Performance Models for Peering Content Delivery Networks

Mukaddim Pathan and Rajkumar Buyya
Grid Computing and Distributed Systems (GRIDS) Laboratory
Department of Computer Science and Software Engineering
The University of Melbourne, Parkville, VIC 3010, Australia
{apathan, raj}@csse.unimelb.edu.au

***Abstract*—A Content Delivery Network (CDN) is expected to provide high performance content delivery, which requires scalable infrastructure to achieve global coverage. The provision of such infrastructure may form a substantial entry barrier for new CDN providers, as well as affecting commercial viability of the existing ones. Peering of CDNs can be a way to allow dynamic infrastructural cooperation between CDNs in a scalable manner, in order to mitigate the impact of flash crowds and to achieve better overall service times. In this paper, we present a Quality of Service (QoS)-driven model to evaluate the user perceived performance of CDN peering relationships. In this model, an overloaded CDN redirects a fraction of its incoming requests to peered CDNs and thereby can avoid the impact of flash crowds. The model-based approach also assists in making concrete QoS guarantee for a given CDN. Our approach endeavors to achieve scalability for a CDN in a user transparent manner.**

I. Introduction

Content Delivery Networks (CDNs) [14] offer fast and reliable Web access services by distributing content to *edge* servers located close to end-users. Running a global CDN is challenging in financial, technical and administrative terms. Moreover, providing Quality of Service (QoS) under unexpected resource shortfalls, such as during flash crowds [1], might be an obstacle for new CDN providers and may also harm the commercial viability of the existing ones. The objective of providing high quality service can be achieved by permitting CDNs to cooperate and thereby providing a means for CDNs to redistribute content delivery between themselves [9][12]. *Peering* between CDNs virtualizes multiple providers and allows flexible resource sharing and dynamic collaboration between autonomous individual CDNs. In such a system, a CDN serves user requests as long as the load can be handled by itself. If the load exceeds its capacity, the overloaded CDN offloads excess requests to the Web servers of its peers. This approach also provides a means to avoid long-term (i.e. periodic traffic pattern during a particular Web event) or short-term (i.e. flash-crowds) bottlenecks [12].

Such peering arrangements are appealing, since it allows individual providers to achieve greater scale and network reach cooperatively than they could otherwise attain individually. However, developing a model capturing the characteristics of end-user requests redirection in peering CDNs is challenging for a number of reasons, which include virtualization of multiple providers and offloading end-user requests from the primary CDN provider to peers based on cost, performance and load. In such a cooperative multi-provider environment, users are redirected across distributed set of Web servers deployed by partnering CDNs as opposed to individual servers belonging to a single CDN. Moreover, limited information about response time or service cost is typically available from individual CDNs, and load balancing control is retained by an individual provider within its own Web servers. Therefore, request-redirections must occur over distributed sets of Web servers belonging to multiple CDN providers, without the benefit of the full information available, as in the single provider case.

The main contributions of this paper are twofold: (1) we introduce analytical models to demonstrate the effects of peering and to predict user perceived performance, and (2) perform sensitivity analysis to study the impact of key performance parameters such as load and measurement errors that can be expected from a real system. The rest of the paper is structured as follows. Section 2 highlights the related work. Section 3 provides a high level description of peering between CDNs. Section 4 presents the analytical performance models. Results are demonstrated in Section 5, which is followed by a decisive evaluation of our approach in Section 6. Finally, Section 7 concludes the paper with a summary of contributions and future work.

II. Related Work

Analyses of previous research reveal a deficient progress to define the frameworks and policies for CDN peering. The reasons for this lack of progress are mainly due to the complexity of the technological problems, legal and commercial operational issues that need to be solved in practice.

An initiative from IETF was the first to propose a Content Distribution Internetworking (CDI) Model [9]. It recommends providing QoS either through using a supervision function or an independent third party to supervise and manage all the CDN peers. However, the CDI model does not define or characterize this supervision. Moreover, it also does not examine the implications of using an independent third party for ensuring QoS guarantees. In the architecture for CDI protocol [15], performance data is interchanged between CDNs before forwarding a request. This has the effect of introducing an overhead to each service response time which is unfortunately not quantified in the paper.

CDN brokering [3] allows a CDN to intelligently redirect end-users dynamically to other CDNs in a domain. Though it provides benefits of increased CDN capacity, reduced cost and better fault tolerance, it does not consider the end-user perceived performance to satisfy QoS while serving requests. Moreover, it demonstrates the usefulness of brokering without evaluating a given CDN's performance.

While the above mentioned research efforts do not explicitly virtualize multiple CDN providers, a peering system in a federated, multi-provider infrastructure has been presented in [2]. The core component of the system is a peering algorithm that directs end-user requests to partner providers to

minimize cost and improve performance. However, peering strategy, resource provisioning and QoS guarantees between partnering providers are not explored in this work.

Cooperative Networking [11] enables cooperation between end-hosts to improve network performance. The main problem with this mechanism is that it is not transparent to users. Hence, it does not permit automated cooperation between CDNs to dynamically share their infrastructure resources.

From the above discussion, it is clear that none of the existing research focuses on providing mechanism to evaluate the QoS performance of a certain provider. Moreover, some of these systems make strong assumptions on the characteristics of applications and do not virtualize multiple providers for cooperative management and delivery of content in a peering environment. Therefore, we develop performance models for peering CDNs, which virtualizes multiple CDN providers, and assists to offload end-user requests from the primary CDN to peers based on load and user perceived QoS performance. Other issues such as cost of migrating content, cache consistency after replication, and added storage requirements fall out of the scope for this paper.

III. Peering Between CDNs

In the peering CDNs architecture [12], a CDN serves user requests as long as the load can be handled (meet QoS) internally. If the load exceeds its capacity, the excess requests are offloaded to the surrogate Web servers of peers. The initiator of each peering negotiation is called a *primary* CDN; while other CDNs who agree to provide their resources are called *peering* CDNs. These roles are fluid and at any time a given CDN may be acting in either primary or peering roles. In some cases, such as when locality is needed to meet service QoS times, it may be operating in both roles.

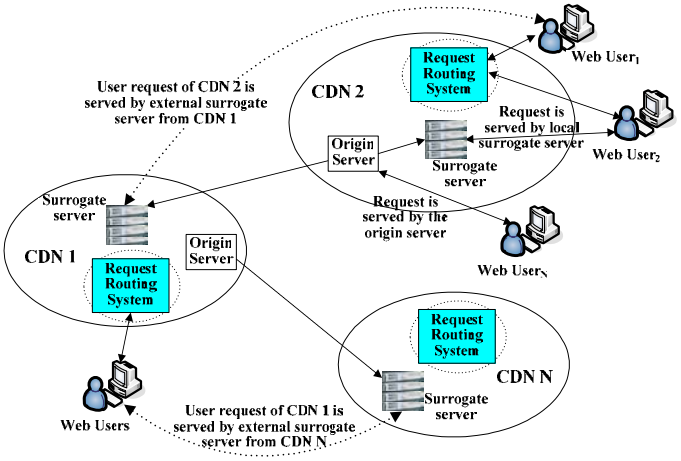


Figure 1: An abstraction of CDN peering.

The result of a peering negotiation between two CDNs is a contract, e.g. Service Level Agreement (SLA) [4], which specifies the peer resources (Web servers, bandwidth etc.) that will be allocated to serve content on behalf of a given primary CDN. The primary CDN directly manages the resources it has acquired, insofar that it determines what content is served and what proportion of the incoming traffic (user requests) is redirected. In Figure 1, we provide an overview of the peering CDNs architecture. End-user requests for content are made to the Request Routing System of the primary CDN. These requests are then forwarded either directly to its Web server(s), or to a peering CDN. In the figure, we observe that some user requests of CDN 2 are served by its local servers or the origin server (on cache miss), whereas others are being served by the external Web servers of a peer, CDN 1. It is important to note that depending on the load any CDN can act as a primary CDN in a peering relationship. For instance, CDN 2 acts as a primary when its users are served by the peers' Web servers. Again, in the same peering relationship, CDN 1 plays the role of a primary CDN when its users are served by external Web servers from CDN N.

A. QoS and SLAs in Peering CDNs

QoS performance can be measured based on users' experience of a service to compare the 'promise' against the 'delivery'. Here, we define quality as:

Let A be a CDN and $\mathbf{S} = \{S_1, S_2, \ldots, S_m\}$ *be the set of services provided by it. Assume that for each service* S_i, S_i^p *is the quality that A promised to offer to the users and* S_i^d *is the actual delivered quality. The QoS for CDN A is* $QoS_A = f(S_i^p, S_i^d)$, *where f is a function to measure the conformance between* S_i^p *and* S_i^d.

Ensuring QoS guarantees requires a means of establishing a set of common quality parameters and establishing which attributes are needed by a particular customer to describe its QoS requirements. These factors are combined in an SLA that both a customer and a provider agree to and that the provider refers to when monitoring its QoS performance. Two examples of QoS parameters that an SLA may specify are: (1) 95% of requests should be served in less than T time units, and (2) a service should be available for at least 99.9% of the time. In this paper, we measure QoS in terms of the *expected waiting time* for a request to be served.

IV. Performance Models

In the single CDN model, Web user arrivals follow a memoryless process with a constant arrival rate over a significant period of time [16]. Internet access workloads are self-similar and heavy-tailed in nature [7][8]. Based on these observations, we model a CDN as an M/G/1 queue (Figure 2) assuming the total processing of the Web servers of a CDN being accumulated through the server and abstracting the request streams coming to the Web servers of a CDN as a single request stream. An M/G/1 queuing system allows to define the service times independent of interarrival times and of one another, and to have a general P.D.F. The mean arrival rate is λ following a Poisson process and the mean service rate is μ following a general distribution. Poisson modeling of user arrivals provides us a good basis for theoretical constructs and mathematical tractability. Moreover, it allows considering the complex network traffic dynamics as a "black box" and helps to estimate parameters (inputs) that are difficult to specify, collect or measure in practice.

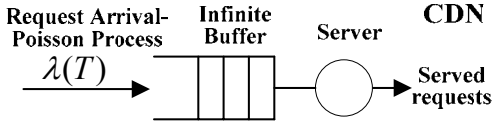


Figure 2: A CDN modeled as an M/G/1 queue.

We use the term 'task' to denote a request arrival and 'task size' to denote its processing requirements. The task size on a CDN's service capacity follows a Bounded Pareto distribution, with the following probability density function (P.D.F), to capture the heavy-tailed nature of Internet access workloads:

$$f(x) = \frac{\alpha k^{\alpha}}{1-(k/p)^{\alpha}} x^{-\alpha-1}, \text{ where } k \le x \le p \text{ and } 0 \le \alpha \le 2$$

The variable α represents the task size variation, k is the smallest possible task size, and p is the largest possible task. By varying the value of α, we can observe moderate ($\alpha \approx 2$) to high ($\alpha \approx 1$) variability in distributions. Let $E[X]$ be the mean service time and $E[X^j]$ be the j-th moment of the service distribution of *tasks*. We have,

$$E[X^j] = \begin{cases} \dfrac{\alpha p^j ((k/p)^\alpha - (k/p)^j)}{(j-\alpha)(1-(k/p)^\alpha)} & \text{if } j \neq \alpha \\ \dfrac{\alpha k^\alpha \ln(p/k)}{(1-(k/p)^\alpha)} & \text{if } j = \alpha \end{cases}$$

Using P-K formula, the expected waiting time is $E[W] = \lambda E[X^2]/2(1-\rho)$, which can be used to measure the waiting time with respect to varying load and task sizes.

Hyper-Exponential Approximation

The Bounded Pareto distribution has all moments finite; however advanced analysis on it is complex due to the difficulties in manipulating the Laplace transforms of the queuing metrics (e.g. waiting time, busy period). Therefore the 'heavy-tailed' Bounded Pareto distribution is approximated with a series of exponential distributions known as Hyper-exponential distributions. Hyper-exponentials preserve the main characteristics of the original distribution, such as heavy tail, first and second moments [5]. We use an n part Hyper-exponential distribution which has the following P.D.F:

$$h_n(t) = \sum_{i=1}^{n} P_i \lambda_i e^{-\lambda_i t} \text{ , where } \sum_{i=1}^{n} P_i = 1$$

We numerically invert the Laplace transform of the waiting time $L_W(s)$ to obtain the P.D.F of the waiting time distribution, $w(t)$ and this is used to obtain the cumulative distribution function (C.D.F) $W(t)$ [13].

A. CDN Peering Model

A conceptual view of the peering CDNs is provided in Figure 3, in which each CDN is modeled as an M/G/1 queue. It is abstracted such that N independent streams of end-user requests arrive at a conceptual entity, the *dispatcher*, following a Poisson process with arrival rate λ_i, $i \in \{1,2,...,N\}$. The dispatcher acts as a centralized scheduler in a particular peering relationship with independent mechanism to distribute content requests among partnering CDNs and assists to assign a fraction of requests of one CDN to its peer(s) in a user transparent manner.

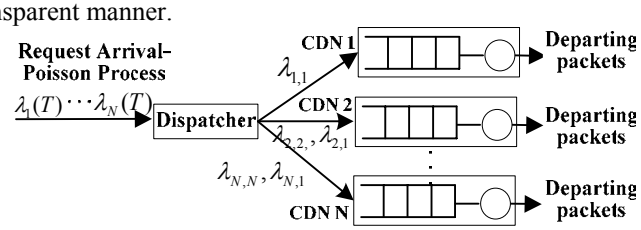


Figure 3: Conceptual view of the peering CDNs.

The request stream is defined as $\lambda_{j,i}$ = request to CDN j for CDN i's content. For $\forall j \neq i$, $\lambda_{j,i}$ denotes redirected user requests, where CDN i is the primary and CDN j is a peer. On the other hand, for $\forall j = i$, $\lambda_{j,i}$ denotes the user requests to a given primary CDN i. For example, request to CDN B for CDN A's content can be denoted as $\lambda_{B,A}$.

Inside each request stream, there is FCFS service. Each request stream is assigned a priority. Here, $p = 1,2,...,P$ priority classes of user-requests are assumed. A peer always prioritizes requests from a given primary CDN over its own user requests. However, if a redirected request (higher priority) arrives to a peer when its own user request (lower priority) is being served, it never interrupts the current service. Thus, this priority discipline is non-preemptive during service quantum of user requests. In addition, no redirection is assumed until primary CDN's load reaches a *threshold* load ($\rho = 0.5$). The redirection policies used are: uniform (ULB), minimum waiting time (MLB), probabilistic (PLB) and weighted (WLB).

Waiting Time

The classical result [6] for non-preemptive head-of-the-line (HOL) priority queue can be used to find the waiting time for the p-th ($p = 1,2,\ldots,P$) priority user request:

$$W_p = \frac{W_0}{(1-\sigma_p)(1-\sigma_{p+1})}, \quad \text{where } \sigma_p = \sum_{i=p}^{P} \rho_i \qquad (1)$$

W_0 is the average delay to a particular priority user request due to other requests found in service. It can be expressed as:

$$W_0 = \sum_{i=1}^{P} \frac{\rho_i E[X_i^2]}{2}$$

where $E[X_i^2]$ is the second moment of service time for a customer from class i [10].

Let us assume that the user requests for the primary CDN belongs to the p-th priority class. The Laplace transform of the waiting time for the primary CDN is denoted as $W^*_p(s)$. Using the known solution from [6] for the distribution of waiting time for each priority group in a priority queue, it can be expressed as,

$$W_p^*(s) = \frac{(1-\rho)s + \lambda_L[1 - \sum_{j=1}^{p-1} \frac{\lambda_{j,j}}{\lambda_L} B_j^*(s)]}{s - \lambda_{p,p} + \lambda_{p,p} B_p^*(s)}, \lambda_L = \sum_{j=1}^{p-1} \lambda_{j,j} \qquad (2)$$

Similarly, for any peer with the priority in the range *1,2 ...,(p-1)* the Laplace transform of waiting time is found by,

$$W_j^*(s) = \frac{(1-\rho)[s + \lambda_{p,p} - \lambda_{p,p} G_p^*(s)]}{s - \lambda_L + \lambda_L \sum_{j=1}^{p-1} B_j^*(s + \lambda_{p,p} - \lambda_{p,p} G_p^*(s))} \qquad (3)$$

Here, $G^*_p(s)$ is the transform for the M/G/1 busy period distribution for a p-priority class, which is expressed as,

$$G_p^*(s) = B_p^*(s + \lambda_{p,p} - \lambda_{p,p} G_p^*(s)) \qquad (4)$$

QoS Performance

The P.D.F of the waiting time distribution, through numerical inversion, is used to observe the expected waiting time. As a primary CDN's request has priority over any peer's own user requests, we use equation (2) for a primary CDN, and ideally use equation (3) for any peering CDN. Though these equations are useful for computation, the iterative expression for $G^*_i(s)$ in (4) is impossible to invert numerically. Therefore, the waiting time experienced by a primary CDN's user requests is found using (2), while (1) is used to find the average expected waiting time for a peer's user requests.

V. RESULTS

For our experiments, we consider a system consisting of three peering CDNs, as shown in Figure 4. We assume that all peers hold the content required to serve redirected requests from a given CDN acting as a primary. In this figure, CDN 1 is shown as a primary, while CDN 2 and CDN 3 are acting as

peers. However, these roles may change over time, as the roles are dynamic and interchangeable depending on load.

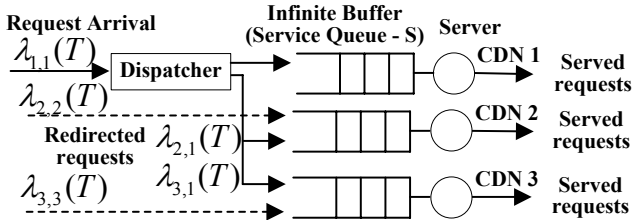


Figure 4: A reference peering scenario.

Each CDN is modeled as an M/G/1 queue with highly variable Hyper-exponential distribution that approximates a heavy-tailed Bounded Pareto service distribution (α, k, p) with variable task sizes. CDNs are arranged according to a non-preemptive HOL priority queuing system. We assume that priority is known upon request arrival at a CDN and that requests are discriminated on the basis of *known* priority. Thus, an incoming request (with priority p) joins the queue behind all other requests with priorities less than or equal to p and in front of all the user requests with priority greater than p.

Table 1. Workload model.

Category	Service Distribution	P.D.F	Range	Parameters
Primary CDN, $0.1 \leq \rho \leq 0.9$	Hyper-exponential	$h_n(t)$ approximating $f(x)$	$x \geq k$	$\alpha = 1.5$ $k = 1010.15$ $p = 10^{10}$
Peer 1, $\rho = 0.5$	Hyper-exponential	$h_n(t)$ approximating $f(x)$	$x \geq k$	$\alpha = 1.5$ $k = 1010.15$ $p = 10^{10}$
Peer 2, $\rho = 0.4$	Hyper-exponential	$h_n(t)$ approximating $f(x)$	$x \geq k$	$\alpha = 2$ $k = 1500.23$ $p = 10^{10}$

The workload model reflects the highly variable and self-similar nature of Web access. Peer 1 and peer 2 are set to $\rho = 0.5$ and $\rho = 0.4$ respectively. These light loads on the peers help to emphasize the performance of a given primary CDN in a peering relationship by tuning its load ($0.1 \leq \rho \leq 0.9$). Table 1 shows the distributions, probability density functions and parameter ranges for the workload model. For our experiments, we consider the expected waiting time as an important parameter to evaluate the performance of a given primary CDN. We also assume an SLA of serving all user requests by the primary CDN in less than *20000* time units.

A. QoS Performance of the Primary CDN

The C.D.F of the waiting time distribution of the primary CDN can be used as a QoS performance metric due to the highly variable nature of the request workload. The waiting time corresponds to the time elapsed by a user request before being served by the CDN. Figure 5 shows the C.D.F of waiting time of the primary CDN (CDN 1 in this case) at different loads, with and without peering. From the figure we see that for a load $\rho = 0.6$, in the non-peering case, there is about *55%* probability that users will have a waiting time less than the threshold of *20000* time units, whereas peering ensures *80%* probability. Therefore, the primary CDN achieves a QoS performance improvement of about *31%* through peering. Again, for a moderate load $\rho = 0.7$, there is about *50%* probability and about *81%* probability that users will have waiting time below the threshold, in non-peering and peering respectively. Thus, it leads to a performance improvement of about *38%*. Similarly, for a heavy load $\rho = 0.9$, the probability in non-peering is about *24%*, which is increased to *70%* in the peering case, providing a performance improvement of about *65%*. Moreover, for loads $\rho > 0.9$, still higher improvement can be predicted using the model. Based on these observations, we posit that peering between CDNs, irrespective of any particular request-redirection policy, achieves substantial QoS performance improvement over the non-peering case.

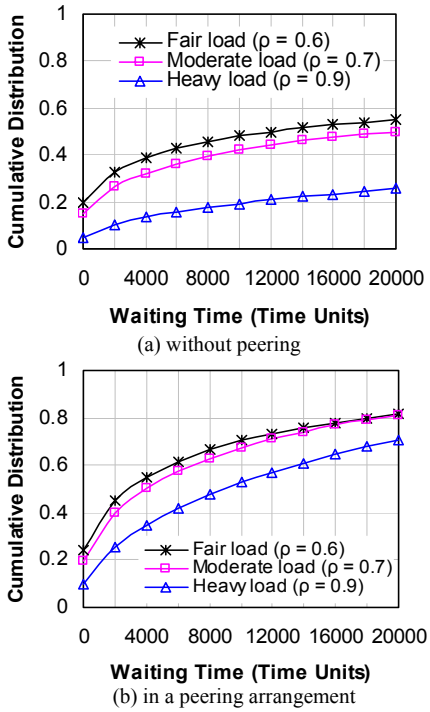


Figure 5: CDF of waiting time of the primary CDN.

B. Impact of Request-Redirection

Without request-redirection, when a given primary CDN's load approaches to *1.0*, the user perceived performance on a given primary CDN tends to infinity. With redirection, the waiting time of the primary CDN decreases as excess requests are offloaded to the peers. However, request-redirection may lead to temporary overload on certain peer(s).

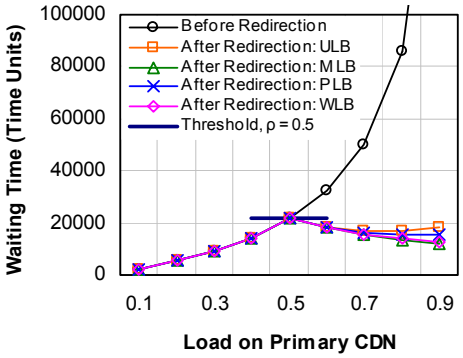


Figure 6: Impact of request-redirection on waiting time.

Figure 6 shows the performance improvement (in terms of waiting time) the primary CDN gains for different request-redirection policies. Here, we compare the waiting time as a

function of system load. It can be noted that a weighted average value of waiting time is presented to capture the effect of request-redirection. For all the four policies mentioned earlier, we observe that substantial performance improvement is achieved in the waiting time when compared to the non-peering case. ULB, PLB and WLB request-redirection policies distribute redirected requests among peers according to certain percentages. Therefore, to some extent they exhibit similar characteristics. In our experiments, ULB uses *50%-50%* distribution of redirected requests to peer 1 and peer 2. The use of MLB by the dispatcher assigns all the redirected requests to peer 2, which has the minimum expected waiting time. Hence, no redirected request is assigned to peer 1. When the dispatcher uses PLB, it leads to a distribution of *40%-60%* to peer 1 and peer 2 respectively. A dispatcher following the WLB policy assigns *80%* of redirected requests to peer 2 (with minimum expected waiting time) and *20%* to peer 1.

Table 2: Reduction on waiting time under different redirection policies.

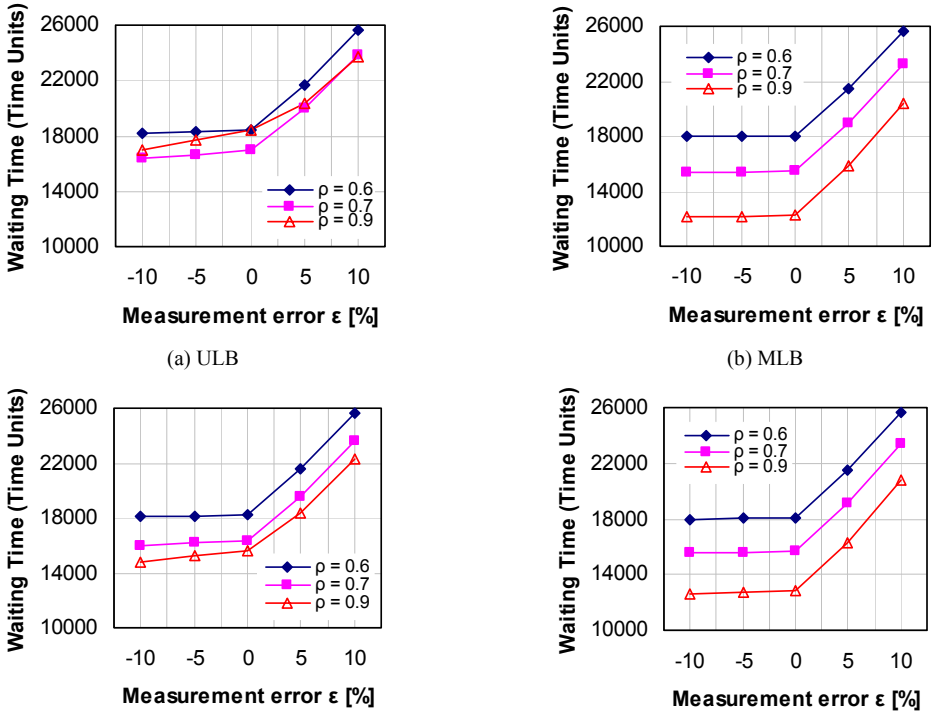
Load on primary CDN	**Reduction in waiting time %**			
	ULB	**MLB**	**PLB**	**WLB**
Fair load, $\rho = 0.6$	*43.20%*	*44.41%*	*43.66%*	*44.24%*
Moderate load, $\rho = 0.7$	*66.31%*	*69.31%*	*67.50%*	*68.91%*
Heavy load, $\rho = 0.9$	*90.52%*	*93.70%*	*91.94%*	*93.39%*

Table 2 summarizes the reduction of waiting time for the primary CDN in peering for different request-redirection policies. Interestingly, results for MLB follow more or less similar trend as other three policies and show good enough performance due to light loads on peers. However, in MLB, there is the risk that the peer with minimum expected waiting time could become overloaded with the redirected requests (herd effect). From the results, it is clear that all the request-redirection policies produce a maximum waiting time less than *20000* time units. This confirms that redirecting only a certain fraction of requests reduces instability and overload in the system because the peers are not overwhelmed by bursts of additional requests.

C. Measurement Errors

The dispatcher makes its redirection decision based on the measured value of a given primary CDN's load. So far we have assumed that perfect information is available for this decision. However, the dispatcher can have inaccurate load information due to delays in receiving the measurements. Therefore, in this section we study the impact of measurement errors on the effectiveness of the redirection policies. Let us denote the measured load of the primary CDN as $\tilde{\rho} = \tilde{\lambda}E[X]$, where $\tilde{\lambda} = \lambda(1 \pm \varepsilon)$ and ε is the percentage of the correct load ρ.

Figure 7 shows the impact of load measurement error on the waiting time for different request-redirection policies. Each curve in the figure denotes an average waiting time over all the requests for different primary CDN load ρ, and the x-axis denotes the measurement error ε, in percent of ρ. In all the four cases, for measurement error $\varepsilon > 0$, the dispatcher assumes the primary CDN's load to be higher than what it is and hence it redirects more requests than the actual load. These extra redirections introduce additional waiting time for the user requests and causes the waiting time to increase linearly from $\varepsilon = 0$. For negative ε, the dispatcher assumes the primary CDN's load to be less than the actual and hence redirects pessimistically. As a result, requests on the primary CDN experience greater expected waiting time for being processed. However, the average of waiting time normalizes it to keep the performance at an acceptable level. It is also observed that in the same scenario, users experience is poorer under positive values of ε when compared to negative ε. We thus conclude that greater accuracy is needed in load measurement of the primary CDN.



(a) ULB

(b) MLB

(c) PLB

(d) WLB

Figure 7: Waiting time for load measurement errors ε for different redirection policies.

VI. Critical Evaluation

Although our approach can be assistive for peering between CDNs, there are a number of challenges, both technical and non-technical (i.e. commercial and legal), that could hinder its rapid growth. These challenges must be dealt to promote CDN peering. For CDNs to peer, they need a common protocol to define the technical details of their interaction as well as the duration and QoS expected during the peering period. The proprietary nature of a CDN to gain competitive advantage in the market may block off the nascence of peering CDNs in commercial domain. Furthermore, there can often be complex legal issues involved (e.g. embargoed or copyrighted content) that could prevent CDNs from arbitrarily cooperating with each other. Finally, there may simply be no compelling commercial reason for a large CDN provider such as Akamai to participate in CDN peering, given the competitive advantage that its network has the most pervasive geographical coverage of any commercial CDN provider. However, it is expected that our approach can be beneficial and applicable in research-based academic CDN domain where the main focus is not on whether such peering *will* emerge in reality, which mostly depends on the key players in commercial CDNs domain that we cannot divine, but rather on whether such peering *could* emerge.

Although the performance models in this context are simplified in order to accommodate the system complexities, we believe that our models provide a foundation for performing effective peering between CDNs though achieving target QoS in service delivery to end-users. Since the peering CDNs retain load-balancing control within their own Web server sets, using our approach a primary CDN can realize the QoS performance it can provide to the end-users, without requiring individual partners to provide expected service performance from it. Our model-based approach is important, since having each CDN provider communicate how it would service millions of potential end-users, would introduce significant scalability issues, and requesting this information from each partnering provider at the user requests time would introduce substantial delays. Thus, we believe that our approach seeks to achieve scalability for a CDN in a user transparent manner.

VII. Conclusions And Future Work

In this paper we have presented an innovative analytical model for peering CDNs. In this model, an overloaded CDN is stabilized by offloading a fraction of the incoming content requests to the peers. Through the presented performance models we have demonstrated the effects of peering and predicted end-user perceived performance from a given CDN. We have also showed that it is easier to meet QoS goals using peering between CDNs, and that any resulting system is more resilient to flash or periodic crowds. Our model-based approach can be used within a live system so that load distribution decisions can be evaluated in the model prior to deployment, improving scalability and reducing reliance on outdated load information. Our approach seeks to achieve scalability for a CDN in a user transparent manner.

Our future work[1] includes performing an advanced system analysis to study the impact of other performance parameters such as network latency and cost of peering. In this regard, we would like to perform realistic experiments in the real-world settings, such as PlanetLab, to validate the methodology presented in this paper. Our future work also includes developing a proof-of-the-concept implementation for demonstrating the real-time application of our approach for peering between CDNs.

We expect that our methodology for modeling peering CDNs and predicting performance of a given CDN provider in a peering arrangement will be a timely contribution to the current content networking trend.

[1] For more information on our efforts on peering CDNs, please visit the project Web site at: http://www.gridbus.org/cdn

References

[1] Adler, S. The SlashDot effect: An analysis of three Internet publications. *Linux Gazette*, 38, 1999.

[2] Amini, L., Shaikh, A., and Schulzrinne, H. Effective peering for multi-provider content delivery services. In *Proc. of 23rd Annual IEEE Conference on Computer Communications (INFOCOM'04)*, pp. 850-861, 2004.

[3] Biliris, A., Cranor, C., Douglis, F., Rabinovich, M., Sibal, S., Spatscheck, O., and Sturm, W. CDN brokering. *Computer Communications*, 25(4), pp. 393-402, 2002.

[4] Bouman, J., Trienekens, J., and Zwan, M. Specification of service level agreements, clarifying concepts on the basis of practical research. In *Proc. of the Software Technology and Engineering Practice Conference*, pp. 169, 1999.

[5] Broberg, J., Zeephongsekul., P., and Tari, Z. Approximating bounded general service distributions, In *Proc. of IEEE Symposium on Computers and Communications*, Jul. 2007.

[6] Cobham, A. Priority assignment in waiting line problems. *Journal of the Operations Research*, 2(1), pp. 70-76, 1954.

[7] Crovella, M. E. and Bestavros, A. A self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6), 1997.

[8] Crovella, M. E., Taqqu, M. S., and Bestavros, A. Heavy-tailed probability distributions in the World Wide Web. *A Practical Guide To Heavy Tails*, Birkhauser Boston Inc., Cambridge, MA, USA, pp. 3-26, 1998.

[9] Day, M., Cain, B., Tomlinson, G., and Rzewski, P. A model for content internetworking. IETF RFC 3466, Feb. 2003.

[10] Kleinrock, L. *Queuing Systems. Vol. II: Computer applications*. John Wiley & Sons, pp. 15-19, 1975.

[11] Padmanabhan, V. N. and Sripanidkulchai, K. The case for cooperative networking. In *Proc. of International Peer-To-Peer Workshop (IPTPS'02)*, 2002.

[12] Pathan, M., Broberg, J., Bubendorfer, K., Kim, K. H., and Buyya, R. An architecture for virtual organization (VO)-based effective peering of content delivery networks, UPGRADE-CN'07, In *Proc. of the 16th IEEE International Symposium on High Performance Distributed Computing (HPDC'07),* Monterey, California, USA, Jun. 2007.

[13] Pathan, M., Broberg, J., and Buyya, R. An approach for QoS-driven performance modeling of peering CDNs. Technical Report, GRIDS-TR-2007-19, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Oct. 2007.

[14] Pathan, M., Buyya, R., and Vakali, A. CDNs: state of the art, insights, and imperatives. *Content Delivery Networks*, R. Buyya et al. (Eds.), Vol. 9, Springer-Verlag, Germany, 2008.

[15] Turrini, E. An architecture for content distribution internetworking. Technical Report. UBLCS-2004-2, University of Bologna, Italy, Mar. 2004.

[16] Willinger, W. and Paxson, V. Where mathematics meets the Internet. *Notices of the American Mathematical Society*, 45(8), pp. 961-970, 1998.