

Distributed Backup Scheduling: Modeling and Optimization

Peter M. van de Ven, Bo Zhang
IBM Thomas J. Watson Research Center
1101 Kitchawan Rd, Yorktown Heights, NY 10598

Angela Schörgendorfer
Google
1600 Amphitheatre Parkway, Mountain View, CA 94043

Abstract—Recent years have seen rapid growth in data storage, magnifying the importance of ensuring data safety by performing regular backups. However, traffic created by such backups can be a significant burden on the underlying communication network. In the present paper we address the tradeoff between frequent backups (increased safety) and reducing the network peak load. We address the problem of shifting backup traffic from peak hours to off-peak hours within the constraints imposed by user connectivity. Backups are scheduled using a distributed protocol characterized by a set of probabilities that indicate the likelihood of a user initiating a backup during a given hour. Given these probabilities, we study the network capacity by investigating the rate at which users can generate data while retaining stable backlog processes. We then derive explicit expressions for the stationary behavior of the backup process, and discuss how to choose the backup probabilities that strike the right balance between a low peak load and data safety. Via simulation experiments we show that this approach is highly successful in reducing costs.

I. INTRODUCTION

The past decade has witnessed explosive growth in the amount of data generated and stored. According to a recent McKinsey report [7], 235 terabytes of data were collected by the US Library of Congress by April 2011, and 15 out of 17 sectors in the United States have more data stored per company than the Library of Congress.

The unprecedented scope of data storage magnifies the challenge of guaranteeing data safety by performing frequent data backups. For large organizations it is essential that their employees perform regular data backups to ensure uninterrupted operations and avoid the potentially immense costs of data loss. Various commercial solutions are available for managing backups in systems with large numbers of users, see [1], [3], [11].

These backup solutions perform data transfer across a network to a centralized backup server (or the cloud), which will be the setting we consider in this paper. Traffic created by data backups can be a significant portion of total network traffic, and the timing of data backups is critical for maintaining a low network peak load. Network connections are typically sized according to peak load, and the size of connections affects their cost, so reducing peak load is an equally crucial component in minimizing total network costs.

A key question arises: How should backups be scheduled to smooth overall network traffic and reduce the effect that backups have on network peak load? In particular, how can

we address the tradeoff between performing frequent backups to improve data safety on the one hand, and the resulting increases in network costs on the other hand? This work originates from a joint project between *IBM Research* and the IBM divisions *Europe Innovations Team* and *Integrated Technology Delivery*, the objective of which was to investigate this tradeoff in the context of the IBM internal data network.

Existing commercial solutions for managing large-scale backup traffic use a distributed approach in which users themselves decide when to initiate a backup, based on local information only. This de-centralized approach is more scalable than a centralized implementation in which the backup server instructs all users when to initiate their backups, as a distributed approach does not require centralized knowledge of the state of the entire system and thus generates less communication overhead.

The distributed backup mechanism under consideration assigns certain *backup probabilities* $\phi(t)$ to each hour t . At the beginning of hour t users will initiate a backup in that hour with probability $\phi(t)$. This decision is made without observing external information such as the current network load, and is independent from the decision of other users. The backup probabilities may be a function of various parameters such as the hour of the day, the time since the previous backup, and the backlog accumulated up since the previous backup.

The network carries extraneous data traffic, in addition to traffic generated by user backups. Assuming non-critical user data, backup traffic is typically delay-tolerant, as postponing backups by a few hours is unlikely to affect the perceived performance of the backup service. Our goal is then to design the backup probabilities for minimizing the network peak load, by scheduling backups when the extraneous load is low (off-peak) and avoiding doing backups when the extraneous load is high (peak).

However, one must be careful to avoid choosing consistently low backup probabilities, effectively prohibiting users from regularly accessing the backup server. To that end, we consider the problem of choosing the backup probabilities for minimizing a cost function of the network load, subject to constraints on the time between successive backups.

To the best of our knowledge, no formal mathematical model of backup scheduling (distributed or otherwise) has been proposed. The contribution of our work is twofold. First, we fill the gap in the research literature on modeling

the distributed backup process by developing a novel two-dimensional Markov chain model and analyzing its stability conditions and stationary behavior. Second, we study the question of how to design the backup probabilities in order to address the aforementioned tradeoff.

We consider a large corporate network of workstations (‘users’). On each workstation a backup client software application is installed and used to arrange users’ data backups in a distributed manner. All data backups are transmitted across the network to a centralized backup server. A user that schedules a backup will transfer only data generated since the previous backup to the server. As a user goes without doing a backup for a long time, it will have more data to transfer. Throughout this paper we use the term ‘user’ to describe both the workstation itself and the backup client software running on the workstation.

Users can only initiate a backup when connected to the network, and user connectivity may vary over time. For example, a user may not be connected outside of working hours. Users that are not connected may still generate data to be backed up at a later time.

The conflict between peak-load shifting and frequent backups is exacerbated by the fact that connectivity and extraneous load are positively correlated in practice. We demonstrate that the optimal choice for the backup probabilities is indeed low for peak hours, and that they are increasing in the time since the previous backup.

When backup probabilities are low, users get few opportunities to schedule a backup and their backlog may grow large as a result. In fact, we show that when backup probabilities decrease too fast with the time since the previous backup, the system may not be able to sustain the rate at which backup traffic is generated. Large backlogs amplify the risk of data loss, a concern that is the main reason for implementing a system for backups in the first place. Given our proposed backup scheme, we are able to express the long-run average backlog, which allows us to quantify the expected amount of data at risk of loss in the case of random workstation crashes.

We are not aware of any study in the literature that rigorously models backup scheduling, or addresses peak-load shifting in this context. For our application, one relevant study is by Sandnes and Huang [10], in which a temporal load balancing strategy for distributed web applications is proposed and analyzed. The effect of delay-tolerant bulk transfers such as backups on Internet traffic is explored in [5], along with a discussion on how to utilize off-peak capacity to perform these transfers. The problem of peak-load shifting is also related to Internet congestion management via raffle-based or time-of-day pricing schemes (see [6] and the references therein), incentivizing commuters to travel at less congested times [8], [9] and peak-shaving in smart grid [4], or reducing manufacturing costs [13]. However, the distributed nature and high delay-tolerance encountered in our setting create unique challenges.

The model studied in this paper shows some resemblance to the single-channel wireless model first proposed in [12]. Both

instances consider a collection of users that share a resource subject to connectivity constraints; in the latter setting the resource represents the wireless medium and the connectivity constraints reflect fading effects. The main differences are the centralized scheduling approach used in [12], and different interference and capacity constraints compared to [12].

The remainder of this paper is structured as follows. In Section II we present the model, and in Sections III and IV we study stability conditions and the stationary distribution of the backup process, respectively. We use these results to address the problem of optimizing distributed backup scheduling in Section V, and we present the corresponding proofs in Section VI. In Section VII this approach is applied to balancing traffic of a data network using real-life data, and we validate our results by simulation experiments, and show that our solution works well in practice. We conclude by discussing future research directions in Section VIII.

II. MODEL DESCRIPTION

We consider N users that connect to a backup server over a data network. Time is slotted, with slots representing hours, and indexed by $t = 0, 1, \dots$. Time slots are further grouped into days of $T = 24$ slots each. For any $k \geq 0$, the k th day corresponds to the slots $\{kT, \dots, (k+1)T - 1\}$. For $u = 0, \dots, T - 1$, we say that slot $kT + u$ corresponds to hour u of day k and denote by $\eta(t) := t \bmod T$ the hour of the day corresponding to slot t . Throughout this paper, we use ‘slot’ and ‘hour’ interchangeably.

Consider the system represented in Figure 1. User i generates $A_i(t)$ units of data (bits) for backup at the beginning of slot t . For each $u = 0, \dots, T - 1$, the $A_i(kT + u)$ ’s are assumed to be independent among hourly slots u , i.i.d. (independent and identically distributed) between days and users, and we denote $a(u) := \mathbb{E}[A_1(u)]$, $u = 0, \dots, T - 1$. In other words, the stochastic process $\{A_i(t), t \geq 0\}$ is periodic with cycle length T . During slot t , $A_i(t)$ bits of data are added to the backlog $B_i(t)$, which represents the total amount of data that user i has not backed up at the end of slot $t - 1$.

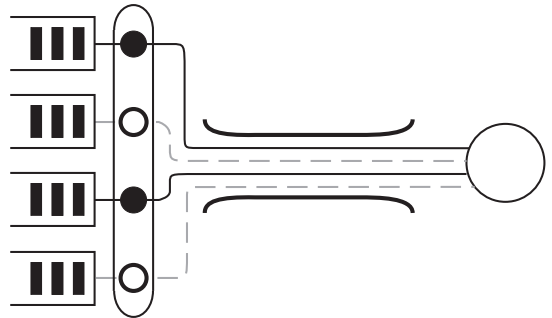


Figure 1. The model consisting of users, data network and backup server. Users may be connected (solid lines) or not connected (dashed lines) to the network. Among connected users, some may schedule a backup (grey circle).

In each slot, a user may transfer data, consisting of the backlog from previous time slots $B_i(t)$ and the newly generated bits $A_i(t)$, to the server. But the user can only do so if

it is connected to the network. Connectivity varies over time and is represented by a stochastic process $\{C_i(t), t \geq 0\}$, indicating whether user i is connected ($C_i(t) = 1$) or not ($C_i(t) = 0$). The connectivity reflects whether a user is currently able to perform a backup. For example, a laptop that is taken home after working hours is considered as being not connected, although it may still generate new backup data. In addition to traffic generated by the backups, the data network carries extraneous traffic $L(t)$. Similar to the data generation process $\{A_i(t), t \geq 0\}$, user connectivity and extraneous traffic are assumed to be periodic with cycle length T , reflecting fluctuations throughout the day. We assume that $A_i(t)$, $C_i(t)$ and $L(t)$ are independent mutually and across slots, and that their distributions depend only on the hour of the day, i.e., $C_i(u) \stackrel{d}{=} C_i(u + kT)$, $i = 1, \dots, N$ and $L(u) \stackrel{d}{=} L(u + kT)$, $k = 0, 1, \dots$; $u = 0, \dots, T - 1$. We assume that all users have the same connectivity probability ($C_i(u) \stackrel{d}{=} C_1(u)$), and denote $c(u) = \mathbb{E}[C_1(u)] = \mathbb{P}(C_1(u) = 1)$, $u = 0, \dots, T - 1$.

We consider a randomized backup scheme, i.e., in each time slot, connected users decide to schedule a backup with a certain probability. Specifically, let $\sigma_i(t) \in \{0, 1\}$ represent whether user i schedules a backup in slot t , and write

$$\sigma_i(t) := \begin{cases} 1 & \text{w.p. } \phi_i(t), \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

A backup scheduled in slot t by user i is successful only if that user is connected during slot t . We assume that backups are completed in the time slot in which they are initiated and that the data newly generated during that time slot are included in the backup. Hence, the backlog of user i evolves according to

$$B_i(t+1) = (A_i(t) + B_i(t))(1 - \sigma_i(t)C_i(t)). \quad (2)$$

In addition to the backlog, we also keep track of the user type $W_i(t)$, the number of days since the previous backup of user i at the beginning of slot t :

$$W_i(t) := \left\lfloor \frac{t}{T} \right\rfloor - \left\lfloor \frac{\tau_i^*(t)}{T} \right\rfloor, \quad (3)$$

with

$$\tau_i^*(t) := \max\{0 \leq \tau \leq t - 1 : \sigma_i(\tau)C_i(\tau) = 1\}. \quad (4)$$

Here $\tau_i^*(t)$ represents the time slot of user i 's most recent backup before slot t , with $\tau_i^*(t) := 0$ if no backup has been completed by time t . We call user i a type- w user at slot t if $W_i(t) = w$ for any $w = 0, 1, 2, \dots$, and remark that a type-0 user has done a backup that day. The user type of user i then satisfies

$$W_i(t+1) = W_i(t)(1 - \sigma_i(t)C_i(t)) + \mathbb{1}_{\{\eta(t+1)=0\}}. \quad (5)$$

Probabilities ϕ_i may depend on properties of user i , but cannot take as input any information about other users in the system due to the distributed nature of the backup scheme. Specifically, we consider backup probabilities that depend on the hour of the day and the user type. That is, $\phi_i(t) = \nu(\eta(t), W_i(t))$ for some function $\nu : \{0, \dots, T - 1\} \times \mathbb{N}_0 \mapsto$

$[0, 1]$, where \mathbb{N}_0 represents the set of all non-negative integers. This formulation covers a wide range of backup policies and, as we shall illustrate, allows for sufficient flexibility to balance regular backups and cost minimization.

The $B_i(t)$ and $W_i(t)$ are closely related, and are readily bounded as

$$(W_i(t) - 1)^+ \hat{a} \leq \mathbb{E}\{B_i(t)|W_i(t)\} \leq (W_i(t) + 1)\hat{a}, \quad (6)$$

where $\hat{a} = \sum_{u=0}^{T-1} a(u)$ represents the expected aggregate traffic generated during a single day. In Lemma 1 we shall present the exact relationship between $B_i(t)$ and $W_i(t)$, which is more involved as this depends on the hour of the last backup.

In Sections III and IV we study the stability conditions and stationary behavior for fixed backup probabilities, respectively, and in Section V we use these results to balance the network load across the day and minimize costs subject to certain constraints on backup frequency. Because the backlog $B_i(t)$ and user type $W_i(t)$ are so closely related, it suffices to only keep track of the hour of the day and the user type: Stability of $W_i(t)$ implies stability of $B_i(t)$ (Section III) and Lemma 1 allows us to translate results on the stationary distribution of $W_i(t)$ to that of $B_i(t)$ (Section IV).

Upon inspection of the evolution of the backlog (2) and user type (5) we see that users are independent from each other. As we shall explain in more detail in the relevant sections, this allows us to obtain network-wide behavior from that of a single user. Thus we shall in Sections III and IV limit ourselves to studying the behavior of a single user, and then broaden the scope to the full user population in Section V.

Specifically, we propose the two-dimensional Markov chain $U := \{(\eta(t), W(t)), t = 0, 1, \dots\}$, where the user type process $W(t)$ satisfies (5) with all the subscripts therein removed, as the first rigorous model on the (distributed) data backup process in the literature. The first dimension of the process $\eta(t)$ has strong statistical correlations with user behavior (i.e., connectivity in our model) and direct cost implications through its correlation with extraneous network traffic. The second component $W(t)$ can be interpreted as the *age of data not backed up*, which is a simple metric for data safety.

We shall derive stability conditions on the backup policy and study the stationary behavior of this Markov chain. This chain is periodic with period T and state space $\Xi := \{(u, w) : u = 0, \dots, T - 1, w \in \mathbb{N}_0\} \setminus \{(0, 0)\}$. State $(0, 0)$ should be excluded from the state space because hour 0 is the first hour of the day and no backup could have been performed before this hour on the same day.

III. STABILITY CONDITIONS

In this section we derive the conditions on the backup policy ν that lead to (in)stability. We assume that users all have the same set of backup probabilities, and because of the users' independence the stability of a single user is equivalent to that of the entire network. We can thus limit ourselves to deriving the stability conditions for the process U .

We first present a sufficient condition for stability. Let

$$m(u, u', w) := \prod_{v=u}^{u'} (1 - c(v)\nu(v, w)), \quad u, u' = 0, \dots, T-1; \\ w = 0, 1, \dots, \quad (7)$$

and define

$$g(u, w) := 1 - m(u, T-1, w)m(0, u-1, w+1), \\ u, u' = 0, \dots, T-1; \quad w = 0, 1, \dots, \quad (8)$$

the probability that a type- w user does at least one backup within the next T time slots starting from hour u . The following proposition provides a sufficient condition on ν for the positive recurrence of the joint process U .

Propositon 1. *If*

$$\alpha(u) := \lim_{w \rightarrow \infty} w(g(u, w) - w^{-1}) \quad (9)$$

exists and $\alpha(u) \in (0, \infty)$ for all $u \in \{0, 1, \dots, T-1\}$, then U is positive recurrent.

All the proofs, including that of Proposition 1, can be found in Section VI.

Using Proposition 1, we can provide a simpler but weaker stability condition by focusing on individual hours.

Corollary 1. *If*

$$\alpha(u) := \lim_{w \rightarrow \infty} w(c(u)\nu(u, w) - w^{-1}) \quad (10)$$

exists for all $u \in \{0, 1, \dots, T-1\}$, and $\alpha(u) \in (0, \infty)$ for at least one $u \in \{0, 1, \dots, T-1\}$, then U is positive recurrent.

An immediate implication of the above result is a sufficient stability condition in the case that ν is constant in the user type.

Corollary 2. *If one of the following conditions holds, then U is positive recurrent.*

- (i) *There exists some $w^* \geq 0$ such that $\nu(\cdot, w) = \nu(\cdot, w^*)$ for all $w \geq w^*$ and there exists some $u \in \{0, \dots, T-1\}$ such that $c(u)\nu(u, w^*) > 0$.*
- (ii) *$\nu(\cdot, w) = \nu(\cdot, 0)$ for all $w \in \mathbb{N}_0$ and there exists a $u \in \{0, \dots, T-1\}$ such that $c(u)\nu(u, 0) > 0$.*

Note that part (ii) implies, for example, stability in the case that $\nu(u, w) = \nu(0, 1) > 0$ (constant backup probability) and users connect at least once per day with a positive probability.

The type of policies considered in Corollary 2 are particularly important in practice, which requires straightforward policies that are easy to implement. If we think of the mapping implied by function ν as a look-up table with width T and an infinite length along the dimension of w , then the results apply to the special case that all rows are identical starting from the $(w^* + 1)$ th row of the table. Equivalently, a backup policy in this case is simply identified by a finite-length $(w^* + 1) \times T$ look-up table. This tabular representation has indeed been implemented in existing commercial backup

software applications, and we discuss this case in more detail in Section VII.

Our next result is a necessary condition for stability on the backup policy ν .

Propositon 2. *U is transient if $\nu(u, w) = o(w^{-1})$ as $w \rightarrow \infty$ for all u .*

Proposition 2 suggests that the backup probabilities cannot decrease faster than the order of the reciprocal of the user type for the system to maintain stability.

IV. STATIONARY ANALYSIS

In this section we assume that the stability conditions from Section III are satisfied, and we analyze the behavior of U in stationarity. As users are independent, the stationary distribution of the stochastic process that describes the behavior of all users has a product-form solution and can be found by multiplying the corresponding probabilities for each individual user (conditioned on the hour of the day).

The stationary distribution π of the process U satisfies the following balance equations:

$$\pi(u, 0) = \pi(u-1, 0) + \sum_{y=1}^{\infty} \pi(u-1, y)c(u-1)\nu(u-1, y), \\ u = 2, \dots, T-1, \quad (11)$$

$$\pi(1, 0) = \sum_{y=1}^{\infty} c(0)\nu(0, y)\pi(0, y), \quad (12)$$

$$\pi(u, w) = \pi(u-1, w)(1 - c(u-1)\nu(u-1, w)), \\ u = 1, \dots, T-1; \quad w \geq 1, \quad (13)$$

$$\pi(0, w) = \pi(T-1, w-1)(1 - c(T-1)\nu(T-1, w-1)), \\ w \geq 1. \quad (14)$$

By manipulating these balance equations we can solve for the stationary distribution of U . We define a product \prod as 1 if its upper bound is smaller than its lower bound, and obtain

$$\pi(u, w) = \pi(0, 1)k_1(u, w), \quad u = 0, \dots, T-1; \quad w \geq 1, \quad (15)$$

$$\pi(u, 0) = \pi(0, 1)k_2(u), \quad u = 1, \dots, T-1, \quad (16)$$

where

$$k_1(u, w) := m(0, u-1, w) \prod_{y=1}^{w-1} m(0, T-1, y), \quad (17)$$

$$k_2(u) := \sum_{v=1}^u \sum_{y=1}^{\infty} k_1(v-1, y)c(v-1)\nu(v-1, y), \quad (18)$$

and with normalizing constant

$$\pi(0, 1) = \left[\sum_{w=1}^{\infty} \sum_{u=0}^{T-1} k_1(u, w) + \sum_{u=1}^{T-1} k_2(u) \right]^{-1}. \quad (19)$$

The above explicit expressions of the stationary probabilities are equivalent to the long-run fraction of time that the process spends in each state. This enables us to evaluate some useful

performance measures. One such performance measure is the long-run average backup rate, i.e., the rate at which backups are completed, irrespective of their size:

$$\sum_{(u,w) \in \Xi} \pi(u,w)c(u)\nu(u,w), \quad (20)$$

This is the reciprocal of the expected time between successive backups.

We can also derive an expression for the expected backlog size of a type- w user at hour u :

$$S(u,w) := \mathbb{E}_\pi [B(t) \mid \eta(t) = u, W(t) = w]. \quad (21)$$

This conditional expectation also serves as an important building block for our analysis in Section V. Here we denote by \mathbb{E}_π and \mathbb{P}_π the expectation and probability with respect to the stationary measure π . Recall that $\hat{a} = \sum_{u=0}^{T-1} a(u)$.

Lemma 1. For any $(u,w) \in \Xi$,

$$\begin{aligned} S(u,w) &= \sum_{u'=0}^{T-1} p(u',u,w) \\ &\times \mathbb{E}_\pi [B(t) \mid \eta(t) = u, W(t) = w, \eta(\tau^*(t)) = u'], \end{aligned} \quad (22)$$

where

$$\begin{aligned} &\mathbb{E}_\pi [B(t) \mid \eta(t) = u, W(t) = w, \eta(\tau^*(t)) = u'] \\ &= \begin{cases} \sum_{v=u'+1}^{u-1} a(v), & w = 0, \\ \sum_{v=u'+1}^{T-1} a(v) + (w-1)\hat{a} + \sum_{v=0}^{u-1} a(v), & w \geq 1, \end{cases} \end{aligned} \quad (23)$$

and

$$\begin{aligned} p(u',u,w) &= \\ &\begin{cases} \sum_{w'=0}^{\infty} \frac{\pi(u',w')}{\pi(u,0)} c(u')\nu(u',w')m(u'+1,u-1,0), & w = 0, \\ \sum_{w'=0}^{\infty} \frac{\pi(u',w')}{\pi(0,1)} c(u')\nu(u',w')m(u'+1,T-1,0), & w \geq 1. \end{cases} \end{aligned} \quad (24)$$

Note that $\mathbb{P}_\pi(\eta(\tau^*(t)) = u' \mid X(t) = (u,w)) = p(u',u,w)$ is independent of w for $w \geq 1$. Using the $S(u,w)$ we can compute the long-run average backup traffic rate:

$$\sum_{(u,w) \in \Xi} \pi(u,w)c(u)\nu(u,w) \left(S(u,w) + a(u) \right). \quad (25)$$

Moreover, the long-run average backlog size at the beginning of a slot equals

$$\mathbb{E}_\pi [B(t)] = \sum_{(u,w) \in \Xi} \pi(u,w)S(u,w). \quad (26)$$

This measures the vulnerability of the data backup mechanism to random system crashes.

V. OPTIMIZATION OF DISTRIBUTED BACKUP SCHEDULING

Having studied the proposed Markov chain model for the backup process in detail, we now look at how to choose the backup policy that best addresses the tradeoff between ensuring regular backups and reducing backup costs. Specifically, the optimal backup policy is obtained by choosing an appropriate load-dependent network cost function and then designing the backup probabilities for minimizing the network costs. We denote by $g : \mathbb{R}_+ \mapsto \mathbb{R}_+$ the cost function that maps the load in a particular time slot to the cost. We assume g to be convex, so minimizing the expected total network costs across time is equivalent to peak-load shifting. We are interested in the long-run average per-day network costs

$$G(\nu) := \sum_{u=0}^{T-1} \mathbb{E}_\pi [g(L(u) + H(u;\nu))], \quad (27)$$

where $H(u;\nu)$ denotes the hour- u load due to backups under policy ν . We include ν in the argument of H to emphasize that the backup policy only affects this part of the network load.

The function $G(\nu)$ is difficult to deal with analytically, and in order to study the network costs we approximate $G(\nu)$ as

$$G(\nu) \approx \tilde{G}(\nu) := \sum_{u=0}^{T-1} g(\mathbb{E}[L(u)] + \mathbb{E}_\pi [H(u;\nu)]), \quad (28)$$

The $\mathbb{E}[L(u)]$ can be obtained from measurements, and $\mathbb{E}_\pi [H(u;\nu)]$ can be expressed in terms of the stationary distribution discussed in Section IV. By exploiting user independence and conditioning on the user type we obtain

$$\begin{aligned} &\mathbb{E}_\pi [H(u;\nu)] \\ &= N \mathbb{E}_\pi [(B(t) + A(t)) \mathbf{1}_{\{C(t)\sigma(t)=1\}} \mid \eta(t) = u] \\ &= N \sum_{w=0}^{\infty} \frac{\pi(u,w)}{\sum_{y=0}^{\infty} \pi(u,y)} (S(u,w) + a(u))c(u)\nu(u,w) \\ &= NT \sum_{w=0}^{\infty} (S(u,w) + a(u))\pi(u,w)c(u)\nu(u,w), \end{aligned} \quad (29)$$

with the last equality following from $\sum_{y=0}^{\infty} \pi(u,y) = 1/T$ for any $u \in \{0, \dots, T-1\}$ and introducing the convention that $\pi(0,0) \equiv 0$. Experience shows that this approximation is sufficiently accurate and works effectively for practical purposes (see Section VII).

We aim to minimize the network costs, while ensuring that users do regular backups. One way to establish regular user backups is by imposing constraints on the expected number of users of different types, say, at the beginning of each day. For some upper bound γ_1 , and some integer w_1 , a constraint of a typical *service-level agreement* form can read $T \sum_{w=w_1}^{\infty} \pi(0,w) < \gamma_1$. This imposes a constraint on the stationary fraction of users that have not done a backup for w_1 days or longer at the beginning of each day. There can be multiple (possibly infinite) constraints of this form in

the optimization formulation. Let $w_j \in \mathbb{N}$ and $\gamma_j \in [0, 1]$, $j = 1, 2, \dots$. We are interested in solving

$$\begin{aligned} & \min_{\nu} \tilde{G}(\nu) \\ & \text{such that } T \sum_{w=w_j}^{\infty} \pi(0, w) \leq \gamma_j, \quad j = 1, 2, \dots \end{aligned} \quad (30)$$

To avoid an infinite sum one may rewrite (30) as $\sum_{w=0}^{w_j-1} \pi(0, w) \geq (1 - \gamma_j)/T$. Similar constraints can be formulated based on the the average backup rate found in (20), or the average backlog size $\mathbb{E}_{\pi}[B(t)]$ found in (26). Note that the constraint in (30) can easily be modified to provide distinct service levels for different users, creating user classes with varying levels of importance. For γ_j small, the optimization (30) may not have a feasible solution.

As we discussed in Section III in relation to Corollary 2, from a practical perspective we are most interested in backup policies identified with a finite-length $(w^* + 1) \times T$ look-up table, i.e., $\nu(\cdot, w) = \nu(\cdot, w^*)$ for all $w \geq w^*$ for some $w^* \geq 0$. If we restrict ourselves to this class of policies and focus on solving for the best one among this class, we may simplify the expressions for the objective function and the constraints in formulation (30). However, the resulting optimization problem is still non-convex with $(w^* + 1) \times T$ decision variables. Despite the fact that the solver cannot guarantee global optimality of its solution (due to the non-convexity), we see in Section VII that it performs remarkably well in practice.

To see that this problem is indeed non-convex, consider the case $T = 2$ and $w^* = 0$, that is, all users have the same backup probabilities $\nu(0, 1)$ (for hour 0) and $\nu(1, 1)$ (hour 1). Denote $h(u) := (1 - c(u)\nu(u, 1))$, then

$$k_1(u, w) = \begin{cases} (h(0)h(1))^{w-1}, & u = 0, \\ h(0)(h(0)h(1))^{w-1}, & u = 1. \end{cases} \quad (31)$$

$$k_2(1) = \frac{c(0)\nu(0, 1)}{1 - h(0)h(1)}. \quad (32)$$

After some calculations we obtain

$$\pi(0, w) = \frac{1}{2} (1 - h(0)h(1)) (h(0)h(1))^{w-1}. \quad (33)$$

In Figure 2 we plot $\pi(0, 2)$ against $\nu(0, 1)$, and observe that it is not convex. In fact, taking its second derivative w.r.t. $\nu(0, 1)$ yields

$$\frac{\partial^2}{\partial \nu(0, 1)^2} \pi(0, 2) = -c(0)^2 (1 - c(1)\nu(1, 1)) < 0, \quad (34)$$

confirming non-convexity.

The non-convexity holds more generally for these stationary probabilities, and since both the objective function and the constraints are comprised of these probabilities, the optimization problem is non-convex.

VI. PROOFS

A. Proof of Proposition 1

Proof: In order to demonstrate positive recurrence of U , we need to define a Lyapunov function that has a negative drift

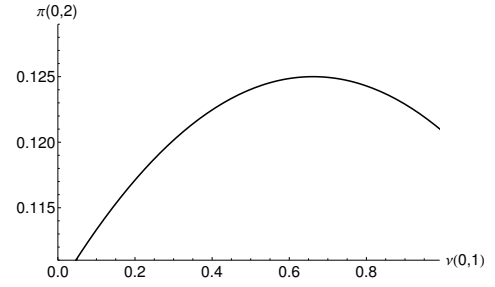


Figure 2. The probability $\pi(0, 2)$ plotted against $\nu(0, 1)$.

outside some closed set of states. We use the linear Lyapunov function $L(v, w) = w$, and consider the T -slot drift

$$\begin{aligned} \Delta_T L(v, w) &:= \mathbb{E}[L(\eta(v+T), W(t+T)) \\ &\quad - L(\eta(t), W(t)) \mid \eta(t) = v, W(t) = w] \\ &= \mathbb{E}[W(t+T) \mid \eta(t) = v, W(t) = w] - w, \end{aligned} \quad (35)$$

where in the last equality we used $\eta(t+T) = \eta(t)$. In order to bound Δ_T , observe from (5) that

$$W(t+T) \leq 1 + W(t) \prod_{s=0}^{T-1} (1 - \sigma(t+s)C(t+s)). \quad (36)$$

Substituting (36) into the expression for the drift (35), we obtain the following bound:

$$\begin{aligned} \Delta_T L(v, w) &\leq -w + 1 + \\ &\quad \mathbb{E}[W(t) \prod_{s=0}^{T-1} (1 - \sigma(t+s)C(t+s)) \mid \eta(t) = v, W(t) = w]. \end{aligned} \quad (37)$$

By conditioning on successive backup attempts and noting that $\{\eta(t+s) < \eta(t)\}$ represents the event that slot $t+s$ takes place the next day from slot t , we may write

$$\begin{aligned} & \mathbb{E}[W(t) \prod_{s=0}^{T-1} (1 - \sigma(t+s)C(t+s)) \mid \eta(t) = v, W(t) = w] \\ &= w \mathbb{P}\left(\bigcap_{s=0}^{T-1} \{\sigma(t+s)C(t+s) = 0\} \mid \eta(t) = v, W(t) = w\right) \\ &= w \prod_{s=0}^{T-1} \mathbb{P}(\sigma(t+s)C(t+s) = 0 \mid \eta(t) = v, W(t) = w, \\ &\quad \sigma(t)C(t) = \dots = \sigma(t+s-1)C(t+s-1) = 0) \\ &= w \prod_{s=0}^{T-1} (1 - c(\eta(t+s))\nu(\eta(t+s), w + \mathbf{1}_{\{\eta(t+s) < \eta(t)\}})) \\ &= w(1 - g(v, w)), \end{aligned} \quad (38)$$

Denote

$$\epsilon := \lim_{w \rightarrow \infty} \min\{1, \frac{w}{2}(g(v, w) - w^{-1})\} > 0, \quad (39)$$

then we know that there exists some $M \in \mathbb{N}$ such that $w(g(x, w) - \frac{1}{w}) > \epsilon$ for all $w \geq M$ and $v \in \{0, \dots, T-1\}$. We introduce the set

$$C := \{(v, w) \in \Xi : w \leq M\}. \quad (40)$$

By the definition of ϵ , we have that, for any $(v, w) \in C^c$,

$$w(g(v, w) - w^{-1}) \geq 2\epsilon > \epsilon, \quad (41)$$

and hence

$$g(v, w) > \frac{1 + \epsilon}{w}. \quad (42)$$

Substituting this into (38) then leads to

$$\begin{aligned} & \mathbb{E}\left[W(t) \prod_{s=0}^{T-1} (1 - \sigma(t+s)C(t+s)) \mid \eta(t) = v, W(t) = w\right] \\ & < w\left(1 - \frac{1 + \epsilon}{w}\right), \quad (v, w) \in C^c. \end{aligned} \quad (43)$$

Using this, we obtain from (37)

$$\Delta_T L(v, w) < 1 - w + w\left(1 - \frac{1 + \epsilon}{w}\right) = -\epsilon, \quad (v, w) \in C^c. \quad (44)$$

Positive recurrence of U then follows from [2, Theorem 2.2.4]. ■

B. Proof of Proposition 2

Proof: Define the linear Lyapunov function $L(x) = x$, the T -slot drift, by

$$\Delta_T L(w) = \mathbb{E}[W(t+T) \mid W(t) = w] - w. \quad (45)$$

Analogous to (36), we may bound

$$W(t+T) \geq (1 + W(t)) \prod_{s=0}^{T-1} (1 - \sigma(t+s)C(t+s)). \quad (46)$$

Combining (45) and (46), we obtain

$$\begin{aligned} \Delta_T L(w) & \geq (1 + w) \\ & \times \mathbb{E}\left[\prod_{s=0}^{T-1} (1 - \sigma(t+s)C(t+s)) \mid W(t) = w\right] - w. \end{aligned} \quad (47)$$

Then since $\nu(u, w) = o(w^{-1})$ for all u and the 0-1 binary $\sigma(t+s)$ is 1 with probability $\nu(\eta(t+s), W(t+s))$ for each $s = 0, \dots, T$, with $W(t+s) \leq w+1$ conditioning on $W(t) = w$, there must exist a number $C > 0$ such that for any $w > C$, $\Delta_T(w) > 0$. Transience then follows from applying [2, Theorem 2.2.4]. ■

C. Proof of Lemma 1

Proof: Denote $X(t) = (\eta(t), W(t))$ and recall that $\tau^*(t)$ denotes the time of the last backup before time t . Conditioning on the hour of the previous backup we obtain

$$\begin{aligned} S(u, w) & = \sum_{u'=0}^{T-1} \mathbb{P}_\pi(\eta(\tau^*(t)) = u' \mid X(t) = (u, w)) \\ & \times [B(t) \mid X(t) = (u, w), \eta(\tau^*(t)) = u']. \end{aligned} \quad (48)$$

The conditional expected backlog size can be written as (23). The distribution of the hour u' , $p(u', u, w) := \mathbb{P}_\pi(\eta(\tau^*(t)) = u' \mid X(t) = (u, w))$ can be obtained by conditioning on the user type at the time of the previous backup. Denote by

$$t' := t - (\eta(t) - u') - W(t)T \quad (49)$$

the time index of the previous backup given that this occurred in hour u' , and write

$$\begin{aligned} & p(u', u, w) \\ & = \sum_{w'=0}^{\infty} \mathbb{P}_\pi(\tau^*(t) = t', W(\tau^*(t)) = w' \mid X(t) = (u, w)) \\ & = \sum_{w'=0}^{\infty} \mathbb{P}_\pi(X(t') = (u', w'), X(t'+1) = (u'+1, 0), \dots \\ & \quad , X(t'+T-u'-1) = (T-1, 0), X(t'+T-u') = (0, 1), \\ & \quad \dots, X(t-1) = (u-1, w) \mid X(t) = (u, w)) \\ & = \sum_{w'=0}^{\infty} \mathbb{P}_\pi(X(t') = (u', w') \mid X(t'+1) = (u'+1, 0)) \\ & \quad \times \mathbb{P}_\pi(X(t'+1) = (u'+1, 0) \mid X(t'+2) = (u'+2, 0)) \\ & \quad \cdot \dots \cdot \mathbb{P}_\pi(X(t'+T-u'-1) = (T-1, 0) \\ & \quad \quad \mid X(t'+T-u') = (0, 1)) \\ & \quad \cdot \dots \cdot \mathbb{P}_\pi(X(t-1) = (u-1, w) \mid X(t) = (u, w)). \end{aligned} \quad (50)$$

The second equality follows from the observation that there exists a unique sample path between two successive backups, fully characterized by the hour of both backups and the corresponding user types. The third equality is due to the fact the time-reversed process is also Markovian. The transition probabilities of the reversed process (for $y = 0, 1, \dots$ and $v = 0, \dots, T-2$) are given as

$$\begin{aligned} & \mathbb{P}_\pi(X(t-1) = (T-1, y) \mid X(t) = (0, y+1)) \\ & = \frac{\pi(T-1, y)}{\pi(0, y+1)} (1 - c(T-1)\nu(T-1, y)), \end{aligned} \quad (51)$$

$$\begin{aligned} & \mathbb{P}_\pi(X(t-1) = (v, y) \mid X(t) = (v+1, y)) \\ & = \frac{\pi(v, y)}{\pi(v+1, y)} (1 - c(v)\nu(v, y)), \end{aligned} \quad (52)$$

$$\begin{aligned} & \mathbb{P}_\pi(X(t-1) = (v, y) \mid X(t) \\ & = (v+1, 0)) = \frac{\pi(v, y)}{\pi(v+1, 0)} c(v)\nu(v, y). \end{aligned} \quad (53)$$

Substituting these probabilities into (50) yields the desired result. ■

VII. NUMERICAL RESULTS

We now consider the data network at a corporate location with $N = 5397$ users who need to transfer data to a remote backup server via the corporation-internal network. We solve the network cost minimization problem formulated in Section V, in which we use a quadratic cost function $g(x) = x^2$ aiming to shift the peak load and balance network load across the day. We measure network traffic and backup traffic during workdays (i.e., excluding weekends and holidays) at the site

over the course of a year to obtain estimates for $\mathbb{E}[L(t)]$, $c(u)$ and $a(u)$. Both extraneous load and connection probabilities have peaks in the slots corresponding to 9-11 am and 1-3 pm. Both decrease significantly outside of office hours, as expected (Figure 3). We find that even though network load does vary slightly from day to day (approximately $\pm 10\%$), most of the variability in load is explained by hourly patterns. However, our model readily extends to multiple days by increasing the number of time slots from $T = 24$ to $T = 5 \times 24$.

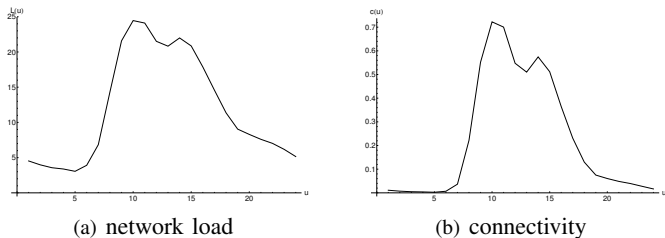


Figure 3. Estimates for the connectivity and network load.

We set $w^* = 5$, so $\nu(\cdot, w) = \nu(\cdot, w^*)$ for all $w \geq w^*$, and choose $\gamma_1 = 0.25$, $\gamma_2 = 0.1$, $\gamma_3 = 0.05$, $\gamma_4 = 0.01$ and $\gamma_5 = 0.002$. The $\nu^*(u, w)$ obtained by solving (30) are shown in Figure 4. As expected, the backup probability is low during peak hours and high at night. Moreover, the optimal backup probabilities are such that users that recently did a backup have a very low backup probability when the extraneous load is high. As the number of days since last backup (w) increases, users are permitted to schedule backups with increasing probability during hours with high extraneous traffic, and eventually may even initiate a backup during peak hours in the afternoon. These slots are favorable due to the positive correlation between the backup probabilities and extraneous load.

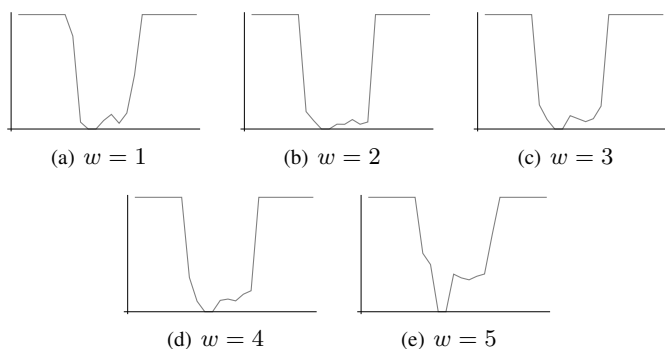


Figure 4. The $\nu^*(u, w)$ plotted against u , $w = 1, \dots, 5$.

We also develop a discrete-event simulation model to assess the performance of different backup policies. We simulate the 5397-user network for 500 days, and are interested in the effect of the backup policies on the network load.

Figure 5 shows the impact of the backup on the hourly load based on the derived optimal policy ν^* . Observe that the

resulting aggregate load on the network is smoothed across peak working hours and the overall peak load is not increased. In contrast, Figure 6 shows the aggregate load in the case that users are allowed to schedule backups with uniform probability ($\nu(u, w) \equiv \kappa$, $\kappa = .2, .4, .8$). The shaded area corresponds to the extraneous load. By allowing users to schedule backups irrespective of extraneous traffic, the backup load increases aggregate traffic on the network during day-time working hours.

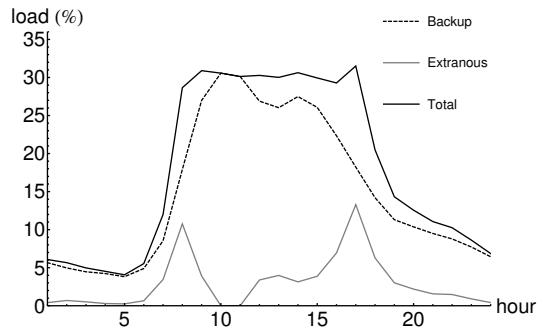


Figure 5. The extraneous load (dashed), load due to backups (gray) and aggregate load (black).

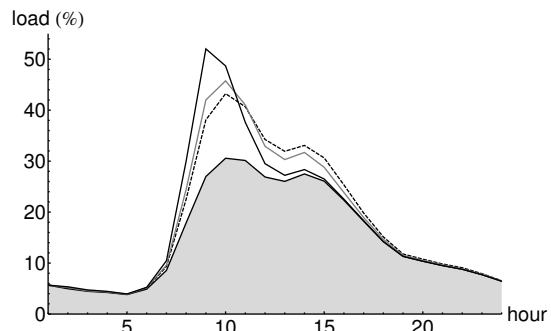


Figure 6. Aggregate load for uniform probability $\kappa = 0.2$ (dashed) $\kappa = 0.4$ (gray) and $\kappa = 0.8$ (black).

In Figure 7 we show the fraction of users of each type, plotted against the hour of the day. At the beginning of the day there are no type-0 users, since no user has yet completed a backup that day. As the day progresses, more users perform a backup and the number of type-0 users increases. The number of type- w users naturally decreases, as users that complete a backup are converted to type-0 users. At the end of each day the number of type- w users, $w \geq 1$, increases because the type of all users increases by 1, see (5).

Finally, the solution presented in this section has been implemented in practice and been shown to perform very well.

VIII. CONCLUSIONS AND OUTLOOK

In this paper we propose a novel model to study backup scheduling for data networks when the backup process is governed by hourly backup probabilities. Using this model we

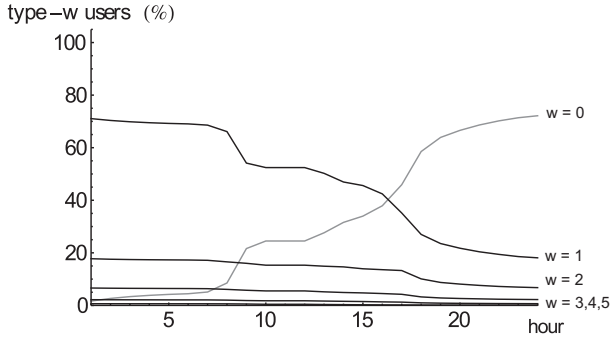


Figure 7. The fraction of users of each type.

explore how much backup traffic the network is able to sustain for given backup probabilities. We compute the stationary distribution of the two-dimensional Markov process of the time since the previous backup and the hour of the day, which is in turn used to compute backup probabilities that minimize network costs subject to regular backup constraints. These methods are tested on a real-life scheduling problem, and the results are validated using a discrete-event simulation.

A. Dependencies in connectivity behavior

Our analysis assumes that a user's connectivity in a given hour is independent from other users and its own connectivity history. Realistically, however, hourly connectivity $C_i(t)$ is correlated with both connectivity of the previous days $C_i(t-kT)$ and previous hours $C_i(t-s)$. The current approach could be extended by explicitly modeling this relationship. Alternatively, we may consider grouping users into user classes according to their connectivity patterns in the past. Many users may only connect during daytime hours, while some users regularly connect during off-peak hours in the evenings or over night. It may then be preferable to let users of the latter type schedule backups during off-peak hours and limit their access to peak-time slots. Finding an optimal schedule for each user class would conceivably improve peak-load reduction above what the current single-class scheme can achieve.

B. Centralized scheduling

The distributed algorithm currently used in practice and discussed in this paper has low communication overhead, but it is unclear how well it does compared to a centralized solution. Alternatively, we may consider a setting in which information on network traffic and the individual users (i.e., connectivity, backlog size, time since last backup, etc.) is available to the backup server, which then makes a centralized decision in each slot on which users can do a backup. Based on predictions of future load and connectivity, the server can then schedule users without relying on a randomized algorithm. Depending on the amount and accuracy of the information known to the server, the scheduling problem within each slot can be seen as a (stochastic) knapsack problem. It would be of interest to analyze the optimal behavior and the potential improvement

over the current distributed scheme, which would however come at the cost of additional communication overhead.

C. Constrained backup server

We have assumed that the backup server can support any number of simultaneous backups irrespective of the backlog of the users involved. Our goal was to design backup probabilities that minimize the network costs by spreading the network load (backup and other traffic) evenly across the day. In practice there may be constraints on the number of simultaneous backups or the amount of traffic that the server can handle, and we have to take this into account while minimizing the costs. Such constraints create dependence between users, and deriving analogues to the results of Sections III-V is not straightforward. Although such limitations on the backup server rarely play a significant role in practice, they are nevertheless interesting to investigate in more detail.

Suppose that in each time slot at most β units of data can be written to the backup server, either due to the capacity constraint on the backup server or the network. If the total amount of backup traffic $H := \sum_{i=1}^N (A_i(t) + B_i(t))\sigma_i(t)C_i(t)$ exceeds β , then each user can only perform a backup proportional to its total backlog size:

$$(A_i(t) + B_i(t))\sigma_i(t)C_i(t) \cdot \frac{\beta}{H}, \quad i = 1, \dots, N.$$

In this case, the dynamics of the backlog process in (2) are no longer valid, and can be rewritten as

$$B_i(t+1) = (A_i(t) + B_i(t)) [1 - C_i(t)\sigma_i(t) \cdot \min\{1, \frac{\beta}{H}\}].$$

Let us assume $A_i(t)$'s are i.i.d. with mean a and $C_i(t)$'s are i.i.d. with mean c . Consider a *constant backup policy*, i.e., $\phi(t) = \nu$ for some $\nu > 0$.

For any vector \mathbf{x} , let $|\mathbf{x}|$ denote its 1-norm. We write $\delta = (\delta_1, \dots, \delta_N) \in \{0, 1\}^N$, $S_i := \{\delta \in \{0, 1\}^N : \delta_i = 1\}$, and $q(\delta) := (c\nu)^{|\delta|}(1 - c\nu)^{N-|\delta|}$. Further, let $i = 1, \dots, N$ and denote $\rho := \beta \sum_{\delta \in S_i} q(\delta) / |\delta| = \beta [1 - (1 - c\nu)^N] / N$, the average backup rate of each user. Note that this holds for any i due to our assumption of all users being statistically identical.

We can again look at stability of the system. Since ν does not depend on $W(t)$ or $\eta(t)$ in this case, the process $U := \{B(t), t = 0, 1, \dots\}$ is Markovian. We have identified the following sufficient condition for stability of U .

Propositon 3. *If $a < \rho$, then U is positive recurrent.*

Due to space limitations the proof of Proposition 3 is omitted. In addition to extending Proposition 3 to the more general assumptions used in Section III, another challenge is to study the stationary behavior of the constrained system. Since users are now dependent, it is not sufficient to study a single user. Although the joint process is difficult to study, the analysis may simplify in certain asymptotic regimes.

REFERENCES

- [1] EMC Corporation, *EMC Backup and Recovery Solutions*, Bedford, MA, USA, 2013, www.emc.com/backup-and-recovery/index.htm.

- [2] G. Fayolle, V. Malyshev, and M. Menshikov, *Topics in the Constructive Theory of Countable Markov Chains*. Cambridge, UK: Cambridge University Press, 1995.
- [3] I.B.M. Corporation, *IBM Tivoli Storage Manager*, Armonk, NY, USA, 2013, www.ibm.com/Tivoli-Storage-Manager.
- [4] A. Ipakchi and F. Albuyeh, "Grid of the future," *IEEE Power Energy Mag.*, vol. 7, no. 2, pp. 52–62, 2009.
- [5] N. Laoutaris, G. Smaragdakis, P. Rodriguez, and R. Sundaram, "Delay tolerant bulk data transfers on the internet," in *Proc. ACM SIGMETRICS '09*, Seattle, WA, 2009, pp. 229–238.
- [6] P. Loiseau, G. Schwartz, J. Musacchio, and S. Amin, "Incentive schemes for internet congestion management: Raffles versus time-of-day pricing," in *Proc. 49th Annual Allerton Conference on Communication, Control, and Computing*, 2011, pp. 103–110.
- [7] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity," McKinsey Global Institute, Tech. Rep., May 2011. [Online]. Available: http://www.mckinsey.com/insights/business_technology/
- [8] D. Merugu, B. Prabhakar, and N. Rama, "An incentive mechanism for decongesting the roads: A pilot program in Bangalore," in *Proc. ACM NetEcon*, Stanford, CA, 2009. [Online]. Available: http://www.stanford.edu/~balaji/papers/NetEcon_final.pdf
- [9] P. Olszewski and L. Xie, "Modelling the effects of road pricing on traffic in Singapore," *Transport Res. A-Pol.*, vol. 39, no. 7–9, pp. 755–772, 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0965856405000479>
- [10] F. Sandnes and Y.-P. Huang, "Temporal load-balancing of web-server traffic," in *Proc. PDCAT '06*, Washington, DC, 2006, pp. 266–271. [Online]. Available: <http://dx.doi.org/10.1109/PDCAT.2006.106>
- [11] Symantec Corporation, *Symantec NetBackup*, Mountainview, CA, USA, 2013, www.symantec.com/netbackup.
- [12] L. Tassiulas and A. Ephremides, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Inf. Theory*, vol. 39, no. 2, pp. 466–478, 1993.
- [13] R. Wu, T. Lee, and E. Hill, "An investigation of the accuracy and the characteristics of the peak-shaving method applied to production cost calculations," *IEEE Trans. Power Syst.*, vol. 4, no. 3, pp. 1043–1049, 1989.