Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Communications 27 (2004) 1401–1411
computer communications
www.elsevier.com/locate/comcom

Modeling content delivery networks and their performance

Benjamin Molina*, Carlos E. Palau, Manuel Esteve

Department of Communication, Polytechnic University of Valencia (UPV), C/ Camino de Vera s/n 46022 Valencia, Spain

Received 31 July 2003; revised 19 May 2004; accepted 21 May 2004

Available online 15 June 2004

Abstract

Content Distribution Networks (CDN) have recently appeared as a method for reducing response times experienced by Internet users through locating multiple servers close to clients. Many companies have deployed their own CDN—and so demonstrating the resulting effectiveness. However, many aspects of deployment and implementation remain proprietary, evidencing the lack of a general CDN model to help the research community analyze different working scenarios. In this paper, we propose a general expression for a content distribution environment and study the performance impact of design variables such as caching hit ratios, network latency, number of surrogates, and server capacity. Our conclusions are supported with simulations results.

Keywords: Content distribution network; Modeling and simulation; Performance evaluation

1. Introduction

Few things compare with the growth of the Internet over recent years. A key challenge for Internet infrastructure has been delivering increasingly complex data of different types and origin to a growing user population. The need to scale has led to the development of clusters [1], global content delivery networks (CDN) [2] and, more recently, P2P structures [3]. However, the architecture of these systems differs significantly, and the differences affect their performance, workloads, and the role that caching can play [4,5].

CDNs are overlay networks across the wide-area Internet which consist of dedicated collections of servers, called surrogates, distributed strategically throughout the Internet. The main aim of the surrogates is to be close to users and provide them with content in a low-latency mode. The surrogates are normally proxy caches that serve cached content directly with a certain hit ratio; the uncached content is previously obtained (if possible) from the origin server before responding. When a client makes a request for content inside a CDN, it is directed to an optimal surrogate, which serves this content within low response time boundaries—at least compared to contacting the origin site [6,7]. CDNs such as Akamai [8] or Digital Island [9] are nowadays used by many websites as they effectively reduce the client-perceived latency and balance load [10]. They accomplish this by serving content from a dedicated, distributed infrastructure located around the world and close to clients. The content is replicated either on-demand, when users request it, or replicated beforehand, by pushing the content on the content servers [11,12]. CDN services can improve client access to specialized content by assisting in four basic areas:

- *Speed*, reducing the response and download times of site objects (e.g. streaming media), by delivering content close to end users.
- *Reliability*, by delivering content from multiple locations; a fault-tolerant network with load balancing mechanisms can be implemented.
- *Scalability*, both in bandwidth, network equipment and personnel.
- *Special events*, by incrementing capacity and peak loads for special situations by distributing content as it is needed [13].

CDNs improve performance and availability of web and some media content by pushing the content towards the network edges and providing replication and replica location services. Intelligent replica placement improves

* Corresponding author. Tel.: +34-96-387-7301; fax: +34-96-387-7309.
E-mail addresses: benmomo@doctor.upv.es (B. Molina), cpalau@dcom.upv.es (C.E. Palau), mesteve@dcom.upv.es (M. Esteve).

0140-3664/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2004.05.012

response time by serving content from a topological location near the client (in terms of network hops), avoiding the congested backbone networks and network access [14]. Replica location services direct requests for objects to nearby replicas by means of redirections through DNS, based on extensive measurements and monitoring of network performance [15]. The overall performance of a CDN is largely determined by its ability to direct client requests to the most appropriate server [10,16,17]. Content providers, such as websites or streaming video sources, contract with commercial CDNs to host and distribute content [18]. They are attractive for content providers because in some cases the responsibility is offloaded to the CDN infrastructure. Most CDNs have servers in ISP points of presence, so clients can access topologically nearby clients with very low latencies. They are capable of sustaining large workloads and flash-crowds due to a large number of servers (Akamai), or few but powerful servers (Digital Island) [11]. The main features of a CDN are:

- Decentralizes content storage by moving content closer to clients.
- Preserves WAN bandwidth by delivering content locally, and maximizes user performance.
- Content management tools help optimize network performance and prioritize mission critical data [19].

CDNs are perfectly integrated in web architecture and the minimum unit managed by them is an object, which are named by URLs. Unlike the web, content providers do not need to manage web servers, since client requests are redirected to replicas hosted by the CDN [12]. CDNs typically host static content (images, advertisements, media clips, etc.) although dynamic content could contain embedded objects served by the CDN [20].

The rest of the paper is structured as follows. Section 2 introduces the motivation and previous work. In Section 3 we present our CDN model starting from previous work. In Section 4 this model is further described and interpreted in an illustrative way for a better understanding. Section 5 presents and explains simulation results obtained from the model. The paper finishes with the conclusions and future work.

2. Motivation and previous work

CDNs are overlay networks on top of the Internet that deliver content to users from the network edges, thus reducing response time compared to obtaining content directly from the origin server, as depicted in Fig. 1.

If client 1 downloads content from a certain site, it traditionally contacts a centralized server, located in the origin site. The communication may traverse several ISPs and WANs, thus being unable to predict content latency and jitter. If the desired content requires some temporal

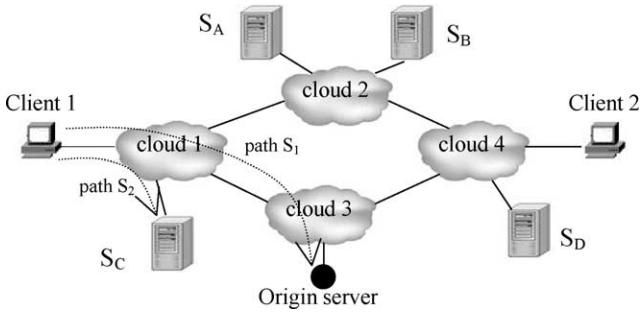

Fig. 1. General CDN scenario on the Internet.

contraints, a decentralized solution is mandatory in order to avoid, or at least reduce, network unpredictability. Following the example of Fig. 1, if client 1 contacts surrogate S_C to retrieve content (path S_2), it will perceive a reduction in the response time compared with contacting the origin server (path S_1). Furthermore, network usage is reduced and optimized on an S_2-path, as backbone traffic is reduced.

As can be seen, reducing response time implies reducing network latency and decreasing server processing time. Due to a lack of a global management of the Internet, companies have traditionally scaled-up with a more powerful server or scaled-out locally in cluster-based architectures. A CDN is a global scale-out approach that tries to reduce network latency by avoiding congestion paths, thus resulting in a reduction of perceived response time. Leading CDN companies have placed from hundreds up to thousands of servers throughout the world, being able to serve content to any client from a nearby surrogate. Correctly managing such huge content networks is extremely important.

Previous research has investigated the use and effectiveness of CDNs, although the proprietary and closed nature of these systems tends to impede investigation [11]. Recent studies confirm that CDNs reduce average download response time, but that DNS redirection techniques add noticeable overhead because of DNS latencies. Most of these studies have been empirical ones, studying and evaluating response times on real CDNs [14,17], effectiveness of DNS redirection [14,21,22], server selection [16,17], or server location [1]. Other papers and contributions have tried to model the behaviour of CDNs using different techniques, but usually simplifying the nature of such systems and losing generality. These studies have focused mainly in two topics: server placement [23] or evaluation of response time. Different techniques have been used for the evaluation of the second parameter: linear time response of web servers [24], water filling schema [25] or queuing theory [26]. This paper starts from basic assumptions in [24] and introduces new features that resemble a more realistic and general model. This way we can study the impact on performance of important parameters such as the caching hit ratio, network latency, number of surrogates and server capacity. Moreover, it is worthwhile establishing some dependencies and relationships among these parameters,

as simulation results are always finite and cover only a limited variability range of the CDN parameters.

3. The CDN model

The CDN model is composed of three main elements, as shown in Fig. 2.

- One *origin server*, placed at a central location
- *P surrogates*, placed somewhere between clients and the origin server, and
- *M client clusters*, dispersed throughout the world. A client cluster is a way of joining multiple single users located within a certain zone. We will analyze this scenario, obtaining therefore results for the whole group and not for single users. However, our general approach enables the easy extraction of a model based on single clients.

As a first approach, we can assume that all client clusters (in the following also clients) are uniformly distributed in a circle with roundtrip-time (RTT) τ_0 around the origin site. The surrogates between origin and clients are τ_d and τ_p away respectively (in terms of RTT). A client generates requests, that are served either by the origin server or surrogates. The mechanism telling the client which is the suitable server to contact is not considered; instead we assume that requests will be routed to the surrogates with a certain probability p. Otherwise, the client contacts the origin server with the opposite probability $(1-p)$.

The normal performance metric considered in all CDN analysis is the mean response time experienced by the users. To start with an initial formula we can consider the following expression for the mean response time:

$$R = pR_{\text{srrgt}} + (1-p)R_{\text{origin}} \tag{1}$$

where R_{srrgt} is the mean response time associated with contacting a surrogate (and be served by it) and R_{origin} is the average response time associated with contacting the origin server (and be served by it). Further on, the mean response time can be represented [1] using a linear model as:

$$R = N\tau + S \tag{2}$$

where N is a scaling factor that incorporates the effect of network loss rates, retransmission and, in general, the volume of exchanged data required for the request; τ is the network latency factor (round-trip-time) associated and S is the request processing time, which will be modelled as a M/M/1 queue system. An M/M/1 queue [27] consists of a FIFO buffer with packets arriving randomly according to a Poisson process and a processor, called a server, that retrieves packets from the buffer at a specified service rate, as depicted in Fig. 3 (Table 1).

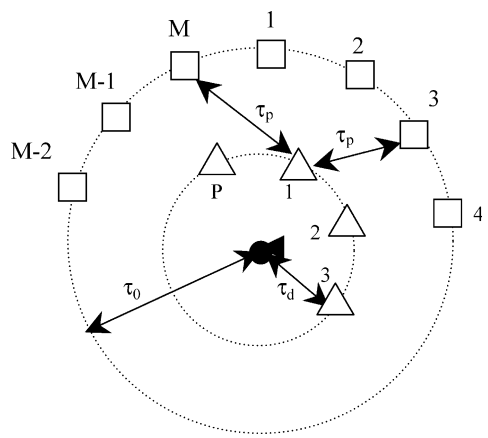

Fig. 2. Simple CDN scenario.

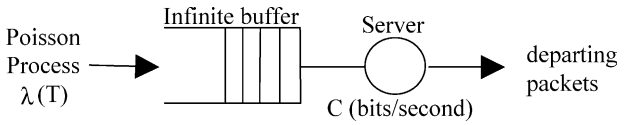

Fig. 3. Model of a M/M/1 Queue.

So it follows for the average response time that:

$$R = p\left[N\tau_p + \frac{1}{\mu_p - \lambda_p}\right] + (1-p)\left[N\tau_s + \frac{1}{\mu_s - \lambda_s}\right] \tag{3}$$

The appearing variables in the above expression will be described separately for a better comprehension.

The variable p is the probability a request will be satisfied by a surrogate. However, a client will not always be routed to the same surrogate. Although theoretically it would be desirable for a client to be redirected to its closest surrogate, it is a fact that the common routing mechanism (DNS redirection) does not always correctly guess client location, thus redirecting it to another surrogate. Even supposing a correct estimation, it could be useful to balance between near surrogates due to overloading conditions. So we can represent the value of p for the i-th cluster as the sum of the probabilities of contacting the P surrogates:

$$p_i = \sum_{j=1}^{P} p_i^j \tag{4}$$

The variable τ_p is the latency associated with contacting the surrogates. Note that this value is highly variable and different for each client cluster. Thus, the latency value for the i-th cluster to the j-th surrogate will be expressed as $\tau_p^{i,j}$.

Table 1
Parameter and expressions for an M/M/1 queue system

Parameter	Expression
mean arrival rate λ	$\lambda = (1/T)$(packets/second)
mean arrival time T	
Mean packet size	$\bar{p}$ (bits/packet)
Service capacity	C (bits/s)
Mean service rate	$\mu = (C/\bar{p})$(pakets/second)
Mean delay	$\bar{W} = (1/(\mu - \lambda))$(seconds/packet) with $\mu > \lambda$

The variables μ_p and λ_p are the mean service rates and the perceived incoming mean arrival rate at each surrogate. Once again, these values may be different per surrogate. Supposing that the i-th cluster generates packets at rate λ_i, then the arrival rate perceived by the j-th surrogate from cluster i will be:

$$\lambda_i^j = p_i^j \lambda_i \tag{5}$$

Due to the stability condition ($\mu_p > \lambda_p$) we could just get a factor $k(k > 1)$ so that $\mu_p = k\lambda_p$. The value of k is important: a low value can limit capacity conditions where processing time is slow; a high value can suppose a practically instantaneous processing time, where only latency values significantly affect the global response time. The same considerations can be applied to the origin site for the variables p, τ_s, μ_s and λ_s. However, this time the origin site is absorbing M exponential distributions from each client cluster, which can be modelled as another exponential distribution with a new arrival rate as the sum of each individual arrival rate.

Before coming to the general expression, it is important to note that the previous formula enables the response time to be obtained for both for a client cluster and for the overall system. The latter case, which will be here considered, is the average of the mean response time obtained for all M clusters:

$$\bar{R} = \frac{1}{M}\sum_{i=1}^{M} R_i \tag{6}$$

So we can present our general expression as:

$$\bar{R} = \bar{R}_s + \bar{R}_0 \tag{7}$$

with the following expressions for $\bar{R}_s$ and $\bar{R}_0$:

$$\bar{R}_s = \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{P}\left[p_i^j\left(N\tau_p^{i,j} + \frac{1}{\mu_p^j - \sum_{l=1}^{M}\lambda_l^j}\right)\right] \tag{8}$$

$$\bar{R}_0 = \frac{1}{M}\sum_{i=1}^{M}\left(1 - \sum_{j=1}^{P} p_i^j\right)\left(N\tau_0^i + \frac{1}{\mu_s - \sum_{l=1}^{M}(1-p_l)\lambda_l}\right) \tag{9}$$

where M is the number of client clusters, P is the number of surrogates, p_i^j stands for the probability of the i-th client cluster contacting the j-th surrogate, N represents the number of required packets for a client-server transaction, $\tau_p^{i,j}$ is the mean roundtrip time between the i-th cluster and the j-th surrogate, μ_p^j stands for the mean service rate of the j-th surrogate, λ_l^j represents the mean arrival rate that the l-th cluster sends to the j-th surrogate, τ_o^j is the mean RTT between the i-th cluster and the origin site, μ_s stands for the mean service rate of the origin site, and $(1 - p_l)\lambda_l$ represents the mean arrival rate that the l-th cluster sends to the origin site.

4. Understanding the model

Before translating our expression into simulation scenarios, it could be of interest to better understand it by using some graphical and mathematical techniques.

As there are M client clusters and P surrogates to serve them, one may ask whether $P = M$, just to associate one optimum surrogate close to each client cluster. However, it is not necessarily true as at the design phase it is extremely difficult to define client clusters. Even so, one may assign various surrogates for the same client cluster ($P > M$), if there is a need to serve content by client profile, which could be the business situation of premium and normal users. If a CDN provider is deploying its network, it is cost-safe to begin with few surrogates ($P < M$) and later grow by aggregating more servers. Independently of the business strategy, technical reasons such as congested servers due to flash crowds suggest serving content from different surrogates.

Let's start from the situation depicted in Fig. 4, composed of 2 client clusters and 3 surrogates.

Client 1 generates requests with a mean rate of λ_1. As it is served by various surrogates, the general approach is to suppose that a certain percentage of requests will be served by surrogate P, and another by surrogate P_2 and P_3. A certain number of requests with probability $1 - p_1 (p_1 = p_1^1 + p_1^2 + p_1^3)$ will be directed to the origin server. As can be logically deducted from the previous picture (Fig. 4), the probabilities of contacting the surrogates vary. Each client will usually contact its closest surrogate, and a hierarchy in the assigned weights by distance is established. So, if client 1 is next to surrogate P_1, then the nearest surrogate is P_2 and then P_3, it follows that $p_1^1 > p_1^2 > p_1^3$. We can therefore construct a matrix that contains all the probabilities of the surrogates, where the row indicates the client cluster

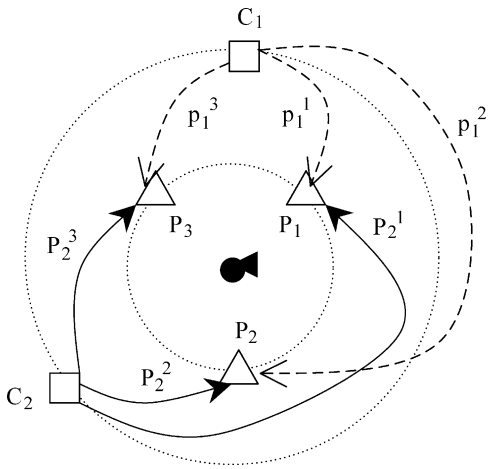

Fig. 4. CDN model for 2 clients and 3 surrogates.

and the column is associated with the correspondent probability of contacting the j-th surrogate. In the case of our system, an example could be:

$$P_p = \begin{bmatrix} 0.4 & 0.1 & 0.2 \\ 0.0 & 0.3 & 0.2 \end{bmatrix} \tag{10}$$

Note that the sum of each row is $\neq 1$. This is because a certain number of requests are directed to the origin server (those that have not been previously cached). So we can construct a vector from the mean caching hit ratios that specify the probability of contacting the origin server by each client. Following the example, we have:

$$\vec{p}_s = \begin{bmatrix} 1-(0.4+0.1+0.2) \\ 1-(0.0+0.3+0.2) \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.5 \end{bmatrix} \tag{11}$$

Generalizing the expression for M clients and P surrogates, we obtain a matrix with M rows and P columns and a vector with M rows.

$$P_p = \begin{bmatrix} p_1^1 & p_1^2 & p_1^3 & \cdots & p_1^P \\ p_2^1 & p_2^2 & p_2^3 & \cdots & p_2^P \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ p_M^1 & p_M^2 & p_M^3 & \cdots & p_M^P \end{bmatrix}_{M\times P} \tag{12}$$

$$\vec{p}_s = \begin{bmatrix} 1-\sum_{j=1}^{P} p_1^j & 1-\sum_{j=1}^{P} p_2^j & \cdots & 1-\sum_{j=1}^{P} p_M^j \end{bmatrix}^T \tag{13}$$

Another aspect to consider is that a client may never contact a specific surrogate ($p_i^j = 0$ for a certain i,j). For example, client 2 will not retrieve any content from surrogate P_1. This is due to the fact that (in terms of latency) the origin server is nearer than surrogate P_1, that is $\tau_p^{2,1} > \tau_0^2$. This relationship can be used as a criterion to assign the probability weights, so that

$$\begin{cases} \text{if } \tau_p^{i,j} < \tau_p^{k,l} \rightarrow p_i^j > p_k^l, & \forall i,k \in [1,\ldots,M],\ \forall j,l \in [1,\ldots,P] \\ \text{if } \tau_p^{i,j} > \tau_0^i \rightarrow p_i^j = 0, & \forall i \in [1,\ldots,M],\ \forall j \in [1,\ldots,P] \end{cases} \tag{14}$$

The behaviour can be illustrated in Fig. 5.

Normally, a client contacts few surrogates—and mainly those that are nearby. A threshold can be assigned introducing a factor $\alpha(\alpha < 1)$ so that $\tau_p^{i,j} < \alpha\tau_0^i$. If the number of surrogates is high, then reducing the value of α also reduces the assigned surrogates. In the example of Fig. 5, a reduction from $\alpha = 1$ to $\alpha = 0.8$ supposes contacting only 3 servers instead of 5. On the contrary, if the number of surrogates is small it should be possible to obtain content from distant surrogates ($\alpha > 1$). This situation is only under conditions of severe congestion in the origin server and nearby surrogates.

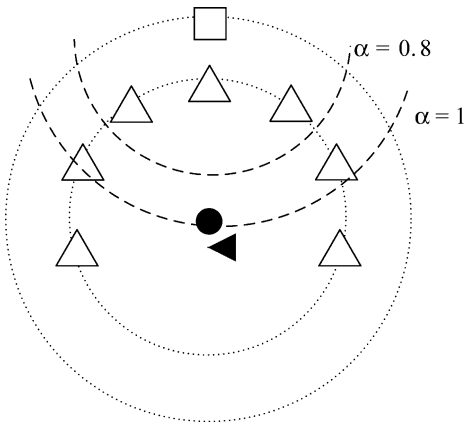

Fig. 5. Latency threshold for a client cluster.

5. Simulating the model

As can be appreciated, the expression for the main response time corresponds to an n-dimensional function represented with a non-fixed value of n. For example, the probability of a client contacting a surrogate, represented by p_i^j, is really a set of variables whose size depends on the number of clients (M) and surrogates (P). Similar behaviour occurs with latencies ($\tau_p^{i,j}$, τ_0^j), capacities (μ_p^j, μ_s) and mean traffic rates (λ^i). So we will simulate a CDN for various values of M clients and P surrogates and we will try to extract some general conclusions. The simulation procedure has to consider the following variables as input parameters:

- The number of clients (M). The clients will be uniformly distributed around the origin server and mutually separated by ($2\pi/M$) radians. To introduce some randomness, the first client is shifted by a random angle (uniformly distributed) between 0 and ($2\pi/M$) radians, and the rest are just placed equidistantly. So it follows:

$$\text{client } 0: \ \alpha_{c0} = \text{rand}\left(0, \frac{2\pi}{M}\right) \tag{15}$$

$$\text{client } i: \ \alpha_{ci} = \frac{2\pi}{M} i + \alpha_{c0}, \qquad i = 0,\ldots,M \tag{16}$$

  Note that the first client begins with index 0, while in previous pictures it had index 1. This is only done to enable a simpler mathematical description. Analogous treatment will happen with the surrogates.
- The number of surrogates (P). The same randomizing procedure as with clients happens with the surrogates:

$$\text{surrogate } 0: \ \alpha_{s0} = \text{rand}\left(0, \frac{2\pi}{P}\right) \tag{17}$$

$$\text{surrogate } j: \ \alpha_{sj} = \frac{2\pi}{P} j + \alpha_{s0}, \qquad j = 0,\ldots,P \tag{18}$$

- The number of necessary retransmissions (N).
- A certain hit ratio per each client cluster ($p_i, i = 1,\ldots,M$).

- A minimum and a maximum value of the latency between each client and the origin server ($\tau_0^{\min}, \tau_0^{\max}$). As each client must not be the same latency away from the server, a random value between this interval will be taken for each client (τ_0^j).
- A minimum and a maximum value of the latency between each surrogate and the origin ($\tau_d^{\min}, \tau_d^{\max}$). All surrogates are not the same value $\tau_d^j (j = 1, \ldots, P)$ away from the origin server, so a random value will be taken between these two boundaries.
- A minimum and a maximum value of the mean traffic rate sent by each client ($\lambda_{\min}, \lambda_{\max}$), as they can have different traffic characterizations. Once again, a random value between minimum and maximum will be taken.
- A threshold factor (α) indicating the area or zone for each client to contact surrogates, as described previously in Fig. 5. Surrogates placed outside this area will not serve any content to the associated client.
- A capacity factor (k) representing the necessary increase with respect to the mean arrival rate, so that the stability condition for an M/M/1 queue is satisfied ($\mu > \lambda$). Though it could be set differently for each surrogate, we will take the same value of the parameter k for all of them.

Fig. 6 shows an example for 8 clients and 4 surrogates. Note that both clients and surrogates do not have the same distance (in terms of latency) from the origin server. Each are inside a ring whose thickness is determined by $[\tau_0^{\min}, \tau_0^{\max}]$ and $[\tau_d^{\min}, \tau_d^{\max}]$, respectively. However, the distance in radians between them is deterministic, as both are uniformly distributed around the ring depending on the values of the parameters M and P.

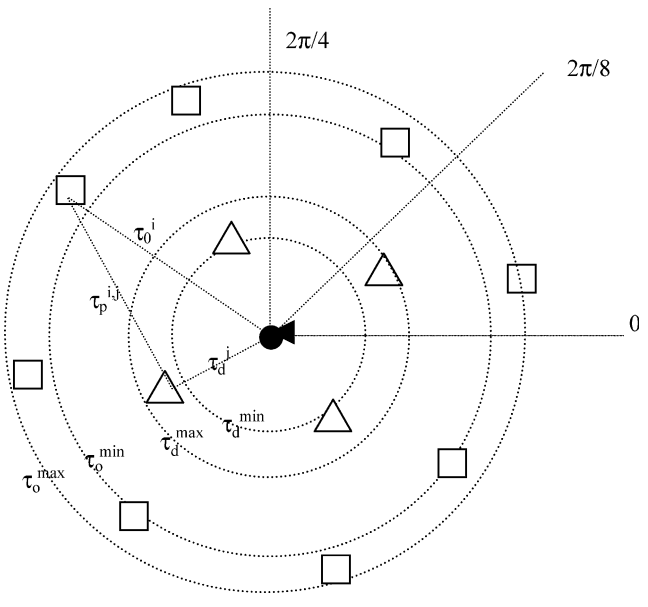

Fig. 6. CDN structure for $M = 8$ and $P = 4$.

With the above geometric distribution the distance between a client i and a surrogate, j can be automatically calculated applying the cosinus theorem, without introducing it as an input parameter. So it follows that:

$$\tau_p^{i,j} = (\tau_o^j)^2 + (\tau_d^j)^2 - 2\tau_o^i \tau_d^j \cos\beta^{i,j} \tag{19}$$

$$\beta^{i,j} = |\alpha_{ci} - \alpha_{sj}| = \left|\left(\frac{2\pi}{M} i + \alpha_{c0}\right) - \left(\frac{2\pi}{P} j + \alpha_{s0}\right)\right| \tag{20}$$

The even property of the cosinus function makes it unnecessary to use the absolute module.

In this way, we can create the following matrix and vector:

$$T_p = \begin{bmatrix} \tau_1^1 & \tau_1^2 & \tau_1^3 & \cdots & \tau_1^P \\ \tau_2^1 & \tau_2^2 & \tau_2^3 & \cdots & \tau_2^P \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \tau_M^1 & \tau_M^2 & \tau_M^3 & \cdots & \tau_M^P \end{bmatrix}_{M \times P} \tag{21}$$

$$\vec{\tau}_s = \begin{bmatrix} \tau_0^1 & \tau_0^2 & \cdots & \cdots & \tau_0^M \end{bmatrix}^T \tag{22}$$

The latency matrix T_p is related with the probability matrix mentioned in the previous chapter P_p, that is, P_p can be automatically obtained from T_p, $\vec{\tau}_s$, α and the *caching hit ratio*. The method to obtain P_p is very simple. For each row i of the matrix T_p a threshold given by $\alpha\vec{\tau}_s(i)$, described graphically in Fig. 5, discriminates between the surrogates being contacted by the i-th client. These method we can construct a mask matrix M_p (dimension $M \times P$) whose elements $m_p^{i,j}$ comply with the following condition:

$$m_p^{i,j} \begin{cases} 1, & \text{if } T_p(i,j) \le \alpha\vec{\tau}_s(i) \\ 0, & \text{if } T_p(i,j) > \alpha\vec{\tau}_s(i) \end{cases} \tag{23}$$

The mask matrix M_p serves as a starting point (together with T_p) for constructing another intermediate matrix P_w that assigns probability weights for a client i to contact a surrogate j. Therefore, the sum of each row in P_w must be 1. The steps for constructing P_w are the following:

1. Create an auxiliary matrix by multiplying the latency matrix T_p with the mask matrix by components This means, it is not a matrix multiplication (note that the matrix dimensions do not allow it), but a component of component multiplication, resulting in a new $M \times P$ matrix similar to T_p, but discriminating between those latencies that are greater than our impossed threshold. So it follows for this auxiliary matrix:

$$\text{aux}^{i,j} = \tau_p^{i,j} m_p^{i,j} \tag{24}$$

2. As each row sum of M_p must be 1, it may seem sensible to divide each element of a row i by the sum of all the elements of the row However, this process would result in a greater weight for a greater distance, which is not our intention; on the contrary, the link between a client and a nearby surrogate must suppose a greater weight than a more

distant surrogate (distant surrogates have been previously avoided). A method to achieve this is by weighting the difference between latency and the sum of latencies in a row:

$$\tilde{p}_w^{i,j} = 1 - \frac{\text{aux}^{i,j}}{\sum_{j=1}^{P} \text{aux}^{i,j}} \qquad p_w^{i,j} = N(\tilde{p}_w^{i,j}) \tag{25}$$

where the N operator implies normalization, that is, the sum of elements within a row must be 1.

Once the simulation function has been described, it is now time to obtain some graphs and interpret their behaviour. Fig. 7 depicts response times for a possible CDN scenario for various values of caching hit ratios and a reduced number of clients and surrogates. Easy parameter relationships have been established to analytically validate the results. For example, the caching hit ratio is the same for all clients sending traffic with the same mean rate. The parameter α has been set to 1 to be sure that each client would contact at least one surrogate. For simplicity, there is no latency ring for clients nor surrogates.

Note that the mean response time can be seen in two ways: (i) as the response times of the surrogates and the origin, or (ii) as the response time associated with latency and process.

The first approach enables an analysis of working conditions and when to send, or not, content to the surrogates; the second approach indicates where response time is mainly affected by network congestion, or server overload.

As can be seen in Fig. 7, all response times follow a linear behaviour. Once identified, further conclusions can be made. The lowest value of mean response time is obtained for the highest caching hit ratio, that is, when all content is served by the surrogates and none of it by the origin server.

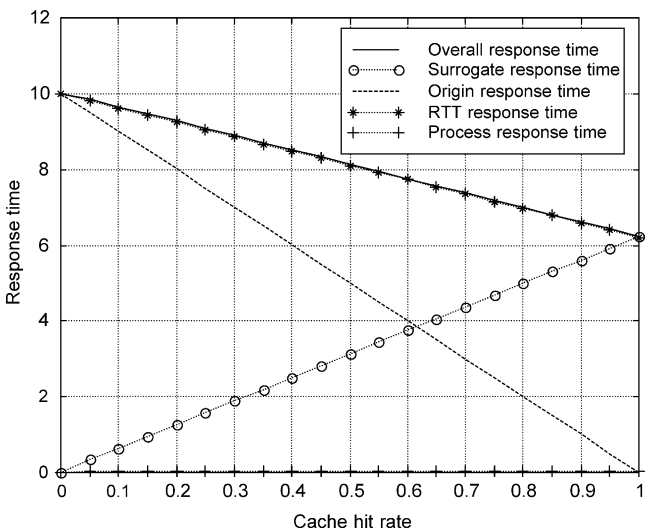

Fig. 7. Response times for $M = 8$, $P = 4$, $N = 5$, $\alpha = 1$, $k = 1.2$, hit_ratio $= 0,\ldots,1$, $\tau_0^{\min} = \tau_0^{\max} = 2$ s, $\tau_d^{\min} = \tau_d^{\max} = 1$ s, $\lambda_{\min} = \lambda_{\max} = 100$.

The reduction of response time is perhaps insignificant in absolute terms, but significant in relative terms (representing a 40% reduction).

Besides, the overall process time is very reduced (nearly imperceptible), so only the roundtrip affects the mean reponse time. Maybe the set value of factor k leads to underload conditions in all servers. Anyway, the latency value of this scenario is considerable for a real CDN.

After the first observations, it would be interesting to support them with a mathematical viewpoint. Let's consider only the response time associated with the roundtrip time ($\bar{R}_t$)

$$\bar{R}_t = \frac{1}{M} \sum_{i=1}^{M} \left\{ \sum_{j=1}^{P} p_i^j N \tau_p^{i,j} + \left(1 - \sum_{j=1}^{P} p_i^j\right) N \tau_0^i \right\} \tag{26}$$

As all clients have the same hit rate p and are the same roundtrip time away from the origin server, it follows that:

$$\left(1 - \sum_{j=1}^{P} p_i^j\right) = (1-p),\ \tau_0^i = \tau_0\ p_i^j = p p_w^{i,j},\ \forall i \tag{27}$$

So $\bar{R}_t$ is simplified to the following expression:

$$\bar{R}_t = (1-p) N \tau_0 + \frac{pN}{M} \sum_{i=1}^{M} \sum_{j=1}^{P} p_w^{i,j} \tau_p^{i,j} \tag{28}$$

Note that the model only considers nearby surrogates for the clients, where we can assume a similar latency for the surrogates a client will contact ($\tau_p^{ij} \cong \tau_p^*$), so it follows that:

$$\bar{R}_t = (1-p) N \tau_0 + pN \tau_p^* \tag{29}$$

corresponds to the equation of a linear function with variable p. As $\bar{R}_t(p=0) = N\tau_0 > N\tau_p^* = \bar{R}_t(p=1)$ we can conclude that for the roundtrip time it is desirable to offload all the content to the surrogates.

Considering now the response time associated with the process required time ($\bar{R}_p$), we start from the following expression:

$$\bar{R}_p = \frac{1}{M} \sum_{i=1}^{M} \left\{ \sum_{j=1}^{P} \left(\frac{p_i^j}{\mu_p^j - \sum_{l=1}^{M} \lambda_l^j} \right) + \frac{\left(1 - \sum_{j=1}^{P} p_i^j\right)}{\mu_s - \sum_{l=1}^{M} (1-p_l)\lambda_l} \right\} \tag{30}$$

As the same capacity factor k is set to all clients, the previous expression can be simplified to:

$$\bar{R}_p = \frac{P+1}{(k-1)\lambda M} \text{ as } \sum_{i=1}^{M} \sum_{j=1}^{P} \frac{p_w^{i,j}}{\sum_{l=1}^{M} p_w^{l,j}} = P \tag{31}$$

The first property of $\bar{R}_p$ that attracts attention is the fact that it is independent of the caching hit rate, that is, it remains constant. This can be observed in Figs. 7 and 8.

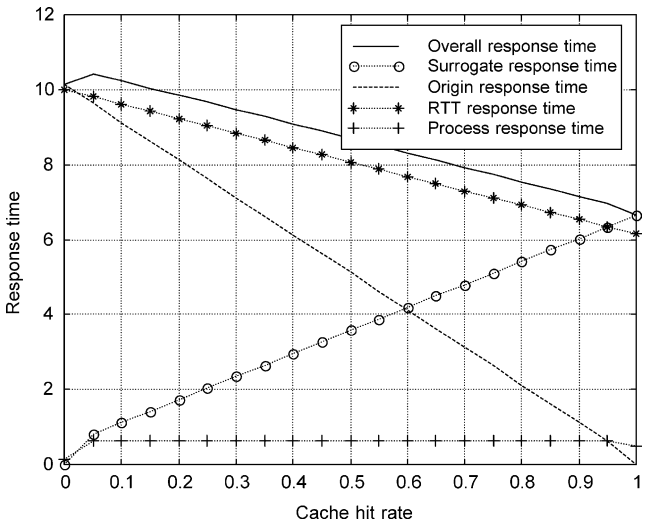

Fig. 8. Response times for $M = 8$, $P = 4$, $N = 5$, $\alpha = 1.01$, $k = 1.01$, hit_ratio $= 0, \ldots, 1$, $\tau_0^{\min} = \tau_0^{\max} = 2$ s, $\tau_d^{\min} = \tau_d^{\max} = 1$ s, $\lambda_{\min} = \lambda_{\max} = 100$.

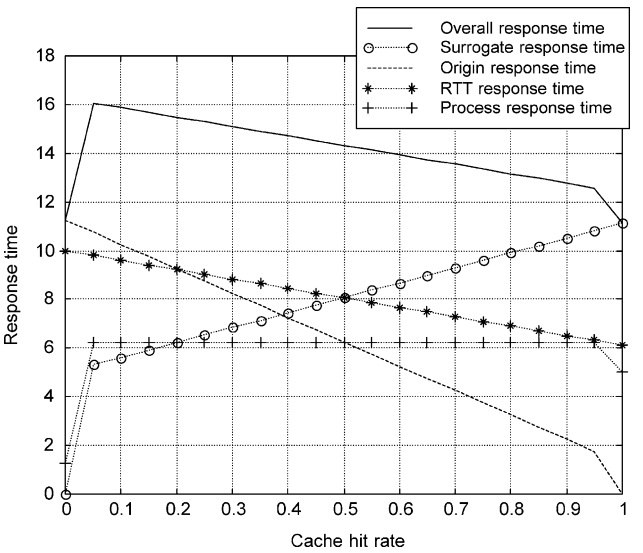

Fig. 9. Response times for $M = 8$, $P = 4$, $N = 5$, $\alpha = 1.01$, $k = 1.001$, hit_ratio $= 0, \ldots, 1$, $\tau_0^{\min} = \tau_0^{\max} = 2$ s, $\tau_d^{\min} = \tau_d^{\max} = 1$ s, $\lambda_{\min} = \lambda_{\max} = 100$.

The previous expression is valid for values of caching hit rates in the interval $]0, \ldots, 1[$. If $p = 0$, then the mean process time is the one required by the origin server:

$$\bar{R}_p = \frac{1}{M}\sum_{i=1}^{M}\left\{\frac{1}{(k-1)\sum_{l=1}^{M}\lambda_l}\right\} = \frac{1}{(k-1)M\lambda} \tag{32}$$

As $\bar{R}_p(p=0) = (1/(P+1))\bar{R}_p(p \neq 0)$, it justifies the notorious jump or discontinuity for the graph of process response time in Fig. 8. Similar behaviour occurs in the CDN scenario depicted in Fig. 7; however, the effect remains unnoticeable as the process time is very reduced.

If $p = 1$, then the mean process time is required by the surrogates:

$$\bar{R}_p = \frac{1}{M}\sum_{i=1}^{M}\left\{\sum_{j=1}^{P}\left(\frac{p_i^j}{\mu_p^j - \sum_{l=1}^{M}\lambda_l^j}\right)\right\} = \frac{P}{(k-1)M\lambda} \tag{33}$$

The discontinuity effect is less perceivable in the graph, as the relative reduction is lower for $p = 1$ than for $p = 0$. Fig. 9 shows the same CDN scenario but with a reduced value of the capacity factor k. This value supposes working at extreme capacity conditions, which is not a real case. However, it clearly shows the discontinuities at $p = 0$ and $p = 1$. Similar graphs could be obtained for different values of M, P and λ.

From previous expressions it is easy to obtain response times associated with contacting just the surrogates and contacting just the origin server, in order to observe in both cases the linear dependency between response time and caching hit ratio.

$$\bar{R}_{\text{srrgt}} = pN\tau_p^* + \frac{P}{(k-1)M\lambda} \tag{34}$$

$$\bar{R}_{\text{origin}} = (1-p)N\tau_0 + \frac{1}{(k-1)M\lambda} \tag{35}$$

A fixed capacity factor k does not realistically represent server load status, as capacity always increments by the same factor—independently of the traffic rate that is being supported, which is not the real case. Though this feature has helped us by simplifying the global expression into interesting partial conclusions, it leads to working scenarios where the response time associated with the server process remains practically constant, independently of other parameters in a possibly real CDN environment.

Real servers have a limited capacity that cannot be increased dynamically. The greater the number of requests supported then the longer the required process time, until a limit of overload is reached where packets may be discarded. On the contrary, a fixed value of k in our model supposes that a huge traffic arrival rate will result in a reduced process response time, as

$$R_p = \frac{1}{\mu - \lambda} = \frac{1}{\lambda(k-1)} \tag{36}$$

To correct this drawback, a fixed capacity must be fixed to all the servers. A worst-case dimensioning will be utilized here. For the origin server, the worst case takes place when the hit ratio p is 0, and therefore it must support all the traffic.

So it follows for the origin service rate (μ_s):

$$\mu_s = k \sum_{i=1}^{M} \lambda_i \tag{37}$$

For the surrogates, the worst case appears when $p = 1$, so clients send all traffic to surrogates. As there are certain weights for the communication, as well as different traffic rates per client, each surrogate has a different perceived arrival rate, which will result in a different capacity. So we can construct a vector for the surrogate service rates such as:

$$\vec{\mu}_p = P_w^T \vec{\lambda} \tag{38}$$

where P_w^T is the transposed probability-weight matrix and $\overrightarrow{\lambda}$ is a vector containing the traffic rate of each client (a random number between $\lambda_{\min}$ and $\lambda_{\max}$). Note that the matrix dimensions ($P \times M$ and $M \times 1$) enable the use of the matrix multiplication, and so resulting in a new vector with dimension $P \times 1$.

Fig. 10 depicts a new CDN scenario where server capacity has been previously fixed using the previous considerations commented before.

It can be seen that process time is very reduced, so overall response time is mostly affected by roundtrip time. This is because the servers are over-dimensioned as a consequence of worst-case design. In fact, this is realistic if the traffic rates of the clients (or the perceived arrival rate at the surrogates) can be estimated. For a CDN that uses Internet as a communication infrastructure, the number of requesting clients is unpredictable. This becomes a problem when flash crowds events occur, which could be simulated in a simplified way through a very reduced value of k. The independent variable for the flash-crowd simulation should be the time, and not the caching hit probability. Fig. 11 depicts a possible simulation where

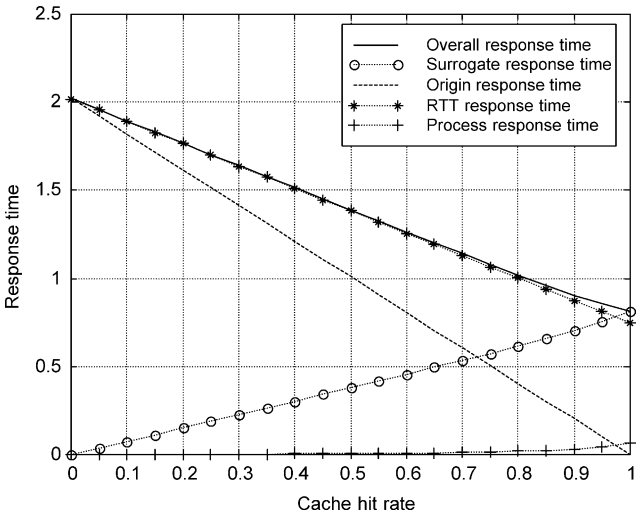

Fig. 10. Response times for $M = 100$, $P = 50$, $N = 5$, $\alpha = 0.7$, $k = 1.1$, hit_ratio $= 0, \ldots, 1$, $\tau_0^{\min} = 0.3$ s, $\tau_0^{\max} = 0.5$ s, $\tau_d^{\min} = 0.3$ s $\tau_d^{\max} = 0.4$ s, $\lambda_{\min} = 50, \lambda_{\max} = 100$.

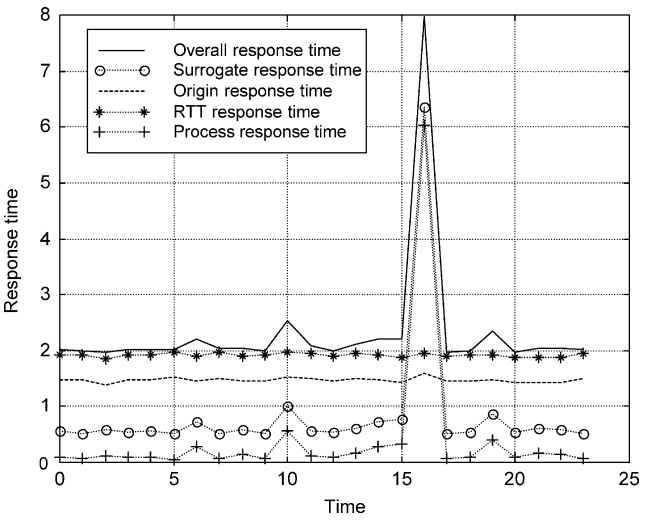

Fig. 11. Response times for $M = 100$, $P = 50$, $N = 5$, $\alpha = 0.7$, $k_{\max} = 1.4$, hit_ratio $= 0.5$, $\tau_0^{\min} = 0.3$ s, $\tau_0^{\max} = 0.5$ s, $\tau_d^{\min} = 0.3$ s $\tau_d^{\max} = 0.4$ s, $\lambda_{\min} = 50, \lambda_{\max} = 100$.

the effective capacity factor k is a random variable, but always bigger than the traffic arrival rate in order to comply with the stability factor. Its variability margins are between a very reduced value (1.001) and the given value of $k(k_{\max})$. Besides, flash crowds affect network link speed, as networks become more congested. This normally supposes that distances (in terms of latency) between clients and surrogates increase, and probably some packets are discarded at an intermediate node. For simplicity, this effect will be simulated through a variability of the variable N (between its given value and the double) that represented the necessary number of retransmissions required to satisfy a request. Fig. 11 shows a possible flash crowd that appears at four o'clock in the afternoon. The reduced absolute value of the response time is not important here, but the relative increment from the mean response time over the whole time interval is important. For this CDN scenario, server overload has mostly determined the mean response time, whereas link bandwith has hardly varied during the simulation. However, it could have been the other way round.

Some other considerations about the graph must be made. A flash crowd can be local to a certain zone, or global over whole content distribution network. The former can be successfully managed, at least partially, by the CDN provider redirecting clients to other distant surrogates, but this would supply low response times by avoiding either congested paths or server overload. A global flash crowd affects the whole system and very little can be done as congestion and overload conditions are everywhere and balancing mechanisms are less successful. While Fig. 11 shows mean response times obtained from multiple surrogates, Fig. 12 depicts a CDN scenario where response time is represented for each surrogate and origin server, as well as the mean response time experienced by a client.

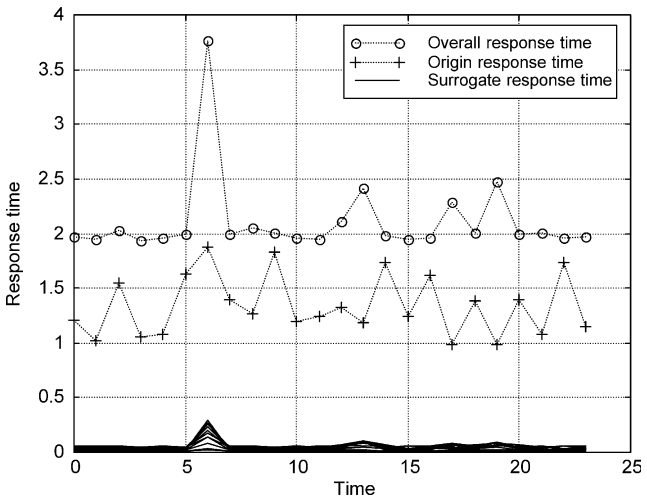

Fig. 12. Response times for $M = 100$, $P = 50$, $N = 5$, $\alpha = 0.7$, $k_{max} = 1.3$, hit_ratio $= 0.5$, $\tau_0^{min} = 0.3$ s, $\tau_0^{max} = 0.5$ s, $\tau_d^{min} = 0.3$ s, $\tau_d^{max} = 0.4$ s, $\lambda_{min} = 50$, $\lambda_{max} = 100$.

Note that, as each client contacts different surrogates and origin servers, it is not easy to obtain the mean response time graph from the other functions, as the probability weights must also be taken into account.

6. Conclusions and future work

CDNs deal with a communication process where network latency and server capacity are decisive parameters in the response time perceived by the user, as well as the system's client redirection ability to match each client with a suitable nearby surrogate. Performance studies have been made by the research community both in empirical and analytical approaches. This article focuses on the latter approach, starting from previous work where a simple model of a CDN is presented. This paper tries to introduce a more realistic model of a CDN, where clients and surrogates are dispersed throughout the world with different link latencies and traffic rates. Server redirection behaviour is better characterized when each client contacts various nearby surrogates with different probability weights based on distance in terms of latency. We have come to the following conclusions:

- It is desirable to offload all the content to the surrogates from the point of view of the roundtrip time, as they are located closer to clients than the origin server. Moreover, the roundtrip time follows under reasonable conditions a linear function dependent on the caching hit rate with a negative slope. This corroborates the previous statement.
- If the servers are dimensioned with a capacity value dependent on just the traffic arrival rate, the caching hit rate does not have any effect on the process time. Under this assumption it is necessary to discriminate in the analysis when the content is served by the origin site (no CDN), only surrogates (100% hit rate), or a mixture of both. If the dimensioning factor is the same for all servers, serving all the content from the origin server produces a lower response from the point of view of the process time.
- Real cases suppose a limited server capacity, selected at design phase on the expected client population. This implies a greater value for process time when the caching hit ratio increases, so the total offload of content is undesirable from this perspective. Therefore, a trade-off between roundtrip time and process time determines the scenario for optimum values of total response time inside a caching hit range.
- Flash crowd effect can be caused by network congestion, server overload, or both. All these scenarios can be simulated through large retransmission values, factor N and roundtrip times τ (for network congestion), and capacity factor k and traffic arrival rate λ (for overload conditions). As the variables are assigned per surrogate, origin server, and client, the flash crowd can be reproduced either locally (only on a few nodes of the network) or globally (on the whole CDN).

As further work, we intend introducing different probability-weight assignment policies, as well as a non-uniform distribution of clients and surrogates in order to obtain dense and sparse clients within the same working scenario. Complementary models of client redirection mechanisms and server placement are to be introduced in future work.

References

[1] P.S.M. Sayal, P. Vingralek, Selection algorithms for replicated web servers, ACM SIGMETRICS Internet Server Perfsormance Workshop, Madison (USA), June, 1998.
[2] D. Verma, Content Distribution Networks, an engineering approach, John Wiley, New York, 2002.
[3] D. Liben-Nowell, H. Balakrishnan, D. Karger, Analysis of the evolution of peer-to-peer systems, ACM Conference on Principles of Distributed Computing, Monterrey (USA), 2002.
[4] S. Gadde, J. Chase, M. Rabinovich, Web caching and content distribution: a view from the interior, Fifth International Workshop on Web Caching and Content Distribution, Lisbon (Portugal), June, 2000.
[5] S. Sariou, K.P. Gummadi, R. Dunn, S. Gribble, H.M. Levi, An analysis on Internet content delivery systems, Fifth Symposium on Operating Systems Design and Implementation, Boston (USA), December, 2002.
[6] A. Barbir, B. Cain, F. Douglis, M. Green, M. Hofmann, R. Nair, D. Potter, O. Spatscheck, Known CDN Request-Routing Mechanisms-draft-cain-cdnp-known-request-routing-01.txt, February, 2001.
[7] V. Cardellini, M. Colajanni, P.S. Yu, Request redirection algorithms for distributed web systems, IEEE Transaction on Parallel and Distributed Systems 14 (4) (2003) 355–368.
[8] Akamai, http://www.akamai.com.
[9] Digital Island, http://www.sandpiper.net.

[10] K.L. Johnson, J.F. Carr, M.S. Day, M.F. Kaashoek, The measured performance of content distribution networks, Fifth International Workshop on Web Caching and Content Distribution, Lisbon (Portugal), June, 2000.
[11] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaram, B. Weihl, Globally distributed content delivery, IEEE Internet Computing September/October (2002) 50–58.
[12] D. Verma, S. Calo, K. Amiri, Policy based management of content distribution networks, IEEE Network, March (2002) 34–39.
[13] J. Jung, B. Krishnamurthy, M. Rabinovich, Flash crowds and denial of service attacks: characterization and implications for cdns and web sites, 11th International Conference of WWW, Honolulu (USA), May, 2002.
[14] Z. Mao, C. Cranor, F. Douglis, M. Rabinovich, A. precise, A precise and efficient evaluation of the proximity of web clients and their local DNS servers, USENIX'02, Monterrey CA (USA), June, 2002.
[15] A. Shaikh, R. Tewari, M. Agrawal, On the effectiveness of dns-based server selection, IEEE INFOCOM'01, Anchorage (USA), April, 2001.
[16] R.P. Doyle, J.S. Chase, S. Gadde, A.M. Vahdat, The trickle-down effect: web caching and server request distribution, Computer Communications 25 (2002) 345–356.
[17] J. Kangasharju, K.W. Ross, J.W. Roberts, Performance evaluation of redirection schemes in content distribution networks, 5th International Workshop on Web Caching and Content Distribution, Lisbon (Portugal), June, 2000.
[18] C. Cranor, et al., Enhanced Streaming Services in a content distribution network, IEEE Internet Computing, July/August (2001) 66–75.
[19] Z. Fei, A novel approach to managing consistency in content distribution networks, 6th International Workshop on Web Caching and Content Distribution, Boston (USA), June, 2001.
[20] A. Biliris, C. Cranor, F. Douglis, M. Rabinovich, S. Sibal, O. Spatcheck, W. Sturm, C.D.N. Brokering, Computer Communications 25 (4) (2002) 393–402.
[21] J. Kangasharju, K.W. Ross, J.W. Roberts, Performance evaluation of redirection schemes in content distribution networks, 5th International Workshop on Web Caching and Content Distribution, Lisbon (Portugal), June, 2000.
[22] B. Krishnamurthy, C. Wills, Y. Zhang, On the use and performance of Content Delivery Networks, ACM SIGCOMM Internet Measurements Workshop, San Diego (USA), August (2001).
[23] C. Cameron, S. Low, D. Wei, High-density model for server allocation and placement, ACM SIGMETRICS'02, Marina del Rey, CA (USA), June, 2002.
[24] D. Agrawal, J. Giles, D. Verma, On the performance of content distribution networks, International Symposium on Performance Evaluation of Computer and Telecommunication Systems, Orlando (USA), July, 2001.
[25] M. Masa, E. Parravicini, Impact of request routing algorithms on the delivery performance of content delivery networks, 22nd IEEE International Performance Computing and Communications Conference, Phoenix (USA), April, 2003.
[26] S. Calo, D. Verma, D. Agrawal, J. Giles, On the Effectiveness of Content Distribution Networks, International Symposium on Performance Evaluation of Computer and Telecommunication Systems, San Diego (USA), July, 2002.
[27] L. Kleinrock, R. Gail, Queueing Systems: Problems and Solutions, Wiley, New York, 1996.

Benjamin Molina received his MSc degree in telecommunication engineering from the Universidad Politecnica de Valencia (UPV) in 2001. He made his awarded final project about voice technologies in Tissat, an ICT company in Valencia, where he worked for a year developing PDA web interfaces and CTI services on top of the Java platform. Later he became a member of the Distributed Real-Time Systems research group of the Departamento de Comunicaciones, at the UPV. Benjamin Molina is currently involved in research projects related to network simulation environments covering content distribution and scalability issues that may affect real implemented networks. His main interest is focussed on multimedia distribution across Internet and the different related technologies, such as: multicast communications, web caching and real-time systems.

Carlos E. Palau received his MSc and PhD (Dr Ing.) degrees, both in telecommunication engineering, from the Universidad Politecnica de Valencia (UPV) in 1993 and 1997, respectively. He is Associate Professor in the Escuela Tecnica Superior de Ingenieros de Telecomunicación at the UPV, and works in the Distributed Real-Time Systems research group of the Departamento de Comunicaciones. He is currently involved in research and development projects for the application of multimedia and real-time technologies to industry, medicine, education, and communications. Dr. Palau is a member of IEEE and IASTED, and is involved in several IPCs of national and international conferences. He has chaired IASTED Communications Systems and Networks 2002 and 2003.

Manuel Esteve received both his MSc in computer engineering and his PhD in telecommunication engineering (Dr Ing) from the Universidad Politécnica de Valencia in 1989 and 1994, respectively. He is Professor in the Escuela Técnica Superior de Ingenieros de Telecomunicacion at the Universidad Politecnica de Valencia (UPV), and he leads the Distributed Real-Time Systems research group of the Departamento de Comunicaciones. He is currently involved in research and development projects for the application of multimedia and real-time technologies to industry, medicine, education, and communications. He is the responsible for the Virtual University at the UPV And Has Co-Chaired EUROMEDIA'01.