This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2014.2308726, IEEE Transactions on Multimedia

Dynamic Request Redirection and Elastic Service Scaling in Cloud-Centric Media Networks

Jianhua Tang, *Student Member, IEEE,* Wee Peng Tay, *Member, IEEE,* and Yonggang Wen, *Member, IEEE*

***Abstract*—We consider the problem of optimally redirecting user requests in a cloud-centric media network (CCMN) to multiple destination Virtual Machines (VMs), which elastically scale their service capacities in order to minimize a cost function that includes service response times, computing costs, and routing costs. We also allow the request arrival process to switch between normal and flash crowd modes to model user requests to a CCMN. We quantify the trade-offs in flash crowd detection delay and false alarm frequency, request allocation rates, and service capacities at the VMs. We show that under each request arrival mode (normal or flash crowd), the optimal redirection policy can be found in terms of a price for each VM, which is a function of the VM's service cost, with requests redirected to VMs in order of nondecreasing prices, and no redirection to VMs with prices above a threshold price. Applying our proposed strategy to a YouTube request trace data set shows that our strategy outperforms various benchmark strategies. We also present simulation results when various arrival traffic characteristics are varied, which again suggest that our proposed strategy performs well under these conditions.**

***Index Terms*—User request redirection, service capacity scaling, cost-aware provisioning, resource allocation, cloud-centric content network, quickest detection**

I. Introduction

There is an urgent need to design networks and protocols that specifically address the many technical challenges introduced by the upsurge in Internet multimedia traffic [1], [2]. Cisco has predicted that Internet traffic will grow with an annual rate of 32% in next few years, i.e., nearly double every 2.5 years [3], with the growth driven mostly by increased demand for video content and mobile data. By the year 2015, video traffic, including TV, Internet, VoD and P2P, will constitute approximately 90% of global Internet traffic [4]. However, according to Broadband Properties Magazine, the annual growth rate of Internet infrastructure is merely 19% [5]. This mis-matched growth between supply and demand may trigger a quality-of-service (QoS) deterioration of network services, with end users starting to experience longer latencies than before.

One feature of multimedia traffic is the higher frequency of flash crowds compared to normal Internet traffic [6]. A flash crowd occurs when there is an unexpectedly high amount of user traffic during a short period of time [7]. For example, [2] records 2501 flash crowds appearing in the CoralCDN over a 4-year span, i.e., almost two flash crowds happen daily on average. The use of content delivery networks (CDNs), pioneered by Akamai [8], allows contents to be pushed towards the edge servers [9], which are closer to the end consumers. However, due to their inherent architectural limitations, existing CDN solutions are inadequate to deal with the exponential growth of multimedia traffic. First, with an increasing amount of dynamic and media contents in the CDNs, web applications are becoming more intensive in computation capability. As a result, the traditional server selection mechanisms (e.g. forwarding requests to the closest server) designed for static contents may no longer be optimal [10]. Second, static resource allocation mechanisms suffer from poor resource dimensioning as multimedia traffic is highly variable, as reflected by the high peak to valley ratio [11] of user traffic. A static allocation based on peak hour traffic may have utilization rates as low as 5% to 10% [10], [12], while an allocation based on average traffic may result in high latency during flash crowds. Therefore, one of our main focus in this work is to develop robust allocation methods for multimedia CDNs that mitigate the under-utilization and high latency problems of existing allocation methods.

The emergence of cloud computing offers a natural way to extend the capabilities of CDNs to support the growth of multimedia contents. In cloud computing, an organization or individual rents remote server resources dynamically, and users can add or remove server capacity at any time to meet their requirements [13]. In a cloud-centric media network (CCMN) architecture [14]–[16], virtual machines (VMs) are carved out of an underlying hybrid cloud, forming a content distribution overlay. The amount of system resources can be dynamically scaled up and down, matching real-time application demands. The introduction of cloud computing in CCMN poses additional technical challenges in its operations. One particular example of importance is the user-redirection mechanism, which should take into consideration the new resource allocation paradigm under cloud computing.

In this paper, we address the problem of dynamically scaling the request allocation rates and service capacities of the VMs, where the request allocation rate of a VM is the number of transaction requests allocated by a dispatcher to the VM per unit time, and the service capacity of a VM is the number of transactions per unit time completed by the VM, typically measured in transactions per second (TPS) [17]. In order to mitigate the under-utilization and high latency problems alluded to earlier, we adopt the quickest detection framework of [18] to detect the onset of flash crowds, and

J. Tang and W. P. Tay are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Email: {jtang4, wptay}@ntu.edu.sg.

Y. G. Wen is with the School of Computer Engineering, Nanyang Technological University, Singapore. Email: ygwen@ntu.edu.sg.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2014.2308726, IEEE Transactions on Multimedia

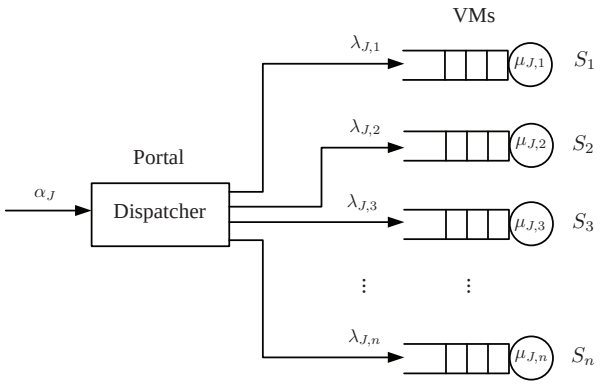


Fig. 1. Request redirection model.

derive optimal allocation and service capacity policies that take into account detection delays as well as false alarms. Our main contributions are the following:

- We develop a systematic framework for dynamic request allocation and service capacity scaling in a CCMN based on quickest detection of changes in the arrival traffic mode, which switches between normal and flash crowd modes depending on the traffic arrival rate. We consider an elastic cost model in which the service capacity of a VM can be elastically scaled to model a cloud computing environment.
- We derive an optimal traffic mode change detection strategy, the optimal request allocation policy, and the optimal service capacity scaling policy under each traffic mode by introducing the concept of a *VM price* for each VM. The price of a VM depends only on the service cost function of the VM itself, and determines the desirability of the VM to be allocated requests. We also show that there exists a *threshold price*, which depends on the traffic arrival rate and the VM service costs. We interpret this threshold price as the maximum price the dispatcher is willing to pay for any VM. We show that the optimal VM selection policy consists of choosing only those VMs with a VM price less than the threshold price.
- We provide simulations that suggest that our proposed dynamic allocation and service capacity scaling mechanism outperforms other existing allocation methods when the arrival request traffic has both normal and flash crowd modes. We test all strategies on a real world YouTube request data set, and we also use extensive simulations to verify their performances when various arrival traffic characteristics are varied.

In the following, we first briefly review prior works that are related to this paper, and summarize some of the notations that are commonly used in the paper.

A. Related Works

In this paper, we develop cost-aware optimal request allocation and service capacity scaling mechanisms in a CCMN under switching arrival traffic modes. In a user request allocation system (cf. Figure 1), a dispatcher aims to redirect incoming service requests to multiple VMs, while each VM receiving requests scales its service capacity in order to jointly minimize a cost function. Here, we assume that the arrival request process consists of a stream of transactions, each of which is to be allocated to a single VM. This reflects practical architectures like *MapReduce*, in which multiple VMs are allocated transactions by the *Map* function at the dispatcher.

The topic of user request allocation or load balancing has been well studied in the literature (see [19], [20] for surveys on this topic). We summarize some of the most relevant works below.

The earliest works on request allocation focus on servers with known static service capacities. Several mechanisms including random balancing and weighted round robin methods [21]–[23] were proposed. Subsequently, server loads [24], [25] and response times [26] are taken into consideration when choosing the servers for redirection, leading to lower service response times. Due to the increasing complexity and heterogeneity of CDNs in terms of available computing, bandwidth and memory resources, several works have proposed request allocation methods based on octree partitioning [27], directed graph models [28], and multilevel graph models [29]. In addition, [30] propose load balancing by redirecting the requests to locations with the cheapest energy or by maximizing energy reduction and minimizing the effect on client-perceived Service Level Agreements (SLAs).

All the above works address the issue of request redirection in a traditional CDN, where the servers have fixed service capacities. In contrast, in a CCMN, the VMs can elastically scale their service capacities to match the packet arrival rates, with the user paying a higher cost for a higher service capacity. In [31], the request allocation problem is considered for a data center, in which a master server splits its request arrivals to all available computing servers, each with a different rate. The authors consider a linear computing cost for each computing server, and their goal is to optimize the service capacity of all servers to minimize the system response time. In our paper, we consider a different request redirection model that is somewhat simpler but more suited to a CDN. Specifically, we assume that there is a dispatcher that splits arrival requests to multiple VMs optimally. Our goal is to minimize a weighted cost function consisting of the system response time, VM computing costs and routing costs. Furthermore, we aim to optimize not only the service capacities but also the allocation rates to each VM. In contrast to [31], which fix the allocation rates in advance, we obtain the somewhat surprising conclusion that not all available VMs should be utilized.

The reference [32] studies the interaction of service capacity scaling with load balancing, which is defined in the strict sense that a performance metric (like mean response time) at all servers with allocated requests have the same value. Equilibrium load balancing and service capacity scaling are derived. Our work in this paper differs from [32] in the following ways. Firstly, we derive optimal allocation rates and service capacities to minimize a cost function similar to [32] but without the strict load balancing constraint. This models a CCMN better since a CCMN is designed for elastic rate allocation without the need to maintain the same

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2014.2308726, IEEE Transactions on Multimedia

equilibrium performance at each VM. Secondly, we do not restrict to monomial power cost functions at the VMs. Instead, we consider a general class of cost functions that is non-decreasing and convex in the service capacity (cf. Section II for examples). This allows us to include various costs like computation, power, storage, and routing costs. Lastly, we allow the arrival traffic to switch between different modes to better model usage patterns in a CCMN, whereas [32] assumes that the arrival traffic statistics are fixed.

Minimizing content retrieval latency using caching and edge servers has been studied in [33], while [34] investigates the joint problem of replica placement and building distribution paths in cloud CDNs. Although finding optimal distribution paths and replica placements are topics of importance and related to the problem of user request redirection, these are not within the scope of this paper. Instead we assume that the redirection destinations have been fixed, and costs like the routing cost and computing cost function at each of these destinations are known. The joint problem of content placement and request redirection has been considered in [35] under various constraints. However, the proposed algorithm is an integer linear program that has high complexity and is sensitive to variations in request traffic, i.e., a new optimization is required at each time step. It is still an open problem to find good approximations to the joint optimization problem that yield reasonable sub-optimal solutions.

B. Notations

Throughout this paper, for each function $f : \mathbb{R} \mapsto \mathbb{R}$, we use $\partial^- f(x)$ and $\partial^+ f(x)$ to denote the left and right derivatives of $f(x)$ with respect to (w.r.t.) x, respectively. For a convex function f, the subdifferential at x is the set $[\partial^- f(x), \partial^+ f(x)]$. We also use the notation $x^+ = \max(0, x)$. For the readers' convenience, we summarize some commonly used notations (which are formally defined in the sequel where they first appear) in Table I.

The rest of this paper is organized as follows. In Section II, we present our system model and problem formulation. We describe the alternating optimization approach to decompose our optimization problem into several sub-problems in section III. In Section IV, we consider the request redirection and service scaling sub-problem and present analytical solutions to it. In Section V, we present simulation results to verify the performance of our proposed strategy. Finally, we conclude the paper in Section VI.

II. Problem Formulation

In this section, we describe our model and system setup, define some notations, and state our assumptions. We consider the problem of redirecting an arrival request process at a dispatcher or load balancer in a CCMN (cf. Figure 1), which tends to experience sporadic bursts of user traffic [7]. Since Poisson processes have been widely used to model request arrivals under both normal and flash crowd traffic conditions [36], [37], we assume that the arrival process is a Poisson process that switches between a normal mode (denoted as N) with arrival rate α_N and a flash crowd mode (denoted as F) with arrival rate $\alpha_F > \alpha_N$ at unknown times [12], [36].

The dispatcher splits the arrival requests into independent sub-processes, which are redirected to a *maximum* of $n > 0$ destination VMs $S_1, \ldots, S_n$. Suppose that the dispatcher has determined that the arrival process is in mode J, where $J \in \{N, F\}$. Let $\lambda_{J,i}$ be the *allocation rate* or the rate of the redirected Poisson arrival process at VM S_i in traffic mode J. A VM is said to be *active* in mode J if the allocation rate $\lambda_{J,i} > 0$ (i.e., the dispatcher may choose to use less than n VMs). Let $\mu_{J,i}$ be the *service capacity* provided by S_i, i.e., the number of arrival transactions that can be serviced by the VM per unit time. Similar to assumptions commonly adopted in the CDN and multimedia cloud literature [31], [38], we assume that the service time is an exponential distribution with rate $\mu_{J,i}$. Each VM in Figure 1 is therefore modeled as an $M/M/1$ queue. To ensure stability, we require that $\lambda_{J,i} < \mu_{J,i}$ for all i and J. Let $\boldsymbol{\lambda}_J = (\lambda_{J,1}, \ldots, \lambda_{J,n})$ and $\boldsymbol{\mu}_J = (\mu_{J,1}, \ldots, \mu_{J,n})$ be vectors of the allocation rates and service capacities respectively under traffic mode J. A pair $(\boldsymbol{\lambda}_J, \boldsymbol{\mu}_J)$ is called a *policy*.

A redirection to VM S_i incurs a *service cost* $\varphi_i(\mu_{J,i}) = s_i(\mu_{J,i}) + c_i$ for each request, where c_i is the unit routing cost, and $s_i(\cdot)$ is the computing cost. For simplicity, we assume that the total routing cost $\lambda_{J,i} c_i$ scales linearly with the allocation rate $\lambda_{J,i}$. The computing cost $s_i(\cdot)$ at each VM is non-decreasing in the service capacity. This models a cloud infrastructure where one or more VMs are initiated for each application, and a larger cost is incurred if more computing resources are requested by each VM. The computing cost includes the cost of the power used by S_i, and the memory or storage costs required. The power cost is typically assumed to be monomial in the service capacity [32], but we do not restrict to such cost functions in this paper. Instead, we make the following assumptions regarding the computing cost.

Assumption 1: For each VM S_i, $i = 1, \ldots, n$, the computing cost $s_i(\cdot)$ has the following properties:

(1) $s_i(\mu) \geq 0$ for all $\mu \geq 0$,
(2) $s_i(\mu)$ is a convex and non-decreasing function of μ.

Assumption 1 models many computing costs of interest in cloud architectures. To the best of our knowledge, all existing commercial cloud platforms like Amazon EC2, Rackspace, Google App Engine and Microsoft Windows Azure, charge flat hourly prices for a fixed amount of resources or each type of instance. An example is the Amazon EC2 Standard On-demand Instance pricing model [39]. Under this plan, four different types of VM instances are available (see Table II). These are called the Small, Medium, Large, and Extra Large instances, with 1, 2, 4, and 8 EC2 Compute Units respectively. Each EC2 Compute Unit has a fixed service capacity of $\bar{\mu}$ transactions per hour, and the price of each unit is $p_{\text{EC2}} = 0.06/\bar{\mu}$ USD[1] per transaction. The cost of an instance S_i can then be modeled by

$$s_i(\mu) = \begin{cases} e_i p_{\text{EC2}}, & \text{if } \mu \leq e_i \bar{\mu}, \\ \infty, & \text{if } \mu > e_i \bar{\mu}, \end{cases} \tag{1}$$

[1]USD is abbreviation for United States dollar.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2014.2308726, IEEE Transactions on Multimedia

TABLE I
SUMMARY OF COMMON NOTATIONS.

Symbol	Definition
α_J	traffic arrival rate to the dispatcher in mode J, $J \in \{N, F\}$
$\lambda_{J,i}$	request allocation rate to VM S_i in traffic mode J
$\mu_{J,i}$	service capacity of VM S_i in traffic mode J
π	average fraction of time that the arrival traffic is in normal mode N
n	maximum number of VMs available to the dispatcher
$s_i(\cdot)$	computing cost of VM S_i, as a function of its service capacity
c_i	unit routing cost to VM S_i
$\varphi_i(\cdot)$	$= s_i(\cdot) + c_i$, service cost of VM S_i
β	weight for the total expected service cost incurred
$\mathcal{T}$	set of stopping times based on the number of request arrivals in each time interval
T_N	stopping time for detecting the switch of arrival traffic from mode N to mode F
T_F	stopping time for detecting the switch of arrival traffic from mode F to mode N
$\delta(T)$	worst case detection delay of stopping time T
$\theta(T)$	mean time between false alarms of stopping time T
c_F	average additional cost incurred per request arrival due to detection delays by T_N or false alarms by T_F
$v(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F)$	average additional cost per unit time due to false alarms by T_N or detection delays by T_F
l_J	mean time duration for each continuous time period that the arrival traffic is in mode J
t_f	maximum amount of time within which false alarms are resolved
p_i	VM price of VM S_i, see (16)

TABLE II
AMAZON EC2 INSTANCES AND PRICES

Name	No. of EC2 Compute Units	Price
Small	1	0.06 USD/hour
Medium	2	0.12 USD/hour
Large	4	0.24 USD/hour
Extra Large	8	0.48 USD/hour

where $e_i = 1, 2, 4$, or 8 depending on the type of instance that S_i belongs to. This cost can be regarded as the "computing cost" if a dispatcher is redirecting application requests to Amazon EC2 Standard On-demand instances, and it clearly satisfies Assumption 1.

The Amazon EC2 pricing model and other similar commercial cloud computing pricing models are designed with retail and enterprise consumers in mind, where lower level costs like power consumption and memory storage costs are transparent to a user, and included in a single price p_{EC2}. For the cloud service providers, to optimize a data center, the computing cost can be more appropriately modeled as $s_i(\mu) = k_i \mu^{a_i}$, where $k_i > 0$ and $a_i > 1$ are positive constants. This cost corresponds to the power used by the VM when its service capacity is μ [40], and is adopted by [32]. In addition to the cost of power consumption, we can also include the cost of processor and storage memory by letting $s_i(\mu) = k_i \mu^{a_i} + r_i \mu$, where $r_i > 0$, and we have assumed that memory or storage cost scales linearly with service capacity [35]. (Such general cost functions are not considered in [32].) Furthermore, in practical systems, the maximum service capacity of any VM is often limited; a fact that is not captured in smooth cost functions. Our model allows us to define cost functions similar to (1), where an infinite cost is incurred when the service capacity exceeds the maximum VM capacity.

Assumption 1 implies that there is a trade off between the average mean response time [41] of the VMs given by

$$\sum_{i=1}^{n} \frac{\lambda_{J,i}}{\alpha_J} \frac{1}{\mu_{J,i} - \lambda_{J,i}},$$

and the total expected service cost incurred, given by

$$\sum_{i=1}^{n} \frac{\lambda_{J,i}}{\alpha_J} \varphi_i(\mu_{J,i}).$$

Let $\beta > 0$ be a fixed weight and for $J \in \{N, F\}$, $\boldsymbol{\lambda}_J = (\lambda_{J,1}, \ldots, \lambda_{J,n})$, $\boldsymbol{\mu}_J = (\mu_{J,1}, \ldots, \mu_{J,n})$, and $\alpha_J > 0$, let

$$f(\boldsymbol{\lambda}_J, \boldsymbol{\mu}_J, \alpha_J) = \sum_{i=1}^{n} \frac{\lambda_{J,i}}{\alpha_J} \frac{1}{\mu_{J,i} - \lambda_{J,i}} + \beta \sum_{i=1}^{n} \frac{\lambda_{J,i}}{\alpha_J} \varphi_i(\mu_{J,i}). \tag{2}$$

We are interested to find policies that minimize a weighted average of $f(\boldsymbol{\lambda}_N, \boldsymbol{\mu}_N, \alpha_N)$ and $f(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F, \alpha_F)$, while taking into consideration the switching traffic modes. To do this, we divide time into equal unit intervals, and suppose that $Z_1, \ldots, Z_{\nu-1}$ are the number of arrivals in each interval when the arrival process is in mode N, and $Z_\nu, Z_{\nu+1}, \ldots$ are the number of arrivals in each interval when the arrival process has switched to mode F. The random variables Z_i, for $i = 1, \ldots, \nu-1$ are independent and identically distributed (i.i.d.) Poisson random variables with rate α_N, while Z_i, for $i \geq \nu$ are i.i.d. Poisson random variables with rate α_F. The index ν is called a change point, and can take values in $[1, \infty]$, with $\nu = \infty$ corresponding to the case where the arrival

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2014.2308726, IEEE Transactions on Multimedia

process stays in mode N throughout. As ν is unknown, it is inferred from the observations $Z_1, Z_2, \ldots$. Let $\mathcal{T}$ be the set of stopping times associated with $Z_1, Z_2, \ldots$ [18]. Each element of $\mathcal{T}$ is a detection policy that can be used by the dispatcher to determine if the arrival process has switched from mode N to mode F. Any detection policy $T \in \mathcal{T}$ has a trade-off in the worst case detection delay [18], [42]

$$\delta(T) = \sup_{t \geq 1} \operatorname{ess\,sup} \mathbb{E}_t[(T - t + 1)^+ \mid Z_1, \ldots, Z_{t-1}],$$

where $\mathbb{E}_t$ is the expectation operator when $\nu = t$, with its mean time between false alarms defined as

$$\theta(T) = \mathbb{E}_\infty[T].$$

One can similarly define the worst case detection delay and mean time between false alarms for the case where the arrival traffic switches from mode F to mode N.

Our goal is to find allocation rates $\boldsymbol{\lambda}_J$ and service capacities $\boldsymbol{\mu}_J$, for $J \in \{N, F\}$, that are optimal in the sense that they minimize a cost function that takes into account the trade-offs in performance at each VM, the service cost incurred at each VM, and the costs incurred due to detection delays and false alarms. Suppose that the arrival traffic switches from mode N to mode F. When there is a delay in detecting the change in the arrival traffic mode, there will be a transient increase in the buffer length at each VM as the underlying arrival rate has increased to α_F. This incurs additional memory or storage cost and an increase in the service response time. We assume that the average additional cost per unit time is given by $c_F(\alpha_F - \alpha_N)$, where c_F is a positive constant that can be interpreted as the average additional cost incurred per request arrival. On the other hand, if a false alarm occurs, the system adopts the policy $(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F)$ even though the policy $(\boldsymbol{\lambda}_N, \boldsymbol{\mu}_N)$ is optimal. This results in higher computing costs at the VMs. We assume that false alarms are resolved within a fixed bounded time period t_f, with the average additional cost per unit time given by a penalty function $v(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F)$ incurred at the active VMs. A similar consideration can be made when the arrival traffic switches from mode F to mode N. For simplicity, we assume that there is an ergodic stochastic process governing the traffic modes. For example, the sequence of traffic modes may be modeled as a continuous time Markov process, with transitions between modes N and F. We do not require full knowledge of this underlying process, except that the average fraction of time that the arrival process is in normal mode N is given by $\pi \in (0, 1)$, which can be estimated from historical data. In Section V, we present simulation results to show how the performance of our proposed algorithm is impacted by the value of π.

Let T_N and T_F be stopping times for detecting the switch of arrival traffic from mode N to mode F, and vice versa, respectively. Consider a long interval consisting of l request arrivals, and let K_J be the set of requests at which the arrival process is in mode J, for $J \in \{N, F\}$. Let D_N and D_F be the set of change points at which the arrival traffic switches from mode N to mode F, and vice versa, respectively. For each $J \in \{N, F\}$ and $i \in D_J$, let d_i be the length of the detection delay incurred by the stopping time T_J. We approximate the detection delays d_i to be independent and identically distributed for each J. Finally, let F_J be the set of arrivals at which T_J incurs false alarms. Assuming that VMs' response times reach equilibrium and are independent of the length of the traffic mode, the average equilibrium cost can be approximated as

$$\begin{aligned} &\frac{|K_N|}{l} f(\boldsymbol{\lambda}_N, \boldsymbol{\mu}_N, \alpha_N) + \frac{|K_F|}{l} f(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F, \alpha_F) \\ &+ \frac{1}{l} \sum_{i \in D_N} d_i c_F(\alpha_F - \alpha_N) + \frac{1}{l} \sum_{i \in D_F} d_i v(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F) \\ &+ \frac{|F_N|}{l} t_f v(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F) + \frac{|F_F|}{l} t_f c_F(\alpha_F - \alpha_N). \end{aligned} \tag{3}$$

For $J \in \{N, F\}$, let l_J be the mean time duration for each continuous time period that the arrival traffic is in mode J. Letting $l \to \infty$, we see that (3) can be approximately upper bounded by (4) below. Our optimization problem is then formulated as

$$\begin{aligned} \text{minimize} \quad & \pi f(\boldsymbol{\lambda}_N, \boldsymbol{\mu}_N, \alpha_N) + (1 - \pi) f(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F, \alpha_F) \\ & + c_F(\alpha_F - \alpha_N) \frac{(1 - \pi)}{\alpha_F} \left(\frac{\delta(T_N)}{l_F} + \frac{t_f}{\theta(T_F)} \right) \\ & + v(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F) \frac{\pi}{\alpha_N} \left(\frac{\delta(T_F)}{l_N} + \frac{t_f}{\theta(T_N)} \right), \\ \text{such that} \quad & \sum_{i=1}^{n} \lambda_{J,i} = \alpha_J, \text{ for } J \in \{N, F\}, \\ & 0 \leq \lambda_{J,i} < \mu_{J,i}, \text{ for } 1 \leq i \leq n, \; J \in \{N, F\}, \\ & T_N, T_F \in \mathcal{T}. \end{aligned} \tag{4}$$

In this paper, we suppose that the penalty function $v(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F)$ is separable across VMs and satisfies the following assumptions.

Assumption 2: Let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_n)$ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$.

(i) The penalty function $v(\boldsymbol{\lambda}, \boldsymbol{\mu})$ has the form

$$v(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{i=1}^{n} \lambda_i v_i(\mu_i) - \xi,$$

where ξ is a positive constant, and for each $i = 1, \ldots, n$, $v_i(\cdot)$ is a non-negative, convex, and non-decreasing function.

(ii) We have $\min v(\boldsymbol{\lambda}, \boldsymbol{\mu}) \geq 0$, where the minimization is over all $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ such that $0 \leq \lambda_i < \mu_i$ for all $i = 1, \ldots, n$, and $\sum_{i=1}^{n} \lambda_i = \alpha_F$.

Assumption 2 covers many, but not all, practical cases of interest. For example, in designing a CCMN system, we can interpret the constant ξ in Assumption 2 as the average cost (storage, service response time, computing cost etc.) incurred per unit time when adopting the policy $(\boldsymbol{\lambda}_N^*, \boldsymbol{\mu}_N^*) = \arg\min f(\boldsymbol{\lambda}_N, \boldsymbol{\mu}_N, \alpha_N)$ during normal traffic mode, while the function $v_i(\mu)$ is the average cost incurred when VM S_i provides a service capacity of μ. Assumption 2(ii) ensures that the penalty function is always positive over all feasible allocation rates and service capacities, which is the case for a practical system. We note that Assumption 2 does not cover the most general case where the penalty function is of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2014.2308726, IEEE Transactions on Multimedia

form $v(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F, \boldsymbol{\lambda}_N, \boldsymbol{\mu}_N)$, which however introduces technical difficulties into the solution of (4) and its interpretation. In Section V, we provide simulation results that suggest that Assumption 2 does not significantly impact the performance of the policies derived from (4), compared to the "perfect" strategy that knows the exact points in time when the traffic switches its mode.

A. Approximation for Detection Delay and False Alarm Frequency

In this section, we briefly review the theory of quickest detection in a non-Bayesian setting, and derive approximations to the change detection delay and mean time between false alarms, from which an approximation to the optimization problem (4) is then obtained.

The problem of quickest detection is to optimally detect a change in the underlying distribution from which a sequence of observations $Z_1, Z_2, \ldots$ is drawn from, subject to certain false alarm constraints. The observations are drawn i.i.d. from distributions Q_0 and Q_1 before and after an unknown change point ν respectively. Since at each time t, we only have access to the previously observed random variables Z_k, for $k \leq t$, the change detection policy is a stopping time $T \in \mathcal{T}$.

A commonly used stopping time is Page's CUSUM test [18], [43] given by

$$T_\sigma = \inf\{k \geq 0 : \max_{1 \leq j \leq k} \sum_{l=j}^{k} L(Z_l) \geq \log \sigma\}, \tag{5}$$

where $L(z) = \log \frac{dQ_1}{dQ_0}(z)$ is the log-likelihood ratio of Q_0 w.r.t. Q_1. It is well known that Page's test is an optimal change detection policy in the sense that for any σ, the test T_σ minimizes the detection delay $\delta(T)$ among all stopping times T satisfying $\theta(T) \geq \theta(T_\sigma)$. The following result follows directly from the optimality of Page's CUSUM test, and Wald's approximations [18], and its proof is omitted.

Proposition 1: Suppose A and B are positive constants, and $L(z)$ has no atoms under Q_0. There exists a threshold $\sigma > 0$ such that an optimal solution to the following optimization problem

$$\begin{aligned} \text{minimize} \quad & A\delta(T) + \frac{B}{\theta(T)} \\ \text{such that} \quad & T \in \mathcal{T}, \end{aligned} \tag{6}$$

has the form (5). Furthermore, if $\sigma \geq 1$, then the optimal detection delay and the mean time between false alarms can be approximated as

$$\delta(T_\sigma) \cong \frac{1}{M_1}\left(\frac{1}{\sigma} - 1 + \log \sigma\right), \tag{7}$$

$$\theta(T_\sigma) \cong \frac{1}{M_0}(\sigma - \log \sigma - 1), \tag{8}$$

and the optimal threshold σ can be approximated as the solution to

$$A(\sigma - \log \sigma - 1)^2 = BM_0M_1\sigma, \tag{9}$$

where for $i = 0, 1$, $M_i = |\mathbb{E}_{Q_i}[L(Z)]|$, and $\mathbb{E}_{Q_i}$ is the expectation operator under the probability distribution Q_i.

We now return to the optimization problem in (4). For any fixed rates $\{\boldsymbol{\lambda}_J, \boldsymbol{\mu}_J \mid j \in \{N, F\}\}$, finding the optimal stopping time T_N is equivalent to solving the optimization problem (6), with $A = (1-\pi)c_F(\alpha_F - \alpha_N)/(\alpha_F l_F)$ and $B = \pi t_f v(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F)/\alpha_N$, while to find the optimal T_F, we set $A = \pi v(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F)/(\alpha_N l_N)$ and $B = (1-\pi)c_F t_f(\alpha_F - \alpha_N)/\alpha_F$. To simplify the mathematics, we use the approximations in Proposition 1 to arrive at the following approximation for the objective function in (4),

$$\begin{aligned} &\mathbf{Q}(\boldsymbol{\lambda}_N, \boldsymbol{\mu}_N, \boldsymbol{\lambda}_F, \boldsymbol{\mu}_F, \sigma_N, \sigma_F) \\ &= \pi f(\boldsymbol{\lambda}_N, \boldsymbol{\mu}_N, \alpha_N) + (1-\pi) f(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F, \alpha_F) \\ &\quad + c_F(\alpha_F - \alpha_N)\frac{(1-\pi)}{\alpha_F}\left(\frac{1}{l_F M_F}\left(\frac{1}{\sigma_N} - 1 + \log \sigma_N\right)\right. \\ &\qquad\qquad \left. + \frac{t_f M_F}{\sigma_F - \log \sigma_F - 1}\right) \\ &\quad + v(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F)\frac{\pi}{\alpha_N}\left(\frac{1}{l_N M_N}\left(\frac{1}{\sigma_F} - 1 + \log \sigma_F\right)\right. \\ &\qquad\qquad \left. + \frac{t_f M_N}{\sigma_N - \log \sigma_N - 1}\right), \end{aligned} \tag{10}$$

where for $J \in \{N, F\}$, σ_J is the threshold corresponding to T_J and $M_J = |\alpha_J \log(\alpha_N/\alpha_F) + (\alpha_F - \alpha_N)|$.

With this approximation, in the rest of this paper, we focus on obtaining the solution to the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \mathbf{Q}(\boldsymbol{\lambda}_N, \boldsymbol{\mu}_N, \boldsymbol{\lambda}_F, \boldsymbol{\mu}_F, \sigma_N, \sigma_F), \\ \text{such that} \quad & \sum_{i=1}^{n} \lambda_{J,i} = \alpha_J, \text{ for } J \in \{N, F\}, \\ & 0 \leq \lambda_{J,i} < \mu_{J,i}, \text{ for } 1 \leq i \leq n,\ J \in \{N, F\}, \\ & \sigma_J \geq 1, \text{ for } J \in \{N, F\}. \end{aligned} \tag{11}$$

We note that to solve (11), the parameters $\pi, \alpha_N, \alpha_F, l_N$, and l_F need to be first estimated from historical data (using for example, the Maximum Likelihood Estimator (MLE) approach [44]), while the remaining parameters c_F and t_f, and penalty function $v(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F)$ need to be chosen appropriately depending on the application. We show an example in Section V.

III. Alternating Optimization

In this section, we apply the alternating optimization method to simplify the optimization problem (11). We partition the variables in (11) into three subsets: $(\boldsymbol{\lambda}_N, \boldsymbol{\mu}_N)$, $(\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F)$ and the thresholds (σ_N, σ_F). We start with a random initial guess $(\boldsymbol{\lambda}_N(0), \boldsymbol{\mu}_N(0), \boldsymbol{\lambda}_F(0), \boldsymbol{\mu}_F(0), \sigma_N(0), \sigma_F(0))$ for the optimization variables respectively. At each iteration t, we perform the following series of optimizations,

$$\begin{aligned} (\boldsymbol{\lambda}_N(t), \boldsymbol{\mu}_N(t)) = \arg \min_{\boldsymbol{\lambda}_N, \boldsymbol{\mu}_N} \mathbf{Q}(\boldsymbol{\lambda}_N, \boldsymbol{\mu}_N, \boldsymbol{\lambda}_F(t-1), \boldsymbol{\mu}_F(t-1), \\ \sigma_N(t-1), \sigma_F(t-1)) \end{aligned} \tag{12}$$

$$\begin{aligned} (\boldsymbol{\lambda}_F(t), \boldsymbol{\mu}_F(t)) = \arg \min_{\boldsymbol{\lambda}_F, \boldsymbol{\mu}_F} \mathbf{Q}(\boldsymbol{\lambda}_N(t), \boldsymbol{\mu}_N(t), \boldsymbol{\lambda}_F, \boldsymbol{\mu}_F, \\ \sigma_N(t-1), \sigma_F(t-1)) \end{aligned} \tag{13}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2014.2308726, IEEE Transactions on Multimedia

$$(\sigma_N(t),\sigma_F(t)) = \arg\min_{\sigma_N,\sigma_F} \mathbf{Q}(\boldsymbol{\lambda}_N(t),\boldsymbol{\mu}_N(t),\boldsymbol{\lambda}_F(t),\boldsymbol{\mu}_F(t), \sigma_N,\sigma_F) \quad (14)$$

where the minimizations in (12) and (13) are over all $\boldsymbol{\lambda}_J$ and $\boldsymbol{\mu}_J$ such that

$$\sum_{i=1}^{n} \lambda_{J,i} = \alpha_J,$$

and

$$0 \le \lambda_{J,i} < \mu_{J,i}, \text{ for } 1 \le i \le n,$$

for $J = N$ and $J = F$ respectively, and the minimization in (14) is over all $\sigma \ge 1$. The minimization in (14) can be computed by solving (9) with the appropriate values of A and B substituted in. In the next section, we derive the solutions to the minimization problems (12) and (13).

Since $\mathbf{Q}(\boldsymbol{\lambda}_N,\boldsymbol{\mu}_N,\boldsymbol{\lambda}_F,\boldsymbol{\mu}_F,\sigma_N,\sigma_F) \ge 0$ and the objective function value is non-increasing at each iteration of the alternating optimization procedure, the iterates converge to a local minimum of $\mathbf{Q}(\cdot)$ as $t \to \infty$. To increase the chance of finding the global minimum, the procedure can be applied to several random initial guesses.

IV. Optimal Request Allocation and Service Capacity

In this section, we derive optimal allocation rates and service capacities for the problems (12) and (13), which correspond to policies the dispatcher adopts after it has determined that the arrival process is in mode N and F respectively. To simplify notations, we drop the iteration index t in the alternating optimization procedure in this section. The problems (12) and (13) are both equivalent to the following optimization problem,

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} \frac{\lambda_i}{\alpha} \frac{1}{\mu_i - \lambda_i} + \beta \sum_{i=1}^{n} \frac{\lambda_i}{\alpha} w_i(\mu_i), \quad (15) \\ \text{such that} \quad & \sum_{i=1}^{n} \lambda_i = \alpha, \\ & 0 \le \lambda_i < \mu_i, \text{ for } 1 \le i \le n, \end{aligned}$$

where we let $\alpha = \alpha_J$, $\lambda_i = \lambda_{J,i}$ and $\mu_i = \mu_{J,i}$, with $J = N$ and F for (12) and (13) respectively. In addition, we let $w_i(\mu) = \varphi_i(\mu)$ and

$$w_i(\mu) = \varphi_i(\mu) + \frac{\pi\alpha_F v_i(\mu)}{(1-\pi)\beta\alpha_N} \Big(\frac{1/\sigma_F - 1 + \log\sigma_F}{l_N M_N} + \frac{t_f M_N}{\sigma_N - \log\sigma_N - 1} \Big),$$

for (12) and (13) respectively. We derive the general form of the optimal solution for (15), which is applicable for both (12) and (13).

A. VM Prices

The form of the optimal solution to (15) is closely related to the price of each VM, which for $i = 1,\ldots,n$, we define as

$$p_i = \min_{\mu>0} \left\{ \frac{1}{\mu} + \beta w_i(\mu) \right\}. \quad (16)$$

Suppose that the dispatcher has to pay one dollar per unit service time, and β dollars per unit cost (routing and computation), then the price of VM S_i is the expected total price that the dispatcher pays to VM S_i. Interpreted in this way, it is natural that the dispatcher should choose VMs with the lowest prices such that the system is stable. Therefore, it follows that there is a threshold price, below which all VMs with a price cheaper than this threshold will be sent redirection requests. These are the active VMs. Furthermore, the VMs will provide service capacities that depend on this threshold price. We show that this intuitive argument holds in Theorems 1.

For each $i = 1,\ldots,n$, and $p > 0$, let

$$g_i(p) = \sup \left\{ \mu : w_i(\mu) + \mu\partial^- w_i(\mu) \le \frac{p}{\beta} \right\}. \quad (17)$$

Since w_i is convex, we have $\partial^- w_i(x)$ is non-decreasing, and $g_i(\cdot)$ is a non-decreasing function. If the computing cost $s_i(\mu)$ is continuously differentiable over all $\mu \in [0,\infty)$, $g_i(p)$ can be found as the implicit solution to

$$w_i(\mu) + \mu w_i'(\mu) = \frac{p}{\beta},$$

where $w_i'(\mu)$ is the first derivative w.r.t. μ. We make use of $g_i(\cdot)$ in Theorem 1 below to characterize the optimal number of active VMs, service capacities and allocation rates. We first prove a result regarding $g_i(p_i)$.

Lemma 1: For $i = 1,\ldots,n$, let $\tilde{\mu}_i > 0$ be such that $p_i = 1/\tilde{\mu}_i + \beta w_i(\tilde{\mu}_i)$. Then, $\tilde{\mu}_i = g_i(p_i)$.

Proof: Since $\tilde{\mu}_i$ is the minimizer of the right hand side (R.H.S.) in (16), there exists a subgradient d such that

$$\frac{1}{\tilde{\mu}_i} = \beta\tilde{\mu}_i d. \quad (18)$$

Let $\mu = g_i(p_i)$. From (17), we have

$$\beta(w_i(\mu) + \mu\partial^- w_i(\mu)) \le \frac{1}{\tilde{\mu}_i} + \beta w_i(\tilde{\mu}_i),$$

which together with (18) yields

$$w_i(\mu) - w_i(\tilde{\mu}_i) \le \tilde{\mu}_i d - \mu\partial^- w_i(\mu). \quad (19)$$

Suppose that $\mu > \tilde{\mu}_i$. Since $w_i(\cdot)$ is non-decreasing, we have from (19) that $\tilde{\mu}_i d \ge \mu\partial^- w_i(\mu)$. This is a contradiction since $\partial^- w_i(\mu) \ge d$ as $w_i(\cdot)$ is convex. Therefore, we must have $\mu \le \tilde{\mu}_i$. Similarly, from (17) and the convexity of $w_i(\cdot)$, we have

$$\beta(w_i(\mu) + \mu\partial^+ w_i(\mu)) \ge \frac{1}{\tilde{\mu}_i} + \beta w_i(\tilde{\mu}_i),$$

and the same argument as above implies that $\mu \ge \tilde{\mu}_i$. The lemma is now proved. ■

Theorem 1: Suppose that Assumptions 1 and 2 hold, and that $p_1 \le \ldots \le p_n < p_{n+1} = \infty$. Then, the optimal service capacities for the optimization problem (15) are

$$\mu_i^* = \begin{cases} g_i(p), & \text{for } 1 \le i \le m^* \\ 0, & \text{for } m^* < i \le n, \end{cases} \quad (20)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2014.2308726, IEEE Transactions on Multimedia

and the optimal allocation rates are

$$\lambda_i^* = \begin{cases} \mu_i^* - \sqrt{\dfrac{\mu_i^*}{p - \beta w_i(\mu_i^*)}}, & \text{for } 1 \le i \le m^* \\ 0, & \text{for } m^* < i \le n, \end{cases} \tag{21}$$

where

$$m^* = \arg \max_{1 \le k \le n} \left\{ p_k \;\middle|\; \sum_{i=1}^{k} \left(g_i(p_k) - \sqrt{\frac{g_i(p_k)}{p_k - \beta w_i(g_i(p_k))}} \right) < \alpha \right\}, \tag{22}$$

and $p \in (p_{m^*}, p_{m^*+1}]$ is such that $\sum_{i=1}^n \lambda_i^* = \alpha$.

Proof: The Lagrangian for the convex optimization problem (15) is

$$L = \sum_{i=1}^{n} \frac{\lambda_i}{\alpha} \frac{1}{\mu_i - \lambda_i} + \beta \sum_{i=1}^{n} \frac{\lambda_i}{\alpha} w_i(\mu_i) - p \sum_{i=1}^{n} (\frac{\lambda_i}{\alpha} - 1) - \sum_{i=1}^{n} b_i \lambda_i - \sum_{i=1}^{n} h_i(\mu_i - \lambda_i),$$

where p, b_i, h_i are Lagrange multipliers, with $b_i \ge 0$, $h_i \ge 0$ for $i = 1, \cdots, n$. For each $i = 1, \ldots, n$, we obtain from $\partial L / \partial \lambda_i = 0$,

$$\frac{\mu_i^*}{(\mu_i^* - \lambda_i^*)^2} = p + \alpha(b_i - h_i) - \beta w_i(\mu_i^*), \tag{23}$$

and the Karush Kuhn Tucker (KKT) conditions yield

$$\lambda_i^* = \left(\mu_i^* - \sqrt{\frac{\mu_i^*}{p + \alpha b_i - \beta w_i(\mu_i^*)}} \right)^+, \tag{24}$$

where we have set $h_i = 0$.

Since $w_i(\mu_i)$ is a convex function, the KKT conditions for subdifferentiable functions [45] give

$$\begin{cases} \lambda_i^* \left(\dfrac{1}{(\mu_i^* - \lambda_i^*)^2} - \beta \partial^+ w_i(\mu_i^*) \right) \le 0 \\ \lambda_i^* \left(\dfrac{1}{(\mu_i^* - \lambda_i^*)^2} - \beta \partial^- w_i(\mu_i^*) \right) \ge 0. \end{cases} \tag{25}$$

If $\lambda_i^* > 0$, we have $b_i = 0$, and from (24), we obtain

$$\frac{1}{(\mu_i^* - \lambda_i^*)^2} = \frac{p - \beta w_i(\mu_i^*)}{\mu_i^*}. \tag{26}$$

Substituting (26) into (25) , we obtain

$$\beta(w_i(\mu_i^*) + \mu_i^* \partial^- w_i(\mu_i^*)) \le p \le \beta(w_i(\mu_i^*) + \mu_i^* \partial^+ w_i(\mu_i^*)), \tag{27}$$

which implies that $\mu_i^* = g_i(p)$. Observe from (15) that if $\lambda_i^* = 0$ for some i, then the optimal μ_i^* can be chosen to be any non-negative value without changing the objective function value. This implies that there are an infinite number of optimal solutions.[2] We can still take $\mu_i^* = g_i(p)$, let $h_i = 0$ and choose b_i appropriately to satisfy the KKT conditions.

We now show that VM S_i is active only if $p > p_i$, and is inactive only if either $g_i(p) = 0$ or $p \le p_i$. The necessary condition for S_i to be active comes from (24), which implies that if $\lambda_i^* > 0$, we have $b_i = 0$ and

$$p > \frac{1}{\mu_i^*} + \beta w_i(\mu_i^*) \ge p_i.$$

Now suppose that $\lambda_i^* = 0$. From (24) and (27), we have

$$p + b_i \le \frac{1}{g_i(p)} + \beta w_i(g_i(p)) \tag{28}$$

$$\le p - g_i(p) \left(\beta \partial^- w_i(g_i(p)) - \frac{1}{g_i(p)^2} \right). \tag{29}$$

where the last equality follows from (17). Since $b_i \ge 0$, we have either $g_i(p) = 0$ or $\beta \partial^- w_i(g_i(p)) - 1/g_i(p)^2 \le 0$. The second condition is equivalent to $g_i(p) \le \tilde{\mu}_i$, where $\tilde{\mu}_i$ is the unique minimizer of the R.H.S. in (16). From Lemma 1, we have $g_i(p) \le g_i(p_i)$, and since $g_i(\cdot)$ is a non-decreasing function, we obtain $p \le p_i$. This implies that in the optimal policy, the VMs are chosen in non-decreasing order of p_i. The number of active VMs needs to meet the constraint $\sum_{i=1}^n \lambda_i^* = \alpha$, hence (22) holds, and the theorem is proved. ■

From Theorem 1, the allocation rates follow a water-filling solution, with the Lagrange multiplier p serving as a threshold price, and only VMs with prices below this threshold are sent redirection requests. In step (12) and (13) of the alternating optimization procedure, the optimal policy can be found by using the following procedure:

1) The dispatcher sorts $\{p_i : i = 1, \ldots, n\}$ in non-decreasing order.
2) The dispatcher chooses a set of VMs using (22), the optimal allocation rates $\{\lambda_i^*\}_{i=1}^n$ using (21), and computes the price threshold p.
3) The dispatcher sends the price threshold p to the chosen active VMs.
4) Each active VM S_i provides service capacity $g_i(p)$.

The complexity of the first step is $O(n \log n)$. To find the price threshold p and the set of active VMs in the second step, a binary search on the sorted array obtained in the first step produces an interval $(p_{m^*}, p_{m^*+1}]$ containing p, and the optimal number of VMs m^*. This has complexity $O(n \log n)$. Assuming that the prices $\{p_i\}_{i=1}^n$ increases at most exponentially fast, a binary search in the interval $(p_{m^*}, p_{m^*+1}]$ takes $O(n)$ iterations. Therefore, the computation complexity at the dispatcher is $O(n \log n)$ for each iteration of the alternating optimization procedure.

B. Bounded Service Capacity

In this section, we consider the special case where each VM S_i, for $i = 1, \ldots, n$, has bounded service capacity $\bar{\mu}_i$, and $\varphi_i(\mu) = v_i(\mu) = \bar{c}_i$. An example of such a system is given by the Amazon EC2 Standard On-demand Instance plan, as described in Section II. We have the following corollary for solving (12), which follows from Theorem 1. A similar result holds for the problem (13).

Corollary 1: Suppose that Assumptions 1 and 2 hold, and the maximum service capacity at VM S_i is $\bar{\mu}_i$ with constant service cost $\varphi(\mu) = \bar{c}_i$, for $i = 1, \ldots, n$. Suppose further that $p_i = 1/\bar{\mu}_i + \beta \bar{c}_i$, for $i = 1, \ldots, n$, are such that $p_1 \le \ldots \le$

[2]The only physically reasonable solution however corresponds to choosing $\mu_i^* = 0$ when $\lambda^* = 0$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2014.2308726, IEEE Transactions on Multimedia

$p_n < p_{n+1} = \infty$. Then, the optimal number of active VMs for the optimization problem (12) is

$$m^* = \arg \max_{1 \le k \le n} 1 \left\{ p_k \left| \sum_{i=1}^{k} \left(\bar{\mu}_i - \sqrt{\frac{\bar{\mu}_i}{p_k - \beta \bar{c}_i}} \right) < \alpha \right. \right\}. \tag{30}$$

Furthermore, the optimal service capacity for $1 \le i \le m^*$ is $\mu_i^* = \bar{\mu}_i$, and the optimal allocation rates are

$$\lambda_i^* = \begin{cases} \bar{\mu}_i - \sqrt{\frac{\bar{\mu}_i}{p - \beta \bar{c}_i}} & , \quad \text{for } 1 \le i \le m^* \\ 0 & , \quad \text{for } m^* < i \le n, \end{cases} \tag{31}$$

where $p \in (p_{m^*}, p_{m^*+1}]$ is such that $\sum_{i=1}^{n} \lambda_i^* = \alpha$.

V. Simulation Results

In this section, we present simulation results to verify the performance of our proposed request allocation and service capacity scaling policy. We test the performance of our algorithm and various benchmark strategies on real YouTube request data collected by [46], and then on simulated request data in order to verify the impact of traffic arrival characteristics on the algorithms' performance. For ease of reference, we call the solution that we derive for the problem (11) via the alternating optimization method described in Section III the Dynamic Request Redirection and Elastic Service Scaling (DRES) strategy. We use the sum of service delay and service cost weighted by β as our performance criteria (cf. (2)).

We compare the performance of the DRES strategy with that of the following benchmark strategies:

- N *strategy.* This strategy assumes that the arrival traffic is always in the normal traffic mode N and the policy it adopts is obtained by minimizing $f(\boldsymbol{\lambda}, \boldsymbol{\mu}, \alpha_N)$. Let the optimal allocation rates be $\boldsymbol{\lambda}^N$ and the service capacities be $\boldsymbol{\mu}^N$. For $J \in \{N, F\}$, the dispatcher redirects $\alpha_J \lambda_i^N / \alpha_N$ amount of traffic to VM S_i, which uses a service capacity of μ_i^N regardless of the arrival rate at that VM. When the arrival traffic is in mode F, the service capacity μ_i^N may be smaller than the actual traffic arrival rate at S_i, leading to a rapidly increasing service response time.
- F *strategy.* This strategy assumes that the arrival traffic is always in flash crowd mode F by adopting the allocation rates and service capacities obtained when minimizing $f(\boldsymbol{\lambda}, \boldsymbol{\mu}, \alpha_F)$. Suppose that the optimal allocation rates are $\boldsymbol{\lambda}^F$ and the service capacities are $\boldsymbol{\mu}^F$. In this strategy, we can guarantee that $\mu_i^F > \lambda_i^F > \alpha_J \lambda_i^F / \alpha_F$ for all $J \in \{N, F\}$. However, when the arrival traffic is in mode N, the policy used is not optimal, leading to higher service costs.
- *Perfect strategy.* We assume that we can perfectly detect the change points when the arrival traffic switches between mode N and mode F. We adopt the N or F policies when the arrival traffic is in mode N or F respectively. This strategy is unrealizable, and serves as a lower bound in our performance comparisons.

In our simulations, we adopt a cubic computing cost model, with $s_i(\mu) = k_c \mu^3$, where $k_c > 0$ for every $i = 1, \ldots, n$. This cost function has been widely used to model the relationship between energy consumption and service capacities [32], [40]. We let the unit routing cost to VM S_i be $c_i = i c_0$, where $c_0 > 0$ is a constant.

TABLE III
Simulation Parameters

Parameter	Value	Parameter	Estimated
$\bar{c}_F$	100	π	0.497
γ	5	α_N	14.33
k_c	10	α_F	44.89
c_0	2	l_N	269.69 min.
t_f	5 min.	l_F	295.83 min.

To compute the DRES strategy, we let the false alarm penalty function in Assumption 2 be $v_i(\cdot) = \gamma s_i(\cdot)$, for each $i = 1, \ldots, n$, and where $\gamma > 0$ is a constant. We choose $\xi = \alpha_N f(\boldsymbol{\lambda}^N, \boldsymbol{\mu}^N, \alpha_N)$, where $(\boldsymbol{\lambda}^N, \boldsymbol{\mu}^N)$ is the policy adopted by the N strategy. We choose γ to be sufficiently large to ensure that Assumption 2 holds. We also let the average additional cost per request during a flash crowd detection delay be $c_F = \bar{c}_F (f(\boldsymbol{\lambda}^F, \boldsymbol{\mu}^F, \alpha_F) - f(\boldsymbol{\lambda}^N, \boldsymbol{\mu}^N, \alpha_N))$, where $(\boldsymbol{\lambda}^F, \boldsymbol{\mu}^F)$ is the policy in the F strategy. The parameters used are given in Table III, while the remaining parameters π, α_N, α_F, l_N and l_F are estimated from the YouTube trace data.

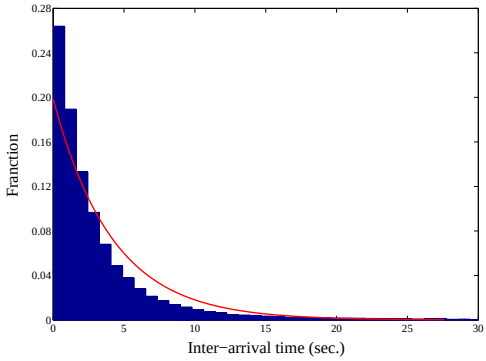
A. YouTube Trace Data

YouTube request trace data was collected by [46] from a campus network over a period of 13 days. A profile of the number of requests per min is shown in Figure 3 for a typical day in the dataset. It is clear from Figure 3 that there are two periods (as indicated by the dotted lines) in which the average traffic arrival rate is significantly higher than that in other periods like the interval [0,400]. In order to define a flash crowd, we first take a running average of the number of requests over 30-minute windows to smooth out the data. Then, we define a flash crowd to be a period of at least 30 minutes, in which the running average traffic arrival rate is not less than 35 requests per minute [36].

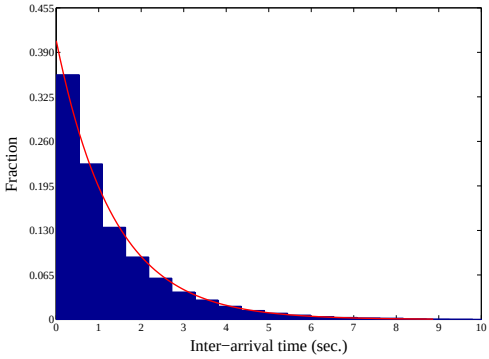
The first 5 days of the YouTube dataset are used for parameter estimation. We find that on most of the days, flash crowds occur twice a day, and each flash crowd has a mean period of $l_F = 295.83$ minutes, while the normal traffic mode has a mean period of $l_N = 269.69$ minutes. We plot the histogram of the inter-arrival times for both normal traffic mode and flash crowd mode for the first 5 days of Youtube trace data in Figure 2. We see that the distribution of the inter-arrival times of the requests approximates the exponential distribution reasonably well, with a better fit for the flash crowd mode than the normal traffic mode. Therefore, it is reasonable to use Poisson arrival processes to model the arrival traffic in both traffic modes. It can then be shown that the MLE for the traffic arrival rate in each traffic mode is given by the average number of requests per minute over the traffic mode period [44]. We estimate that $\alpha_N = 14.33$ per minute and $\alpha_F = 44.89$ per minute. Finally, the fraction of time occupied by the normal traffic mode is given by $\pi = 0.497$.

We first show the objective function value (10) in each iteration of the alternating optimization procedure used to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2014.2308726, IEEE Transactions on Multimedia



(a) Normal traffic mode.



(b) Flash crowd mode.

Fig. 2. Histogram of inter-arrival times under different traffic modes. The red curve shows a fitted exponential distribution.

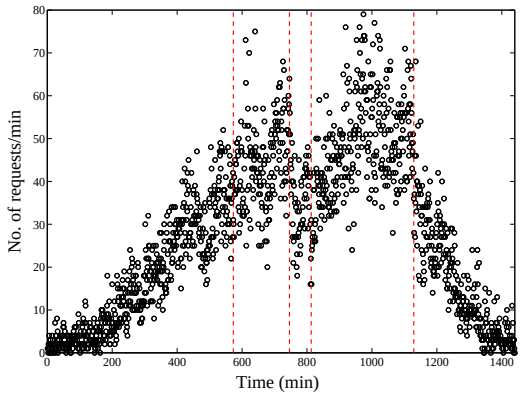


Fig. 3. Rate of Youtube requests over a typical one day period, starting at 3:40am and ending at 3:40am next day. The dotted lines indicate two periods in which the average traffic arrival rate is significantly higher than other times.

compute the optimal policies for the DRES strategy. It can be seen from Figure 4 that the procedure converges in less than 10 iterations.

We next evaluate the performance of the different strategies using arrival request data from the remaining 8 days not used in the parameter estimation. It can be seen from Figure 5 that DRES outperforms the N and F strategies over a wide range of β values. Our simulations suggest that since current request allocation and service capacity strategies implemented

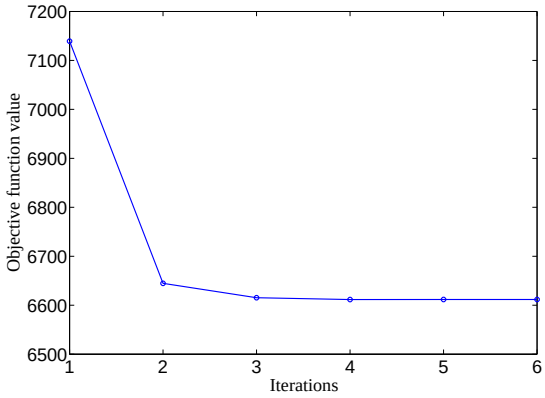


Fig. 4. Convergence of the alternating optimization procedure for computing the DRES strategy.

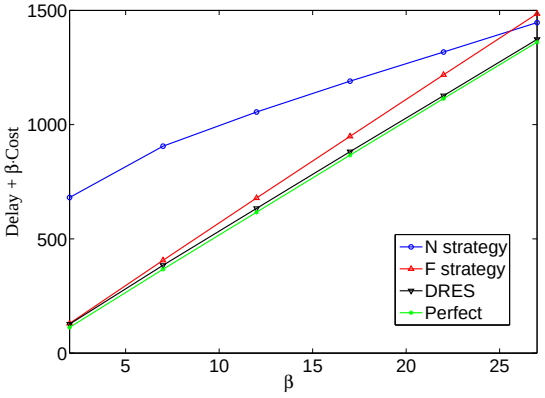


Fig. 5. Performance simulated with YouTube data.

in CDNs are similar to one of the N or F strategies, they are poorly equipped to handle multiple traffic modes, with either an upsurge in service delay during flash crowd traffic mode or an under-utilization of resources in normal traffic mode.

B. Simulated Arrival Requests

In this subsection, we simulate various arrival traffic characteristics, which cannot be tested using the YouTube data set, in order to verify the performance of our proposed algorithm. In each simulation run, we consider an interval of 1440 minutes, and randomly generate two change points that are $1440(1-\pi)$ apart. The arrival traffic in between the two change points correspond to flash crowd traffic, and is generated using a Poisson process with rate α_F. The arrival traffic in the rest of the interval is in normal mode, and is generated using a Poisson process with rate α_N.

We let $\beta = 25$, and use the same parameters $\pi = 0.497$ and $\alpha_N = 14.33$ estimated from the YouTube data set in our simulations, while α_F is varied to simulate different arrival rate ratios α_F/α_N. In Figure 6, we show the performance when we estimate the value of π to be $\hat{\pi} = 0.2, 0.497$ or 0.8 in the DRES strategy. It can be seen that the DRES strategy out-performs the N and F strategies for all arrival rate ratios, even when $\hat{\pi} \neq \pi$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2014.2308726, IEEE Transactions on Multimedia

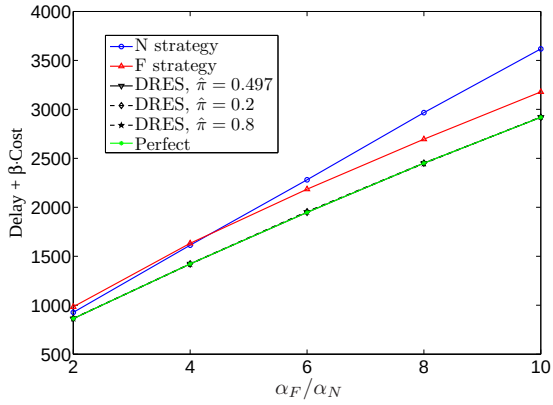


Fig. 6. Performance comparison under different arrival rate ratios.

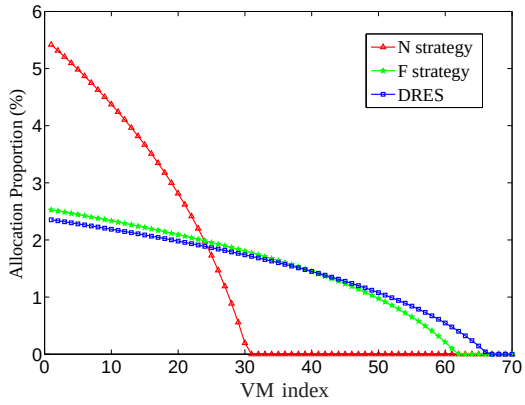


Fig. 8. Allocation proportions for different strategies in the flash crowd mode.

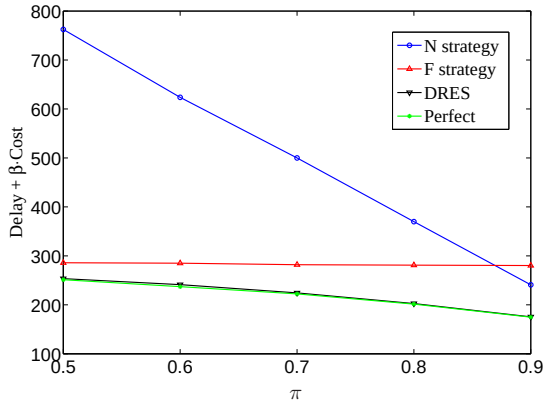


Fig. 7. Performance comparison when π changes.

Next, we fix the arrival rates α_N and α_F to be those estimated from the YouTube data set and let $\beta = 5$, but vary the proportion π of normal mode traffic. It can be seen from Figure 7 that DRES again outperforms the N and F strategies. In Figure 8, we compare the normalized allocation rates for the different strategies. We see that DRES spreads the arrival requests more evenly amongst the VMs during flash crowd traffic arrivals. Although the VMs with a higher index have higher routing costs, DRES still redirects more traffic to these VMs than the F strategy because it tries to mitigate the additional penalty incurred when a false alarm occurs, which is ignored by the F strategy. On the other hand, if DRES determines that the arrival is in N mode, it adopts the same rates as the N strategy. In this case, it redirects requests to far fewer VMs than the F strategy, which uses the same rates regardless of the traffic arrival statistics. This allows the DRES strategy to be more energy efficient by turning off non-active VMs during N mode traffic arrivals.

VI. Conclusion

The emergence of cloud computing technologies allow the design of elastic cost-aware user redirection mechanisms that scale flexibly with the arrival traffic. In this paper, we derive optimal redirection policies under a cloud-centric framework, by jointly detecting the arrival traffic mode and adapting the allocation and service capacities accordingly. We show that the optimal redirection policy involves choosing those VMs with the lowest prices, up to a threshold price. We also show how to compute the optimal allocation and service capacities of the active VMs. The simulation result shows that the proposed mechanism performs better than other benchmark strategies.

In this paper, we have investigated redirection strategies for a single dispatcher. In a data center, multiple dispatchers have to cooperate with one another in redirecting their request arrivals. In future research, we will study the problem of cooperation amongst multiple dispatchers and develop distributed redirection and service scaling strategies under different traffic conditions.

Acknowledgement

The authors would like to thank the associate editor and anonymous reviewers for their comments and suggestions that have helped to improve this manuscript.

This work was supported in part by NTU Start-Up Grant (RG11/31), a research grant from Microsoft Research Asia and a research grant from Cisco Systems, Inc.

References

[1] H. Yin, X. Liu, F. Qiu, N. Xia, C. Lin, H. Zhang, V. Sekar, and G. Min, "Inside the bird's nest: measurements of large-scale live vod from the 2008 olympics," in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, ser. IMC '09. New York, NY, USA: ACM, 2009, pp. 442–455.

[2] P. Wendell and M. J. Freedman, "Going viral: flash crowds in an open CDN," in *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, ser. IMC '11. ACM, Nov. 2011, pp. 549–558.

[3] Cisco Systems, "Cisco visual networking index: Forecast and methodology, 2010-2015," Cisco Systems, Technical Report, 2011.

[4] ——, "Cisco visual networking index: Global mobile data traffic forecast update, 2010-2015," Cisco Systems, Technical Report, 2011.

[5] L. Kingsle, "Responding to the exaflood," *Broadband Properties*, p. 24, June 2008.

[6] C. Yishuai, C. Changjia, and G. Dan, "A measurement study of online internet video views through a large-scale, search engine," in *Cyber-Enabled Distributed Computing and Knowledge Discovery, 2009. CyberC '09. International Conference on*, 2009, pp. 180–183.

[7] S. Ranjan and E. Knightly, "High-performance resource allocation and request redirection algorithms for web clusters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 9, pp. 1186–1200, 2008.

[8] Akamai Technologies, 2013. [Online]. Available: http://www.akamai.com/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2014.2308726, IEEE Transactions on Multimedia

[9] A.-J. Su, D. R. Choffnes, A. Kuzmanovic, and F. E. Bustamante, "Drafting behind Akamai: Inferring network conditions based on CDN redirections," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1752–1765, 2009.

[10] F. Lo Presti, C. Petrioli, and C. Vicari, "Distributed dynamic replica placement and request redirection in content delivery networks," in *Proc. 15th Int. Symp. Modeling, Analysis, and Simulation of Computer and Telecommunication Systems MASCOTS '07*, 2007, pp. 366–373.

[11] M. Kasbekar, "On efficient delivery of web content," Akamai Technologies, Technical Report, 2010.

[12] R. Buyya, M. Pathan, and A. Vakali, *Content Delivery Networks*. Berlin, Germany: Springer, 2008.

[13] U. Sharma, P. Shenoy, S. Sahu, and A. Shaikh, "A cost-aware elasticity provisioning system for the cloud," in *Proc. 31st Int Distributed Computing Systems (ICDCS) Conf*, 2011, pp. 559–570.

[14] Y. Wen, G. Shi, and G. Wang, "Designing an inter-cloud messaging protocol for content distribution as a service (CoDaas) over future internet," in *Proc. the 6th International Conference on Future Internet Technologies 2011(CFI11)*, 2011.

[15] W. Zhang, Y. Wen, Z. Chen, and A. Khisti, "QoE-Driven cache management for http adaptive bit rate streaming over wireless networks," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1431–1445, 2013.

[16] Y. Jin, Y. Wen, K. Guan, D. Kilper, and H. Xie, "On monetary cost minimization for content placement in cloud centric media network," in *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, 2013.

[17] J. Gray, B. Good, D. Gawlick, P. Homan, and H. Sammer, "One thousand transactions per second," in *In Proc IEEE Compcon '85 Conf*, 1985, pp. 96–101.

[18] H. Poor and O. Hadjiliadis, *Quickest Detection*. Cambridge, U.K.: Cambridge University Press, 2008.

[19] K. D. Devine, E. G. Boman, R. T. Heaphy, B. A. Hendrickson, J. D. Teresco, J. Faik, J. E. Flaherty, and L. G. Gervasio, "New challenges in dynamic load balancing," *Applied Numerical Mathematics*, vol. 52, pp. 133–152, Feb. 2005.

[20] K. Gilly, C. Juiz, and R. Puigjaner, "An up-to-date survey in web load balancing," *World Wide Web*, vol. 14, no. 2, pp. 105–131, 2011.

[21] N. R. Mahapatra and S. Dutt, "Random seeking: a general, efficient, and informed randomized scheme for dynamic load balancing," in *Proc. 10th Int. Parallel Processing Symp., IPPS '96*, 1996, pp. 881–885.

[22] M. Jaseemuddin, A. Nanthakumaran, and A. Leon-Garcia, "TE-Friendly content delivery request routing in a CDN," in *Proc. IEEE Int. Conf. Communications ICC '06*, vol. 1, 2006, pp. 323–330.

[23] D.-C. Li and F. M. Chang, "An in-out combined dynamic weighted round-robin method for network load balancing," *Comput. J.*, vol. 50, no. 5, pp. 555–566, Sep. 2007.

[24] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 10, pp. 1094–1104, 2001.

[25] C.-M. Chen, Y. Ling, M. Pang, W. Chen, S. Cai, Y. Suwa, and O. Altintas, "Scalable request routing with next-neighbor load sharing in multi-server environments," in *Proc. 19th Int. Conf. Advanced Information Networking and Applications AINA 2005*, vol. 1, 2005, pp. 441–446.

[26] R. L. Carter and M. E. Crovella, "Server selection using dynamic path characterization in wide-area networks," in *Proc. IEEE Sixteenth Annual Joint Conf. of the IEEE Computer and Communications Societies INFOCOM '97*, vol. 3, 1997, pp. 1014–1021.

[27] T. Minyard, Y. Kallinderis, and K. Schulz, "Parallel load balancing for dynamic execution environments," in *Comput. Methods Appl. Mech. Engrg*, 1996, pp. 96–0295.

[28] J. D. Teresco, M. W. Beall, J. E. Flaherty, and M. S. Shephard, "A hierarchical partition model for adaptive finite element computation," *Comput. Methods Appl. Mech. Engrg*, vol. 184, pp. 269–285, 1998.

[29] C. Walshaw and M. Cross, "Multilevel mesh partitioning for heterogeneous communication networks," *Future Generation Comput. Syst*, vol. 17, pp. 601–623, 2001.

[30] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, ser. SIGCOMM '09, 2009, pp. 123–134.

[31] X. Nan, Y. He, and L. Guan, "Optimal resource allocation for multimedia cloud based on queuing model," in *Proc. IEEE 13th Int Multimedia Signal Processing (MMSP) Workshop*, 2011, pp. 1–6.

[32] L. Chen, N. Li, and S. H. Low, "On the interaction between load balancing and speed scaling," in *ITA Workshop*, 2011.

[33] M. Bjorkqvist, L. Y. Chen, M. Vukolic, and X. Zhang, "Minimizing retrieval latency for content cloud," in *Proc. IEEE INFOCOM*, 2011, pp. 1080–1088.

[34] F. Chen, K. Guo, J. Lin, and T. La Porta, "Intra-cloud lightning: Building CDNs in the cloud," in *Proc. IEEE INFOCOM*, 2012, pp. 433–441.

[35] X. Qiu, H. Li, C. Wu, Z. Li, and F. C. M. Lau, "Cost-minimizing dynamic migration of content distribution services into hybrid clouds," in *Proc. IEEE INFOCOM*, 2012, pp. 2571–2575.

[36] I. Ari, B. Hong, E. L. Miller, S. A. Brandt, and D. D. E. Long, "Managing flash crowds on the Internet," in *Proc. 11th IEEE/ACM Int. Symp. Modeling, Analysis and Simulation of Computer Telecommunications Systems MASCOTS 2003*, 2003, pp. 246–249.

[37] Y. Chen, B. Zhang, and C. Chen, "Modeling and performance analysis of P2P live streaming systems under flash crowds," in *Proc. IEEE Int Communications (ICC) Conf*, 2011, pp. 1–5.

[38] K. Hosanagar, R. Krishnan, M. D. Smith, and J. Chuang, "Optimal pricing of content delivery network (CDN) services," in *HICSS*, 2004.

[39] Amazon Elastic Compute Cloud, 2013. [Online]. Available: http://aws.amazon.com/ec2/

[40] S. Kaxiras and M. Martonosi, *Computer Architecture Techniques for Power-Efficiency*, 1st ed. Morgan and Claypool Publishers, 2008.

[41] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. New Jersey, U.S.: Prentice Hall, 1992.

[42] G. Lorden, "Procedures for reacting to a change in distribution," *The Annals of Mathematical Statistics*, vol. 42, no. 6, pp. 1897–1908, 1971.

[43] G. V. Moustakides, "Optimal stopping times for detecting changes in distributions," *The Annals of Statistics*, vol. 14, no. 4, pp. pp. 1379–1387, 1986.

[44] H. V. Poor, *An introduction to signal detection and estimation (2nd ed.)*. New York, NY, USA: Springer-Verlag New York, Inc., 1994.

[45] D. Bertsekas, A. Nedić, and A. Ozdaglar, *Convex Analysis and Optimization*, ser. Athena Scientific optimization and computation series. Athena Scientific, 2003.

[46] UMassTraceRepository, 2013. [Online]. Available: http://traces.cs.umass.edu/index.php/Network/Network

Jianhua Tang (S'11) received his BEng degree in Communication Engineering from Northeastern University, China, in June 2010. He is currently a Ph.D. candidate at the School of Electrical and Electronic Engineering, Nanyang Technological University. His research interests include cloud computing, mobile computing and green communication system.

Wee Peng Tay (S'06-M'08) received the B.S. degree in Electrical Engineering and Mathematics, and the M.S. degree in Electrical Engineering from Stanford University, Stanford, CA, USA, in 2002. He received the Ph.D. degree in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2008. He is currently an Assistant Professor in the School of Electrical and Electronic Engineering at Nanyang Technological University, Singapore. His research interests include distributed decision making, data fusion, distributed algorithms, communications in ad hoc networks, machine learning, and applied probability.

Dr. Tay received the Singapore Technologies Scholarship in 1998, the Stanford University Presidents Award in 1999, and the Frederick Emmons Terman Engineering Scholastic Award in 2002. He is the coauthor of the best student paper award at the 46th Asilomar conference on Signals, Systems, and Computers. He is currently serving as a vice chair of an Interest Group in IEEE MMTC, and has served as a technical program committee member for various international conferences.

Yonggang Wen (S'99-M'08) is an assistant professor with school of computer engineering at Nanyang Technological University, Singapore. He received his PhD degree in Electrical Engineering and Computer Science (minor in Western Literature) from Massachusetts Institute of Technology (MIT), Cambridge, USA. Previously he has worked in Cisco to lead product development in content delivery network, which had a revenue impact of 3 Billion US dollars globally. Dr. Wen has published over 90 papers in top journals and prestigious conferences. His latest work in multi-screen cloud social TV has been featured by global media (more than 1600 news articles from over 29 countries) and recognized with ASEAN ICT Award 2013 (Gold Medal) and IEEE Globecom 2013 Best Paper Award. He serves on editorial boards for IEEE Transactions on Multimedia, IEEE Access Journal and Elsevier Ad Hoc Networks. His research interests include cloud computing, green data center, big data analytics, multimedia network and mobile computing.