Optimization of Content Delivery Networks server placement

Anthony Busson , Daniel Kofman, Jean-Louis Rougier

Ecole Nationale Supérieure des Télécommunications*
46 rue Barrault, 75013 Paris, France.
[anthony.busson, daniel.kofman, jean-louis.rougier] @enst.fr

January 24, 2003

Abstract

Content Delivery Networks (CDN) became in the last years an important element of the Internet architecture. Between their most important benefits, we can cite the enhancing of the quality of service (QoS) perceived by users and the reduction of the cost of content delivery. This is accomplished by placing servers that host content copies (or that forward streamed content) near the users. A user may then download a copy of the required content element from the closest server, thus with a better QoS. Documents are not delivered end-to-end any more. They are delivered once to the servers and then from the servers to the closest users, thus, under correct design, reducing the delivering cost.

In this paper, we consider the problem of optimizing CDN server placement. First, we propose a new modeling approach of the Internet and the CDNs based on Stochastic Geometry. Different point processes on the plane are used to represent the topology of the Internet (Interconnection of Autonomous Systems) and the placement of CDN servers. Then, we propose a cost function reflecting the server costs and the quality of service perceived by the users. The cost function is indeed a functional of the above introduced point processes. The optimization of the cost function allows us to quantify the optimal number of servers and their optimal placement. The impact of the different parameters on the system is also evaluated. We first analyze a system based on only one type of servers. Then, we analyze the more general case where different server classes and object placement policies are considered. For the last case, we are able to compare different policies for content placement that can take into account the popularity of different content elements, and to optimize the design of these policies.

1 Introduction

The success of the Internet services, like the Web, has motivated in recent years the emergence of Content Delivery Networks (CDNs), targeted to improve the quality and to reduce the cost of content delivery. A CDN is composed of a set of servers, called surrogate servers, which are deployed on the Internet (overlay approach). The surrogates host copies of web content (we concentrate in this paper of the Web service, even though most of the presented results can be applied to other Internet services). Thus, a user may transparently download a nearby copy of this content rather than fetch it from the, maybe distant, server of the content provider. On one hand, the QoS perceived by the users when downloading objects stored in the CDN servers is thus dependent on the number and placement of the surrogates. On the other hand, the cost of the CDN is also dependent on these parameters, so an optimization of the CDN topology is required. In this paper

*This work is supported by the french government funded project GEORGES (http://www.telecom.gouv.fr/rnrt/projets/pgeorges.htm)

we propose a new modeling approach for CDNs that allows us to optimize their topology when considering a realistic cost function.

The problem of placement of CDN servers has already been addressed in related works. In the usual approach, the different components of the model, as the network topology, the location and the number of users and the quality of service parameters are supposed to be known a priori (see, for example, [1], [2], [3] and [4]). More elaborate parameters such as the consumed bandwidth or the read, write and storage costs are sometimes also taken into account ([5], [6]).

Unfortunately, the location of the users or the object popularity most of the time cannot be known in advance and vary in time. Moreover, this static approach leads to NP hard optimization problems which may be difficult to optimize when a large network, as the Internet, is considered. Therefore, in order to quantify the optimal deployment of CDN servers, a macroscopic model is more appropriate.

In this paper, we propose a random geometric model for CDNs based on point process distributed in the plane. A similar modeling approach has proven to be efficient to optimize other types of telecommunication systems (See, for example, [7], [8], [9] and [10]). A survey of the use of stochastic geometry may be found on [11].

In our study, marked point processes are used to represent the Autonomous Systems (ASes) of the Internet and the servers of the CDN.

We introduce a cost function which is constituted of two terms : the first term represents the servers cost and the second one reflects the quality of service perceived by the users. The cost function is indeed a functional of the above introduced point processes. We are able to compute the explicit formula for this cost function which depends on a reduced number of parameters. The optimization of the cost function allows us to quantify the optimal number of servers. The impact of the different parameters on the system is also evaluated.

In our approach, the complexity of the optimization problem is independent of the considered network size.

We first analyze a system based on only one type of servers. Then, we analyze the more general case where different server classes and object placement policies are considered. Each server class hosts a subset of the CDN objects. For the last case, we are able to compare different policies for content placement that can take into account the popularity of different content elements, and to optimize the design of these policies.

2 Stochastic Geometry model of CDNs

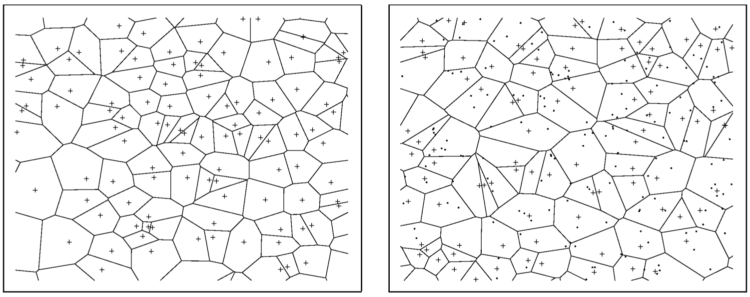
In this section we present an analytical model of CDNs based on the Stochastic Geometry theory.

A marked Poisson point process distributed in $\mathbb{R}^2$, denoted by N, with intensity λ ($\lambda \in \mathbb{R}^+$), represents the Autonomous Systems (ASes) of the Internet. More precisely, each point of the point process represents an AS.

Two marks are associated with each point of N. Let x be a point of N. The first mark associated with x, denoted by m_x, represents the number of requests per time unit which emanate from the users of the AS located at x. We denote by $\overline{m}$ the mean value of m. The second mark indicates whether there is a surrogate in the corresponding AS or not. This mark has a Bernouilli law of parameter p ($0 < p < 1$). Each mark process is independent and identically distributed (i.i.d.) and both mark process are independent. We denote by N_p and N_{1-p} two point process obtained as a thinning of N using the second mark process. The process N_p (resp. N_{1-p}) represents the points of N for which the corresponding mark is equal to 1 (resp. 0). By the properties of the Poisson point process, N_p and N_{1-p} are independant Poisson point processes.

Without lost of generality, the point processes N, N_p and N_{1-p} are observed in the open ball of radius R and centered at the origin of the plane. This ball is denoted by $B(o, R)$. R is considered as the radius of the Internet.

A user in an AS is supposed to download the documents of the CDN from the closest surrogate server with regard to the Euclidian distance. For any point x of N representing an AS, all the



(a) Poisson point process and its Voronoï cells

(b) The process N_p, its tessellation and the process N_{1-p}

Figure 1: Poisson point processes and Voronoï tessellations

users of this AS will fetch the documents from the server in a given AS (represented by a point of N_p). For each server s, we can thus define a domain in the plane that contains all ASes for which the users will download the documents from s. For a point y of N_p where a server is placed, this domain is the set of points of $\mathbb{R}^2$ which are closest to y than to any other point of N_p. This domain is called the Voronoï cell of y with regards to N_p and is denoted by $V_y(N_p)$. The set of Voronoï cells for all the points of N_p forms a tessellation of the plane. A Poisson point process and the corresponding Voronoï tessellation are represented in Figure 1(a). In Figure 1(b), we plot the process N_p (the crosses), its tessellation and the points of N_{1-p} (the points).

3 Optimization of a CDN with homogeneous servers

3.1 Cost function

The ideal solution would be to deploy a server in all the ASes. All the users would thus perceive a good quality of service in terms of delay and bandwidth. However, the servers have a cost, so a tradeoff must be found between the number of servers to deploy and the quality of service seen by the users. Consequently, our cost function has two components. We partially reuse here the structure of the cost function described in [1].

The first component expresses the annoyance perceived by the users. This component increases with the distance between the users and the servers and thus decreases with the number of servers. It corresponds to the sum, for all points of N, of a function f of the distance between the point of N and its closest server in the CDN (this distance is zero if there is a server of the CDN in the considered AS) multiply by the number of requests per second issued in this point.

The second component is the servers cost. In a first approach, we associate to all the servers the same cost α.

The cost for a sample of our stochastic model has the form :

$$Cost = \sum_{x \in N_p} \alpha + \sum_{y \in N} m_y f(z_y),$$

where z_y is the Euclidian distance between a point y of N and its closest point of N_p. The cost function that we consider is the mean value of this function. The computation of the cost function is given in the appendix. In the case $f(0) = 0$ we obtain[1]:

[1] The computation presented in the appendix do not require any hypothesis on the value of $f(0)$. We present here this formula because all our numerical examples suppose $f(0) = 0$.

$$Cost = \lambda \pi R^2 \left[p\alpha + (1-p)\overline{m}\lambda p 2\pi \int_0^{+\infty} r f(r) \exp\{-\lambda p \pi r^2\} dr \right].$$

3.2 Computing function f

Nunber of BGP hops	Number of ASes
ASes reachable by 1 AS hops	49
ASes reachable by 2 AS hops	498
ASes reachable by 3 AS hops	5 906
ASes reachable by 4 AS hops	4 971
ASes reachable by 5 AS hops	1 902
ASes reachable by 6 AS hops	469
ASes reachable by 7 AS hops	81
ASes reachable by 8 AS hops	6
ASes reachable by 9 AS hops	1

Figure 2: Number of ASes reachable in terms of the number of hops

The QoS perceived by a user depends strongly on the number of ASes in the path between the user and the server. Indeed, the bottlenecks of the network are often at the the ASes interconnection points. Nowadays, there exists some mechanisms which allow the DNS server to redirect a request to the closest server with regards to the number of BGP hops (Border Gateway Protocol, see [14]).

We will use the number of BGP hops as the metric of the QoS perceived by the users. Thus, we need to compute a function f (see the cost function in the previous section) which is close to the real distribution of the number of BGP hops between the ASes of the Internet. In a first step, we estimate the cumulative distribution function of the number of hops between two ASes randomly chosen in the Internet. In a second step, we compute f such that the distribution of f applied to two points of N randomly chosen corresponds to the distribution estimated in the first step.

These two steps are detailed as follows: We assume that the number of ASes between two given ASes increases with the Euclidian distance between these ASes. The users of a AS are then served by the closest server with regard to the Euclidian distance as explained in Section 2.

In order to choose the most realistic function f representing the number of hops between the ASes we based our estimation on real BGP tables of various ASes. We obtained from [17] the data indicating the number of ASes reachable from a given AS as a function of the number of BGP hops. We computed the average of the 43 tables obtained from the site. The result is shown in Figure 2. We computed the cumulative distribution function of the number of hops which separate two given ASes. This estimation approach is a first approximation, in our future works we shall improve it. In particular, we plan to consider a more accurate description of the connectivity of the ASes (from the data we used, it follows that the ASes have a larger connectivity than the real one when considering the whole network).

Let r be the random variable representing the Euclidian distance between two points randomly chosen among the set of points of N. We compute the function f such that the distribution of $f(r)$ corresponds to the distribution of the c.d.f. previously computed. Since N is a Poisson point process, knowing that a given number of points of N are in $B(o, R)$, these points are independently and uniformly distributed in $B(o, R)$. We are then able to compute the distribution of r and to evaluate the function f. The function f is given in Figure 6(b), where x and y are the two points of N randomly chosen.

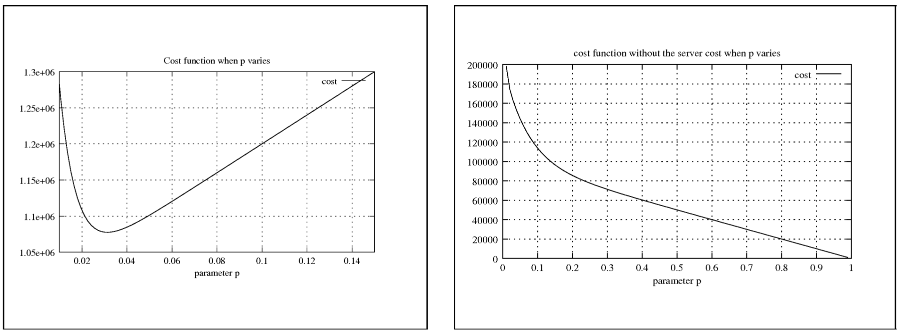
3.3 Optimization, numerical results

We optimized the cost function introduced in Section 3.1 with the function f defined in Section 3.2. In this case, the cost function is bounded and then it has at least one minimal value when $p \in [0, 1]$. In the numerical cases that we have studied, we found that there is a unique value of $p \in]0, 1]$ which minimizes the cost function.

In this section, unless explicit indication, we fix the following parameters: the number of ASes equals 10 000 ASes[2] (this means that $\lambda = \frac{10000}{\pi}$ for $R = 1$), $\alpha = 300$ and $\overline{m} = 100$.

In Figure 3(a), we draw the cost function in terms of p. We observe that the optimal value of p is close to 0.03. This is the optimal proportion of ASes in which a server of the CDN should be deployed. From p, we can easily deduce the optimal number of servers of the CDN which is $p\lambda\pi$. The number of servers is an important element of the deployment cost.

In Figure 3(b), we represent the cost function when $\alpha = 0$ (we neglect the server costs) in order to evaluate the impact of the number of servers on the QoS. The slope of this function increases with p (decreases in absolute value). There is a great benefit of increasing p for small values of this parameter: in this case, when a new server is deployed, the content is brought closer for a set of ASes. For large values of p, the impact of increasing p becomes negligible (for $p > 0.3$ in Figure 3(b)). Indeed, in this case, almost all the ASes are already at most at one hop of an AS with a server and adding a new server is just beneficial for the AS where the new server is deployed.



(a) Cost function

(b) Cost function when the servers cost equals 0

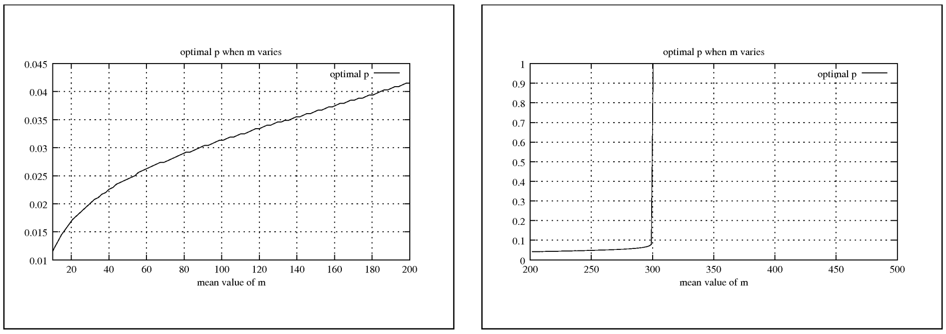
Figure 3: Cost functions

3.3.1 Impact of $\overline{m}$

As it can be seen in the explicit formula obtained in Section 3.1, the cost function is insensitive to the distribution of m, it depends only on its mean, $\overline{m}$.

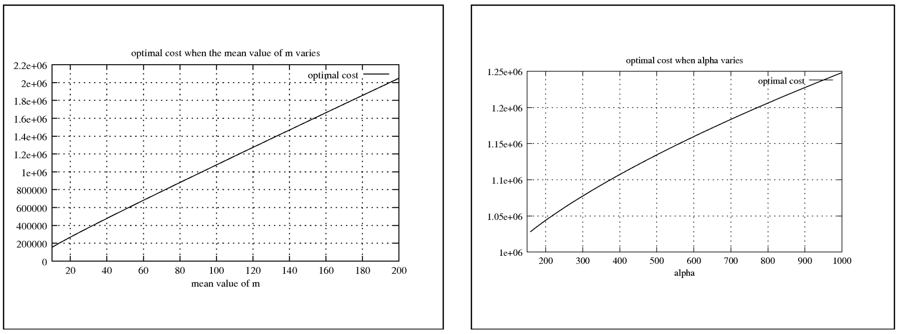
Figure 4(a) shows the optimal value of p when $\overline{m}$ varies from 5 to 200. We observe that the curve is concave, which corresponds, as we explain in the following, to the same phenomena that allowed us to interpret Figure 3(b). It is clear that p increases with $\overline{m}$ since the second term of the cost function increases with $\overline{m}$ and decreases with p. For small values of $\overline{m}$ and then of p, an increase of $\overline{m}$ implies an important increase of p since adding a server will benefit more requests per ASes ($\overline{m}$ has increased) for several ASes (those for which the new server becomes the closest one). For large values of $\overline{m}$ and then of p, an increase of $\overline{m}$ implies a less important increase of p since adding a server will benefit more requests per ASes of a reduced number of ASes (for p near 1, only the requests from the AS to which a server has been added will perceived some benefit).

[2] Which is a realistic value for the present Internet.



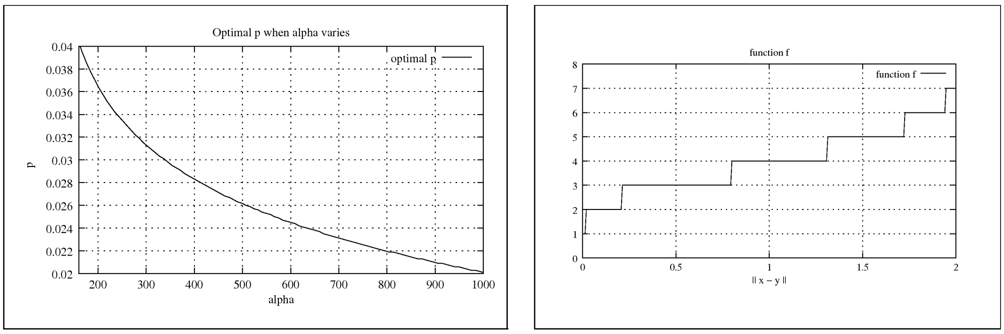
(a) Impact of $\overline{m}$ when $\overline{m} < \alpha$ (b) Impact of $\overline{m}$ when $\overline{m} > \alpha$

Figure 4: Impact of $\overline{m}$ on the optimal value of p



(a) Impact of $\overline{m}$ on the cost function at the optimal (b) Impact of α on the cost function at the optimal

Figure 5: Impact of $\overline{m}$ and of α on the cost function



(a) Impact of α (b) The step function f

Figure 6: Impact of α on p and the function f

In Figure 4(b), we observe that when $\overline{m}$ reach α, the optimal value of p becomes 1. Indeed, the cost of a server becomes less expensive than the cost of downloading content from a neighborhood AS. In the rest of the paper, we shall consider only the case where $\overline{m} < \alpha$.

In figure 5(a), we plot the cost function at the optimum when $\overline{m}$ varies. We observe that the cost is almost linear with $\overline{m}$.

3.3.2 Impact of α

We evaluate here the impact of the servers cost α on the optimal proportion of ASes which host a server. As expected, the optimal value of p decreases with the servers cost (as shown in the Figure 6(a)). The convexity of the curve can be explained by using the same arguments that in the previous section. In Figure 5(b), the cost function at the optimum is plotted. It is concave with the server costs which is related with the fact that the optimal p is convex in terms of α.

4 Optimization of a CDN with heterogeneous servers

In this Section, we extend the previous model to consider the hererogeneous servers case and to evaluate different content placement policies that can depend on content popularity. Usually, content elements hosted by the CDN are not placed in all the servers. Therefore, the quality of service perceived by a user depends on the location of the servers where the required content element is hosted.

In our extended model, we consider several classes of servers. Each class of servers is in charge of a set of objects (content elements). An object is hosted in all the servers of a given class.

We model the placement of the servers of a given class i by a Bernouilli thinning of parameter p_i of the Poisson process N, that we denoted by N_i. There is an independent and homogeneous thinning of N per class. We denote by n the number of classes. These $N_1, ..., N_n$ are independent Poisson Point process of intensities $\lambda_i = p_i\lambda, i = 1, ...n$. In other words, N_i describes the ASes where the servers of class i are deployed. We propose a global cost function that takes into account the class where the objects are hosted (and then the placement policies) and the different costs of the servers of different classes. For instance, the server costs may depend on the storage capacity of the servers as we shall see in Section 4.2.

We define:

$$Cost = \sum_{i=1}^{n} \mathbb{E}\left[\sum_{x \in N_i} \alpha_i\right] + \sum_{i=1}^{n} q_i \mathbb{E}\left[\sum_{y \in N} m_y f(z_y^i)\right],$$

where q_i is the probability that a requested object is stored in the servers of class i and z_y^i is the distance between a point y of N and its closest point of N_i (the closest class i server). In the following sections, the probabilities q_i will be computed for different objects popularities and placement policies.

4.1 Examples of placement policies

We present here two placement policies for the case of two classes of servers. The results can be easily extended to any number of classes. We denote by K ($K \in \mathbb{N}^*$) the total number of objects stored by the CDN.

4.1.1 Object popularity based policy

The first policy is based on the popularity of the objects hosted by the CDN. Among the K objects, the d ($d \in \mathbb{N}^*$) most popular objects are stored in class 1 servers and the other objects are stored in class 2 servers. Basically the idea is that we should have for this case many class 1 servers hosting a reduce number of very popular objects and then will be cheap and a few class 2 servers hosting most of the objects and then being expensive.

Several studies have shown that the distribution of the object popularities is well modeled by a Zipf law of parameter 1 (see [15] and [16]). In these studies, the popularity of a given object means exactly the ratio between the number of requests for this object and the total number of requests for all the objects.

If the objects are ordered by popularity from 1 to K (the most popular is the first object) and if we note X the number associated to an object following this order, we have :

$$\mathbb{P}(X = j) = \frac{C}{j}, \forall j = 1, .., K$$

where C is a non-negative constant.

The probability q_i that a requested object is hosted in class i servers is then given by:

$$q_1 = \sum_{j=1}^{d} \mathbb{P}(X = j), \text{ and}$$

$$q_2 = \sum_{j=d+1}^{K} \mathbb{P}(X = j)$$

4.1.2 Random Policy

With this policy, the CDN does not take into account the object popularities. Each object is uniformly and independently placed among the different classes. If we consider just two classes of servers and if d objects are stored in class 1 servers then:

$$q_1 = \frac{d}{K} \text{ and } q_2 = \frac{K-d}{K}$$

4.2 Cost of the servers

We split the cost of a server into two terms. The acquisition cost plus the maintenance cost of a class i server is given by a constant γ_i. Moreover, we assume that the servers cost depends on the capacity; therefore, on the potential number of hosted objects. For each object stored in class i servers, we add a constant cost β_i. The global cost of a server of class i is then:

$$\alpha_i = \gamma_i + d_i \beta_i,$$

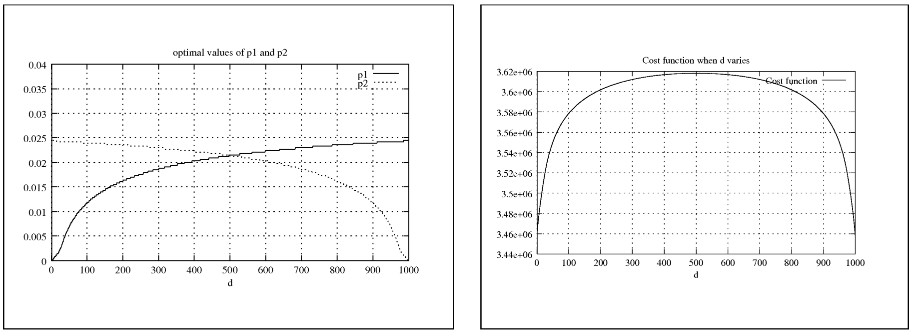
where d_i is the number of objects stored by the servers of class i.

4.3 Numerical results for the heterogeneous server case

With the structure of the extended cost function, the parameter p_i for each server class can be optimized separately. Therefore, we do not measure the impact of the parameters α nor $\overline{m}$ for this cost function since the results are similar to those presented in Section 3.3. We evaluate here the impact of the parameters of the two policies on the cost functions and on the optimal number of servers. We also compare both policies. In the rest of the paper, we assume that the total number of objects is equal to 1000 ($K = 1000$).

4.3.1 Impact of the number of objects hosted in class 1 servers

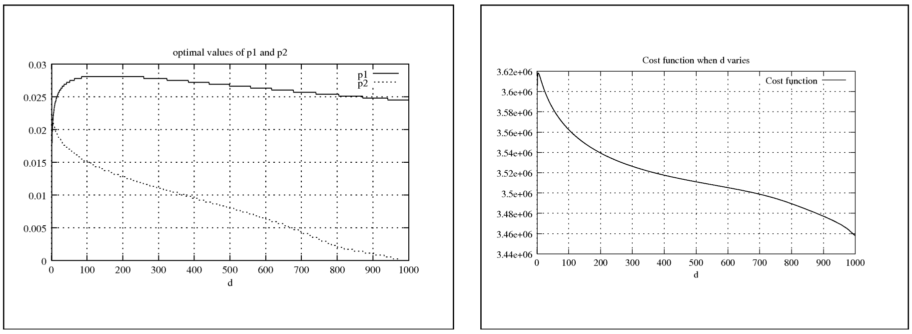
In Figures 7(a) and 7(b), we show the impact of d on the optimal values p_1, p_2 (the optimal proportions for the servers of class 1 and 2) and on the cost for the random policy. For these numerical cases, the parameters are $\gamma_1 = \gamma_2 = 600$, $\beta_1 = \beta_2 = 1$ and $\overline{m} = 300$. Since the



(a) Optimal values of p_1 and p_2 when d varies

(b) The cost function when d varies

Figure 7: Optimal values of p_1 and p_2 and the corresponding cost function for a random policy



(a) Analysis of the popular policy

(b) The cost function when d varies for a popular policy

Figure 8: Optimal values of p_1 and p_2 and the corresponding cost function for a popular policy

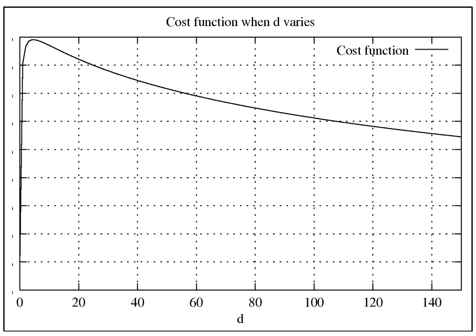
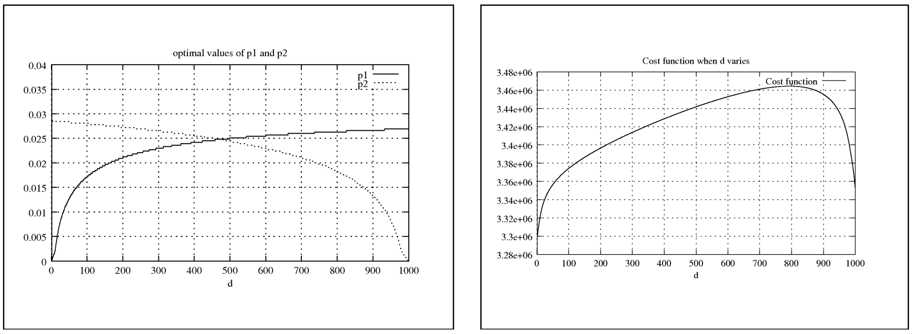


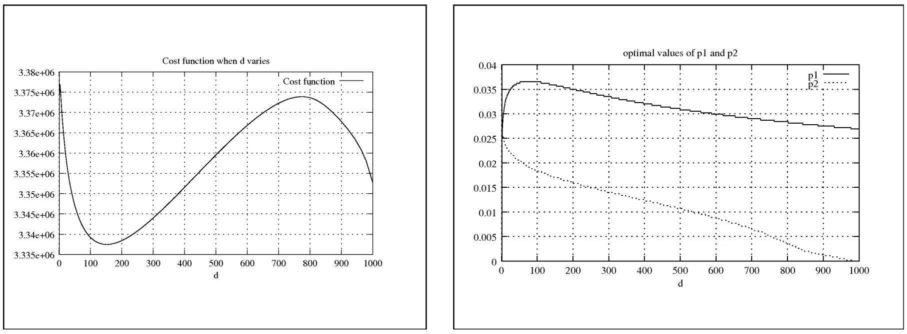
Figure 9: Zoom of the cost function for a popular policy and $d < 150$



(a) Optimal values of p_1 and p_2 for the random policy when d varies

(b) The cost function when d varies and for a random policy

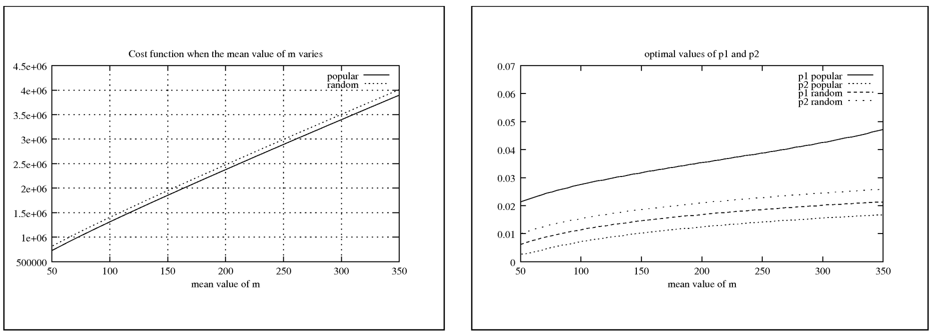
Figure 10: Optimal values and cost function for a random policy for two different server costs



(a) The cost function when d varies and for a popular policy

(b) Optimal values of p_1 and p_2 for the popular policy when d varies

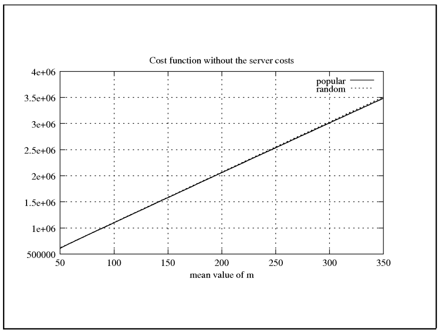
Figure 11: Optimal values and cost function for a popular policy and for two different cost of servers



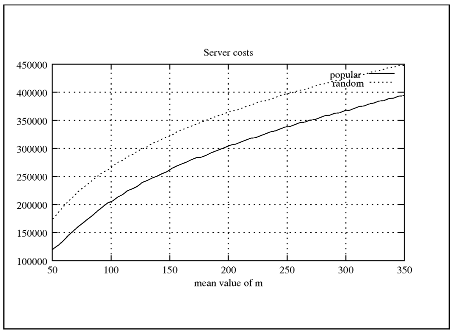
(a) The cost functions for the two policies

(b) Optimal values of p_1 and p_2 for the two policies

Figure 12: Optimal values of p_1 and p_2 and the corresponding cost functions



(a) Cost function without the server costs



(b) Server costs

Figure 13: Detail of the cost function

servers cost parameters are equal for the two classes, for a random policy, the cost function and the optimal values p_1 and p_2 are symmetric. The best placement of the objects consists in placing all the objects in the same class of servers. This minimize the number of servers and then the acquisition part of the cost ; the rest of the cost function being insensitive to the placement policy in this model.

In Figures 8(a) and 8(b), we plot the optimal values of p_1, p_2 and the corresponding cost function for the popular policy and for the same parameters than in the random policy. The best placement of the objects among the two classes is still the placement of all the objects in the same server class (see Figure 9 which is a zoom of Figure 8(b) around $d = 0$). Positive very small values of d represent a bad case since a lot of class one servers have to be deployed and the fix part of the server cost increases much more than the decrease of the second term of the cost function.

We conclude that when the servers cost parameters are identical for the different classes, the optimal is to have just one class. We observe that the curves are very asymetric in this case.

We analyze now the impact of the asymmetry of the servers cost parameters on the optima. We consider a first class of servers, less expensive than the second class, but with a limited storage capacity, and thus, where the storage cost is greater. We take the following parameters : $\gamma_1 = 300$, $\gamma_2 = 600$, $\beta_1 = 1$, $\beta_2 = 0.5$ and $\overline{m} = 300$. For the random policy, the cost function at the optimal is asymmetric, but the optimal is still reached for all the objects placed in the servers of the same class (class 2 in this case), as illustrated in Figure 10(a) and 10(b).

When a popular policy is used, there is minimum and a maximum of the cost function as a function of d, as shown in Figure 11(a). The optimal configuration consists in placing the 150 most popular objects in the class 1 servers. As illustrated in Figure 11(b), at the optima for d the number of ASes that must be deployed is about 360.

After simple computations we obtained that class 1 servers are in charge of approximately 75% of the requests from the users. The number of servers of class 2 is smaller, about 160. These servers are in charge of just 25% of the requests and they store 85% of the objects for a limited cost, indeed a small number of class 2 servers is deployed and the storage cost of these servers is smaller ($\gamma_2 = \frac{\gamma_1}{2}$). But this does not penalize significantly the QoS since only the most unpopular objects are hosted in this class of servers.

We observe that there is also a worst case for the placement of the objects among the two classes for $d = 780$. In this case, the servers of class 1 respond to 96% of the requests, which require a great cost of storage. Moreover, servers of class 2, which have an expensive acquisition cost, must be deployed for just 4% of the requests.

4.3.2 Comparison of the two policies of placement

We compare the two policies in order to evaluate the gain to use the popular policy. In Figure 12(a), the costs at the optima are shown. The cost of the random policy is slightly greater than the cost of the popular policy. But, if we consider the number of servers of class 1, represented in Figure 12(b), we note that this number is more important for the popular policy. As explained in the previous Section, a great number of servers of class 1 will be in charge of 75% of the requests, providing a good QoS for these requests. A small number of servers of class 2, more expensive but with cheaper storage cost, hosted the 85% of the unpopular document, whereas a great number of these expensive servers is required if the random policy is used. The global QoS perceived by the users is approximately the same for both policies as illustrated in Figure 13(a) where we have plot the cost function without the server costs. The benefit between the two policies is due to the server costs as shown in Figure 13(b). The server costs are $12\% - 40\%$ more expensive for the random policy.

So, the important conclusion is that, for the same QoS level, a CDN implementing a placement policy based on popularity generates a cheaper network in terms of acquisition and management costs.

5 Conclusion

In this paper, we have addressed the CDN server placement optimization problem. The innovative modeling approach we proposed allows to formulate the aforementioned optimization problem using realistic cost functions with a reduced number of parameters which do not depends on the size of the network. The approach thus scales for large networks.

More precisely, the involved parameters are the intensity of a point process used to model the ASes of the Internet, the mean number of requests per unit time per AS, the servers cost and a function which represents the QoS perceived by the users. We obtained the optimal number of ASes where a CDN server must be deployed and the impact of the number of requests and of the servers cost on the optima.

We have modeled and analyzed the cases of homogeneous and heterogeneous servers. In the second case, different policies for object placement among the server classes can be studied with our model. As an example, we compared two particular policies. The first one is based on the object popularity and the second one consists in randomly placing the objects among the server classes. The popular policy proved to be more efficient than the random one. We highlighted that for the popular policy, it exists an optimal repartition of the objects among the server classes that minimizes the global cost.

This work is being enhanced and extended in two directions. First, we are evaluating more precisely the distribution of the number of BGP hops between the ASes. Second, we are including in the cost function a term that represents the cost of distributing and updating the content to the surrogates.

A Appendix : Computation of the cost function

In this section, we present the complete computation of the cost function for the homogeneous server case. The computation for the heterogeneous case is almost the same and requires a more complicated notation, so it is not presented here for the sake of clarity. The reader interested in these computations (as well as in a more detailed presentation of the computations for the homogeneous case) can see [18]. For an introduction to the stochastic geometry and for a presentation of the different formulae used in this Appendix, the reader may see [12] and [13].

The cost function introduced in Section 3:

$$Cost = \mathbb{E}\left[\sum_{x\in N_p\cap B(o,R)} \alpha\right] + \mathbb{E}\left[\sum_{y\in N\cap B(o,R)} m_y f(z_y)\right],$$

can be rewritten as

$$Cost = \mathbb{E}\left[\int B(o,R)\alpha N_p(dx)\right] + \mathbb{E}\left[\int_{B(o,R)}\int_{V_x(N_p)} m_y f\left(\|y-x\|\right) N(dy)N_p(dx)\right].$$

Since α is a constant, we have for the first term of the cost :

$$\mathbb{E}\left[\int B(o,R)\alpha N_p(dx)\right] = \lambda_p \pi R^2 \alpha$$

For the second term of the cost, since $N = N_p + N_{1-p}$

$$\mathbb{E}\left[\int_{B(o,R)}\int_{V_x(N_p)} m_y f\left(\|y-x\|\right) N(dy)N_p(dx)\right] = \mathbb{E}\left[\int_{B(o,R)} m_y f(0) N_p(dy)\right]$$
$$+\mathbb{E}\left[\int_{B(o,R)}\int_{V_x(N_p)} m_y f\left(\|y-x\|\right) N_{1-p}(dy)N_p(dx)\right]$$

The marks m_x are i.i.d., thus,

$$\mathbb{E}\left[\int_{B(o,R)} m_y f(0) N_p(dy)\right] = \lambda p \pi R^2 \overline{m} f(0)$$

For the other term, we have,

$$\begin{aligned}
\mathbb{E}\left[\int_{B(o,R)}\int_{V_x(N_p)} m_y f\left(\|y-x\|\right) N_{1-p}(dy)N_p(dx)\right] &= \lambda p \int_{B(o,R)} \mathbb{E}\left[\int_{V_0(N_p)} m_y f\left(\|y\|\right) N_{1-p}(dy)dx\right] \\
&= \lambda(1-p)\int_{B(o,R)} \mathbb{E}^o_{N_{1-p}}\left[m_{Z_0} f\left(\|Z_0\|\right)\right] dx \\
&= \lambda(1-p)\pi R^2 \overline{m}\mathbb{E}^o_{N_{1-p}}\left[f\left(\|Z_0\|\right)\right] dx
\end{aligned}$$

The first equality is obtained by Campbell's formula, the second is the application of the Neveu's exchange formula. $\mathbb{E}^o_{N_{1-p}}[.]$ is the expectation under the Palm measure with respect to the point process N_{1-p} and Z_0 is the point of N_p which is the closest to the origin of the plane.

The point processes N_p and N_{1-p} being independent, we have,

$$\begin{aligned}
\mathbb{E}^o_{N_{1-p}}\left[f(\|Z_0\|)\right] &= \mathbb{E}[f(\|Z_0\|)] \\
&= \mathbb{E}\left[\int_{\mathbb{R}^2} f\left(\|z\|\right) \mathbb{1}_{\{N_p(B(o,\|z\|))=0\}} N_p\left(dz\right)\right] \\
&= \lambda p \int_{\mathbb{R}^2} f\left(\|z\|\right) \mathbb{P}^o_{N_p}\left(N_p\left(B\left(-z,\|z\|\right)\right)=0\right) dz \\
&= \lambda p \int_{\mathbb{R}^2} f\left(\|z\|\right) \exp\{-\lambda p \pi \|z\|^2\} dz \\
&= \lambda p 2\pi \int_0^{+\infty} r f\left(r\right) \exp\left\{-\lambda p \pi r^2\right\} dr
\end{aligned}$$

The equalities are obtained from Campbell's Formula and Slyvniak's theorem.
In the case where $f(0) = 0$, as in the example studied in this paper, we obtained :

$$Cost = \lambda \pi R^2 \left[p\alpha + (1-p)\overline{m}\lambda p 2\pi \int_0^{+\infty} r f(r) \exp\{-\lambda p \pi r^2\} dr \right].$$

References

[1] L. Qiu, V. Padmanabhan, G. Voelker *On the Placement of Web Server Replicas*, In Proc. of 20 th IEEE INFOCOM, Anchorage, USA, April 2001.

[2] P. Krishnan, D. Raz et Y. Shavitt *The Cache Location Problem*, IEEE/ACM Trans. on Networking, 8(5):568–582, October 2000.

[3] P. Radoslvov, R. Govindan, D. Estrin *Topology Informed Internet Replica Placement*, In Proceedings of the International Workshop on Web Caching and Content Distribution, June 2001.

[4] S. Guha, K. Munagala, and A. Meyerson. Hierarchical placement and network design problems. Proceedings of 41st IEEE Symposium on Foundations of Computer Science, 2000.

[5] Konstantinos Kalpakis, Koustuv Dasgupta, Ouri Wolfson *Optimal Placement of Replicas in Trees with Read, Write, and Storage Costs.* IEEE Transactions on Parallel and Distributed Systems 12(6): 628-637 (2001).

[6] A. Venkataramani, M. Dahlin, and P. Weidmann. *Bandwidth constrained placement in a WAN.* In Proceedings of the 20th International Conference on Distributed Computing Systems, Aug 2001.

[7] F. Baccelli, M. Klein, M. Lebourges and S. Zuyev, *Stochastic Geometry and Architecture of Communication Networks.* Telecommunication Systems, 7, pp. 209-227, 1997.

[8] F. Baccelli and B. Blaszczyszyn. *On a coverage process ranging from the boolean model to the poisson Voronoï tessellation, with applications to wireless communications.* Adv. Appl. Prob., 33(2), 2001.

[9] F. Baccelli, D. Kofman, J.L. Rougier. *Self-Organizing Hierarchical Multicast Trees and their Optimization.* IEEE Infocom'99, New-York (USA), March 1999.

[10] A. Busson, J-L Rougier, D. Kofman *Analysis and Optimization of Hierarchical Reliable transport protocols*, Procceedings of Internet Teletraffic Congress (ITC 17), December 2001, Salvador (Brazil).

[11] Stochastic Geometry : A Tool for Modeling Telecommunication Networks *http://www.di.ens.fr/ mistral/sg/*

[12] D. Stoyan, S. Kendall, J. Mecke *Stochastic Geometry and its Applications, second edtion*, J. Wiley and Sons.

[13] D. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes.* Springer Series in Statistics, Springer Verlag New-York, 1988.

[14] Y. Rekhter, T. Li, *A Border Gateway Protocol 4 (BGP-4).* RFC 1771. 1995.

[15] L. Breslau et al., *Web caching and Zipf like distribution : Evidence and implications*, In procceedings of IEEE Infocom , March 1999, New York.

[16] C. Cunha, A. Bestavros and M. Crovella. *Characteristics of WWW Client based traces.* Technical Report TR-95-010, Boston University, Computer Science Dept., USA, 1995.

[17] CIDR report *http://bgp.potaroo.net*

[18] A. Busson, *Modélisation Spatiale des Réseaux, PhD Thesis*, ENST, December 2002, to be printed, www.enst.fr/~abusson.