Topology-Informed Internet Replica Placement [*]

Pavlin Radoslavov
pavlin@catarina.usc.edu

Ramesh Govindan [†]
govindan@isi.edu

Deborah Estrin [‡]
estrin@isi.edu

ABSTRACT

Recently, several studies have looked into the problem of replicated server placement on the Internet. Some of those studies have demonstrated that there exists a replica placement algorithm that can perform within a factor of 1.1–1.5 of the optimal solution. However, this particular greedy algorithm requires detailed knowledge about network topology, and knowledge about expected client locations on the topology. One of these previous studies has also looked at topology-informed replica placement. They consider placing replicas at highly connected nodes in the Autonomous System level graph. In this paper we extend their work by investigating the performance of topology-informed placement on Internet router-level topology. In our evaluation, we consider approximated policy-based paths, and examine the sensitivity of our results to different client placements. We find that topology-informed replica placement methods can achieve average client latencies which are within a factor of 1.1–1.2 of the greedy algorithm, but only if the placement method is designed carefully.

Keywords

Content Distribution Network, replica placement algorithms, router fanout

1. INTRODUCTION

Content Distribution Networks (CDNs) [2, 10, 6] replicate Web content in an effort to reduce client access latency. This kind of replication can also reduce network overhead. However, the efficacy of content distribution can crucially depend on the placement of these replicas, and on the relative location of the client population.

In the past, there have been several studies that have addressed the problem of replica placement on the network and its impact on network performance [12, 18, 11, 13]. A number of replica placement methods have been proposed and studied. Two of the studies [18, 11] have considered a greedy placement strategy [1] which, compared to the computationally expensive optimal solution, performs remarkably well in practice (within a factor of 1.1–1.5), and is relatively insensitive to imperfect input data. Unfortunately, this greedy placement requires knowledge about the client locations in the network, and all pairwise inter-node distances, which information in many cases may not be available.

One of the previous studies [11] considers also topology-informed replica placement, where nodes are selected as replicas in decreasing order of their node degree [2]. Their results suggest that this method can perform almost as good as the greedy placement. However, due to lack of more detailed network topology, this particular study uses only Autonomous Systems (AS) topologies (real-world and generated) where each node represents a single AS, and a node link corresponds to AS-level BGP peering.

In this paper we extend their evaluation of fanout-based replica placement in several ways. First, instead of using a coarse-grained AS topology derived from BGP AS paths information [16], we have in our possession an approximate router-level Internet topology [8] which we use to obtain more detailed and accurate results. Second, instead of shortest-path routing, we generate router-level paths using approximate models of inter-AS routing policy [20]. With their technique, each router from the router-level topology is mapped to the AS it belongs to (based on that router IP address), and then AS-level shortest-path routing is combined with router-level shortest-path towards the next-hop AS. Finally, we look into results sensitivity by considering various client placement models, and some other topologies.

[*]This work was supported by the Defense Advanced Research Projects Agency under grant DABT63-98-1-0007. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency.

[†]P. Radoslavov and R. Govindan are with the USC/Information Sciences Institute, 4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90292-6695, USA.

[‡]D. Estrin is with the Information Sciences Institute and UCLA.

[1]The particular greedy placement is also very similar to the one in [12].

[2]For the rest of the paper we will use interchangeably the terms *node degree* and *node fanout* to represent the number of links connecting a node with its neighbors. Also, we will use the term *well-connected node* to indicate a node that has a large fanout.

Our main findings are:

- In most cases the router-level fanout placement is almost as good as the greedy placement (within a factor of 1.1–1.2).
- A fanout-based replica placement method needs to be carefully designed to be efficient. For example, if we select first a well-connected AS and then we select a router within that AS, we must be very careful which particular router is selected.

Our conclusions do not depend on client locations. Only if the number of clients is very small, then there is a significant performance difference between the fanout-based replica placement and the greedy placement. The results are true also for random graphs, generated and real-world AS topologies, but do not apply for overlay topologies such as Mbone [14]. At the end of the paper we present a possible explanation when and why those results may hold.

The rest of the paper is organized as follows. In Section 2 we discuss some of the previous work. In Section 3 we describe the particular replica and client placement models we consider in this paper. Section 4 contains the performance evaluation results. Conclusions and future work are in Section 5.

2. RELATED WORK

A number of papers have addressed the problem of Web server replica or cache placement impact on performance. Note that some of the recently published papers are independent studies, but have notable similarity in problem formulation and final results. The replica or cache placement problem can be modeled after the *center placement problem*, a well-known problem in graph theory, and in particular two of its variations: *the facility location problem*, or *the minimum K-median problem* [5]. A number of approximate solutions have been proposed in the past [23], but they are either very computationally expensive, or are difficult to apply in practice.

Krishnan et al. [12] study the problem of placing transparent en-route network caches (TERCs), and in particular how various placement methods can be used to reduce the network traffic or the average access latency. Unlike our work where we assume the replicas can be placed anywhere in the network, their work allows the caches to be only on the path between a client and the server.

Qiu et al. [18] consider the problem of placement strategies for Web server replicas within the context of CDNs that offer Web server hosting services. They propose several placement algorithms, including a simple greedy placement which we use in this paper, and which is very similar to the greedy algorithm in [12]. They find that this greedy algorithm performs very well in practice (typically within a factor of 1.1–1.5 of the optimal solution). Further, its performance is relatively insensitive to imperfect input data such as client locations and network topology information. However, this study does not consider node-fanout based placement.

The study by Jamin et al. [11] is similar to [18]. Their work examines the impact of the number of replicas on the performance of various replica placement methods. Their main finding is that, regardless of the placement method, increasing the number of replicas is effective in reducing client download time only for a very small number of replicas. They also discuss an AS-level fanout-based placement, in which replicas are placed within ASs in decreasing order of node degree on the AS topology. The results suggest that the AS-level fanout-based placement can perform almost as well as the greedy placement. Our study is centered around this finding, and we try to verify it through more detailed simulations by using router-level Internet topology, instead of only AS-level topology, and by exploring in more details the impact of various replica and client placement methods.

3. REPLICA AND CLIENT PLACEMENT MODELS

In this section we describe the replica placement models we are interested at, and are evaluated later in Section 4. We use each of those models to place a number of replicas on the topology, so we could eventually reduce the client access latency and the overall network overhead (compared to a single-server solution). We also describe the client placement models that we use to select a number of nodes as clients. Those models are used in Section 4.3.2 to perform the client-impact sensitivity evaluation. Before presenting the replica and client placement models, we describe the client-replica assignment we assume.

3.1 Client-Replica Assignment

In this paper we assume that each client selects the closest (in number of hops) replica. Indeed, it is possible to consider a more sophisticated scheme where a client selects in real-time the replica that offers the lowest latency, but for simplicity we ignore such schemes. The second assumption we make is that we do not limit the number of a clients that can be assigned to a replica. Both assumptions are similar to those in some of the previous work ([18, 12, 11]). One of the arguments to support the latter assumption is that typically it is much easier to increase the capacity of a particular replica (*e.g.,* by creating a cluster of replicas at the same location), than deploying a new replica at different location for the sake of reducing other replicas' load. The latter assumption does not impact our conclusions even in the presence of flash crowds, because, as we demonstrate later in the paper, our results are robust to variations in client locations and client population size.

3.2 Replica Placement Models

In this paper we consider the following replica placement methods. The first method has been proposed in some of the previous work (see Section 2). In our study we use it as a base for comparison.

- *Greedy placement.* The greedy placement we choose is same as the greedy algorithm described in [18] and [11]. The basic idea is to choose the replicas one-by-one, a subject to a greedy selection: at each step we evaluate all nodes in the topology and choose the one that, if we place a replica there, the resulting network overhead will be minimized. The process is repeated until all replicas have been chosen. The input to this method is all pairwise inter-node distances, and the client placement locations.

- *Max-router fanout placement.* Given a network topology and the fanout of each node, we choose the replicas one-by-one in decreasing order of their node degree until all replicas have been chosen. The intuition behind this method is that the nodes with large fanout are eventually the closest (on average) to all other nodes, and therefore they are a better choice for replica location.

- *Max-AS/max-router fanout placement.* This method assumes that each node/router has been assigned to some AS, and that all ASs have been connected into an AS-level topology. If R is the number of replicas to select, first we select the R ASs that have the largest fanout (on the AS-level topology). Then, within each selected AS, we choose the router that has the largest router-level fanout. Similar to the *max-router fanout* placement, the intuition is that the selected nodes will be closer to the rest of the nodes.

- *Max-AS/min-router fanout placement.* This method is similar to the *max-AS/max-router fanout* placement, except that instead of selecting the router with largest fanout within each of the chosen ASs, we select the router with the smallest fanout. This placement may not make sense for practical purposes, but we need to consider it to evaluate the sensitivity of network performance to replica placement within an AS. Note that for the rest of the paper, when we use the term *fanout-based placement*, we do not include the *max-AS/min-router* fanout placement, unless stated otherwise.

- *Random placement.* In this method the replicas are chosen at random with uniform probability among all nodes in the topology. We consider it as an "upper-bound" placement method in a sense that an efficient replica placement method should always be better than the random placement.

Unlike the previous work described in Section 2, we do not consider some of the existing optimal solutions that have been proven to be always within a small factor of the most optimal solution. As we mentioned earlier in Section 2, the greedy placement has been shown to perform very well in practice, within a factor of 1.1–1.5 of the computationally intensive optimal solutions. Therefore it is a reasonable choice for our needs and we can use it as a base of comparison.

3.3 Client Placement Models

To investigate the sensitivity of replica placement performance to client locations, we look into several client placement models. Our goal is not to explore all possible client placements, but to consider the extreme cases, along with the random case, because the extreme cases can give us the boundary of expected performance.

The first model we look into is the *random client placement*, where the client nodes are selected at random with uniform probability.

We also look into the extreme client placement as defined in [17], namely *extreme affinity* and *extreme disaffinity*. The extreme affinity model places the clients as close as possible to each other; the extreme disaffinity model places the clients as far as possible from each other. The particular algorithm we use to place a number of clients on a graph according to the affinity/disaffinity model is described in [24]. Below is a brief summary of that algorithm. The first client is selected at random among all nodes. Then, we assign to each node n_i that is not selected yet the probability $p_i = \frac{\alpha}{w_i^\beta}$, where w_i is the closest distance between node n_i and a node that is already selected as a client, α is calculated such that $\sum_{n_i} p_i = 1$, and β is the parameter that defines the degree of affinity or disaffinity. After a node is chosen to be a client, the probabilities of the remaining nodes are recomputed and the process is repeated until the desired number of clients is selected. Similar to [24], in our experiments we use $\beta = 15$ and $\beta = -15$ for extreme affinity and disaffinity respectively.

To verify our results with real-world data, we use Web server access logs to create the population of clients. In particular, we collect the unique IP addresses of all clients that have accessed the same Web server within some period of time. Then, we run a traceroute to each of the client addresses. Finally, we intersect each of the traceroute paths with the Internet map to find the last-hop router toward a Web client that is on that map. The set of all last-hop routers is our *web clients set* that can be used to represent the population of the real-world Web clients.

4. PERFORMANCE EVALUATION

In this section we present the main results from our evaluation. In particular, we use numerical simulations to compute the relative network performance. As part of our evaluation we look into the impact of various factors on performance: replica and client placement, client number, network topology. First we describe the metric space, and then we present the results when we vary each of the input factors we consider.

4.1 Metric Space

The two particular metrics we are interested at are *average client latency* and *overall network overhead*. For simplicity, we assume that the latency between two nodes is proportional to the number of link-hops between them. A similar assumption has been used in a previous work [18].

Indeed, [9] shows that router-level hops correlate well with observed latency. That work points out that the number of ASs in the path to a destination has a higher correlation to latency. However, that study is several years old and Internet has evolved since then. Further, a more recent study has measured 50–70% correlation between network hop and round-trip time [15], and its authors claim that the router-level number of hops is more meaningful as a latency metric [3]. Finally, due to lack of information, we assume that the bandwidth capacity of all links is same. Obviously, those assumptions are not perfect, but without detailed network measurements this is the best we can do. Hence, the average client latency across all clients c can be computed as:

$$AveClientLatency = \frac{\sum_{clients(c)} Dist(c, Replica(c))}{NumberOfClients}$$

where $Replica(c)$ is the replica node for client c, and $Dist(c, Replica(c))$ is the distance between them in number of hops.

For similar reasons as above, we also assume that the overall network overhead is proportional to the number of link-hops used to disseminate the data from the replicas to all clients. At the same time, we ignore the network overhead to distribute the data from its original location to each of the replicas, because this overhead may be a small fraction of the network overhead to distribute the data from the replicas to a large number of clients. Hence, the overall network overhead for all clients can be computed using the following formula:

$$NetworkOverhead = \sum_{clients(c)} Dist(c, Replica(c))$$

In our evaluation, we are not interested in the absolute client latency or absolute network overhead metrics. Instead, we are interested in the *relative client latency* or *relative network overhead* of each replica placement method versus the greedy placement. Based on our assumptions, we have $AveClientLatency = \frac{NetworkOverhead}{NumberOfClients}$, therefore it is easy to see that when we perform *relative* comparison between two replica placement methods using the same set of clients, then the relative average client latency will be the same as the relative network overhead. Therefore, for the rest of the paper we use a single metric we call *efficiency ratio* to compute the relative performance between two replica placement methods. We always use the greedy placement as a base for comparison, hence the efficiency ratio of method M_i can be computed as:

$$EffRatio(M_i) = \frac{NetworkOverhead(M_i)}{NetworkOverhead(Greedy)}$$

[3]We also show, in Section 4.3.4, that if we consider latency defined in terms of AS-level hops, our findings hold even stronger.

4.2 Simulation Setup

For most of our simulations (except for those described in Section 4.3.4 where we look at the topology impact factor), we use a real-world router-level topology. The topology information was collected by using a large number of traceroute requests sent over the Internet [8, 22]. The resulting topology had 102639 nodes and 142303 links. Then we recursively removed all nodes that have a fanout of one to obtain a topology we call *Internet core*. The reason that we truncate the original topology is to remove the long, "skinny" branches that do not represent well the network connectivity at the edges, but are an artifact from the particular methodology used to obtain the topology information.

To obtain more realistic results, instead of using shortest-path routing, we use AS-level hierarchical routing as described in [20]. With their technique, first each router from the router-level topology is mapped to the AS it belongs to, based on that router IP address and the AS-level topology [4] at the time the router-level topology data was collected. After that, the AS-level shortest-path routing is computed. Finally, to compute the router-level path between two nodes, the AS-level path is followed and within each AS the router-level shortest path is used to reach the closest node that belongs to the next-hop AS.

Topology	Nodes	Links	Diam.	Ave. dist.	Ave. fanout
Internet core	27646	67310	26	8.3	4.9
Random graph	19596	40094	16	7.2	4.1
Power-law graph	10091	23253	9	3.2	4.6
AS	4830	9077	11	3.7	3.8
Mbone	4179	8549	26	10.1	4.1

Table 1: Metrics of used topologies

Within each set of simulations we fix the number of clients and vary the number of replicas, or vice-versa. The number of replicas varies between 1 and 50; the client population size varies as a fraction of the number of all nodes between 0.005 and 0.2. The replica and the client placement methods, as described in Section 3, are the other input to the simulations. In all simulations we use 100 different sets of algorithmically chosen clients (except for the Web-derived clients when we have 3 sets), and we average the results among all trials. The results we show are for the 95% confidence interval (note that in most cases this interval is very small and can be seen as a single dot) [4]. To create the set of Web-derived clients, we use the access logs of a busy Web server for three consecutive days, and we apply the technique described in Section 3.3 to compute the nodes on the router-level topology that represent the Web clients. The number of unique client addresses for each of the three days is 37401, 40833, 43558 respectively. Those clients, after the intersection of the traceroute paths with the router-level topology, are represented by 4015, 4158 and 4264 unique nodes on the Internet-core map (approximately 15% of all nodes).

[4]We looked also into the min-max interval, and it was almost unnoticeable for the *max-router fanout* and *max-AS/max-router* placement methods.

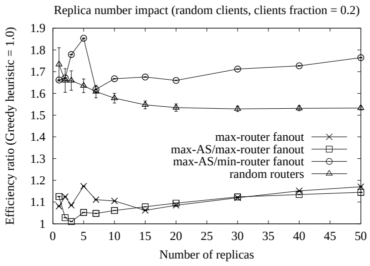

Figure 1: Internet core: replica placement impact (random clients)

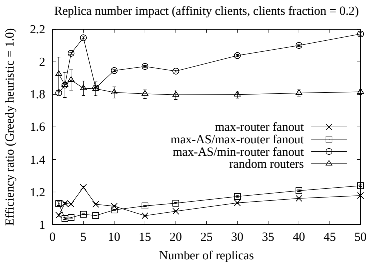

Figure 2: Internet core: replica placement impact (extreme affinity clients)

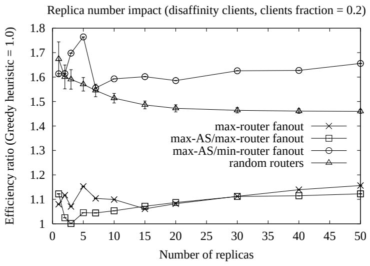

Figure 3: Internet core: replica placement impact (extreme disaffinity clients)

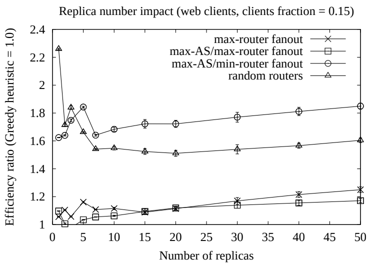

Figure 4: Internet core: replica placement impact (web clients)

We tried also some generated topologies, and some other real-world maps (see Section 4.3.4). Table 1 summarizes some of the metrics of all topologies.

4.3 Network Efficiency Results

First we present the results for different replica placement methods, which are of most interest to us. Then we look at how the client placement may have impact on performance. Finally, we look into other factors such as client number and network topology.

4.3.1 Replica Placement Impact

To evaluate the replica placement impact, we assume a fixed number of randomly placed clients (20% of all nodes), and the number of replicas varies between 1 and 50. Then we compute the relative network efficiency for different replica placement methods (as described in Section 3.2), by using the greedy algorithm results as the base for comparison (1.0).

The results from this simulation are on Figure 1. The first observation we can make is that both *max-router fanout* and *max-AS/max-router fanout* placement methods perform very well, within a factor of 1.1–1.2 of the greedy placement, regardless of the number of replicas. This result is our first confirmation that the fanout-based placement methods perform well even on Internet router-level topology. On the other hand, the *max-AS/min-router fanout* placement performs even worse than random replica placement. This, to some extent, is a surprising result, because we expected that the AS fanout is the major factor that has impact on performance. Clearly, placement is very sensitive to the actual selection of routers within ASs–selecting the highest fanout AS alone for placing a replica is not sufficient.

4.3.2 Client Placement Impact

To evaluate the client placement impact on the results, first we consider the extreme cases of *affinity* and *disaffinity*. The particular model we use was described already in Section 3.3. Figure 2 and Figure 3 show the results for extreme affinity and extreme disaffinity respectively (the rest of the setup is same as in the case of random client placement in Section 4.3.1). Here again we can see that the *max-router* and *max-AS/max-router* fanout-based placement methods perform remarkably well within a factor of 1.1–1.2 of the greedy placement.

The results with the *web-clients* are on Figure 4. Similar to the extreme affinity and extreme disaffinity client placement, with web-clients the fanout-based placement methods perform equally well.

It is interesting to note that, unlike the greedy placement, the fanout-based replica placement methods do not take client locations into account, yet they can perform very well over a wide range of client placements (including realistic placements).

4.3.3 Client Number Impact

The next question we want to answer is how the client population size impacts the performance. In this set of simulations we fix the number of replicas to 50, and then we vary the fraction of nodes that are clients in the range 0.005–0.2. Figure 5, Figure 6 and Figure 7 show the results

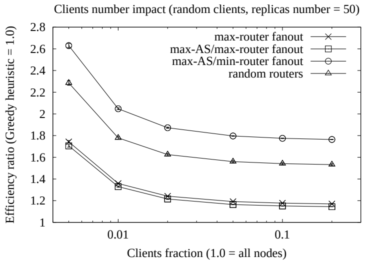

Figure 5: Internet core: client number impact (random clients)

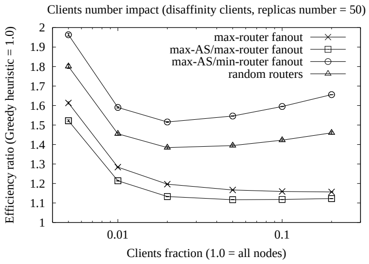

Figure 7: Internet core: client number impact (extreme disaffinity clients)

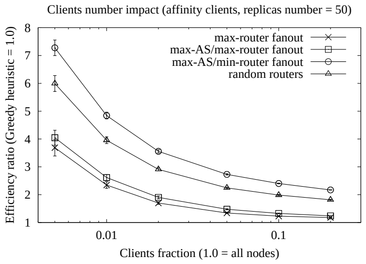

Figure 6: Internet core: client number impact (extreme affinity clients)

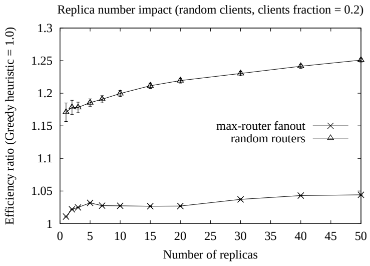

Figure 8: Random graph: replica placement impact (random clients)

for various client placement: random, extreme affinity and extreme disaffinity respectively.

We observe that when the number of clients is small, the fanout-based placement methods do not perform very well. This is especially true for extreme affinity client placement. For moderate and large client number, the fanout-based placements perform much better, as expected [5]. Another observation we can make is that the performance difference is larger with extreme affinity of the clients. This result is not unexpected, because when all clients are clustered together in some part of the network, there is relatively low probability that there will be a node with large fanout in their proximity that will be selected as a replica (unless the number of clustered clients is very large and altogether they cover a notable fraction of the network).

From the above results we can see that the client population size has impact on performance only when the number of clients is small. Only then the fanout-based placements do not perform as well as the greedy placement.

4.3.4 Network Topology Impact

The next factor we consider that may have impact on performance is the type of topology. First, we repeat the same simulations with two different types of generated topologies. The first one is a random graph (see Table 1 for some of its metrics, as well for the metrics of the other topologies), generated by the GT-ITM topology generator [3]. The second one is a power-law graph [6] created by a generator based on the algorithm described in [1]. A recent study shows that this topology qualitatively resembles both the AS and the router-level topologies [21]. Obviously, we do not have ASs over the generated topologies, therefore we have to use shortest-path routing; for the same reason we cannot apply the *max-AS/max-router* or the *max-AS/min-router* replica placement methods.

The results for the random graph with randomly placed clients (20% of all nodes) when we vary the number of replicas are on Figure 8. We can see that the difference between the greedy placement and the *max-router fanout* placement is even smaller (within a factor of 1.05). On the other hand, even random replica placement can perform within a factor of 1.25 which is much lower compared to the Internet-core results.

The results for the the power-law graph are on Figure 9. As expected, the fanout-based placement performs very well, while the random placement performs notably worse. We tried also a real-world AS-level topology itself [4], and the results were very similar to the results for the generated power-law graph.

[5]The *random* and *max-AS/min-routers* placement performance for extreme disaffinity of the clients as a function of the client population size may seem a little bit unusual because it is not monotonically increasing or decreasing. This behavior can be explained by the fact that the results can be influenced significantly by various factors if the number of clients is very small (of the order of the number of replicas)

[6]One of the characteristics of the power-law graphs is that the node fanout distribution can be described by a power law: $f_d \propto d^{\beta}$ where f_d is the frequency of out-degree d, and β is a constant.

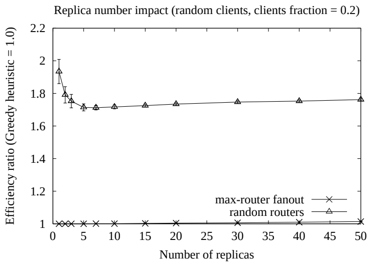

Figure 9: Power-law graph: replica placement impact (random clients)

For all topologies, the extreme affinity and extreme disaffinity client placement results were similar to the random client placement results.

However, we should note that this result is not universal. When we tried the Mbone [14] overlay topology [22], we found that the *max-router fanout* replica placement is not better than the random placement. In fact, in most cases it was even worse. We believe the reason is that the connectivity of the Mbone topology is very sparse compared to the other topologies we have considered (compare the topology size, topology diameter and average inter-node distances of all topologies on Table 1). As a result, the average distance from a well-connected node to the rest of the nodes is relatively large, therefore such node is not a good choice to be a replica.

Finally, we should note that when we repeated the Internet-core simulations with using shortest-path routing instead of the approximated AS-policy routing, we did not see any notable difference. From this we can conclude that the impact of the routing on relative performance is insignificant.

4.4 Results Discussion

In this section we discuss our findings, and try to explain the reasons behind some of them.

Our main finding is that the fanout-based placement methods can perform remarkably well. Unlike more sophisticated methods such as the greedy placement that takes into account the client locations to compute the appropriate replica placement, the fanout-based placement does not require any knowledge about expected client locations. This is a very significant simplification, because it basically suggests that we do not need a dynamic and adaptive replica placement that requires knowledge about expected client locations. In other words, as long as the replicas are placed in some "key" locations in the network, then the expected performance would be reasonably good.

If we abstract from the particular replica placement methods and ask the question "if we had to select a single node as a replica, what would be the best node to select", the answer would be the node that is as close as possible to all clients. Typically, if a node has a large fanout, it means that it is a one-hop away from a large number of nodes, and therefore probabilistically it is close to a large number of clients as well. The AS and the Internet topologies have the characteristics of power-law graphs [7, 19]. One of the characteristics of those topologies is that they have a small number of nodes with very large fanout which nodes are apparently just few hops away from all other nodes. In other words, the high-fanout nodes are the "key-location" nodes for the power-law graphs that in most cases are very close to the rest of the nodes.

Here someone may ask the question why did we get similar results for the random graph which is not a power-law graph? The answer to that question may be in the fact that the majority of the nodes in the random graph have similar fanout, while overall random graphs have very high *topology expansion* (defined as the growth of neighborhood size as a function of distance) [19]. The combination of these two factors eventually means that all nodes are just few hops away from each other, and then choosing any node to be a replica will be a good solution. This observation also explains why the random replica selection performs much better for random graph compared to power-law graph (see Figure 8 and 9).

Another observation is that a two-level fanout-based placement such as the *max-AS/max-router* placement method can perform very well. One possible speculation here may be that the AS fanout is the factor that matter, *i.e.,* that choosing any node within an AS with a large fanout will be a good solution. However, the results for the *max-AS/min-router* replica placement show that router selection based on the AS-level fanout only is not sufficient: the router-level fanout must be considered as well. One possible explanation to this is as follows. Typically, an AS with large fanout has inside a large number of nodes, and some of those nodes may not be very well connected [7]. Therefore, choosing such not well connected node as a replica may not be beneficial at all to reduce client latency or network overhead.

Finally, we should note that in most cases the *max-AS/min-router fanout* replica placement performs slightly better compared to the *max-router fanout* placement. The reason for this, we believe, is that the *max-AS/min-router fanout* placement spreads the replicas among a number of ASs, by placing no more than one replica inside each AS. On the other hand, the *max-router fanout* placement does not have this property, therefore it may place a number of replicas very close to each other without adding significant benefit to the clients.

5. CONCLUSIONS AND FUTURE WORK

In this paper we consider the problem of replica placement for Content Distribution Networks. In particular, given a number of replicas and a network topology, where should we place them such that the average client latency and the overall network overhead are minimized. Previous studies

[7]Indeed, when we compared the number of nodes assigned to each AS versus the AS fanout, on average the relation can be approximated with a straight line on the log-log scale, which means that each of the few ASs with the largest fanout contains a large number of nodes inside.

have proposed a simple greedy placement algorithm that performs very well (within a factor of 1.1–1.5) compared to the optimal solution. Unfortunately, this algorithm requires knowledge about expected client locations, and all pairwise inter-node distances.

We look into few other replica placement methods and compare them with the greedy placement. Interestingly, fanout-based placement methods that choose the nodes with the maximum fanout perform very well (within a factor of 1.1–1.2) compared to the greedy placement. Unlike the greedy placement, the fanout-based placements do not require knowledge about client locations, and do not need detailed network topology information. The only information needed is the set of nodes with the largest fanout (including eventually, in case of Internet, the AS-level fanout as well). Further, unlike the greedy placement solution, a fanout-based solution does not tend to change significantly even on a long-range time scale.

The implication of those finding is not that we want to add various replicas to the high-fanout routers to achieve good performance. Typically, those routers are very busy, and it may be technically impossible to add any services to them. One possible solution would be to build a cluster of replicas that are connected to the network in a point that is as close as possible to a high-fanout router. An interesting problem then would be to investigate how the addition of such clusters of replicas would change the traffic volume at the high-fanout routers and at the rest of the network, and how the change in traffic may have impact on performance.

Someone may argue that CDN providers may possess detailed topology information, and may have reasonable estimation about expected client location, therefore to compute the replica locations a greedy algorithm would be preferred. While it may be true, it is not difficult to imagine that in many situations that information may not be available, or may be inaccurate (*e.g.,*, right after major topology or routing reconfiguration). In that case, the fanout-based heuristic can be a quick and easy solution, and that solution could be refined later with the help of more sophisticated algorithms.

The fanout-based placement methods are not universal. They do seem to perform very well on power-law and random graphs, but may not be a good solution if most of the nodes have similar, relatively low fanout. Investigating in more details the particular network topology characteristics that may have impact on the fanout-based replica placement performance is part of our future work.

6. ACKNOWLEDGMENTS

Hongsuda Tangmunarunkit suggested the *Max-AS/min-router fanout* replica placement, and provided us with the AS-level overlay mapping for the Internet-core topology. The Internet-core topology was obtained from the SCAN project [22]. The AS-level information was obtained from the NLANR server [4], which itself collected it from the University of Oregon Route Views Project [16]. Finally, we would like to thank the anonymous reviewers as well for their valuable comments.

7. REFERENCES

[1] W. Aiello, F. Chung, and L. Lu. A Random Graph Model for Massive Graphs. In *Proc. of the 32nd Annual Symposium on Theory of Computing*, 2000.

[2] Akamai. http://www.akamai.com/.

[3] K. L. Calvert, M. B. Doar, and E. W. Zegura. Modeling Internet Topology. *IEEE Communications Magazine*, June 1997.

[4] UCSD/San Diego Supercomputer Center. The NLANR Project. http://moat.nlanr.net/Routing/rawdata/.

[5] Pierluigi Crescenzi and Viggo Kann (Editors). A compendium of NP optimization problems. http://www.nada.kth.se/~viggo/problemlist/.

[6] Digital Island. http://www.digitalisland.com/.

[7] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On Power-Law Relationships of the Internet Topology. In *Proceedings of the ACM SIGCOMM'99*, Cambridge, Massachusetts, USA, August 1999.

[8] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for Internet Map Discovery. In *Proceedings of the IEEE Infocom 2000*, Tel-Aviv, Israel, March 2000.

[9] James Gwertzman. Autonomous Replication in Wide-Area Internetworks. BA thesis, Harvard College, Cambridge, Massachusetts., 1995.

[10] Inktomi. http://www.inktomi.com/.

[11] Sugih Jamin, Cheng Jin, Anthony R. Kurc, Danny Raz, and Yuval Shavitt. Constrained Mirror Placement on the Internet. In *Proceedings of the IEEE Infocom 2001*, Anchorage, Alaska, USA, April 2001.

[12] P. Krishnan, Danny Raz, and Yuval Shavitt. The Cache Location Problem. *IEEE/ACM Transactions of Networking*, 8(5):568–582, October 2000.

[13] Bo Li, Mordecai J. Golin, Giuseppe F. Italiano, Xin Deng, and Kazem Sohraby. On the Optimal Placement of Web Proxies in the Internet. In *Proceedings of the IEEE Infocom 1999*, pages 1282–1290, New York, USA, March 1999.

[14] Michael R. Macedonia and Donald P. Brutzman. MBone Provides Audio and Video Across the Internet. *IEEE Computer*, April 1994.

[15] Katia Obraczka and Fabio Silva. Network Latency Metrics for Server Proximity. In *Proceedings of the IEEE Globecom 2000*, San Francisco, California, USA, November 2000.

[16] University of Oregon Route Views Project. http://www.antc.uoregon.edu/route-views/.

[17] Graham Phillips, Scott Shenker, and Hongsuda Tangmunarunkit. Scaling of Multicast Trees: Comments on the Chuang-Sirbu scaling law. In *Proceedings of the ACM SIGCOMM'99*, Cambridge, Massachusetts, USA, August 1999.

[18] Lili Qiu, Venkata N. Padmanabhan, and Geoffrey M. Voelker. On the Placement of Web Server Replicas. In *Proceedings of the IEEE Infocom 2001*, Anchorage, Alaska, USA, April 2001.

[19] Pavlin Radoslavov, Hongsuda Tangmunarunkit, Haobo Yu, Ramesh Govindan, Scott Shenker, and Deborah Estrin. On Characterizing Network Topologies and Analyzing Their Impact on Protocol Design. Technical Report 00-731, University of Southern California, Dept. of CS, February 2000.

[20] Hongsuda Tangmunarunkit, Ramesh Govindan, Scott Shenker, and Deborah Estrin. The Impact of Routing Policy on Internet Paths. In *Proceedings of the IEEE Infocom 2001*, Anchorage, Alaska, USA, April 2001.

[21] Hongsuda Tangmunarunkit, Ramesh Govindan, Scott Shenker, Sugih Jamin, and Walter Willinger. Network Topologies, Power Laws, and Hierarchy. Work in progress.

[22] USC/ISI. The SCAN Project. http://www.isi.edu/scan/.

[23] Vijay Vazirani. *Approximation Methods.* Springer-Verlag, 2001.

[24] Tina Wong and Randy Katz. An Analysis of Multicast Forwarding State Scalability. In *Proceedings of the 8th IEEE International Conference on Network Protocols (ICNP 2000)*, Osaka, Japan, November 2000.