TOL k at IDC i in the beginning of time slot t. Clearly, $Q_{ik}(t)$ depends on how much traffic of TOL k shifted to IDC i, and how much served by IDC i. We have the following s-queue dynamics

$$Q_{ik}(t+1) = \max\left[Q_{ik}(t) - R^t_{ik}T_p, 0\right] + D^t_{i^{'}ik}, \tag{5}$$

where $R^t_{ik} = \mu_{ik}S^t_{ik}$, i.e., the average service rate for TOL k in IDC i in time slot t.

Following the above TOL queue dynamics, there can be different TOL load shifting models. One model is that how much IDC i can serve, how much to shift from IDC $i^{'}$ in a time slot t. By this way, TOL load shifting is closely coupled with TOL capacity allocation. As showed later, the proposed Benchmark II scheme and the OrgQ scheme follow this method, i.e., by setting $B^t_{i^{'}ik} = \mu_{ik}\hat{S^t_{ik}}$ in each time slot t. Here both $B^t_{i^{'}ik}$ and $\hat{S^t_{ik}}$ are control variables. Note that we cannot set $B^t_{i^{'}ik} = \mu_{ik}S^t_{ik}$ since S^t_{ik} is random. Another TOL load shifting model is to decouple TOL load shifting from TOL capacity allocation. In this case, $D^t_{i^{'}ik}$ has no direct relation with $\mu_{ik}\hat{S^t_{ik}}$. The later proposed Benchmark I and SubQ follow this model. We will discuss the advantages of the former one later.

TABLE 1
Main Notations

Input	Including direct inputs and notations of parameters that lead to direct inputs		
T_p	Length of a time slot, i.e., time interval for server provisioning	T_s	Length of a sub-slot
j (k)	Index of classes of SENs (TOLs) ($j = 1, \ldots, J$, and $k = 1, \ldots, K$)	$\mu_{ij}(\mu_{ik})$	Service speed per unit of capacity by IDC i for SEN j (TOL k)
$\Pi_i^h(\Pi_i^l)$	Set of classes of SENs (TOLs) that can be served by i	$\Upsilon_{i'i}$	Set of classes of TOLs that can be shifted from i' to i
$\Gamma_j(\Gamma_k)$	Set of IDCs that can serve SEN j (TOL k)	α_i^t	Electricity price of IDC i in time slot t
$D_j^{t\tau}$	Traffic of SEN j in sub-slot τ of time slot t, with mean λ_j^t and deviation σ_j^t (Take λ_j^t and σ_j^t as the inputs)		
D_k^t	Traffic arrival size of TOL k in time slot t, with mean $\lambda_k T_p$ (Take λ_k as the input in StoS)		
Output	Including control variables (CVs) in the vector $\mathbf{X}^t = \{C_i^t, r_{ij}^t, B_{i'ik}^t, \hat{S_{ik}^t}\}$ and indirect outputs which are based on control variables		
C_i^t	#active servers (or capacity) of IDC i in t (**CV**)	r_{ij}^t	Load dispatch ratio to IDC i of SEN j in time slot t (**CV**)
$B_{ii'k}^t$	Bandwidth between IDC i' and i for TOL k in t (**CV**)	$\hat{S_{ik}^t}$	Expected capacity at IDC i of TOL k in t, ($E(S_{ik}^{t\tau})$, **CV**)
$S_{ij}^{t\tau}$	Capacity demand (per sec) in sub-slot τ of time slot t at IDC i by SEN j, $S_{ij}^{t\tau} = r_{ij}^t D_j^{t\tau}/(\mu_{ij} T_s)$ (Indirect output)		
$S_{ik}^{t\tau}$	Capacity (per sec) received by TOL k in sub-slot τ of time slot t at IDC i, i.e., $r_{ik}^t(C_i^t - \sum_{j\in\Pi_i^h} S_{ij}^{t\tau})$ (Indirect output)		

We first consider a TOL provisioning scheme that does not leverage the information of $Q_k(t)$ and $Q_{ik}(t)$. We further propose an algorithm that uses $Q_k(t)$ only. Our last algorithm considers both $Q_k(t)$ and $Q_{ik}(t)$. For all the algorithms we proposed, the queue dynamics of $Q_k(t)$ and $Q_{ik}(t)$ follow (4) and (5), respectively. Note that we don't model queue dynamics for every sub-slot.

3.4 Control Variables and Main Constraints

In summary, our control variables include the maximum available capacity of IDC i in time slot t, i.e., number of active servers C_i^t, SEN traffic dispatching ratios, r_{ij}^t, link bandwidth assigned to TOL k between i' and i, $B_{i'ik}^t$, expected capacity for TOL k at IDC i, $\hat{S_{ik}^t}$. We can write the control variable vector as $\mathbf{X}^t = \{C_i^t, r_{ij}^t, B_{i'ik}^t, \hat{S_{ik}^t} | i', i = 1, \ldots, N; j = 1, \ldots, J; k = 1, \ldots, K\}$.

A control variable vector $\mathbf{X}^t$ needs to satisfy the server provisioning constraints ($C_i^t \le C_i^m$), load dispatching constraints for SENs, overloading probability constraints for SENs, bandwidth allocation constraints for TOLs, and capacity allocation constraints for both SENs and TOLs (which are equivalent to the capacity sharing constraints of TOLs). Let Λ^t denote the set of $\mathbf{X}^t$ that satisfy the five types of constraints in time slot t. We list the main notations in Table 1.

3.5 Power Consumption and Cost Model

According to [25], [26], power consumption of a server (processor) running at a speed $s \in [0, 1]$ is

$$P(s) = \upsilon s^{\kappa} + 1 - \upsilon, \tag{6}$$

where the exponent $\kappa \ge 1$, and typically takes a value of 1 or 2 [26]. In this paper, we choose $\kappa = 1$, i.e., a linear power-speed model. $1 - \upsilon$ represents the power consumption in the idle state, which is around 0.6, and hardly lower than 0.5 [3].

Consider an IDC i. In a time slot t, there are C_i^t active servers, and the total capacity demand in sub-slot τ is $\sum_{j\in\Pi_i^h} S_{ij}^{t\tau} + \sum_{k\in\Pi_i^l} S_{ik}^{t\tau}$. Clearly, the power consumption of servers at IDC i in time slot t, P_i^t, has an expectation as

$$E[P_i^t] = (1 - \upsilon) C_i^t + \upsilon \sum_{j\in\Pi_i^h} r_{ij}^t \lambda_{ij}^t + \upsilon \sum_{k\in\Pi_i^l} \hat{S_{ik}^t}. \tag{7}$$

We consider energy costs of an IDC i as the product of power consumption and its electricity price α_i^t, which is different for different time slots and different IDCs.

3.6 Load Shifting Costs

We consider cross-IDC load shifting for TOLs. We do not model SEN load shifting among IDCs since it is usually not desirable due to excessive delay incurred. However, in some scenarios, e.g., datacenter overloading, SENs may be shifted from an IDC to another one. Thus, when performing TOL load shifting, there is a risk that the potential SEN load shifting is delayed, especially when TOLs take a large link bandwidth between two IDCs. To reduce such a risk, we use a piece-wise linear cost function with increasing slopes to model the shifting costs for TOLs. Let $\phi_{i'i}^t$ denote the shifting costs in time slot t between IDC i' and i, we have

$$\phi_{i'i}^t = \max\left(a_{ii'}^{\vartheta} \frac{\sum_{k\in\Upsilon_{i'i}} B_{i'ik}^t}{B_{i'i}} + b_{ii'}^{\vartheta} \right), \vartheta = \{1, \ldots, \theta\}, \tag{8}$$

where $\frac{\sum_{k\in\Upsilon_{i'i}} B_{i'ik}^t}{B_{i'i}}$ is the link bandwidth occupation ratio by TOLs. We have $a_{ii'}^1 \le \cdots a_{ii'}^{\vartheta} \cdots \le a_{ii'}^{\theta}$, i.e., an increasing slope with the link bandwidth for TOLs, which captures the increasing risk of delaying SEN load shifting. $b_{ii'}^{\vartheta}$ can be interpreted as other fixed bandwidth costs such as link construction or maintenance costs. $\phi_{i'i}^t$ is a convex function on $B_{i'ik}^t$, and thus on $\mathbf{X}^t$, since it is the pointwise maximum of a set of affine functions. The model is widely considered by previous works, e.g., in [37]. With slight modification, our work can also incorporate other shifting costs such as routing energy costs [21] and bandwidth costs [23].

4 Cost-Optimal Benchmarks

In this section, we design benchmark schemes, which do not leverage any TOL queue information, to evaluate the latter proposed queue-based joint resource provisioning scheme. In the benchmark schemes, our objective is to minimize the time average of the total costs of N IDCs, while satisfying the

QoS (overloading probability) requirements for SENs and stabilizing all o-queues and s-queues of TOLs. We first design benchmarks in a stationary and ergodic setting. For a stationary and ergodic stochastic process, the joint probability distribution of system states does not change with time. We use Ω to denote a set of system states. Ω have steady time distributions according to the assumption on stationary and ergodic setting. In a generic time slot t, the system stays in a state ω, $\omega \in \Omega$. A system state ω characterizes a unique set of system input parameters, among which electricity prices α_i^t, and SEN traffic statistics λ_j^t and σ_j^t, vary over different time slots.

Let π_ω denote the steady distribution of ω, i.e., the time fraction of staying in state ω. Let $\mathbf{X}^\omega$ denote the control variable associated with the state ω, which is in the set Λ^ω. Clearly, $\mathbf{X}^\omega$ is the same for different time slots with the same state ω. Further, let $g^\omega()$ denote the average cost function in state ω. The optimization problem can be written as

$$\min\ g_e = \sum_{\omega\in\Omega} \pi_\omega E[g^\omega(\mathbf{X}^\omega)] \tag{9}$$

$$\text{s.t.}\ \sum_{\omega} \pi_\omega \sum_{i\in\Gamma_k} B^\omega_{i'ik} > \lambda_k, \tag{10}$$

$$\sum_{\omega} \pi_\omega E(R^\omega_{ik}) > \sum_{\omega} \pi_\omega B^\omega_{i'ik}, \tag{11}$$

$$\mathbf{X}^\omega \in \Lambda^\omega, i \in \Gamma_k, k = 1 \ldots K, \tag{12}$$

where the objective function is the average total costs. The LHS and RHS of (10) are the average service rate and the average arrival rate of each o-queue of TOL k. R^ω_{ik} is the average service rate by IDC i for TOL k in state ω, which is random for the same state ω, since S^ω_{ik} is random given a state ω. We have $E(R^\omega_{ik}) = \mu_{ik}\hat{S^\omega_{ik}}$. The LHS and RHS of (11) are the average service rate and the average arrival rate of each s-queue of TOL k at IDC i. The conditions of (10) and (11) are to guarantee the stability of each o-queue and s-queue, respectively. We have the Lemma 1 for the property of (9)-(12).

Lemma 1. *(9)-(12) is a convex optimization problem.*

Proof. Please refer to the supplemental material, available in http://doi.ieeecomputersociety.org/10.1109/TCC.2014.2382118. □

The solution to (9)-(12) can be computed efficiently. We name the solution Benchmark I, which can be used to establish the cost bounds of the latter proposed SubQ. We also modify (9)-(12) by setting $B^\omega_{i'ik} = \mu_{ik}\hat{S^\omega_{ik}}$. Then the constraints (10) and (11) will be replaced by $\sum_\omega \pi_\omega \sum_{i\in\Gamma_k} \mu_{ik}\hat{S^\omega_{ik}} \geq \lambda_k$, $\forall k$. Moreover, the TOL load shifting constraint in Benchmark II becomes $\sum_{k\in\Upsilon_{i'i}} \mu_{ik}\hat{S^\omega_{ik}} \leq B_{i'i}$. We name the solution to the modified optimization problem as Benchmark II, which is used to establish the cost bounds of the latter proposed OrgQ, since they have the same feasibility set Λ^ω (i.e., Λ^t for OrgQ).

Both Benchmark I and II work in a stationary and ergodic setting and require system distribution information π_ω. We next design a benchmark scheme that does not require such information and can thus work in a non stationary ergodic setting.

We first define a Lagrangian function associated with problem (9)-(12) as

$$\begin{aligned} L(\vec{\nu},\vec{\gamma},\vec{X}) = &\sum_{\omega\in\Omega} \pi_\omega E[g^\omega(\mathbf{X}^\omega)] - \sum_{k=1}^{K} \nu_k \left(\sum_{\omega\in\Omega} \pi_\omega \sum_{i\in\Gamma_k} B^\omega_{i'ik} - \lambda_k \right) \\ &- \sum_{k=1}^{K} \sum_{i\in\Gamma_k} \gamma_{ik} \left[\sum_{\omega\in\Omega} \pi_\omega \left(\mu_{ik}\hat{S^\omega_{ik}} - B^\omega_{i'ik} \right) \right], \end{aligned} \tag{13}$$

where $\vec{X} = \{\mathbf{X}^\omega | \omega \in \Omega\}$, $\mathbf{X}^\omega \in \Lambda^\omega$. $\vec{\nu} = (\nu_1, \ldots, \nu_K)$ and $\vec{\gamma} = (\gamma_{ik} | i \in \Gamma_k, k = 1, \ldots, K)$ are the two sets of the Lagrangian multipliers. Note that $\vec{\nu} \geq 0$ and $\vec{\gamma} \geq 0$. The dual problem of (9)-(12) is defined as

$$\max_{\vec{\nu}\geq 0, \vec{\gamma}\geq 0} \min_{\vec{X}} L(\vec{\nu},\vec{\gamma},\vec{X}). \tag{14}$$

To solve the dual problem, we first consider (13). For given lagrangian multipliers $\vec{\nu}$ and $\vec{\gamma}$, the problem is separable for different system states. Thus, we can solve the following problem for a given state ω,

$$\begin{aligned} \min_{\mathbf{X}^\omega} &E[g^\omega(\mathbf{X}^\omega)] - \sum_{k=1}^{K} \nu_k \left(\sum_{i\in\Gamma_k} B^\omega_{i'ik} - \lambda_k \right) \\ &- \sum_{k=1}^{K} \sum_{i\in\Gamma_k} \gamma_{ik} \left(\mu_{ik}\hat{S^\omega_{ik}} - B^\omega_{i'ik} \right) \\ &\text{s.t.} \quad \mathbf{X}^\omega \in \Lambda^\omega. \end{aligned} \tag{15}$$

The dual problem (14) can be solved using a stochastic subgradient algorithm, which updates Lagrangian multipliers iteratively. Take $\vec{\nu}$ as an example, it is updated by

$$\nu_k^{n+1} = \left[\nu_k^n + \frac{1}{n} \left(-\sum_{i\in\Gamma_k} B^{\omega^n\,*}_{i'ik} + \lambda_k \right) \right]^+, \tag{16}$$

where n denote the nth iteration, i.e., nth time slots in our case, $-\sum_{i\in\Gamma_k} B^{\omega^n\,*}_{i'ik} + \lambda_k$ is a stochastic subgradient w.r.t. ν_k^n, where $B^{\omega^n\,*}_{i'ik}$ is in the optimal solution to (15). Similarly, γ_{ik} can be updated as $\gamma_{ik}^{n+1} = [\gamma_{ik}^n + \frac{1}{n}(-\mu_{ik}S^{\omega^n\,*}_{ik} + B^{\omega^n\,*}_{i'ik})]^+$. It can be proven that the Lagrangian multiplier ν_k^n converges to the optimal solution of the dual problem (14) by the update in (16). Since the original problem (9)-(12) is convex, there is no duality gap. We omit the proof here since this scheme mainly serves as a benchmark and is less focused.

We name the above proposed scheme as joint resource provisioning with stochastic subgradient-based trough filling, abbreviated as StoS. StoS converges to the optimal solution of problem (9)-(12) in a stationary ergodic setting. Thus it can achieve the optimal costs with queue (both o-queues and s-queues) stability assured in a stationary ergodic setting. StoS can work in non stationary ergodic systems. Lagrangian multiplier $\vec{\nu}$ and $\vec{\gamma}$ has practical properties. Both of them can be considered as prices, which increase as service rate being smaller than the average arrival rate, i.e., bandwidth/capacity under-provisioning, and decrease vice-versa. Moreover, one can tune the average service rate for TOLs, i.e., by adjusting λ^k in (10), to control the TOL queue delay.

All the above benchmark schemes, i.e., Benchmark I and II, and StoS do not leverage TOL queue information, which may not have desirable TOL queue delay performance. We next propose OrgQ, which leverages TOL queue information in resource provisioning.

5 TOL Queue-Based Schemes

In this section, we propose schemes that leverage TOL queue backlog information to design trough-filling. The key intuition is that when a class of TOLs has a large queue length, more IDC capacity is allocated to it to reduce the queue length. Nevertheless, TOL load shifting costs and energy costs are considered as tradeoffs.

We first consider a scheme which leverages both the o-queue's and each s-queue's length information of each class of SENs to perform trough-filling, named joint resource provisioning with both the o-queue and s-queue based trough filling, abbreviated as SubQ. SubQ is based on the back pressure routing algorithm, which is often used in network resource allocation where queue stability needs to be assured. The intuition of SubQ is to shift more traffic of a class of TOLs to an IDC if the difference between the length of the o-queue and that of the s-queue of this class of TOLs in the IDC is large. In SubQ, capacity allocation is based on current s-queue length of each class of TOLs, which is decoupled from TOL load shifting. SubQ can be formulated by the following optimization problems.

I. *Bandwidth allocation for TOLs:*

$$\begin{aligned}\min_{B^t_{i'ik}} \; & \sum_{k=1}^{K}\sum_{i\in\Gamma_k}[-Q_k(t)+Q_{ik}(t)]B^t_{i'ik}T_p \\ & + V\sum_{i'=1}^{N}\sum_{i=1,i\neq i'}^{N}\max_{1\le\vartheta\le\theta}\left(a^{\vartheta}_{ii'}\frac{\sum_{k\in\Upsilon_{i'i}}B^t_{i'ik}}{B_{i'i}}+b^{\vartheta}_{ii'}\right) \\ & \sum_{k\in\Upsilon_{i'i}}B^t_{i'ik}\le B_{i'i},\ k=1,\ldots,K, i=1,\ldots,N.\end{aligned} \tag{17}$$

II. *Capacity allocation for SENs and TOLs:*

$$\begin{aligned}\min_{\mathbf{X}^t} \; & -\sum_{k=1}^{K}\sum_{i\in\Gamma_k}Q_{ik}(t)\mu_{ik}\hat{S^t_{ik}}T_p \\ & + V\sum_{i=1}^{N}\alpha^t_i\left[(1-\upsilon)C^t_i+\upsilon\sum_{j\in\Pi^h_i}r^t_{ij}\lambda^t_{ij}+\upsilon\sum_{k\in\Pi^l_i}\hat{S^t_{ik}}\right] \\ & \text{s.t.}-C^t_i+\sum_{j\in\Pi^h_i}r^t_{ij}\lambda^t_{ij}+f(\delta_i)\sqrt{\sum_{j\in\Pi^h_i}{r^t_{ij}}^2{\sigma^t_{ij}}^2}\le 0 \\ & \sum_{j\in\Pi^h_i}r^t_{ij}\lambda^t_{ij}+\sum_{k\in\Pi^l_i}\hat{S^t_{ik}}\le C^t_i, \\ & \sum_{i\in\Gamma_j}r^t_{ij}=1,\ C^t_i\le C^m_i, \\ & i=1,\ldots,N,\ j=1,\ldots,J,\ k=1,\ldots,K,\end{aligned} \tag{18}$$

Equation (17) is a back pressure routing problem with routing costs. Clearly, (17) is a convex optimization problem. In the algorithm, the difference between the o-queue's length and a s-queue's length of a class of TOLs is taken as a weight. In case of a larger weight, more traffic is shifted from the o-queue to the s-queue. V is a control parameter to tune the tradeoff between TOL load shifting costs and o-queue length (i.e., queue delay). A large V leads to less TOL traffic from o-queue shifted. In (18), $\mathbf{X}^t$ does not include $B^t_{i'ik}$, $k=1,\ldots,K$ and $i=1,\ldots,N$. V is also a control parameter to tune the tradeoff between energy costs and s-queue length. Equation (18) is also a convex optimization problem. SubQ has the same control variable set as Benchmark I. We can use Benchmark I to establish the costs and queue delay bounds of SubQ (Please refer to the supplemental material, available in the online supplemental material.). SubQ may not be cost-effective in distributed IDC environments. This is mainly because TOL load shifting in SubQ is decoupled from capacity allocation. Thus, location diversity of electricity prices is not leveraged well in SubQ.

To overcome the disadvantage of SubQ, we further design a scheme that leverages the o-queue backlog information of each class of TOLs to design trough-filling. The intuition of the scheme is to make TOL load shifting coupled with IDC capacity allocation, and IDC capacity allocation is based on o-queue length of each class of TOLs. When o-queue length of a class of TOLs is large, more capacity may be assigned at each IDc that can serve this class of TOLs.

In each time slot t, observe current queue backlog $Q_k(t)$, $k=1,\ldots,K$, α^t_i, λ^t_j, and σ^t_j, $i=1,\ldots,N$, and $j=1,\ldots,J$. Perform the following optimization scheme, named joint resource provisioning with o-queue-based trough filling, abbreviated as OrgQ.

I. *Bandwidth allocation for TOLs:*
Set $B^t_{i'ik}=\mu_{ik}\hat{S^t_{ik}}$, $k=1,\ldots,K$.

II. *Capacity allocation:*

$$\begin{aligned}\min_{\mathbf{X}^t} \; & -\sum_{k=1}^{K}Q_k(t)\sum_{i\in\Gamma_k}\mu_{ik}\hat{S^t_{ik}}T_p \\ & + V\sum_{i=1}^{N}\alpha^t_i\left[(1-\upsilon)C^t_i+\upsilon\sum_{j\in\Pi^h_i}r^t_{ij}\lambda^t_{ij}+\upsilon\sum_{k\in\Pi^l_i}\hat{S^t_{ik}}\right] \\ & + V\sum_{i'=1}^{N}\sum_{i=1,i\neq i'}^{N}\max_{1\le\vartheta\le\theta}\left(a^{\vartheta}_{ii'}\frac{\sum_{k\in\Upsilon_{i'i}}\mu_{ik}\hat{S^t_{ik}}}{B_{i'i}}+b^{\vartheta}_{ii'}\right) \\ & \text{s.t.}-C^t_i+\sum_{j\in\Pi^h_i}r^t_{ij}\lambda^t_{ij}+f(\delta_i)\sqrt{\sum_{j\in\Pi^h_i}{r^t_{ij}}^2{\sigma^t_{ij}}^2}\le 0 \\ & \sum_{j\in\Pi^h_i}r^t_{ij}\lambda^t_{ij}+\sum_{k\in\Pi^l_i}\hat{S^t_{ik}}\le C^t_i \\ & \sum_{i\in\Gamma_j}r^t_{ij}=1,\ \sum_{k\in\Upsilon_{i'i}}\mu_{ik}\hat{S^t_{ik}}\le B_{i'i},\ C^t_i\le C^m_i, \\ & i',i=1,\ldots,N,\ j=1,\ldots,J,\ k=1,\ldots,K.\end{aligned} \tag{19}$$

Equation (19) is a convex optimization problem. Thus at the beginning of each slot, $\mathbf{X}^t$ can be determined efficiently.

The intuition of OrgQ is clear. When queue length $\sum_{k=1}^{K}Q_k(t)$ is large, OrgQ has incentives to allocate a larger capacity to TOLs to reduce the o-queue length. When the costs are relatively large or queue length is small, OrgQ allocates a smaller capacity to TOLs to reduce the costs. Similar

to SubQ, the parameter V balances the TOL queue length and the costs. If V is large, OrgQ allocates a smaller capacity, and vice versa.

In OrgQ, TOL bandwidth allocation is closely coupled with capacity allocation. Thus OrgQ is expected to be more cost-effective than SubQ. In addition, capacity allocation in SubQ is based on s-queues, while capacity allocation in OrgQ is based on o-queues. The total number of o-queues is much smaller than that of s-queues. Thus, a higher statistical multiplexing gain can be achieved by OrgQ. OrgQ may also have a smaller overall queue delay than SubQ. This is because the s-queue delay by OrgQ is negligible, while the s-queue delay by SubQ may be relatively larger due to the competition among multiple s-queues. We will numerically compare the performance among StoS, OrgQ, and SubQ, and demonstrate the properties and insights behind each scheme.

We analyze the performance of OrgQ, including both the cost and queue delay performance. We use Benchmark II to establish the performance of OrgQ. Let $\mathbf{X}_{em}^t$ and g_{em}^t denote the decision variable vector and cost in time slot t of Benchmark II, respectively. Clearly, $\mathbf{X}_{em}^t$ takes values from the same feasibility set Λ^t, as in OrgQ. The time average value of g_{em}^t, denoted by g_{em}^*, is no less than g_e^*, because TOL load shifting in Benchmark II is sub-cost-optimal. We use $g_q^t(\mathbf{X}^t)$ to denote the cost by OrgQ in time slot t. Introduce a new parameter μ_i, $i = 1, \ldots, N$, which is equal to $\max\{\mu_{ik} | k \in \Pi_i^l\}$, i.e., maximum unit service rate for TOLs in IDC i. We have the following proposition.

Proposition 1. *In a stationary ergodic system, OrgQ stabilizes the o-queues for a given parameter V. In addition, an upper bound on average o-queue length is*

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K}E[Q_k(t)] \leq \frac{(\sum_{i=1}^{N}\mu_i C_i^m T_p)^2 + \sum_{k=1}^{K} D_k^{m2} + 2Vg_{em}^*(\epsilon)}{2\epsilon T_p}. \quad (20)$$

Further, the average costs achieved by OrgQ is upper bounded as

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}E[g_q^t(\mathbf{X}^t)] \leq g_{em}^* + \frac{(\sum_{i=1}^{N}\mu_i C_i^m T_p)^2 + \sum_{k=1}^{K} D_k^{m2}}{2V}, \quad (21)$$

where $g_{em}^*(\epsilon)$ is the average costs of Benchmark II with TOL traffic arrival rate $\vec{\lambda}^l + \mathbf{1}\epsilon$, and $\epsilon > 0$.

Proof. Please refer to the supplemental material, available in the online supplemental material. □

We have established the bound on the sum length of o-queues of OrgQ. We next show OrgQ stabilizes each s-queue.

Corollary 1. *In a stationary ergodic system (characterized by electricity prices and traffic statistics of SENs), OrgQ stabilizes each s-queue.*

Proof. See the supplemental material, available in the online supplemental material. □

It can be envisioned that both Benchmark II and OrgQ have a small s-queue delay, because in each time slot, each s-queue's traffic arrival rate is no larger than the expected service rate. When there is no service rate randomness, the service delay of a job in a s-queue is no larger than 1.

6 Implementation Issues and Caveats

In our schemes, as discussed, the decision-maker needs gathering input information in the beginning of each time slot. The messaging delay is about tens of milliseconds, which is similar to [22]. Each IDC sends only a few parameters to the decision-maker. Since each time slot can be tens of minutes, the messaging overhead is negligible. Moreover, the decision overhead is also negligible since the convex optimization problems can be solved efficiently. The executions of our schemes involve server provisioning, SEN load dispatching, TOL load shifting, and SEN/TOL capacity allocation. Server provisioning is performed in a large time scale, i.e., every tens of minutes. Thus, the overhead in turning on/off servers is inconsiderable. SEN load dispatching is performed in a small time scale based on instantaneous traffic. The destination IDCs and load dispatching ratios are fixed during each time slot. A front-end portal just follows the ratios to shift SEN traffic to destination IDCs. Cross-IDC TOL load shifting is practical and widely used in existing schemes [2], [21], [22], [23], [24]. Thus, our schemes do not require additional change of hardware or software in IDCs.

Similar to a lot of research in the literature, we do not consider the effect of the possible virtualization. When one runs several applications on the same server, the performance versus power curve becomes more difficult to quantify [22]. This is a good future research topic.

7 Performance Evaluation

In this section, we evaluate the performance of proposed SubQ and OrgQ by simulations based on Matlab with CVX, a Matlab-based convex optimization tool [38], and PERL. We first focus on the total costs and TOL delay performance. We later use real traffic trace to study the performance of SENs.

7.1 Total Costs and Delay Performance of TOLs

7.1.1 Simulation Setup

We consider $N = 5$ IDCs in different locations. There are $J = 10$ classes of SENs. There are $K = 10$ classes of TOLs randomly originated in one of the five IDCs. The sets of IDCs that can serve each class of SEN j and TOL k, i.e., Γ_j and Γ_k, are chosen randomly, respectively. We choose the length of each time slot, T_p, to be 10 minutes, and length of each sub-slot, T_s, to be 0.1 second. In each time slot, we first generate traffic of each class of SEN j with a Gaussian distribution. In each time slot, the mean and standard deviation of traffic of SEN j randomly take values from 100 to 200, and from 50 to 150, respectively. The traffic arrival rate of each class of TOL k randomly takes value from 0 to 2,000. Thus, the average traffic arrival rate of each class of TOL is 1,000. For simplicity, we set μ_{ij}, i.e., service rate offered by IDC i to SEN j as 1. Further, μ_{ik}, i.e., service rate received by TOL k at IDC i, randomly takes value from 5 to 10. Thus, the load demand of SENs and TOLs are comparable,

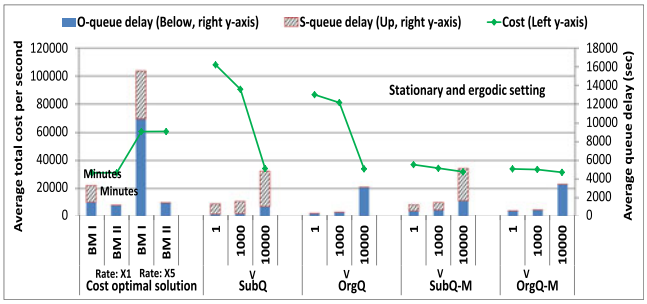


Fig. 1. Comparing costs and queue delay among Benchmark I and II (cost-optimal benchmarks), SubQ and OrgQ in a stationary and ergodic setting.

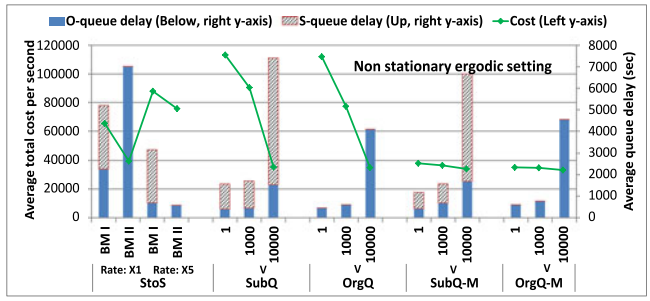


Fig. 2. Comparing costs and queue delay among StoS, SubQ and OrgQ in a non stationary ergodic setting.

roughly 53 versus 47 percent. We set the maximum number of servers at each IDC as 2,000. Each server has a maximum (normalized) speed as 1. Each IDC requires an overloading probability constraint δ of 0.05. Further, the bandwidth constraint between two IDCs, i.e., $B_{ii'}$, randomly takes value from 500 to 1,500. Note that when $i' = i$, there is no load shifting constraint. In our simulation, we choose a large value as $2 * 10^6$. Idle power consumption of each server ν is set as 0.6. We let electricity price α_i of each IDC i randomly take value from 5 to 15 in each time slot. Load shifting costs between two IDCs for TOLs follow the proposed piece-wise linear cost model. Two segments are considered with the link bandwidth utilization by TOLs of 0.5 as the point of inflection. Further, $(a^1_{i'i}, b^1_{i'i})$ and $(a^2_{i'i}, b^2_{i'i})$ are set as (500, 500) and (1,000, 250), respectively. We first consider a stationary ergodic setting where there exists the minimum total costs. In the stationary ergodic setting, we assume there are in total 50 different system states. Note that one system state is characterized by a pair of $(\lambda_j, \sigma_j, \alpha_i | j = 1, \ldots, J, i = 1, \ldots, N)$. We then consider a non stationary ergodic setting, where in each time slot, $(\lambda_j, \sigma_j, \alpha_i | j = 1, \ldots, J, i = 1, \ldots, N)$ randomly takes values as specified above. There is no limit on the unique pairs of $(\lambda_j, \sigma_j, \alpha_i | j = 1, \ldots, J, i = 1, \ldots, N)$.

7.1.2 Simulation Results

We compare performance of OrgQ, and SubQ to BM I and BM II (two different benchmarks on cost-optimal solutions) in Fig. 1, where we omit the performance of StoS since it converges to BM I or BM II in a stationary ergodic setting. For BM I and BM II, we consider different allocated bandwidth and service rates for TOLs, given the same TOL traffic arrival rate. That is, with TOL traffic arrival rate $\vec{\lambda^l} = (\lambda_k | k = 1, \ldots, K)$, we replace λ_k in (10) by $1 * \lambda_k$, and $5 * \lambda_k$, respectively, which offer service rates that are equal to, and five times larger than a TOL queue arrival rate, respectively. Thus we can obtain different results on costs and delay performance by BM I and BM II. Correspondingly, we also modify OrgQ and SubQ, as specified later.

First, let's examine the case of $1 * \vec{\lambda^l}$, i.e., marked by 'Rate: X1' in Fig. 1. It is observed that the queue delay of both BM I and BM II is large (In this case we use the unit of minutes.). BM I has a slightly larger o-queue delay and a much larger s-queue delay than BM II. This is obvious since in BM II, the traffic arrival rate of each s-queue in each time slot is no more than the service rate. The service delay is thus no more than 1 second. In our simulation, we observe that the average s-queue delay of BM II is negligible compared to the o-queue delay. Note that the average s-queue delay of BM II can be larger than 1,s in some cases. This is due to service rate randomness in each time slot. When we increase service rate to $5 * \vec{\lambda^l}$, we observe that queue delay for both BM I and BM II is much smaller, but with a much larger cost. BM II outperforms BM I since it can achieve a similar cost with a obviously smaller queue-delay.

Performance of OrgQ and SubQ is also reported in Fig. 1. For both OrgQ and SubQ, we vary V from 1 to 10,000. When V increases, costs of OrgQ and SubQ decrease, while both the average o-queue delay and average s-queue delay increase. We observe that with the same V, costs of OrgQ are slightly smaller than that of SubQ. The average o-queue delay of OrgQ is larger than that of SubQ, while the average s-queue delay of OrgQ is negligible. The average s-queue delay of SubQ is relatively large, compared to its o-queue delay. When V is large, we observe that costs of SubQ and OrgQ are both close to those of BM I and BM II (with a service rate of $1 * \vec{\lambda^l}$), while the queue delay of both the two schemes are still significantly smaller than that by both BM I and BM II (with service rate of $1 * \vec{\lambda^l}$).

We observe that when V is small, i.e., $V = 1$, although queue delay of both OrgQ and SubQ is small, costs are large, which is not desirable for saving costs. When V is large, queue delay increases considerably for both of them. To improve the tradeoff between costs and delay, we modify OrgQ and SubQ by introducing the following constraints. For OrgQ, in the optimization problem (19), we add the constraint $\sum_{i \in \Gamma_k} \mu_{ik} \hat{S^t_{ik}} T_p \leq Q_k(t), k = 1, \ldots, K$. For SubQ, in the optimization problem (17) and (18), we add the constraints $\sum_{i \in \Gamma_k} B^t_{i'ik} T_p \leq Q_k(t), k = 1, \ldots, K$, and $\mu_{ik} \hat{S^t_{ik}} T_p \leq Q_{ik}(t)$, respectively. Obviously, those constraints make bandwidth or capacity allocation more coupled with the current queue length. In other words, it avoids the case that TOLs receive a large capacity even when their queue length is small. We also study the performance of the modified SubQ and OrgQ in Fig. 1. We observe that costs of the modified OrgQ and SubQ are close to BM I and BM II (with a service rate of $1 * \vec{\lambda^l}$) with different V, which indicates that the modified queue-based schemes are both cost-effective. We further observe that the modified OrgQ has a slightly larger queue delay (represented by o-queue delay) than the original OrgQ. The delay of O-queue of the modified SubQ is also larger than that of the original scheme. But the s-queue's delay gets slightly smaller. This is because the

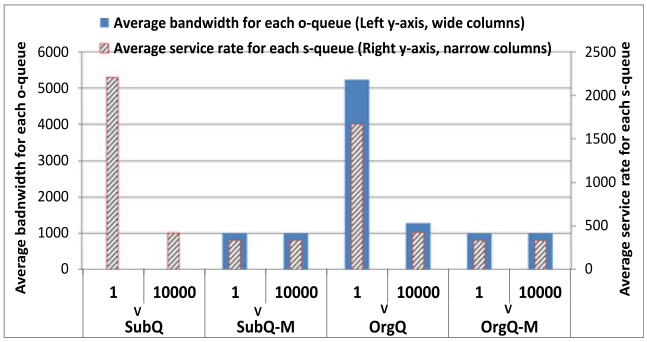


Fig. 3. Average service rate of original queues and sub-queues by SubQ and OrgQ.

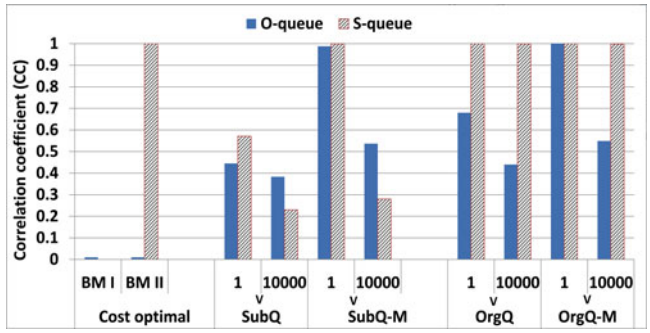


Fig. 4. Correlation coefficient between queue length and service rate.

traffic arrival rate of each s-queue in the modified SubQ is smaller (since the service rate of each o-queue is smaller in the modified SubQ). The overall queue delay gets slightly larger by the modified SubQ. Comparing the modified SubQ and OrgQ to BM I and BM II with a service rate of $5 * \vec{\lambda^l}$, we observe that costs of the modified SubQ and OrgQ are much lower, and the o-queue's delay and the s-queue's delay are also much lower when V is small, i.e., 1.

We also compare the performance among StoS, SubQ, and OrgQ in a non stationary ergodic setting in Fig. 2. The simulation setting is described in "Simulation setup". We observe that the costs and delay trend under different V are similar to that in Fig. 1, although costs of each scheme is higher. We observe that StoS can have a small TOL queue delay with a service rate of $5 * \vec{\lambda^l}$, especially for BM II. In this case, costs of StoS are higher than that of the cost-optimal solution presented by Fig. 1. Compared to StoS, the modified SubQ and OrgQ, especially OrgQ, can achieve a smaller or similar TOL queue delay with a much smaller cost. Fig. 2 shows the modified OrgQ is efficient in saving costs and reducing TOL queue delay with a proper V in a non stationary ergodic setting.

We next explain the insights behind the observations. We mainly examine two aspects: one is average service rate assigned, which discloses how much resource is allocated; the other is the correlation coefficient (CC) between the queue length and the service rate, which discloses how efficiently the resource is allocated. Without loss of generality, we consider the same stationary ergodic setting as in Fig. 1.

7.1.3 Explaining the Observations

We first examine the average service rate for both the o-queues and the s-queues of each scheme by Fig. 3. The average service rate of the o-queue of TOL k in a time slot t is the sum of bandwidth allocated, i.e., $\sum_{i \in \Gamma_k} B^t_{i'ik}$, while the average service rate of a s-queue of TOL k by IDC i is $\mu_{ik} S^t_{ik}$. In Fig. 3, we further average the service rates over all time slots simulated and all classes of TOLs. First, the average service rate for each o-queue by BM I or BM II is the same as the average TOL traffic rate, so does the sum of service rate of all s-queues of TOL k. Thus, we omit them in Fig. 3 and only consider OrgQ and SubQ. We observe that SubQ has a large average service rate for each o-queue, which is because that a large bandwidth is assigned to IDC $i^{'}$, i.e., the IDC that a class of TOL k originates, and theoretically there is no constraint for it (in our simulation, we set the constraint as $2 * 10^6$). This explains why the o-queue delay of SubQ is smaller than OrgQ. When $V = 1$, for both SubQ and OrgQ, the average service rates of both the o-queue and s-queue are large. For example, the average service rate of OrgQ is five times larger than the average TOL traffic arrival rate, i.e., 1,000. This is why when V is small, costs of SubQ and OrgQ are large. The modified SubQ and OrgQ always have an average service rate of the o-queues that is close to the average TOL arrival rate. The average s-queue service rate is always around 330. Note that in our simulations, each class of TOLs has roughly three sub-queues on average. Thus, the sum of service rate over all sub-queues for a class of TOL is around 1,000. This result explains why the modified SubQ and OrgQ always have small costs, i.e., the modified queue-based schemes only allocate resource that can rightly guarantee TOL queue stability.

Queue delay not only depends on the average service rate, but also the efficiency of the service rate allocation. We use a metric named correlation coefficient to evaluate service rate allocation efficiency. Consider two series of n elements of X and Y as x_i and y_i where $i = 1, \ldots, n$. CC between X and Y is calculated by

$$\frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}.$$

CC can take a value from -1 to 1, where 0 indicates there is no relation between X and Y, and a negative (positive) value indicates there exists decreasing (increasing) relationship between them. To calculate CC between the the o-queue of TOL k and its service rate, we can replace x_i and y_i by $Q_k(t)$ and $\sum_{i \in \Gamma_k} B^t_{i'ik}$, respectively. Correspondingly, we can use $Q_{ik}(t)$ and $\mu_{ik}\hat{S^t_{ik}}$ for the s-queue of TOL k at IDC i. A larger CC indicates that the service rate assignment is more efficient, which results in a smaller queue delay.

From Fig. 4, we observe that the cost-optimal solution has a CC of 0.01 for the o-queue in both BM I and II. For the s-queue, CC by BM I is almost 0. While for BM II, CC is close to 1, which explains why BM I have both a large o-queue delay and a s-queue delay, and BM II have a large o-queue delay. CC of the s-queue by BM II is 1 because the current queue length is always the same as or proportional to the assigned service rate. For SubQ, when $V = 1$, CC for the o-queue is smaller than 0.5, while the average CC of the s-queues is larger than 0.5. This is because in SubQ, service rate for the o-queue also depends on the

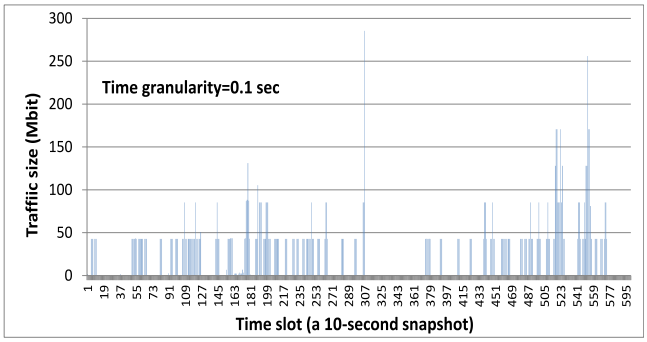


Fig. 5. Busty traffic pattern of real traffic trace used in simulation.

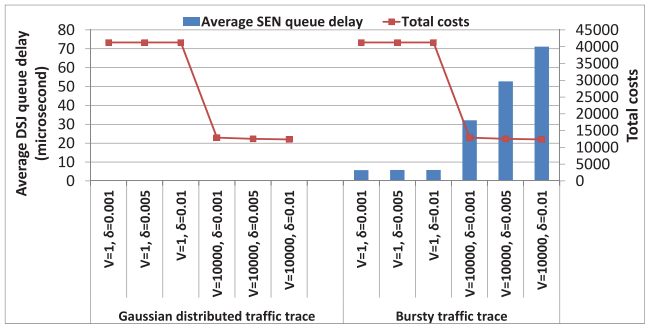


Fig. 6. Average SEN queue delay and total costs with Gaussian and bursty traffic trace, respectively.

s-queue length. However, when $V = 10{,}000$, CC of the s-queues is smaller than that of the o-queues. This is because, the average service rate in this case for each s-queue is small, as shown in Fig. 3. Note that the rate assignment for each sub-queue of SubQ also follows a threshold-based policy. We observe that each sub-queue receives a non-zero service rate with a large time interval. In a consequence, CC of each s-queue becomes smaller. OrgQ has a larger average CC than SubQ of both the o-queues and the s-queues. Moreover, CC of the s-queues by OrgQ is almost 1, which is the same as by BM II. We also observe that the modified SubQ and the modified OrgQ have a much larger CC than SubQ and OrgQ, respectively. This is because in the modified schemes, service rate is constrained by the current queue length. Thus, service rate is more correlated with the queue length. This result also implies that the modified queue-based schemes are more efficient in resource provisioning.

By Fig. 1, we also observe that costs of SubQ is slightly larger than OrgQ. This is because, first, TOL load shifting in SubQ is decoupled from IDC capacity allocation. Thus, electricity price diversity is not leveraged by TOL load shifting to reduce total IDC costs. Second, capacity allocation for SubQ is based on s-queues, while OrgQ is based on o-queues. Since the number of s-queues is much (roughly three times) larger than the number of o-queues. OrgQ can achieve a higher statistical multiplexing gain than SubQ, which leads to a smaller energy cost. Note that this is also the reason that the overall queue delay of OrgQ is smaller than that of SubQ. That is, SubQ has a relatively large s-queue delay because less statistical multiplexing gain is achieved in capacity allocation.

In summary, OrgQ, especially the modified version, can achieve the best tradeoff between costs and delay performance. With V is properly tuned, the modified OrgQ can achieve a cost that is close to the cost optimal solution, with a much smaller o-queue delay and negligible s-queue delay.

7.2 Real Bursty Trace Based Simulations on SENs

In this subsection, we study performance of SENs. We use a real datacenter traffic trace, which is from a commercial datacenter operated by a large cloud service provider in the U.S. We have 15-day's log of its Hadoop distributed file system (HDFS). The HDFS log records the information of the received packets, including the packet size and time-stamp, in a time granularity of 1 millisecond. The data does not differentiate SENs and TOLs (In fact, to differentiate them without application-layer information is itself a challenging issue.). We consider the traffic trace as of SENs. We consider the received packet size as traffic, instead of the number of received packets. We observe the packet-size based traffic exhibits a obvious bursty pattern, i.e., repeated cycles of a consecutive time of low traffic volume followed by a period of large traffic volume, as illustrated by Fig. 5. In practice, bursty SEN traffic is difficult to provision. Thus the simulation results are conservative.

To simulate multiple IDCs and multiple classes of SENs, we treat each day's data as one class of SENs. We consider $T_s = 30$ min, and T_p of 0.1 sec. We repeat each day of data to obtain 5,000 time slots. In each time slot, we assume that traffic arrival rate of each class of TOLs randomly takes a value between 0 and 200, which leads to a load that is close to the load demand of SENs. Total capacity of each IDC is set as 600. The other simulation setting is the same as in Fig. 1. To make a comparison, we also consider Gaussian distributed traffic trace, which takes the same mean and variance in each time slot as the real traffic case. We consider the proposed OrgQ scheme in the simulations.

We evaluate SEN queue delay in Fig. 6. We assume that the service delay of a SEN is 100 μs. If there is no overloading at an IDC i, a SEN arriving the IDC i is served immediately with a delay of 100 μs. When overloading occurs in a sub-slot, the excessive SENs stay in a queue and will be served in the following sub-slots. Thus, queue delay is incurred and capacity of future sub-slots will be used. We study the queue delay of SENs under different overloading probabilities, and different values of V. We first plot the average SEN queue delay and total costs, with real bursty traffic trace and Gaussian distributed traffic trace. For the bursty traffic trace, we observe that when $V = 1$, SEN queue delay is much lower than that of the case of V = 10,000, with a much larger total cost. This is because a smaller V results in a larger total capacity. In this case, SEN queue delay is similar with different overloading probability constraints, because capacity demand by TOLs is more than that of SENs. When $V = 1$, average SEN queue delay is only about 5 μ, which is much smaller than the service latency, i.e., 100 μs. When $V = 10{,}000$, different overloading probability constraints of δ lead to different average SEN queue delay. When $\delta = 0.01$, average SEN queue delay is 70 μs, which is still desirable compared to service latency. For Gaussian distributed traffic, we observe that average SEN queue delay is negligible when the overloading probability is equal to 0.01, 0.005, or 0.001, with both $V = 1$ and

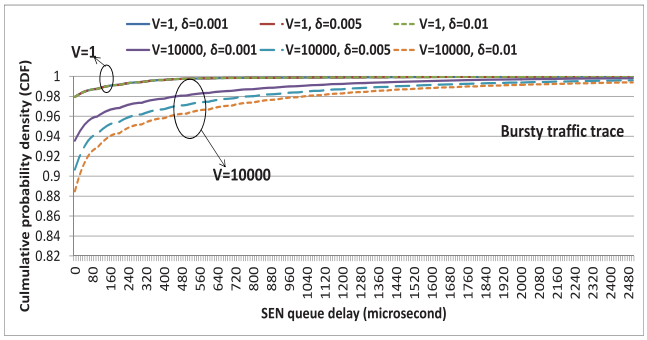


Fig. 7. Distribution of SEN queue delay with bursty traffic trace.

$V = 10{,}000$. Thus, a smooth SEN traffic will result in a much better delay performance than a bursty traffic trace. We also plot the distribution (cumulative probability density function) of SEN queue delay in Figs. 7 and 8 of bursty traffic trace and Gaussian distributed traffic trace, respectively. It is observed that SEN queue delay distribution of bursty traffic has a longer tail than that of the Gaussian distributed traffic. With bursty traffic trace, it is observed that when $V = 1$, SEN queue delay is 0 with a probability of about 0.98, while such a probability is almost 1 with Gaussian distributed traffic.

In summary, bursty traffic leads to a higher SEN queue delay than a smooth traffic trace like Gaussian distributed trace. With bursty traffic, a smaller overloading probability is needed to achieve the same SEN queue delay as with a smooth traffic. Note that in our simulations, we consider received packet size as traffic. A real traffic trace based on number of requests would be much smoother. Therefore, our bursty traffic trace based simulation is conservative.

8 Conclusions

We study joint resource provisioning schemes for SENs and TOLs for distributed IDCs. We consider traffic dynamics of both SENs and TOLs with different time scales. Resource provisioning is performed by joint server provisioning, SEN load dispatching, TOL load shifting, and capacity allocation at different time granularities. In a large time scale, server provisioning is performed jointly with configuring load dispatching/shifting and capacity allocation. In a small time scale, instantaneous load dispatching/shifting is performed. Meanwhile, TOLs are provisioned based on the remaining capacity of each IDC when capacity for SENs is guaranteed. We design different schemes that require different system information. StoS can achieve the smallest cost. But the queue delay is large since it does not leverage any TOL queue information. OrgQ and SubQ can achieve a much smaller queue delay with slightly larger costs. OrgQ outperforms SubQ in cost mainly because its load shifting is closely coupled with capacity allocation, which leverages electricity price diversity. Further, OrgQ has a smaller delay since its s-queue delay is negligible, while SubQ has a relatively large s-queue delay. In conclusion, OrgQ can achieve a good tradeoff between queue delay and costs.

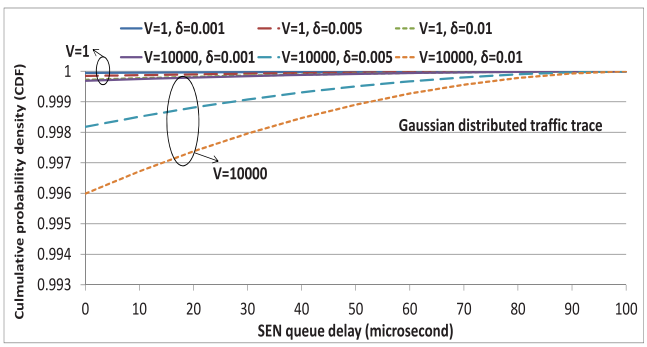


Fig. 8. Distribution of SEN queue delay with Gaussian traffic trace.

Acknowledgments

The authors thank the editors and all the reviewers for their insightful comments and suggestions, which significantly improved the quality of the paper. They also thank Dr. Ernest Tsui from AT&T Labs for his proofreading. Dan Xu and Xin Liu were partially supported by UC Davis Chancellor's fellowship, and US National Science Foundation (NSF) grants 1423542 and 1147930. Zhisheng Niu is sponsored in part by the National Basic Research Program of China (973GREEN: 2012CB316001). The work was partially performed when the first two authors visited Tsinghua National Lab for Information Science & Technology, Beijing. Dan Xu was with the Department of Computer Science, University of California, Davis, when most of the work was performed.

References

[1] Y. Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, "Managing server energy and operational costs in hosting centers," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2005, pp. 303–314.

[2] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[3] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao, "Energy-aware server provisioning and load dispatching for connection-intensive internet services," in *Proc. 5th Usenix Symp. Netw. Syst. Des. Implementation*, 2008, pp. 337–350.

[4] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," in *Proc. IEEE INFOCOM*, 2011, pp. 1098–1106.

[5] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. H. Andrew, "Greening geographical load balancing," in *Proc. ACM SIGMETRICS Joint Int. Conf. Measurement Model. Comput. Syst.*, 2011, pp. 233–244.

[6] T. Lu, M. Chen, and L. L. H. Andrew, "Simple and effective dynamic provisioning for power-proportional data centers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no.06, pp. 1161–1171, Jun. 2013.

[7] J. Tu, L. Lu, M. Chen, and R. K. Sitaraman, "Dynamic provisioning in next-generation data centers with on-site power production," in *Proc. ACM 4th Int. Conf. Future Energy Syst.*, 2013, pp. 137–148.

[8] H. Qian and D. Medhi, "Server operational cost optimization for cloud computing service providers over a time horizon," in *Proc. 11th USENIX Conf. Hot Topics Manage. Internet, Cloud, Enterprise Netw. Services*, 2011, p. 4.

[9] B. Guenter, N. Jain, and C. Williams, "Managing cost, performance, and reliability tradeoffs for energy-aware server provisioning," in *Proc. IEEE INFOCOM*, 2011, pp. 1332–1340.

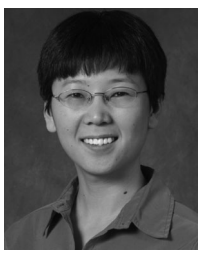
[10] D. Xu, X. Liu, and B. Fan, "Minimizing energy cost for internet-scale datacenters with dynamic traffic," in *Proc. IEEE 19th Int. Workshop Quality Workshop*, 2011, pp. 1–2.

[11] D. Xu, X. Liu, and B. Fan, "Efficient server provisioning and offloading policies for internet datacenters with dynamic load demand," in *IEEE Trans. on Comput.*, vol. PP, no. 99, p. 1.

[12] D. Xu, "Green internet datacenters with dynamic and diverse traffic," Doctoral dissertation, Univ. California, Davis, CA, USA, 2011.

[13] D. Xu and X. Liu, "Geographic trough filling for internet datacenters," in *Proc. IEEE INFOCOM Mini-Conf.*, 2012, pp. 2881–2885.
[14] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. J. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in *Proc. IEEE INFOCOM*, 2012, pp. 1431–1439.
[15] M. Polverini, A. Cianfrani, S. Ren, and A. V. Vasilakos, "Thermal-aware scheduling of batch jobs in geographically distributed data centers," *IEEE Trans. Cloud Comput.*, vol. 2, no. 1, Jan–Mar. 2014.
[16] M. A. Rodriguez and R. Buyya, "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds," *IEEE Trans. Cloud Comput.*, vol. 2, no. 2, pp. 222–235, Apr.–Jun., 2014.
[17] R. Urgaonkar, U. C. Kozat, K. Igarashi, and M. J. Neely, "Dynamic resource allocation and power management in virtualized data centers," in *Proc. IEEE Netw. Oper. Manage. Symp.*, 2010, pp. 479–486.
[18] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, "Renewable and cooling aware workload management for sustainable data centers," in *Proc. ACM SIGMETRICS/PERFORMANCE Joint Int. Conf. Measurement Model. Comput. Syst.*, 2012, pp. 175–186.
[19] H. Xu, C. Feng, and B. Li, "Temperature aware workload management in geo-distributed datacenters," in *Proc. USENIX Int. Conf. Autonomic Comput.*, 2013, pp. 303–314.
[20] A. Beloglazov and R. Buyya, "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1366–1379, Jul. 2013.
[21] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *Proc. ACM SIGCOMM Conf. Data Commun.*, 2009, pp. 123–134.
[22] R. Stanojevic and R. Shorten, "Distributed dynamic speed scaling," in *Proc. IEEE INFOCOM*, 2010, pp. 426–430.
[23] N. Buchbinder, N. Jain, and I. Menache, "Online job-migration for reducing the electricity bill in the cloud," in *Proc. 10th Int. IFIP TC 6 Conf. Netw.*, 2011, pp. 172-185.
[24] M. Lin, Z. Liu, A. Wierman, and L. L. H. Andrew, "Online algorithms for geographical load balancing," in *Proc. Int. Green Comput. Conf.*, 2012, pp. 1–10.
[25] N. Bansal, K. Pruhs, and C. Steins, "Speed scaling for weighted flow times," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2007, pp. 805–813.
[26] A. Wierman, L. L. H. Andrew, and A. Tang, "Power-aware speed scaling in processor sharing systems," in *Proc. IEEE INFOCOM*, 2009, pp. 2007–2015.
[27] F. Xu, F. Liu, H. Jin, and A. V. Vasilakos, "Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions," *Proc. IEEE*, vol. 102, no. 1, pp. 11–31, Jan. 2014.
[28] L. Wang, F. Zhang, J. A. Aroca, A. V. Vasilakos, K. Zheng, C. Hou, D. Li, and Z. Liu, "GreenDCN: A general framework for achieving network energy efficiency in data centers," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 1, pp. 4-15, Jan. 2014.
[29] L. Wang, F. Zhang, A. V. Vasilakos, C. Hou, and Z. Liu, "Joint virtual machine assignment and traffic engineering for green data center networks," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 3, pp. 107-112, 2013.
[30] L. Wang, F. Zhang, K. Zheng, A. V. Vasilakos, S. Ren, and Z. Liu, "Energy-efficient flow scheduling and routing with hard deadlines in data center networks," in *Proc. IEEE 34th Int. Conf. Distrib. Comput. Syst.*, 2014, pp. 248–257.
[31] Z. Shao, X. Jin, W. Jiang, M. Chen, and M. Chiang, "Intra-data-center traffic engineering with ensemble routing," in *Proc. IEEE INFOCOM* 2013, pp. 2148–2156.
[32] H. Zhang, B. Li, H. Jiang, F. Liu, A. V. Vasilakos, and J. Liu, "A framework for truthful online auctions in cloud computing with heterogeneous user demands," in *Proc. IEEE INFOCOM*, 2013, pp. 1510–1518.
[33] L. Mashayekhy, M. Nejad, D. Grocu, and A. V. Vasilakos, "Incentive-compatible online mechanism for resource provisioning and allocation in cloud," in *Proc. IEEE 7th Int. Conf. Cloud Comput.*, 2014, pp. 3112–319.
[34] L. Lu and P. Varman, "Workload decomposition for power efficient storage systems," in *Proc. Conf. Power Aware Comput. Syst.*, 2008, p. 13.
[35] H. Xu and B. Li, "Cost efficient datacenter selection for cloud services," in *Proc. IEEE Int. Conf. Commun. China,*, 2012, pp. 51–56.
[36] D. Niu, C. Feng, and B. Li, "Pricing cloud bandwidth reservations under demand uncertainty," in *Proc. 12th ACM SIGMETRICS/PERFORMANCE Joint Int. Conf. Meas. Model. Comput. Syst.*, 2012, pp. 151–162.
[37] B. Fortz and M. Thorup, "Internet TE by optimizing OSPF weights," in *Proc. IEEE INFOCOM*, 2000, pp. 519–528.
[38] M. Grant and S. Boyd, CVX: Matlab software for disciplined convex programming. (2015, Jan.). Version 2.1 [Online]. Avaialble: http://cvxr.com/cvx/download/

Dan Xu received the PhD degree in computer science from the University of California, Davis, in 2011. Since then, he has been a senior member of technical staff at AT&T Labs, CA. His research interests are in the areas of datacenter network resource provisioning and energy efficiency design, and 3G/4G mobile networks performance modeling and analysis.



Xin Liu received the PhD degree in electrical engineering from Purdue University in 2002. She is currently a professor in the Computer Science Department at the University of California, Davis. Her research is on wireless communication networks. She received the Best Paper of Year Award from the Computer Networks Journal in 2003. She received the US National Science Foundation CAREER Award in 2005. She received the Outstanding Engineering Junior Faculty Award from the College of Engineering, UC Davis, in 2005. She has been a Chancellor's fellow since 2011. She is a member of the IEEE.

Zhisheng Niu graduated from Beijing Jiaotong University, China, in 1985, and received the ME and DE degrees from Toyohashi University of Technology, Japan, in 1989 and 1992, respectively. During 1992-1994, he was with Fujitsu Laboratories Ltd., Japan, and in 1994 joined with Tsinghua University, Beijing, China, where he is currently a professor at the Department of Electronic Engineering and the deputy dean of the School of Information Science and Technology. His major research interests include queueing theory, traffic engineering, mobile Internet, radio resource management of wireless networks, and green communication and networks. He has been an active volunteer for various academic societies, including the director for Conference Publications (2010-2011) and the director for Asia-Pacific Board (2008-2009) of IEEE Communication Society, Membership Development Coordinator (2009-2010) of IEEE Region 10, councilor of IEICE-Japan (2009-2011), and a council member of Chinese Institute of Electronics (2006-2011). He is currently a distinguished lecturer (2012-2015) and the chair of Emerging Technology Committee (2014-2015) of IEEE Communication Society, a distinguished lecturer (2014-2016) of IEEE Vehicular Technologies Society, a member of the Fellow Nomination Committee of IEICE Communication Society (2013-2014), a standing committee member of Chinese Institute of Communications (CIC, 2012-2016), and an associate editor-in-chief of IEEE/CIC joint publication China Communications. He received the Outstanding Young Researcher Award from Natural Science Foundation of China in 2009 and the Best Paper Award from IEEE Communication Society Asia-Pacific Board in 2013. He also co-received the Best Paper Awards from the 13th, 15th, and 19th Asia-Pacific Conference on Communication (APCC) in 2007, 2009, and 2013, respectively, International Conference on Wireless Communications and Signal Processing (WCSP'13), and the Best Student Paper Award from the 25th International Teletraffic Congress (ITC25). He is currently the chief scientist of the National Basic Research Program (so called "973 Project") of China on "Fundamental Research on the Energy and Resource Optimized Hyper-Cellular Mobile Communication System" (2012-2016), which is the first national project on green communications in China. He is a fellow of both the IEEE and IEICE.