Performance Analysis of Server Sharing Collectives for Content Distribution

Daniel Villela, *Student Member*, *IEEE*, and Dan Rubenstein, *Member*, *IEEE*

Abstract—Demand for content served by a provider can fluctuate with time, complicating the task of provisioning serving resources so that requests for its content are not rejected. One way to address this problem is to have providers form a collective in which they pool together their serving resources to assist in servicing requests for one another's content. In this paper, we determine the conditions under which a provider's participation in a collective reduces the rejection rate of requests for its content—a property that is necessary for such a provider to justify its participation within the collective. We show that all request rejection rates are reduced when the collective is formed from a homogeneous set of providers, but that some rates can increase within heterogeneous sets. We also show that, asymptotically, growing the size of the collective will sometimes, but not always, resolve this problem. We explore the use of thresholding techniques, where each collective participant sets aside a portion of its serving resources to serve only requests for its own content. We show that thresholding allows a more diverse set of providers to benefit from the collective model, making collectives a more viable option for content delivery services.

Index Terms—Information services, network servers, modeling.

1 Introduction

Content providers profit from servicing their clients' requests for their content. If a provider's serving resources (e.g., servers and bandwidth) are insufficient, it will be forced to turn away a large number of requests during periods when the content reaches its peak in popularity. The amount of serving resources needed during a peak period, however, is often much larger than what would be needed on a regular basis. Hence, a provider that provisions resources for these peak periods will pay for equipment that sits by idly most of the time, reducing profits.

A recent solution used by many providers has been to contract third party content distribution networks (CDN) that host and service their content during peak periods. The provider, however, pays the CDN for its assistance, which, again, can reduce profits. Instead of relying on CDNs during peak periods, an overlooked alternative is for groups of providers to form *collectives* and host one another's content. When the demand for content that originated at provider A peaks, exceeding its own serving abilities, it can redirect requests to other members of the collective whose available serving resources can handle these requests. In return, when the demand for content that originates at some other provider peaks, provider A's serving resources can be used to help serve this other content.

It is well-known that systems that pool together resources can outperform the performance of their individual components. For instance, load can more easily be balanced among the pooled resources and overloads (dropping requests) are less likely to occur [1], [2], [3], [4], [6]. The problem we consider here, however, has an important distinction from these traditional works and from the CDN model: *Each service provider "profits" only from requests for its own content.* While the collective more efficiently serves the aggregate demand (over all providers), because a provider's resources may be used to help serve other providers' content, there may not be enough resources in the collective to serve its own client demand. This leaves open the possibility that the rejection rate of requests for an individual provider's content can be higher within the collective than if that provider operated in isolation. Since the provider profits only from requests for its originating content, this increase in rejection rate can deter its participation in the collective. The Content Distribution Internetworking (CDI) chart [7] at the IETF describes similar concepts and requirements for interconnection of content networks to collectives. The CDI model, however, lacks a performance analysis of the benefits of such a system.

In this paper, we identify from a performance perspective when these collectives are a viable alternative. In particular, we address the following questions:

- Under what conditions do all provider participants in a collective benefit from their membership in the collective?
- Are there any mechanisms that can be introduced into the collective architecture that will increase the range of conditions under which all participating providers benefit?

To enable us to focus on the performance aspects of this question, we start at the point where a set of providers have agreed to form a collective, have made copies of one another's content, and can redirect requests for a particular

- *The authors are with the Department of Electrical Engineering, Columbia University, 500 W. 120th St., New York, NY 10027.*
E-mail: {dvillela, danr}@ee.columbia.edu.

Manuscript received 12 Dec. 2003; revised 8 Oct. 2004; accepted 27 Feb. 2005; published online 20 Oct. 2005.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0234-1203.

content object to any server within the collective with sufficient available capacity. When no server has available capacity, the request is dropped.

For each provider in the collective, we compare the rejection rate for its content (that it originated) when served within the collective to when it serves its content in isolation. Each provider is described in terms of its capacity (number of jobs it can serve simultaneously) and its intensity (the rate of requests for its content divided by the rate at which it serves requests). We find that collectives reduce rejection rates of **all** provider participants by several orders of magnitude when the collective is formed from a homogeneous set of providers with identical capacities and intensities. However, even slight variations in intensity among providers yield heterogeneous collectives in which the lower intensity participants achieve significantly lower rejection rates in isolation than within the collective.

We next consider whether all providers' needs can be met by growing the size of the collective, i.e., can the rejection rate be brought arbitrarily close to zero by simply increasing the membership to the collective? We identify a simple rule that is a function of the average intensity and the average capacity that determines whether the rejection rate converges to zero or to a positive constant. A convergence to zero implies that all providers would benefit from participating in very large collectives. However, when the rate converges to a positive constant, some providers may still be better off participating in isolation.

To accommodate providers whose rejection rates are lower in isolation, we consider the application of thresholding techniques within the collective. Thresholding allows each provider to set aside a portion of its serving resources to be used exclusively to service its own clients' requests. We demonstrate that often, by appropriately setting thresholds, all providers in a collective will experience lower rejection rates than when they operate in isolation, even if this property did not hold within the threshold-free version of that collective. Our work demonstrates that, from a performance standpoint, collectives that utilize thresholds often offer a viable, cheaper alternative to overprovisioning or utilizing CDN services.

The rest of the paper is structured as follows: In Section 2, we briefly overview related work. In Section 3, we present our general model for server collectives. In Section 4, we investigate performance of content delivery services for fixed-rate sessions when considering collective arrangements. We similarly evaluate elastic file transfers in Section 5. Section 6 evaluates a suite of thresholding techniques. We conclude and elaborate on open issues in Section 7.

2 Related Work

Several works analyze systems that pool server resources to improve various performance aspects of content delivery. For instance, studies [3], [4], [6] investigate the practical challenge of maintaining consistency among distributed content replicas. The study in [8] investigates the placement of content in the network to minimize delivery latencies. Other studies [1], [2] investigate load sharing policies. These approaches keep the processing load on a set of hosts relatively balanced while keeping redirection traffic levels low. In the Oceano project [9], a provider owns and maintains a pool of servers that can be deployed to service businesses of various customers. After servers are allocated to customers, each server is used exclusively by the customers to whom it was allocated and cannot be shared. An analytical study of systems in which servers are spawned upon cutoff points of a single service demand appears in [10]. A model in which a resource manager distinguishes users into classes that can share a resource first appears in [12]. More recently, the study in [11] presented an algorithm to protect such classes from overloads.

The goal in these previous works differs from ours in that there is no notion of individual, competing objectives as there is within a server collective. In other words, in these other works, the only objective is to improve the greater good of the entire system, whereas, in our work, each provider has its own objective of minimizing the rejection rate of its own content.

The problem of alleviating rapid and unpredictable spikes in request demands ("flash crowds") has generated much attention recently. Jung et al. propose a reassignment of servers within a CDN infrastructure to handle such events [13]. Recent proposals in this area [14], [15], [16] solve this problem using peer-to-peer methods, in which clients communicate directly with one another to retrieve the desired content. Here, clients have nothing to gain by serving content. The effectiveness of these approaches simply relies on the goodwill of those who receive content to also transmit the content to others when requested to do so.

The Content Distribution Internetworking (CDI) charter at the IETF is an initiative whose direction is closer to our work. The CDI model concentrates on the definition of requirements and concepts that allow interconnection of CDNs for content delivery across different content networks [7]. The performance of these systems, however, has not yet been analyzed. The analysis of server collectives presented in this paper can apply to the CDI model.

3 Collectives General Model

In this section, we develop the model (Table 1) that allows us to explore the fundamental performance tradeoff that collectives offer to content providers. Namely, we investigate if participating in a collective reduces the rejection rate of requests for a provider's content. To perform our investigation, we develop a model that is simple, elegant, and amenable to a performance analysis.

Our model of a server collective consists of commodities, content providers, clients, servers, and sessions. *Commodities* are the content/information goods offered by content providers. For instance, multimedia lectures are the commodities of an online course offering. The *content provider* (or simply *provider*) is the entity that offers commodities to customers via the Internet. A *client* requests commodities, and a *server* interfaces with the network to deliver commodities to clients. When a server accepts a client request for a commodity, a *session* (or content transfer) is initiated to deliver the content from the server to the client. A server's ability to deliver content is constrained by

TABLE 1
Main Variables and Main Parameters of the Model

$\mathcal{V} = \{v_1, v_2, ..., v_n\}$	set of commodities
$\mathcal{S} = \{s_1, s_2, ..., v_n\}$	set of servers
$\mathcal{Y} = \{y_1, y_2, ..., y_n\}$	set of content providers
$p_{n,i}$	rejection rate of provider y_i's content in a collective with n servers
N_i	number of simultaneous transfers of provider y_i's content
k_i	maximum capacity on provider y_i's server
λ_i	arrival rate of requests for provider y_i's content
B_i	time for processing commodity v_i
ρ_i	intensity due to demand for provider y_i's content
h_i	provider y_i's threshold applied to requests for content other than its content
D1–threshold	threshold as a function of active number of sessions associated with content other than a provider's own content
D2–threshold	threshold as a function of remaining available capacity
D3–threshold	threshold as a function of number of redirected sessions
d_n	average completion time of sessions in a collective with n providers

factors such as its processing capabilities (CPU cycle consumption) and its access link bandwidth.[1]

To analyze the performance of collectives, we assume that a set of providers has already agreed to form a collective and has distributed each commodity to all servers within the collective. A request can be served if there is a server that can immediately process the job associated with that request. Our model assumes that the network core is well-provisioned such that the server's processing capabilities or its access link to the network are what limit the number of jobs that can be served simultaneously. Hence, a client's location in the network does not affect the server's ability to serve that client.

An example of how a collective, once established, can reduce the rate at which requests for a provider's content are rejected is depicted in Fig. 1. Servers s_1, s_2, and s_3 are deployed by three distinct content providers in both Figs. 1a and 1b. The number of sessions a server can host simultaneously is indicated by the number of boxes. Shaded boxes indicate an active session and each clear box is a resource that is available to process a session. In Fig. 1a, three different providers operate in isolation (i.e., they do not participate in a collective and do not host one another's content). The server labeled s_3 cannot service both of the two arriving requests and is forced to drop a request. A logical view of the collective containing these three servers is shown in Fig. 1b. Here, server s_3 redirects the request it cannot service itself to server s_2, which has the capacity to process the job associated with that request. As a result, by participating in the collective, fewer requests for s_3's content are rejected.

1. In practice, licensing restrictions can place additional limitations on the server side. For instance, RealNetworks [19] offers its basic streaming server (free of charge) with a maximum capacity of five simultaneous streams. Their $1,999-dollar license server (Helix Server—starter) has a maximum capacity of 25 simultaneous streams. The maximum number of sessions is then given by the maximum transmission throughput divided by the average transmission rate of delivery sessions.

The rejection rate is the metric used to evaluate a provider's content delivery service. While collectives can be used to handle sudden spikes of demand, our focus is on spikes that last for non-negligible portions of time, where the rejection rate can be determined by observing the steady state statistics of the serving system.

We consider a set of content providers $\mathcal{Y} = \{y_1, y_2, \ldots y_n\}$, where a given provider's commodities are files whose lengths are described by i.i.d. random variables. The set of servers that belong to provider y_i are modeled as a single serving system, s_i. We refer to any commodity that originated at provider y_i as v_i without loss of generality. We develop separate models for two classes of content. The first class consists of *fixed rate* transfers, such as streaming audio or video, where each transfer consumes a *fixed* amount of server bandwidth per unit time, such that the length of a session is independent of the number of files served concurrently by the server. The second class consists of *elastic transfers*, such as data files, where the amount of bandwidth consumed per transfer per unit time is *inversely proportional* to the number of files served at that time by the server. In both classes, the number of files that a server will simultaneously transmit is bounded to ensure that transfer rates proceed above a minimum rate. The maximum number k of simultaneous sessions of a server is the server's capacity. The factors (CPU cycle consumption, access link bandwidth, etc.) that constrain the ability to service sessions typically determine the server's capacity.

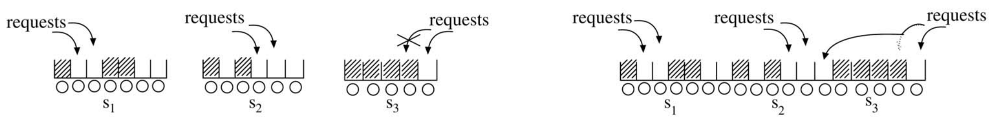
We define *homogeneity* as the property that, for a collective, all providers' intensities are equal and all providers' values of the maximum number of sessions a provider operating in isolation can service (i.e., k_i for the ith provider) are equal as well. Thus, a collective is said to be homogeneous when this property holds, otherwise, the collective is said heterogeneous. Later in this paper, we will see that this property plays an important role in identifying collectives within which providers achieve smaller rejection rates than are achieved in isolation.

We assume that a provider chooses to participate in a collective as long as the rejection rate of requests for its content is lower than when the provider operates in isolation (comparative criterion).

4 Collectives under Fixed-Rate Transfers

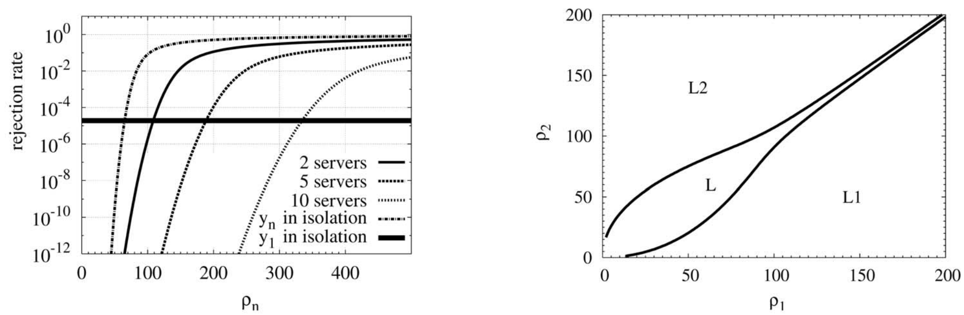
Here, we construct and evaluate a model in which content delivery sessions consist of fixed-rate transfers such as delivery of streaming video.

For fixed rate transfers, we make a simplifying assumption that each commodity (across providers) requires the same rate of transfer, such that each server s_i is capable of hosting a fixed number k_i of sessions simultaneously, where this number is independent of the set of commodities currently being hosted. The request rate for each provider's commodities, $v_i \in \mathcal{V}$, is modeled as a Poisson process with rate λ_i and each request receives service immediately if the number of simultaneous sessions is smaller than the maximum k_i. The service times for instances of transfers of commodity v_i are i.i.d. random variables B_i with mean $E[B_i]$ and are independent across the set of all commodities.



(a) (b)

Fig. 1. Collective example compared to systems in isolation.



(a) (b)

Fig. 2. Evaluation of collectives under the model of fixed-rate transfers. (a) Rejection rate in collectives scenario compared to servers in isolation. (b) Areas of comparison between two-server collectives and servers in isolation.

Since arrivals of requests for content delivery are modeled as Poisson processes, each serving system is an $M/G/k/k$ queuing system. If the server in a collective cannot host an arriving request for its commodity, the server forwards the request to an available server (when one exists) in the collective. Otherwise, the request is dropped. Note that, if a server operates in isolation, then, when it has no additional room to service a request, the request must be dropped. Note that the collective can also be modeled as an $M/G/k/k$ queuing system with arrival rate $\sum_{i=1}^{n} \lambda_i$, mean service time $(\sum_{i=1}^{n} \lambda_i E[B_i])/(\sum_{i=1}^{n} \lambda_i)$, and that can service up to $k = \sum_{i=1}^{n} k_i$ sessions simultaneously. The product $\lambda_i E[B_i]$ is the *intensity* ρ_i of provider y_i.

4.1 Computing the Rejection Rate of a Collective

We define $p_{n,i}$ to be the rejection rate of provider y_i's content in a collective composed of n servers (e.g., a set of providers $\{y_1, y_2, \dots y_n\}$ employing servers $s_1, \cdots, s_n$).[2] A single provider operating in isolation from other providers has rejection rate denoted by $p_{1,i}$, or simply p_1. For provider y_i in isolation, $i = 1, \dots n$, the Erlang loss formula (also known as Erlang B formula) applies directly [17]:

$$p_1 = \frac{\rho_1^{k_1}/k_1!}{\sum_{j=0}^{k_1} (\rho_1)^j / j!}, \tag{1}$$

where $\rho_1 = \lambda_1 E[B_1]$.

We extend this formula to the rejection rate of a two-server collective $p_{2,i}$, $i = 1, 2$. First, we define the random variables N_i, $i = 1, 2$ that describe the number of sessions for each of the commodities v_i, $i = 1, 2$. Since such a loss system is a symmetric queue [18], the stationary distribution for each state $P(N_1 = x, N_2 = z)$, where x (z) is the number of commodities of type v_1 (v_2) actively being processed, can be expressed in product-form: $P(N_1 = x, N_2 = z) = \pi_x \pi_z c_2$, where $\pi_x = \rho_1^x / x!$, $\pi_z = \rho_2^z / z!$, and c_2 is a normalizing constant such that $\sum_{x \geq 0, z \geq 0, z+x \leq k_1+k_2} \pi_x \pi_z c_2 = 1$. Hence, we find that the rejection rate of a two-server collective is

$$\begin{aligned} p_2 &= P(N_1 + N_2 = k_1 + k_2) \\ &= \sum_{x=0}^{n} P(N_1 = k_1 + k_2 - x, N_2 = x) \\ &= \sum_{x=0}^{n} \frac{1}{x!} \rho_1^x \frac{1}{(k_1 + k_2 - x)!} \rho_2^{k_1+k_2-x} c_2 \\ &= (\rho_1 + \rho_2)^{k_1+k_2} \frac{1}{(k_1 + k_2)!} c_2, \end{aligned} \tag{2}$$

where $\rho_i = \lambda_i E[B_i]$, $i = 1, 2$. Noting that (2) is in the form of (1) with ρ_1 replaced by $\rho_1 + \rho_2$ and k_1 replaced by $k_1 + k_2$, we can recursively repeat this process to compute the rejection rate for a collective composed of n providers:

$$p_n = \frac{1}{(\sum_{j=1}^{n} k_j)!} \left(\sum_{j=1}^{n} \rho_j \right)^{\sum_{j=1}^{n} k_j} c_n, \tag{3}$$

where c_n is once more the normalizing constant for this distribution such that

$$\sum_{x_1 \geq 0, x_2 \geq 0, \dots, \sum_{j=1}^{n} x_j \leq \sum_{j=1}^{n} k_j} P(N_1 = x_1, \dots, N_n = x_n) = 1.$$

4.2 Evaluation of Scenarios under Fixed-Rate Transfers

We begin by evaluating collectives as a function of number of participating providers and the provider intensities. Fig. 2a plots rejection rates of n-provider systems p_n, $n = 2, 5, 10$, where each server can simultaneously service 100 commodities ($k_i = 100$, $i = 1, 2$). The results are a direct application of (3).

2. Whenever $\forall y_i, y_j \in \mathcal{Y}, i \neq j, p_{n,i} = p_{n,j}$, $p_{n,i}$ can be written simply as p_n.

First, we fix the configurations of providers $y_1, y_2, \ldots, y_{n-1}$ while we vary the traffic intensity of provider y_n. We are interested in observing how the rejection rate of a provider in a collective of size n is affected when a single provider's rate is varied. We set $\rho_i = 65$ for each provider whose intensity is fixed. This value is chosen such that requests for v_i exhibit a rejection rate of approximately 10^{-5} when y_i operates in isolation, $1 \leq i < n$. On the x-axis, we vary ρ_n, the provider intensity for provider y_n. The y-axis plots rejection rates for various system configurations. The curve labeled "y_n in isolation" depicts the rejection rate of requests for commodities v_n when provider y_n operates in isolation. The constant line at $p_1 \approx 10^{-5}$, labeled "y_1 in isolation," plots the rejection rate of requests for provider y_1s content when y_1 operates in isolation (results for providers $y_2, \cdots, y_{n-1}$ are identical). The remaining curves labeled "n servers," $n = 2, 5, 10$, depict the rejection rate for all providers that participate in an n-provider collective. From the figure, we observe that:

- For the range of values of ρ_n shown, the rejection in the collective for commodity v_n is smaller than its rejection rate in isolation. Therefore, regardless of its own intensity, a provider is willing to participate in a collective when the other providers have sufficiently low intensities.
- Ranges for ρ_n exist, $0 \leq \rho_n \leq 110$, $0 \leq \rho_n \leq 175$, and $0 \leq \rho_n \leq 330$ for $n = 2, 5, 10$, respectively, where the rejection rate in the collective for commodity $v_i, i < n$ is smaller than the rejection rate when s_i operates in isolation. Hence, the collective benefits provider y_i, $i < n$ even when all other providers in the collective have larger provider intensities. In particular, to achieve a given rejection rate, ρ_n needs to be increased significantly as n increases.
- When ρ_n assumes higher values, the dropping probability for commodities becomes excessive and, according to the described criteria, y_i, $1 \leq i \leq n$, would prefer to withdraw its participation from a collective.

Fig. 2b graphically depicts the "areas of participation willingness" for two providers. In this figure, we consider a system with two providers, y_1 and y_2, where $k_i = 100, i = 1, 2$. A two-server collective generates a two-dimensional picture as shown in Fig. 2b. We vary ρ_1 and ρ_2 on the x and y-axis, respectively. The top curve that goes from the bottom-left to the top-right of the graph is the set of values of (ρ_1, ρ_2), where content provider y_1 experiences the same rejection rate regardless of whether it operates in isolation or participates within the collective, i.e., $p_{1,1} = p_2$. Above this curve, $p_{1,1} < p_2$, i.e., provider y_1's rejection rate is lower when operating in isolation. Conversely, below this curve, $p_{1,1} > p_2$, i.e., provider y_1's rejection rate is lower when forming a collective with provider y_2. Similarly, the lower curve that runs from the bottom left to the top right is formed from the points where $p_{1,2} = p_2$. Below this curve, $p_{1,2} < p_2$ and, above, $p_{1,2} > p_2$.

In Fig. 2b, each area is labeled to indicate the "willingness" of y_1 and y_2 to form a collective. A label of "Li," $i = 1, 2$, indicates an area in which rejection rate for y_i's content is smaller in a collective than in isolation, i.e., $p_2 < p_{1,i}$. The pair of labels "L1," "L2," are replaced by "L," for simplicity. We see that there is a significant region in which both providers can benefit simultaneously by participating in a collective. In this case, both rejection rates are each lower when both form a collective than when each provider operates in isolation. We refer to a win-only property when, for the providers of a collective, every provider's rejection rate is smaller in the collective than in isolation. We note, however, that both providers tend to benefit only in "near-homogeneous" configurations (in this case, defined by the line $\rho_2 = \rho_1$), especially when intensities range from moderate to high. As the difference in intensities widens, the win-only property ceases to hold. In fact, we observe a rapid increase in the rejection rate in Fig. 2a for a collective of two servers as ρ_n is increased in the range $85 \leq \rho_n \leq 150$. In addition, we note that there is no area in which the rejection rates of both content providers are higher in the shared system than in isolation.

4.3 Asymptotic Limits of Collectives: The ρ/k Factor

We turn to the performance of an n-server collective as the number of servers n tends to ∞. In practice, a very large collective requires also very large storage units in each of the collective participants since all participants must store all other participants' commodities. There are, however, important insights to be gained from studying the impact on performance as a collective grows. In fact, our results reveal that a rejection rate for providers forming a collective can be close to the asymptotic limit, even for a small number of providers forming the collective.

We assume that there are n_c different classes of providers, where all provider systems in the same class exhibit the same provider intensity ρ_i and have the same bound k_i on the number of jobs that can simultaneously be serviced. We let f_i represent the fraction of providers in class $i, 1 \leq i \leq n_c$. Equation (3) can be reformulated as

$$p_n = \frac{\dfrac{\left(\sum_{j=1}^{n_c}(nf_j\rho_j)\right)^{\sum_{j=1}^{n_c} nk_jf_j}}{\left(\sum_{j=1}^{n_c} nk_jf_j\right)!}}{\sum_{i=0}^{\sum_{j=1}^{n_c} nf_jk_j} \dfrac{\left(\sum_{j=1}^{c_n}(nf_j\rho_j)\right)^i}{i!}}, \tag{4}$$

which is simplified via the constants $\hat{\rho} = \sum_{j=1}^{n_c} f_j\rho_j$ and $\hat{k} = \sum_{j=1}^{n_c} f_jk_j$ to

$$p_n = \frac{(n\hat{\rho})^{n\hat{k}}/(n\hat{k})!}{\sum_{j=0}^{n\hat{k}}(n\hat{\rho})^j/j!}.$$

As n tends to ∞, the asymptotic limit of the rejection rate is [20]:

$$\lim_{n\to\infty} p_n = \begin{cases} 0, & \text{if } \hat{\rho}/\hat{k} \leq 1 \\ 1 - \hat{k}/\hat{\rho}, & \text{if } \hat{\rho}/\hat{k} > 1. \end{cases} \tag{5}$$

Fig. 3 illustrates the behavior of an n-server collective as n grows large. For clarity of presentation, we show only the case where only one class of servers exists, i.e., $n_c = 1$, in which each provider's system can simultaneously handle $k_1 = k = 100$ transmissions. In this case, the average intensity is simply referred as $\rho = \rho_1 = \hat{\rho}$. In Fig. 3a, we vary the number of participating providers along the x-axis,

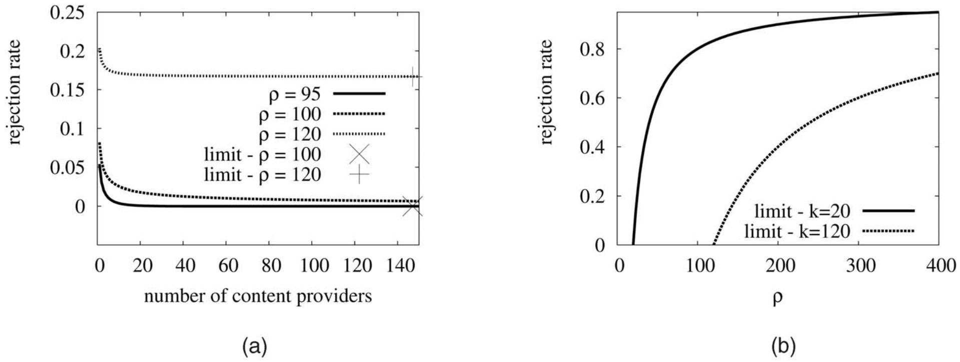


Fig. 3. Asymptotic observations on rejection rate for fixed-rate transfer collectives. (a) Rejection rate when increasing the number of providers. (b) Asymptotic rejection rate for $n \rightarrow \infty$.

plotting the rejection rate along the y-axis, where each curve depicts a collective with the corresponding intensity ρ. We use (3) to plot the curves for $\rho = 95$, $\rho = 100$, and $\rho = 120$. The asymptotic limits are marked with two distinct points. Using (5), the asymptotic limit for $\rho = 100$ is effectively 0, and for $\rho = 120$, the limit is approximately $1/6$. The asymptotic limits observed here fit the claims of (5). In particular, for $\rho < k$, the rejection rate converges to 0, whereas for $\rho > k$, we observe the rejection rate converging to the limit $1 - k/\rho$. The figure demonstrates (for a homogeneous collection of providers) that the providers benefit from joining a collective with a small number of providers, and that increasing the number of providers in the collective further reduces rejection rates, but at a rate that diminishes quickly.

In Fig. 3b we apply (5) to plot the rejection rate of a collective for the limit as the number of providers grows infinitely large as a function of ρ. The curve on the left is the asymptotic rejection rate for the case where $k = 20$. We consider only $\rho \geq k = 20$. The curve on the right is for the case where $k = 120$. Again, we consider only $\rho \geq k = 120$. We notice for $k = 120$ a slower increase in the asymptotic rejection rate. We see that the rejection rate converges more slowly and less abruptly (i.e., there is less of a knee) for larger values of k.

In conclusion, even when participants in a collective are homogeneous but individually overloaded ($\rho_i > k_i$), their rejection rate for requests for their content remains bounded (below) by $1 - k/\rho$.

5 Collectives for Elastic Transfers

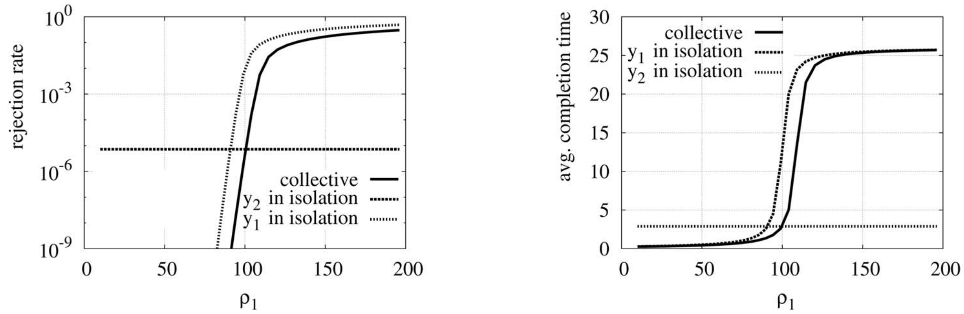
In the previous section, we focused on performance benefits for providers forming a collective where their content distribution services are fixed-rate transfers. In this section, we focus on performance benefits for providers forming a collective where their content distribution services are elastic transfers such as file transfers using TCP/IP. We consider collectives that receive traffic whose service rate is proportional to $1/w_n$, where w_n is the number of customers served simultaneously across all n servers that comprise the collective. For such a model, load is equally balanced across the collective. This can be accomplished in practice, for instance, with parallel downloading technology [21], which is commonplace today in peer-to-peer downloading tools such as Kazaa.

We again consider a model in which client arrivals to each provider are described by a Poisson process and the load imposed by each commodity is described by a general distribution. Using the same argument as before, we collapse the model to the case where each provider offers a single commodity and uses a single server. The processor sharing (PS) service model applies here since the transmission rate is inversely proportional to the number of requests in service. We assume that each server s_i bounds the minimum rate at which it will transmit data to clients by bounding the number of clients accepted by a constant, k_i, and will turn away (or redirect) any additional clients requesting service. In the context of parallel downloading, the sum of all fractional components serviced should add up to $\sum_{i=1}^{n} k_i$. Therefore, a server is modeled as an $M/G/1/k_i/PS$ queue and a collective modeled by an $M/G/1/k/PS$ queue, where $k = \sum_{i=1}^{n} k_i$. We assume, without loss of generality, that work for delivering y_i's content in isolation is processed at a rate equal to the maximum number of simultaneous sessions k_i and for a collective at rate k. As before, we assume that jobs are never queued when service is unavailable, but are simply turned away (dropped). In addition to rejection rate as a metric, we also measure job completion time.

The amount of work necessary for a job processed by the server collective is a random variable U. For an n-server collective, the servicing time $\hat{B}_n$ for each individual job in the collective is $\hat{B}_n = U/k$. Since the servicing time $\hat{B}_1$ obtained with a single provider in isolation is $\hat{B}_1 = U/k_1$, $\hat{B}_n$ is a function of the servicing time, $\hat{B}_1$, such that $\hat{B}_n = k_1\hat{B}_1/k$. While the distribution of an unbounded queuing system that uses a processor sharing discipline is known to be geometric [17], for a finite queuing system, we find the following law:

$$p_n = (\lambda E[\hat{B}_n])^k \frac{1 - \lambda E[\hat{B}_n]}{1 - (\lambda E[\hat{B}_n])^{k+1}}, \qquad (6)$$

where $\lambda = \sum_{j=1}^{n} \lambda_j$ (λ_j is the request rate for commodity v_j) and $E[\hat{B}_n]$ is the mean servicing time of a job processed by the collective. Here, the intensity is $\rho = \lambda E[U]$, where $E[U] = kE[\hat{B}_n]$.



(a) (b)

Fig. 4. Evaluation of a collective for elastic transfers. (a) Rejection rate under a processor sharing discipline. (b) Mean completion time under a processor sharing discipline.

Since, for elastic transfers, the service rate is inversely proportional to the number of simultaneous transfers, the completion time is also a metric of interest. For a collective formed from n providers, the expected completion time of a session d_n is obtained using Little's Law, in terms of the fraction of requests accepted to the system $(1-p_n)\lambda$. We let the number of concurrent, servicing sessions be a random variable N. Using Little's law, we find,

$$d_n = \frac{\bar{w_n}}{(1-p_n)\lambda}, \tag{7}$$

where $\bar{w}_n$ is the expected number of simultaneous sessions, $\bar{w}_n = \sum_{z=0}^{k} zP(N=z)$.

5.1 Numerical Evaluation

Here, we study conditions under which providers are *willing* participants within a collective that consist of providers whose transfer rates are elastic. As in Section 3, we again compare a provider's rejection rate when forming a collective with other providers to its rejection rate when operating in isolation. For instance, Fig. 4 illustrates a scenario in which we vary ρ_1 and maintain ρ_2 fixed at $\rho_2 = 91$. Here, $k_1 = k_2 = 100$. Figs. 4a and 4b plot the rejection rate and average completion time, respectively, of the providers in collectives as well as of providers in isolation as a function of ρ_1. The curves labeled "collective," "y_1 in isolation," and "y_2 in isolation," respectively, plot p_2, $p_{1,1}$, and $p_{1,2}$ (the last being constant). We observe the rejection rate of providers within the collective to be almost 4 orders of magnitude smaller than $p_{1,1}$ when ρ_1 and ρ_2 are approximately equal. However, the benefits become marginal with increasing $|\rho_1 - \rho_2|$. This again supports the intuition that a collective is useful for elastic transfers only when the intensities imposed by providers are approximately the same.

We observe from Fig. 4b that the conclusions for completion time similar to those obtained for the rejection rate. We vary the intensity ρ_1 along the x-axis. The average completion time is shown along the y-axis. The curves labeled "collective," "y_1 in isolation," and "y_2 in isolation," respectively, plot the average completion time for each provider within the collective, for provider y_1 in isolation, and provider y_2 in isolation. The average completion time for a request to provider y_2's commodity is reduced significantly when $\rho_1 \leq 80$. This reduction occurs because jobs for provider y_2's content are likely to receive treatment from y_1's underutilized resources. In the range $80 \leq \rho_1 < 100$, provider y_1's average completion time is reduced significantly. Thus, this range is important because provider y_1 and y_2 both benefit from participating in the collective. For $\rho_1 > 100$, provider y_1 can still obtain dramatically lower completion times within a collective, as low as a sixth of its completion time in isolation, but provider y_2's average completion time is greater than in isolation. Therefore, provider y_2 will not be willing to form a collective.

Fig. 5 depicts willingness areas for elastic transfer services computed via application of (6). The parameters ρ_1 and ρ_2 are varied respectively along the x and y-axis. We use the same labeling convention as in Section 4.2. In comparison to Fig. 2b, we see that a collective in elastic environments favors both providers for a wider variation of their respective intensities when both intensities are small (i.e., the bubble in the bottom-left corner is bigger). However, when intensities are large, the difference in provider intensities over which both providers are willing to form a collective is reduced. Intuitively, this may be due to the fact that, when the system is under high loads, the average completion time of a job increases. Bringing in additional capacity (but with a proportional load) does not reduce completion times of admitted jobs significantly when load is high. It will, however, reduce completion times when load is light. Since a collective formation is most useful when intensities are high (e.g., in Fig. 5, the bulb does not stretch as much as in Fig. 2), we conclude that collectives can tolerate more heterogeneity in systems when

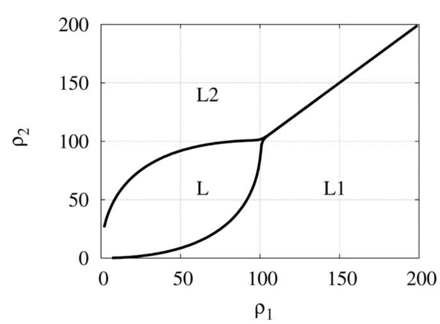


Fig. 5. Area of benefit for collectives under processor sharing discipline.

servicing fixed-rate requests than when servicing elastic-rate requests.

6 Resource Bounding with Thresholds

Here, we evaluate *thresholding* techniques as a means to limit the amount of server resources that a provider contributes to a collective. We show how thresholding can be used to bound rejection rates of providers, thereby encouraging participation within a collective system. Such schemes can be used in either fixed-rate or elastic transfers. Here, we perform the analysis using fixed-rate transfers.

Let h_i be a *threshold*, $0 \leq h_i \leq k_i$, for server s_i such that s_i refuses any requests to service other provider's commodities whenever it is actively servicing h_i commodities of other providers. This guarantees that the provider will maintain space to simultaneously service at least $k_i - h_i$ requests of its own commodity at any given time. We call this threshold type D1. We also evaluate a second thresholding technique, named D2-thresholding. A D2 type threshold denies a request at server s_i for another provider's commodity v_j, $j \neq i$, whenever the available space for commodity transmission at server s_i falls below h_i (i.e., $k_i - x_i$, where x_i is the number of sessions in service). An advantage of D2 over D1-thresholding is a stronger protection for a provider service when its intensity is high, since it can reject requests for commodities other than its own even if no job associated with these commodities is in service. For both types of thresholding, setting $h_i = 0$, $1 \leq i \leq n$ is equivalent to n providers operating in isolation and setting $h_i = k_i$ for all i is equivalent to a collective described in previous sections where providers fully share their resources.

We also consider a third thresholding technique, D3-thresholding, in which a provider's threshold is a function of its average number a of sessions redirected to other provider's servers. Under D3-thresholding, a provider accepts a number h of sessions for content other than its own, such that $h \leq a + \tilde{a}$, where $\tilde{a}$ is a tolerance value beyond the average number a of redirected sessions. This type of tolerance value allows providers to redirect requests when their average number of redirected sessions is greater than a but still within a range defined by the tolerance value $\tilde{a}$. Besides, allowing for variation in the level of tolerance set by the value of $\tilde{a}$, this formulation permits a "bootstrap" of the process of redirecting sessions when the average number of redirected sessions is zero, i.e., when $a = 0$. If a provider's ability to redirect requests to other providers' servers is bounded by a, then, when a equals zero, a provider does not permit utilization of its resources.

All three types of thresholding can be implemented with or without *switching*. When switching is implemented, a job for provider i's that is assigned to a server $j \neq i$ can be moved back to i as soon as there is an available process at server i, i.e., jobs are always moved to their preferred server when there is room. In practice, enabling switching entails additional overhead, but, for analytical purposes, the model is more tractable. We shall see shortly (comparing simulation results) that enabling switching has little impact on rejection rate, so that the analytical results obtained when switching is allowed are a good approximation for models in which switching is not permitted.

6.1 Analytical Evaluation of D1 Thresholding

D1-thresholding is a coordinate-convex sharing policy, thus, a product-form solution is still valid under general service time distributions [11]. However, for a simplified, comprehensible analysis, we assume that service times are exponentially distributed at rate μ_i, $1 \leq i \leq n$.[3] This allows us to model the two-provider collective as a truncated Markov chain with states described by the pair (N_1, N_2), where N_i is the number of sessions servicing commodity v_i in the system, $i = 1, 2$, $0 \leq N_1 \leq k_1 + \min(k_2 - N_2, h_2)$, and $N_2 \leq k_2 + \min(k_1 - N_1, h_1)$. A crucial difference from previous models is that, here, since a provider's decision to accept a request depends on whether the commodity being requested belongs to the provider, the rejection rates for the differing commodities can differ within a collective. The Markov chain transitions are as follows:

- From $(x-1, z)$ to (x, z) with rate λ_1, for
$$1 \leq x \leq k_1 + \min(k_2 - z, h_2).$$
- From (x, z) to $(x-1, z)$ with rate $x\mu_1$, for
$$1 \leq x \leq k_1 + \min(k_2 - z, h_2).$$
- From $(x, z-1)$ to (x, z) with rate λ_2, for
$$1 \leq z \leq k_2 + \min(k_1 - x, h_1).$$
- From (x, z) to $(x, z-1)$ with rate $z\mu_2$, for
$$1 \leq z \leq k_2 + \min(k_1 - x, h_1).$$

A product-form solution is derived:

$$P(N_1 = x, N_2 = z) = \pi_{x,z} = \pi_x \pi_z c_2,$$

where $\pi_x = \rho_1^x / x!$, $\pi_z = \rho_2^z / z!$, and c_2 is a normalizing constant such that $\sum_{x=0}^{k_1+h_2} \sum_{z=0}^{k_2+\min(h_1, k_1 - x)} \pi_{x,z} = 1$.

We use the probabilities $\pi_{x,z}$ to compute the probabilities of all states for which $N_i = k_i + \min(k_j - w,\ h_j)$, given that $N_j = w$, $i = 1, 2$, $j = 1, 2$, $j \neq i$. The rejection rate of provider y_1's content in the collective, $p_{2,1}$, is:

$$\begin{aligned} p_{2,1} &= \sum_{x=k_i-h_i}^{k_i+h_j} \pi_{x,k_i+k_j-x} + \sum_{z=0}^{k_j-h_j-1} \pi_{k_i+h_j,z} \\ &= \sum_{x=k_i-h_i}^{k_i+h_j} \frac{(\lambda_i/\mu_i)^x}{x!} \frac{(\lambda_j/\mu_j)^{k_i+k_j-x} c_2}{(k_i+k_j-x)!} + \\ &\quad \frac{(\lambda_i/\mu_i)^{k_i+h_j}}{(k_i+h_j)!} \sum_{z=0}^{k_j-h_j-1} \frac{(\lambda_j/\mu_j)^z}{z!} c_2. \end{aligned} \tag{8}$$

The rejection rate of provider y_2's content in the collective, $p_{2,2}$, is obtained in analogous manner (not shown here for space limitations).

Fig. 6 depicts respective rejection rates experienced for requests of content of both providers y_1 and y_2 for various intensities and threshold levels under D1-thresholding with

3. Note that our reduction of a provider's server system to a single server with a single commodity still holds without loss of generality.

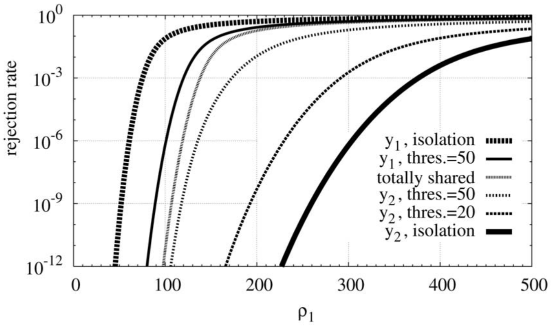


Fig. 6. Respective rejection rate experienced on requests for providers y_1 and y_2s$'$ content and different thresholds.

switching, where providers y_1 and y_2 apply the same threshold, i.e., $h_1 = h_2$. Each of them has a server with total capacity for $k_i = 100$, $i \in \{1, 2\}$, concurrent sessions. On the x-axis, we vary ρ_1. Instead of fixing ρ_2, we set $\rho_2 = 0.2\rho_1$ such that the intensities that both providers contribute to the collective increase along the x-axis, but $y_1{}'$s intensity remains much larger than that of y_2. The various curves depict rejection rates for the two commodities for differing threshold levels. The curves for the systems in isolation are represented with thicker lines. The curve labeled "totally shared" is the rejection rate for both commodity requests when the thresholds are set to maximum values $h_1 = h_2 = 100$. The remaining curves' labels indicate the provider whose content's rejection rate is plotted and the value to which the threshold is set.

The most important conclusion here is that changing the threshold value can lead to significant variations in rejection rate for a collective. In fact, when the intensity of a provider is small enough that the provider can meet its desired rejection rate operating by itself, that provider can then use thresholds in a collective to allow other content providers the use of its resources without raising its own rejection rate above the undesired level.

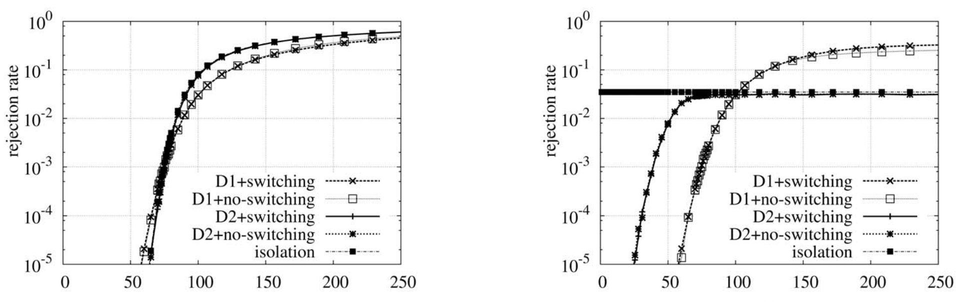
6.2 Comparison of Thresholding Techniques

We resort to simulation to evaluate nonswitching and D2 and D3-threshold configurations. We model arrivals by a Poisson process, but service times here are described by a lognormal distribution, as has been observed in practice [22], [23], [24], [25]. The probability density function of the lognormal distribution is given by $\frac{e^{-(\log(x)-\mu)^2/(2\sigma^2)}}{x\sigma\sqrt{2\pi}}$, where $\log(x)$ is the natural logarithm and μ and σ are the standard parameters used within the lognormal distribution. We used the mean and standard deviation of 26 and 46 (minutes) observed in [22], respectively, to derive the parameters μ and σ.

Based on measurements from [22], we conduct simulations with $\lambda = 3.5$ requests per minute and $E[B] = 26$ minutes giving a value $\rho_2 = 91$ for provider $y_2{}'$s content. Fig. 7 plots rejection rates obtained from the previous analysis ((8), for the case of D1 thresholding with switching) and simulations (for the other cases) in which $h_i = 40$ for $i = 1, 2$. Fig. 7a plots rejection rates for provider $y_1{}'$s content and Fig. 7b plots rejection rates for provider $y_2{}'$s content. Here, ρ_1 is varied along the x-axis in both Figs. 7a and 7b. The curves in each figure labeled "D1+switching," "D1+nonswitching," "D2+switching," and "D2+nonswitching" depict the various collective thresholding configurations formed by alternating between the use of nonswitching and switching methods and between the use of D1 and D2 thresholding techniques.

In Fig. 7a, we observe little difference in rejection rate for provider $y_1{}'$s content for the varied configurations. In Fig. 7b, we observe the rejection rates of provider $y_2{}'$s content. The horizontal line depicts the rejection rate for provider $y_2{}'$s content when y_2 operates in isolation. For $\rho_1 < 40$ the two curves with sharp increases correspond to the collective with D2-thresholds. The two curves are indistinguishable except for $\rho_1 > 100$ where the switching system exhibits a rejection rate that is slightly lower than the nonswitching system. The two curves with sharp increases in the range $60 < \rho_1 < 100$ correspond to the collective with D1-thresholds. When operating in the collective, the rejection rate for provider $y_2{}'$s content drops by as much as three orders of magnitude in the range where $\rho_1 < 50$ using D2 and $\rho_1 < 100$ using D1-thresholds. In the range $\rho_1 > 100$, the rejection rates for provider y_2 converges to the rejection rate when provider y_2 operates in isolation. This is because ρ_2 is sufficiently high to make it unlikely that the number of sessions delivering provider $y_2{}'$s content places the available space at $y_2{}'$s server below its threshold h_2. In contrast, when using D1-thresholding, the exhibited rejection rates are much lower than those exhibited when using D2 thresholding when $\rho_1 < 100$.

We further observe that there is little difference in the results obtained when ongoing session switching is enabled from when it is not. This suggests that analytical results for



(a) (b)

Fig. 7. The use of D1-thresholding and D2-thresholding/switching. (a) Rejection rate of provider $y_1{}'$s content. (b) Rejection rate of provider $y_2{}'$s content.

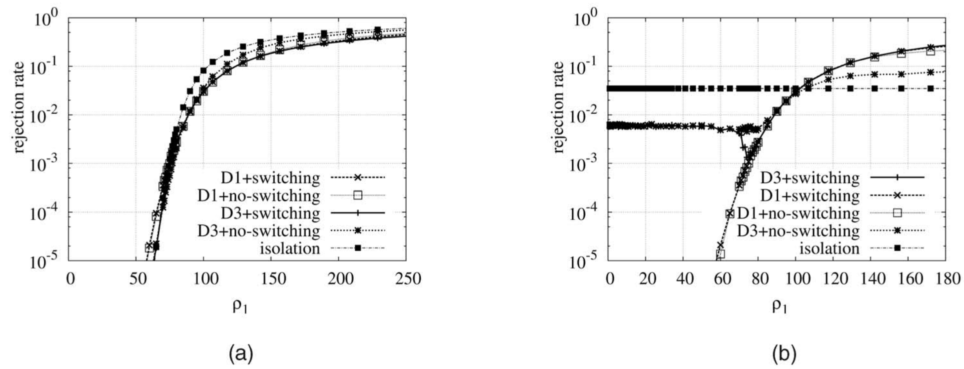


Fig. 8. Comparison of D1-thresholding and D3-thresholding/switching. (a) Rejection rate of provider y_1's content. (b) Rejection rate of provider y_2's content.

rejection rate from a switching system can be used to approximate the rejection rate within nonswitching systems.

Fig. 8 depicts a comparison between the performance of D1 and D3-thresholding. We apply the same simulation experiments used to compare performance of D1 and D2 thresholding. Provider y_2's intensity is kept constant at $\rho_2 = 91$, while ρ_1 is varied. To remain consistent with previous parameters, both providers y_1 and y_2 are set to configurations in which $k_i = 100$ and $h_i = 40$, $i \in \{1, 2\}$. The tolerance value is $\tilde{a} = 10$. Fig. 8a shows the rejection rate of provider y_1's content, whereas Fig. 8b shows the rejection rate of provider y_2's content.

We observe that the main difference in the performance obtained with D3-thresholding compared to using D1-thresholding (and, also, to using D2-thresholding) is that D3-thresholding only allows a reduction of rejection rate of provider y_2's content by no more than one order of magnitude in the range where ρ_1 is low, such as $\rho_1 < 50$. In this range, provider y_2 can only reduce its rejection rate up to a limit approximately given by $\tilde{a}$. This happens because, for low values of ρ_1, provider y_2 does not redirect much of its incoming requests, hence, the average number a of redirected sessions for provider y_1 is low and the threshold is limited by $a + \tilde{a}$. In contrast, in the range $50 < \rho_1 < 80$, provider y_1 can offer larger thresholds to provider y_2. As a result, we observe a larger reduction of the rejection rate of provider y_2's content from its rejection rate when provider y_2 is in isolation. For high values of ρ_1, rejection rates obtained with D3-thresholding for both providers are similar to ones respectively obtained using D1-thresholding. This comes as a result of the use of the extra allowance $\tilde{a}$ which reserves space in providers for content different than their own commodities in the same fashion as D1-thresholding does.

6.3 Extending Heterogeneity

Thresholding encourages providers to participate in collectives with performance benefits where the same providers are unwilling to participate in a collective without any thresholding. Applying the comparison between rejection rate obtained in isolation and the rejection rate obtained in a collective, we can find the potential areas of interest for two providers, y_1 and y_2, when using the same pair of thresholds. We wish to determine if provider y_1 performs better in isolation than in a collective with y_2, and vice versa, given their respective intensities, ρ_1 and ρ_2, and threshold values, h_1 and h_2 (D1-thresholding). We perform an exploration similar in form to that used in Section 4.2, but now the collectives are formed with restrictions given by the two thresholds, h_1 and h_2, and the rejection rates are obtained using (8). We show in Fig. 9 how forming a collective can bring benefits to both providers under more heterogeneous conditions for providers y_1 and y_2. The intensity ρ_1 is varied along the x-axis and the intensity ρ_2 is plotted along the y-axis. The area between the curves labeled "$h_1 = h_2 = 100$" is the set of values for ρ_1 and ρ_2 where both y_1 and y_2 share without restrictions. The area between curves labeled "$h_1 = h_2 = 4$" and "$h_1 = h_2 = 2$" indicate values of ρ_1 and ρ_2 for which both providers are using the same value of threshold, i.e., only up to four slots or two slots, respectively, can be used to serve a content for the other provider. All these areas have in common that both providers have smaller rejection rates than their rejection rates in isolation, i.e., these are "win-only" areas. We conclude from the shown curves that thresholding indeed extends the heterogeneity tolerated for establishing collectives. However, the two providers do not necessarily need to have the same threshold values.

6.4 Optimal Thresholding

Motivated by the results in previous sections, we consider an ideal scenario in which a provider, y_i, selects its optimal threshold as a function of the providers' intensities imposed

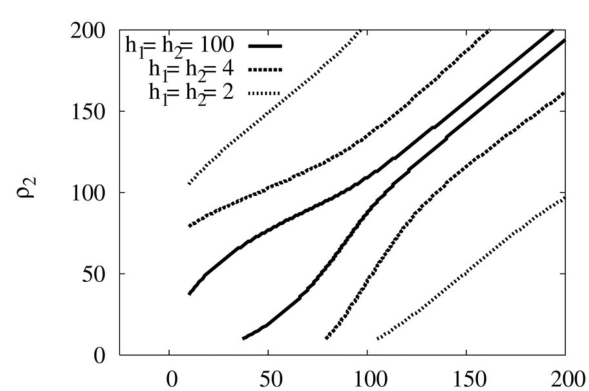


Fig. 9. Extending heterogeneity using thresholds.

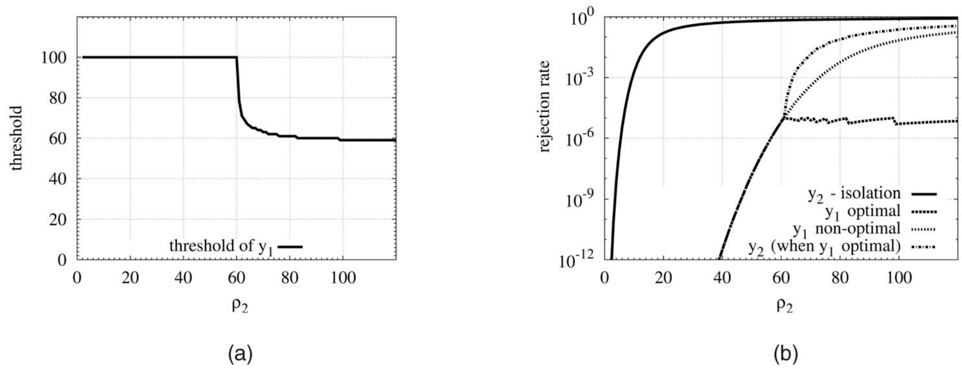


Fig. 10. Optimal thresholding. (a) Threshold adjustment. (b) Rejection rates.

on the collective. By doing so, it contributes the maximum amount of its own server resources (i.e., the highest threshold possible) without the rejection rate for its own commodity, v_i, exceeding a value l_i.

We apply our analysis of D1-type thresholding systems with switching to a two-provider collective in which providers y_1 and y_2 accept a maximum of $k_1 = 100$ and $k_2 = 20$ concurrent sessions, respectively. Provider y_1 receives a fixed intensity of $\rho_1 = 20$. Provider y_1 adjusts its threshold h_1 to the maximum integer value (via (8)) such that the rejection rate for provider y_1's content remains below the value of $l_1 = 10^{-5}$. In this case, provider y_1 relaxes its condition for willingness to participate in the collective and requires only that its content's rejection rate remains below l_1. Provider y_2 enables its server to fully share its resources, i.e., $h_2 = k_2 = 20$.

In Fig. 10a, we vary ρ_2 in the x-axis and plot the largest value of h_1 along the y-axis as a function that maintains $p_{2,1} < l_1$. We see that, for $\rho_2 \leq 60$, the threshold remains at 100. As ρ_2 crosses 60, the threshold drops rapidly, then continues to reduce, but at a much slower rate. In Fig. 10b, we vary ρ_2 along the x-axis and the rejection rate along the y-axis. Fig. 10b shows rejection rates of various configurations as a function of ρ_2. The left-most curve plots the rejection rate of provider y_2's content when y_2 operates in isolation (obtained from (1)). The remaining three curves (which differ only when $\rho_2 > 60$) plot, from top to bottom, the rejection rate of provider y_2's content when participating in the collective with optimal thresholding, the rejection rate for all commodities when participating in the collective without thresholding (obtained from (2)), and the rejection rate of provider y_1's content when participating in the collective with the optimal thresholding.

The bottom curve verifies that, with thresholding, rejection rates of provider y_1's content remain below $l_1 = 10^{-5}$. By comparing the remaining two curves from the collective to the curve for the case where y_2 is in isolation, we see that, even with thresholding, participating in the collective significantly reduces provider y_2's rejection rate. We see that, while thresholding increases the rejection rate for provider y_2's content in comparison to a threshold-free collective, provider y_1 is willing to participate in the collective only when the thresholding is applied, and provider y_2 experiences a rejection rate that is orders of magnitude smaller than if y_2 operates in isolation. Such adjustments permit a provider to set a target rejection rate to not be exceeded.

7 Conclusion

We have analyzed the performance of resource sharing via the formation of server collectives as a means to reduce rejection rates in content distribution services. Providers can benefit by participating in collectives but should avoid situations in which their resources are overused servicing requests on the behalf of other collectives members, worsening the delivery quality of their own content. Our analysis and simulation via fundamental queuing models yields the following results and insights:

- We modeled fixed and elastic rate transfers within collectives and compared the rejection rates and completion times of these transfers to the case where providers operate in isolation. We then used our models to determine the conditions under which a provider benefits from participating in collectives. In particular, we determine the conditions for which all participants simultaneously benefit from their participation in collectives.
- Even a small degree of heterogeneity among participants in a collective can lead to situations in which one or more providers achieve a lower rejection rate for their content by operating in isolation. An expected consequence is that such providers would refrain from participating in collectives in these unfavorable circumstances.
- In some circumstances, we observe significant reduction in rejection rates of collectives in comparison to systems in isolation. For instance, we show a four-order-of-magnitude reduction in rejection rates when comparing an isolated system to a two-server collective. Furthermore, a 10-server collective has a seven-order-of-magnitude reduction in comparison to an isolated system. As the number of servers increases, the relative reduction of rejection rate becomes less dramatic.
- We found asymptotic results as the number of collective providers tends to ∞. If the factor ρ/k given by the average provider intensity ρ and the maximum number of concurrent sessions in the system k is less than one, then the system's rejection

rate is 0 in the limit. Otherwise, the rejection rate converges to $1 - k/\rho$.

- When demands on providers' contents are high, composing a collective (without thresholding) can reduce the rejection rate of all participants for a greater variation in intensities among participating systems supporting fixed-rate transfers than can be tolerated within systems supporting elastic-rate transfers.
- We analyzed three thresholding techniques that enable heterogeneous sets of server systems (different intensities and numbers of slots) to form a collective in which requests for all participants' commodities are dropped at a rate lower than when the systems operate in isolation. We show that, in conjunction with thresholding, the ability to dynamically swap a transmission to the server that profits directly from the servicing of the content has little impact on the rejection rate. Thresholding therefore encourages providers to participate in collectives who otherwise would not do so, extending the range of heterogeneity in providers for which server collectives are applicable.

ACKNOWLEDGMENTS

The authors would like to thank Ed Coffman for valuable discussions and comments on this article and Predrag Jelenkovic and Vishal Misra for valuable discussions regarding this work. They are also thankful to Ward Whitt for pointing out a reference on asymptotic behavior of loss systems. This material was supported in part by the US National Science Foundation under Grant No. ANI-0117738 and CAREER Award No. 0133829. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the US National Science Foundation. Daniel Villela received scholarship support from CNPq-Brazil (Ref. No. 200168/98-3).

REFERENCES

[1] D. Eager, E. Lazowska, and J. Zahorjan, "A Comparison of Receiver-Initiated and Sender-Initiated Adaptive Load Sharing," *Performance Evaluation*, vol. 16, May 1986.
[2] D. Eager, E. Lazowska, and J. Zahorjan, "Adaptive Load Sharing in Distributed Systems," *IEEE Trans. Software Eng.*, vol. 12, May 1986.
[3] M. Dahlin, R. Wang, T. Anderson, and D. Patterson, "Cooperative Caching: Using Remote Client Memory to Improve File System Performance," *Proc. Symp. Operating Systems Design and Implementation*, 1994.
[4] G. Voelker, H. Jamrozik, M. Vernon, H. Levy, and E. Lazowska, "Managing Server Load in Global Memory Systems," *Proc. ACM Sigmetrics*, June 1997.
[5] D. Villela and D. Rubenstein, "Performance Analysis of Server Sharing Collectives for Content Distribution," *Proc. Int'l Workshop Quality of Service*, pp. 41-58, June 2003.
[6] G. Voelker, E. Anderson, T. Kimbrel, M. Feeley, J. Chase, A. Karlin, and H. Levy, "Implementing Cooperative Prefetching and Caching in a Global Memory System," *Proc. ACM Sigmetrics Conf.*, June 1998.
[7] M. Day, B. Cain, G. Tomlinson, and P. Rzemski, "A Model for Content Internetworking (CDI)," RFC 3466, IETF, Feb. 2003.
[8] J. Kangasharju, J.W. Roberts, and K.W. Ross, "Object Replication Strategies in Content Distribution Networks," *Proc. Sixth Int'l Workshop Web Caching and Content Delivery*, June 2001.
[9] IBM Research, "Oceano Project," http://www.research.ibm.com/oceanoproject/, 2000.
[10] L. Golubchik and J.C.S. Lui, "Bounding of Performance Measures for Threshold-Based Systems: Theory and Application to Dynamic Resource Management in Video-on-Demand Servers," *IEEE Trans. Computers*, vol. 74, no. 4, pp. 50-63, Apr. 1995.
[11] G.L. Choudhury, K.K. Leung, and W. Whitt, "Efficiently Providing Multiple Grades of Service with Protection against Overloads in Shared Resources" *AT&T Technical J.*, pp. 353-372, Apr. 1995.
[12] J.S. Kaufman, "Blocking in a Shared Resource Environment," *Trans. Comm.*, vol. COM-29, pp. 1474-1481, Oct. 1981.
[13] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNS and Web Sites," *Proc. WWW Conf.*, May 2002.
[14] V.N. Padmanabhan, H.J. Wang, P.A. Chou, and K. Sripanidkulchai, "Distributing Streaming Media Content Using Cooperative Networking," *Proc. Int'l Workshop Network and Operating Systems Support for Digital Audio and Video*, May 2002.
[15] T. Stading, P. Maniatis, and M. Baker, "Peer-to-Peer Caching Schemes to Address Flash Crowds," *Proc. First Int'l Workshop Peer-to-Peer Systems*, Mar. 2002.
[16] R.J. Bayardo, Jr., A. Somani, D. Gruhl, and R. Agrawal, "Youserv: A Web Hosting and Content Sharing Tool for the Masses," *Proc. WWW-2002*, 2002.
[17] S. Ross, *Stochastic Processes.* John Wiley & Sons, Inc., 1983.
[18] R.W. Wolff, *Stochastic Modeling and the Theory of Queues*, chapter 6, pp. 334-341. Prentice Hall, 1988.
[19] Real Networks, Inc., http://www.realnetworks.com, 2004.
[20] K. Ross, *Multiservice Loss Models for Broadband Telecommunications Networks*, Springer, 1995.
[21] J. Byers, M. Luby, and M. Mitzenmacher, "Accessing Multiple Mirror Sites in Parallel: Using Tornado Codes to Speed up Downloads," *Proc. IEEE INFOCOM*, 1999.
[22] J.M. Almeida, J. Krueger, D.L. Eager, and M.K. Vernon, "Analysis of Educational Media Server Workloads," *Proc. Int'l Workshop Network and Operating SystemsSupport for Digital Audio and Video*, pp. 21-30, June 2001.
[23] J. Padhye and J. Kurose, "An Empirical Study of Client Interactions with a Continuous-Media Courseware Server," *IEEE Internet Computing*, Apr. 1999.
[24] M. Chesire, A. Wolman, G.M. Voelker, and H.M. Levy, "Measurement and Analysis of a Streaming-Media Workload," *Proc. Third USENIX Symp. Internet Technologies and Systems*, Mar. 2001.
[25] S. Jin and A. Bestavros, "Gismo: Generator of Streaming Media Objects and Workloads," *Performance Evaluation Rev.*, 2001.

Daniel Villela received the electrical engineering diploma (cum laude) from Universidade Federal do Rio de Janeiro, Brazil, the MSc degree in electrical engineering from Universidade Federal do Rio de Janeiro, Brazil, and the MPhil degree in electrical engineering from Columbia University, New York. He is currently a graduate research assistant and a PhD candidate in electrical engineering at Columbia University. His interests are in the performance of networked systems and algorithms for distributed computing and distributed communication systems. He has been awarded a scholarship for his doctoral study at Columbia University from Capes/Brazil and a SIGCOMM Student Travel award. He is a student member of the IEEE.

Dan Rubenstein received the BS degree in mathematics from MIT, the MA degree in math from UCLA, and the PhD degree in computer science from the University of Massachusetts, Amherst. He has been an assistant professor of electrical engineering and computer science at Columbia University since 2000. He is generally interested in the design and performance of networked systems, and also has interests in the security and integrity of such systems, as well as the general design of distributed communication algorithms. He has received an US National Science Foundation CAREER Award, an IBM Faculty Award, an IEEE ICNP Best Paper Award, and (as a student) an ACM SIGMETRICS Best Student Paper Award. He is a member of the IEEE.