RC22467 (W0205-214) May 30, 2002
Computer Science

IBM Research Report

Reduction in Round Trip Time by Using Content Distribution Networks

Dakshi Agrawal, James R. Giles, Dinesh Verma
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

Reduction in Round-Trip Time by Using Content Distribution Networks

May 6, 2002

Abstract

It is well-known that a content distribution network can reduce the client-perceived response time of a web site by bringing the content closer to the clients. The reduction in client-perceived response time is strongly dependent on the average round-trip time experienced by the clients. In this paper we quantify the reduction in round-trip time that can be experienced by placing surrogate servers in a network.

Keywords: Content Distribution Network, Performance Analysis, Round-Trip Time, Bandwidth Utilization.

1 Introduction

The objective of this paper is to quantify the reduction in the average round-trip time that can be experienced by utilizing a content distribution network (CDN). By bringing content closer to the clients, a content distribution network reduces the average round-trip time between clients and the content, which in turn strongly influences the client-perceived response time. The client-perceived response time is an important metric for distribution networks, and an estimate of the reduction in the average round-trip time can be used for performance prediction and capacity planning of such networks.

An estimate of the reduction in the average round-trip time is also needed for many engineering decisions such as designing web-applications that will be deployed over content distribution networks. For example, consider a shopping cart or a directory application, where a part of the application will be distributed on a content distribution network while the rest of the application will reside on the origin server due to security concerns. In a related paper, we have shown that distributing such applications may not provide significant benefits if the reduction in round-trip time is below a certain threshold [1, 2]. Conversely, if an estimate of the reduction in round-trip time is known, then a given architecture for splitting an application can be evaluated for the potential acceleration that it may provide. However, in such design problems, the topology of the network, client population profiles, et cetera, are unknown *a priori*. Thus, there is a demand, and a challenge, of providing a rough estimate of the reduction in round-trip time which is applicable in a broad set of network conditions.

It is clear that the exact reduction in the average round-trip time due to a content distribution network will depend on many factors such as the underlying network topology, network-link conditions, and the overall distribution of the client population. Several of these factors are either not precisely known or change frequently. This makes any exact calculation of the reduction in the average round-trip time impractical. Therefore, the objective of this paper is to provide an *estimate* for this reduction for a wide variety of link configurations and ensembles of parameters.

To the best of our knowledge, there is no prior analytical study that estimates the reduction in the average round-trip time obtained by using a content distribution network. There are a few studies in which the

performance of a deployed content distribution network is measured by several clients placed in the Internet [3]. Due to the complexity of arranging such clients, so far these studies have been limited in their scope. Other closely related work is the work on the server placement problem from which some inference can be made regarding the reduction in average round-trip time [4, 5, 6, 7]. Specifically, this body of research work addresses the following question: Given a network topology, and a budget for M surrogate servers, what is the best placement strategy for these servers for a given performance metric. Li *et al.* propose a dynamic programming solution to obtain the optimum placement of servers under the constraint that the network topology is a tree rooted at the origin server [6]. However, they do not quantify the reduction in round-trip time which may result from such placement. Such quantification is further addressed in two different settings by Qiu, Padmanabhan, and Voelker [7], and by Krishnan, Raz, and Shavitt [4]. Qui, Padmanabhan, and Voelker present several heuristic algorithms for the surrogate server placement problem on a general network graph and compare the relative performances of these algorithms by simulating different network topologies and client populations. Krishnan, Raz, and Shavitt consider logs from a specific web-server and show that by utilizing nine surrogate servers the *total traffic* can be reduced by roughly 30%. They also analytically compute the relative reduction in traffic for some simple cases where the network is modeled by regular topologies. However, the focus of their study is to show that their greedy algorithm performs well in comparison to the optimum algorithm, and they do not adequately address the question of reduction in round-trip time.

The objective of this paper is to provide a methodology to evaluate the reduction in round-trip time and to provide an estimate of the reduction which is applicable to a broad category of networks. The paper is organized as follows. In Section 2, we present our simplifying assumptions and the network model used herein. In Section 3, our general strategy to estimate the reduction in round-trip time is outlined. Next, in Section 4, we estimate the reduction in the round-trip time for several combinations of client population and link-delay profiles. Finally, in Section 5, we discuss some of the ramification of our results and outline the limitations of this work.

1.1 Contributions of This Paper

In this paper, we evaluate the performance using *acceleration factor* defined as $\Pi = 1 - \rho_c/\rho_s$, where ρ_c is the average round-trip time in the presence of a content distribution network and ρ_c is the average round-trip time without a content distribution network.

We show that a spanning tree of a given network graph can be decomposed into *chain graphs*. Furthermore, the acceleration factor on the spanning tree equals a weighted average of acceleration factors on its constituent chain graphs. We show that these two facts when combined have the following important implications (precise conditions for these implications to hold true are discussed in the text):

1. Regardless of network topology, client-population and link-delay profiles, the acceleration factor on a spanning tree can always be computed by taking a weighted average of acceleration factors on its constituent chain graphs. Therefore, it is important to study the acceleration factor on chain graphs.
2. A histogram of the acceleration factors, computed by considering a wide set of chain graphs obtained by varying length, client-population profile, and link-delay profile, can provide an estimate of the acceleration factor on a *generic* network graph. For example, if most of the constituent chain graphs of a given network graph lie in the considered set of chain graphs, then the mean of the histogram is likely to be a optimistic estimate of the acceleration factor on the whole network.

Motivated by these results, we computed acceleration factors for a wide set of chain graphs by using both analytical and numerical methods. These computations show that for the set of chain graphs considered in this paper, with one surrogate server on each chain graph, the acceleration factor is likely to lie in the range $[0.4, 0.8]$ with the mean around 0.65. This in turn implies that as long as the number of surrogate servers in the content distribution network is much less than the total number of client clusters, an optimistic estimate for the acceleration factor is 0.65.

One limitation of our study is that although we consider a fairly large set of chain graphs, it is not clear if these chain graphs are representative of chain graphs that constitute the Internet. We are currently collecting data to obtain the statistics of chain graphs on the Internet. Our future plan is to use the methodology developed in this paper on the gathered statistics.

One corollary of our work is that the placement of surrogate servers to optimize consumed bandwidth may not be optimum to reduce the client-perceived response time. Furthermore, either placement may not balance the load among servers. Thus, several potential benefits of a content distribution network require conflicting placement of surrogate servers.

2 Assumptions and Models

Consider an application hosted from a single origin server. Assume that the shortest-path routing algorithm employed by the Internet is stable. Under this assumption, routes from the origin server to its clients' clusters would make a tree graph. Let $\mathcal{V}$ be the set of all vertexes of this graph and let $\mathcal{B}$ be the set of all branches. Each vertex in the graph represents a cluster of clients, and each branch represents the average networking delay between two client clusters located at the two ends of this branch. Given an assignment of client population $U(v)$ for each vertex $v \in \mathcal{V}$, and an assignment of networking delay $D(b)$ for each branch $b \in \mathcal{B}$, it is straight-forward to compute the *average* round-trip time ρ_s as

$$\frac{\sum_{v \in \mathcal{V}} U(v) \Big(\sum_{b \in P(v,s(v))} D(b) \Big)}{\sum_{v \in \mathcal{V}} U(v)}. \tag{1}$$

Here, $P(v_1, v_2)$ denotes the shortest path between v_1 and v_2, and $s(v)$ denotes the vertex where the server serving clients at v is located. In the case of a single server $s(v) = v_0$, where v_0 is the root of the tree graph.

A content-distribution network will place surrogate servers at a few vertexes in the tree graph. Assume that in the presence of a content distribution network, clients contact the server which is closest to them and which lies *on the shortest path* from them to the origin server. For a given placement of surrogate servers, (1) can be used to compute the *average* round-trip time ρ_c in presence of a content-distribution network. Note that since now there are multiple servers in the network, $s(v)$ will depend on v.

Thus, using (1), we can compute ρ_c and ρ_s, the average round-trip time with and without a content distribution network respectively, and compute the *response-time acceleration factor* $\Pi = 1 - \rho_c/\rho_s$ provided by a given content distribution network on a given tree graph.

Before proceeding further, let us examine the two assumptions made thus far. First, we assumed that the underlying network topology was given by a tree graph, and second, we assumed that clients contact only the surrogate server that lies on the shortest path between them and the origin server. The first assumption eliminates some of the possible routes which can potentially be used by the clients to reach surrogate servers. Under the second assumption, the clients do not necessarily contact the server closest to them. Thus, both assumptions can *potentially reduce* the benefits provided by a content distribution network.

However, we expect these assumptions to have little impact on our qualitative results. Krishnan, Raz, and Shavitt conducted a detailed trace study and found that not only are the routes relatively stable, but also the traffic flow in different parts of the tree is stable [4]. This implies that for content distribution networks employing *transparent en-route caches*, both assumptions are valid [4]. On the other hand, for non-transparent surrogate servers, clients need to be steered to the closest server which requires an extensive monitoring and request routing subsystems. These systems themselves add to the overall user response time [8], and thus deteriorate performance. In another study, Qiu, Padmanabhan, and Voelker compared the performance of two algorithms for placing servers in content distribution networks [7]. The first algorithm works with a spanning graph of the original network, and places servers using only the information available in the spanning graph, while the second algorithm exploits all available information. The first algorithm also assumes that the clients can only go to the servers located on the shortest path from them to the origin

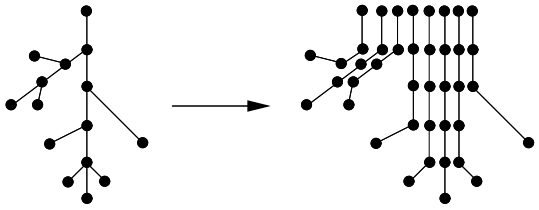

Figure 1: Decomposing of a tree graph into chain graphs

server. Their results show that the median performances of the two algorithms are within a few percent of each other. This result implies that a good estimate of reduction in round-trip time can be obtained just by considering the tree topologies.

2.1 Decomposition of Tree Graphs into Chain Graphs

Consider Figure 1 which shows the decomposition of a tree graph into its constituent chain graphs. A *chain graph* is a tree graph in which the degree of each vertex is equal to two, except for the two end vertexes which have degree one. As the Figure 1 shows that in this decomposition each leaf vertex generates a chain graph which follows the shortest path from that leaf vertex to the root vertex. Thus a tree graph with l leaf vertexes is decomposed into l constituent chain graphs. The constituent chain graphs of a tree graph share vertexes and edges with each other. The client population on a shared vertex is divided equally among the chain graphs sharing the vertex. On the other hand, the delay along an edge of a chain graph is the same as the delay along that edge in the original tree graph. In this subsection, we will show that the acceleration factor Π provided by a content distribution network for a tree graph is intimately related to the values of the acceleration factor on its constituent chain graphs.

Consider a tree graph $\mathcal{N}$ with l leaf nodes. It is easy to see that the average round-trip time on $\mathcal{N}$ (with a single origin server at the root) is given by a weighted sum of the average round-trip time on its constituent chain graphs:

$$\rho_s = \frac{U(1)\rho_s(1) + U(2)\rho_s(2) + \cdots + U(l)\rho_s(l)}{U(1) + U(2) + \cdots + U(l)}$$

where $U(i)$ is the total client population of the i-th chain graph and $\rho_s(i)$ is the average round-trip time on the i-th chain graph (with a single origin server at the root).

Now consider a content distribution network on $\mathcal{N}$ with p surrogate servers. Assume that the p surrogate servers are placed optimally. Recall that we assume that a client contacts the closest surrogate server located on the shortest path between it and the origin server. This assumption implies that the surrogate server for each client is located on the chain graph to which the client belongs. Hence, even in the presence of a content distribution network, the average round-trip time on $\mathcal{N}$ is given by a weighted sum of the average round-trip times on its constituent chain graphs:

$$\rho_c = \frac{U(1)\rho_c(1) + U(2)\rho_c(2) + \cdots + U(l)\rho_c(l)}{U(1) + U(2) + \cdots + U(l)}$$

where $U(i)$ is the total client population of the i-th chain graph and $\rho_c(i)$ is the average round-trip time for clients on the i-th chain graph in presence of the content distribution network.

The following calculation shows that the acceleration factor provided by a content distribution network on $\mathcal{N}$ is a weighted average of the acceleration factor provided by the content distribution network on its

constituent chain graphs.

$$\begin{aligned}\Pi &= 1 - \frac{\rho_c}{\rho_s} \\ &= 1 - \frac{U(1)\rho_c(1) + U(2)\rho_c(2) + \cdots + U(l)\rho_c(l)}{U(1)\rho_s(1) + U(2)\rho_s(2) + \cdots + U(l)\rho_s(l)} \\ &= \frac{U(1)\rho_s(1)\Pi(1) + \cdots + U(l)\rho_s(l)\Pi(l)}{U(1)\rho_s(1) + \cdots + U(l)\rho_s(l)} \\ &= \frac{w(1)\Pi(1) + w(2)\Pi(2) + \cdots + w(l)\Pi(l)}{w(1) + w(2) + \cdots + w(l)} \end{aligned} \tag{2}$$

where $\Pi(i)$ is the acceleration factor provided by the content distribution network for the i-th chain graph and $w(i) = U(i)\rho_s(i)$ is the weight assigned to the i-th chain graph.

The number of leaf nodes, and therefore the number of chain graphs, in a network varies widely from a few tens for corporate intranets to a few thousand for the Internet. This paper is concerned with reasonably large networks with at least a few hundred constituent chain graphs. For such networks (2) has many implications:

1. If a small number of chain graphs account for most of the total weight, then we should treat these chain graphs separately and optimize the performance of the content distribution network on them due to their relatively large weight.

2. In computing the weighted average of acceleration factors for several hundred chain graphs, we can ignore a set of chain graphs without affecting the value of Π significantly provided that the combined weight of the ignored chain graphs is negligible compared to the overall weight.

   For example, consider a network graph whose constituent chain graphs have acceleration factor in the range $[0.1, 0.7]$. Let's assume that 95% of the total weight belongs to chain graphs whose acceleration factor lies in the interval $[0.2, 0.5]$. A simple calculation shows that if we ignore all chain graphs whose acceleration factor lies outside the range $[0.2, 0.5]$, then the computed value of Π will be from 4% to 12% away from its true value.

3. Let $\Pi(i) \in [\Pi_{\min}, \Pi_{\max}]$ for all values of i. Using (2), it follows that $\frac{w(1)\Pi_{\min} + \cdots + w(l)\Pi_{\min}}{w(1)+\cdots+w(l)} \leqslant \Pi$ and $\Pi \leqslant \frac{w(1)\Pi_{\max} + \cdots + w(l)\Pi_{\max}}{w(1)+\cdots+w(l)}$ so that:

   $$\Pi_{\min} \leqslant \Pi \leqslant \Pi_{\max}. \tag{3}$$

   The above inequality says that the acceleration factor for the whole tree graph falls in the same range as the acceleration factor for its constituent chain graphs.

4. Consider a tree graph $\widetilde{\mathcal{N}}$ which is obtained by ignoring a few chain graphs, whose combined weight is an insignificant fraction of the total weight, from the tree graph $\mathcal{N}$. Let the acceleration factor for chain graphs in $\widetilde{\mathcal{N}}$ be in the range $[\widetilde{\Pi}_{\min}, \widetilde{\Pi}_{\min}]$. Combining the points 2 and 3 discussed above, we get

   $$\widetilde{\Pi}_{\min} \lessapprox \Pi \lessapprox \widetilde{\Pi}_{\max}.$$

3 Foundation of Our Analysis

The decomposition of a tree graph into chain graphs as discussed above provides a foundation for our strategy to evaluate acceleration factor for a *generic* tree graph—regardless of network topology, and client population and link-delay profiles, the performance of content distribution on a tree graph is tied to the performance of surrogate servers on chain graphs.

The rest of this paper focuses on analyzing the optimum performance of *one* surrogate server on different types of chain graphs. Our goal is to calculate the acceleration factor in presence of one *optimally placed*

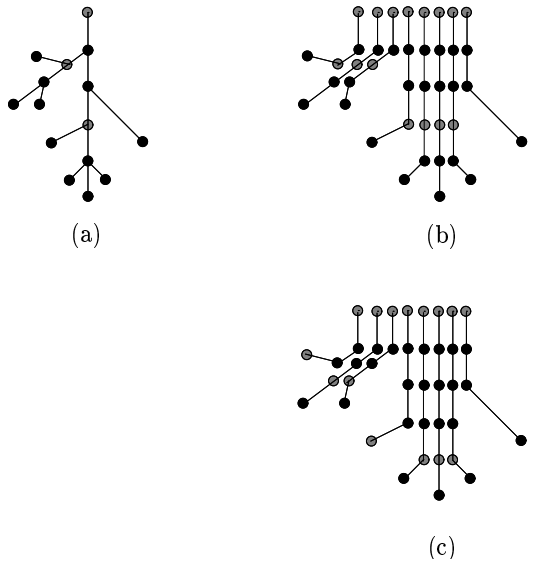

Figure 2: The performance of content distribution networks on chain graphs provides an upper bound on the performance of content distribution networks on tree graphs. Content distribution servers are shown by the circles shaded lightly.

surrogate server for the widest possible varieties of chain graphs. As argued in the next paragraph, this calculation will give us an *optimistic* estimate of the acceleration factor on tree graphs provided that the number of surrogate servers in the content distribution network is small enough so that when placed optimally on the tree graph, most of the constituent chain graphs get no more than one surrogate server.

Consider the optimum placement of p surrogate servers on a tree graph $\mathcal{N}$ with l constituent chain graphs. Let $\widetilde{\mathcal{N}}$ be the tree graph with $\tilde{l}$ constituent chain graphs, $\tilde{l} \leqslant l$, obtained from $\mathcal{N}$ by discarding all branches with more than one surrogate server. Let $\widetilde{\Pi}_1, \ldots, \widetilde{\Pi}_{\tilde{l}}$ be acceleration factors on the chain graphs of $\widetilde{\mathcal{N}}$ when surrogate servers are placed *optimally on the tree graph* $\mathcal{N}$. Let the acceleration factors be $\widehat{\Pi}_1, \ldots, \widehat{\Pi}_{\tilde{l}}$, respectively on these chain graphs, when one surrogate server is placed *optimally on each chain graph*. Since the optimum placement of servers on $\widetilde{\mathcal{N}}$ may not be the optimum placement for individual chain graphs, it follows that $\widehat{\Pi}_i \geqslant \widetilde{\Pi}_i$, for $i = 1, \ldots, \tilde{l}$, and therefore, $\widehat{\Pi}_{\min} = \min(\widehat{\Pi}_1, \ldots, \widehat{\Pi}_{\tilde{l}}) \geqslant \widetilde{\Pi}_{\min} = \min(\widetilde{\Pi}_1, \ldots, \widetilde{\Pi}_{\tilde{l}})$, and $\widehat{\Pi}_{\max} = \max(\widehat{\Pi}_1, \ldots, \widehat{\Pi}_{\tilde{l}}) \geqslant \widetilde{\Pi}_{\max} = \max(\widetilde{\Pi}_1, \ldots, \widetilde{\Pi}_{\tilde{l}})$. Thus the calculation of acceleration factor in the presence of one optimally placed server on chain graphs gives an optimistic estimate of the range $[\widetilde{\Pi}_{\min}, \widetilde{\Pi}_{\max}]$.

The following example should clarify these concept further:

Example 1 Figure 2(a) shows a tree graph with one origin server (at the root vertex) and two surrogate servers. The surrogate servers in this tree graph are placed optimally assuming that the client population is one unit at each vertex and the branch delay is one unit for each branch. It can be verified that the average round-trip time without the surrogate servers is $\frac{48}{14} = 3.4286$. With the two optimally placed surrogate servers, the average round-trip time reduces to $\frac{20}{14} = 1.4286$ with an acceleration factor of 0.583.

Figure 2(b) shows that decomposition of this tree graph in to chain graphs. It results in seven chain graphs with one surrogate server and one origin server, and one chain graphs with just the origin server. It turns out that the optimum placement of servers for the tree graph is not optimum for chain graphs. The optimum placement of surrogate servers for chain graphs is shown in Figure 2(c). With this placement, the weighted average round-trip time is 1.143 with an acceleration factor of 0.667. This round-trip time is slightly lower than 1.4286, the average round-trip time on the tree graph.

Also note that the range of acceleration factor for the optimum placement of servers on the tree graph is

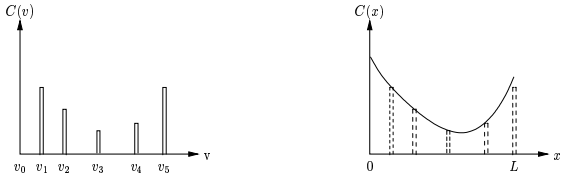

Figure 3: Continuous approximation of a chain graph by a line network

[0.58, 0.72]. When one surrogate server is placed optimally on each chain graph, then the range of acceleration factor turns out to be [0.70, 0.80] which is an optimistic estimate. ∎

4 Acceleration Factor for Chain Graphs

It turns out that the analysis on even a simple chain graph can be too cumbersome. For example, for a chain graph of length n with two congested branches, there are a total $n(n-1)/2$ possible places where these congested branches could be located. Each possible location of congestion gives a different performance. In order to deal with this multitude of possibilities, we will take two approaches. In the first approach, we employ a continuous approximation of chain graphs (see Figure 3) which enables an analytical evaluation of the acceleration factor for well-behaved client population profiles and link-delay profiles. In the second approach, we generate a large number number of chain graphs using certain distributions for client population and link delay, and evaluate the acceleration factor numerically. These two approaches prove to be complementary to each other and illuminate different aspects of the problem.

4.1 Analytic Computation of Acceleration Factor on Continuous Approximation of Chain Graphs

The continuous approximation of a chain graph replaces the chain graph by a line with continuously varying client population and link delay per unit length as shown in Figure 3. Assume that the line goes from $x = 0$ to $x = L$. Also assume that the server is at one end of the line at $x = 0$. Let $U(x)$ be the client population function which gives client population at point x, and let $d(x)$ be the link delay per unit length at point x. Given $U(x)$ and $d(x)$, we can calculate the average round-trip time for a single origin server as follows:

$$\rho_s = \frac{\int_0^L \left(\int_0^y d(x)dx \right) U(y)dy}{\int_0^L U(y)dy}. \tag{4}$$

Note the similarity of (4) and (1). Essentially the above equation can be derived by replacing the summations of terms in (1) by integration of the appropriate quantities, and thereby obtaining a continuous approximation. Without loss of generality, we can assume that the client population function $U(x)$ is normalized so that $\int_0^L U(y)dy = 1$.

Similarly, we can compute the round-trip time in the presence of one surrogate server. Let us assume that there is a surrogate server at $x = \xi$. Also assume that the clients contact a surrogate server *topologically* closest to them as opposed to contacting a server between them and the origin server. This assumption can be expected to reduce the computed average round-trip delay. Therefore it counterbalances the earlier assumption of clients contacting only the servers between them and the origin server. As shown in Figure 4, clients located in R_1 contact the origin server and clients located in R_2 and R_3 contact the surrogate server.

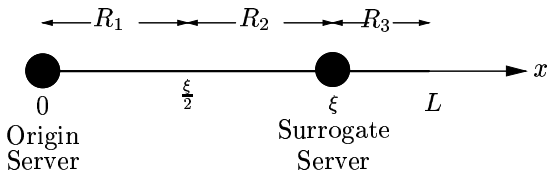

Figure 4: For one surrogate server, clients are divided into three groups

The average round-trip time in this case is given by:

$$\begin{aligned}\rho_c &= \rho_{c_1} + \rho_{c_2} + \rho_{c_3} \\ &= \int_0^{\frac{\xi}{2}} \Big(\int_0^y d(x)\ dx \Big) C(y)\ dy \\ &\quad + \int_{\frac{\xi}{2}}^{\xi} \Big(\int_y^{\xi} d(x)\ dx \Big) C(y)\ dy \\ &\quad + \int_{\xi}^{L} \Big(\int_{\xi}^{y} d(x)\ dx \Big) C(y)\ dy. \end{aligned} \tag{5}$$

In the following, we will consider two client population profiles: uniform density and exponentially decreasing density. Uniform density remains constant on the line, while exponential density decreases rapidly as we move away from the origin server. Along with these client densities, we will consider two link-delay profiles: uniform delay along the line and a step congestion on the line. The following subsections correspond to the four possible combinations of these client population profiles and link-delay profiles.

Note that there are other possibilities of densities that one might consider. We have evaluated several such possibilities and have found that the qualitative results derived below hold in general.

4.1.1 Uniform Client Population with Uniform Link Delay

In this case, we will assume that clients are uniformly distributed over the line network and the link delay is constant. Since the client population is normalized, $\int_0^L U(y)dy = 1$, we have $U(x) = 1/L$. Without loss of generality assume that $d(x) = 1$. Using (4), the average round-trip time for a single origin server is given by:

$$\rho_s = \int_0^L \Big(\int_0^y 1\ dx \Big) \frac{1}{L}\ dy = \frac{L}{2}.$$

Now consider the case when a surrogate server is placed at $x = \xi$. Since the clients contact the server topologically closest to them, the average round-trip time for this case is given by:

$$\begin{aligned}\rho_c &= \rho_{c_1} + \rho_{c_2} + \rho_{c_3} \\ &= \int_0^{\xi/2} \Big(\int_0^y 1\ dx \Big) \frac{1}{L}\ dy \\ &\quad + \int_{\xi/2}^{\xi} \Big(\int_y^{\xi} 1\ dx \Big) \frac{1}{L}\ dy \\ &\quad + \int_{\xi}^{L} \Big(\int_{\xi}^{y} 1\ dx \Big) \frac{1}{L}\ dy \end{aligned}$$

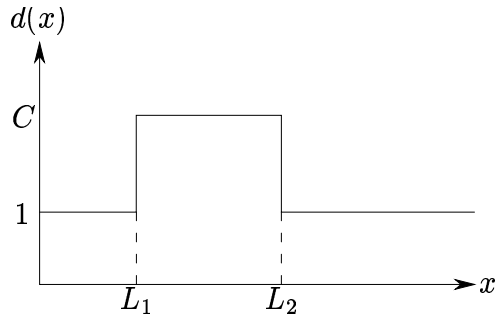

Figure 5: Step congestion of magnitude C extending from L_1 to L_2

which simplifies to

$$\rho_c = \frac{3\xi^2/4 + L^2/2 - L\xi}{L}.$$

It is easy to check that the average round-trip time ρ_c can be minimized by putting the surrogate server at $\xi = 2L/3$, which results in $\rho_c = L/6$. Therefore, for a uniform client population and a uniform link delay, the maximum value of acceleration factor is $1 - L/6/L/2 = 2/3$. This simple case provides many other insights.

- For maximum acceleration factor corresponding to $\xi = 2L/3$, 2/3 of the clients contact the surrogate server and the remaining 1/3 contact the origin server. Thus, the placement optimum for reducing round-trip time does not balance load among servers.
- Since $d(x) = 1$, we can think of $d(x)$ as the bandwidth consumed by the hosted application and Π as a measure of bandwidth that can be saved by utilizing surrogate servers: bandwidth saved in this case is 2/3. Again, the placement for optimum bandwidth saving does not result in balanced load on content distribution servers.

4.1.2 Exponential Client Population with Uniform Link Delay

In this case, we assume that the line network is infinitely long and the client population is given by $U(x) = \lambda \exp(-\lambda x)$. Assume that the link delay is uniform given by $d(x) = 1$. The average round-trip time for a single origin server located at $x = 0$ is given by:

$$\rho_s = \int_0^\infty y\lambda \exp(-\lambda y)\ dy = \frac{1}{\lambda}.$$

In the presence of a surrogate server at $x = \xi$, we can compute the average round-trip time by using (5). After some tedious algebra we get,

$$\rho_c = \Big(1 + 2\exp(-\lambda\xi) - 2\exp(-\lambda\xi/2)\Big)/\lambda.$$

Thus the acceleration factor in this case is $2\exp(-\lambda\xi/2) - 2\exp(-\lambda\xi)$. The maximum value of acceleration is 1/2 for $\xi = 2\text{kn}2/\lambda$.

4.1.3 Uniform Client Population with a Step Congestion

In this subsection, we will compute the possible reduction in round-trip time in the presence of a step congestion. Assume that the client population $U(x)$ is uniform and is given by $U(x) = 1/L$. Also assume

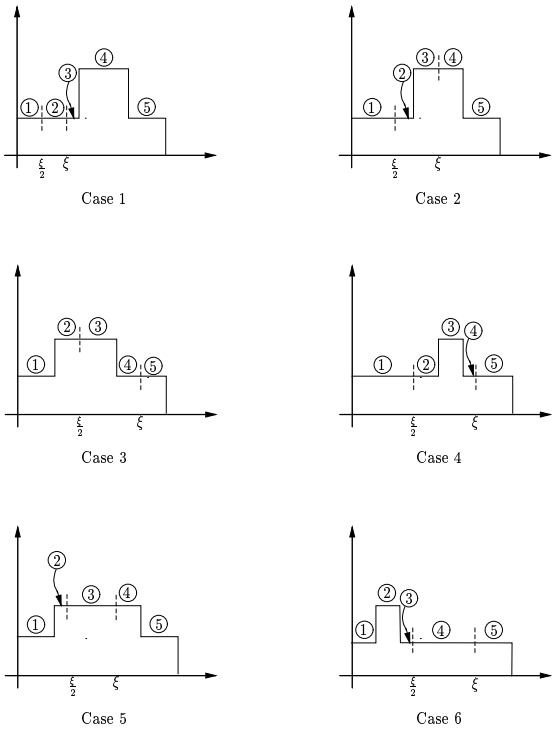

Figure 6: Six possible relative positions of the surrogate server and the step congestion

that the link delay $d(x)$ has a step congestion of magnitude C between $x = L_1$ and L_2 (see Figure 5). It turns out that without loss of generality, we can take $L = 1$ and assume that $0 < L_1 < L_2 < 1$.

Simple calculations show that for a single origin server at $x = 0$, ρ_s is given by:

$$\rho_s = \frac{1}{2}(1 - L_2)^2 - L_1^2 + L_1 + \frac{C}{2}\Big(L_1^2 - L_2^2 + 2(L_2 - L_1)\Big).$$

Next, consider a surrogate server at $x = \xi$ in addition to the origin server at $x = 0$. Due to the presence of the step congestion, the average round-trip time depends on the relative positions of the congestion and the surrogate server, that is, on the relative values of L_1, L_2, and ξ. Different relative values of L_1, L_2, and ξ give rise to six cases which are shown in Figure 6. In each case, the client population can be divided into five regions and the formulas for the average round-trip time ρ_c can be computed. For the sake of brevity, we will omit the details of these computations.

We used these formulas to compute the optimum placement of the surrogate server for different values of L_1 and L_2, and to evaluate the corresponding reduction in the average round-trip time. For $C = 2$ and 10, and for different congestion lengths, $L_2 - L_1$, the possible acceleration factors Π are given in Table 1. Each row of this table corresponds to a given length of step congestion. The range indicated for Π is obtained by moving the step congestion to different places on the line.

4.1.4 Exponential Client Population with a Step Congestion

In this subsection, we will examine the possible reduction in the average round-trip time for exponentially decaying client population given by $U(x) = \lambda e^{-\lambda x}$ in the presence of a step congestion. For a single server

Table 1: The range of optimistic Π

$C = 2$	
$L_2 - L_1$	Π
0.05	0.65–0.69
0.10	0.63–0.59
0.25	0.59–0.60
0.50	0.58–0.59
0.75	0.61–0.66

$C = 10$	
$L_2 - L_1$	Π
0.05	0.53–0.47
0.10	0.46–0.54
0.25	0.37–0.63
0.50	0.37–0.63
0.75	0.54–0.46

Table 2: The range of optimistic Π

$C = 2$	
$L_2 - L_1$	Π
0.10	0.49–0.51
0.50	0.47–0.53
1.00	0.46–0.59
2.00	0.43–0.54
8.00	0.41–0.50

$C = 10$	
$L_2 - L_1$	Π
0.10	0.42–0.55
0.50	0.23–0.58
1.00	0.29–0.62
2.00	0.43–0.68
8.00	0.50–0.78

located at $x = 0$, the average round-trip time is given by:

$$\rho_s = \frac{1}{\lambda}\Big(1 + (C-1)e^{-\lambda L_1} - (C-1)e^{-\lambda L_2}\Big).$$

The details are algebraically involved and they will be skipped here.

Next consider the case when there is an additional surrogate server at $x = \xi$. It turns out that the average round-trip time ρ_c in this case also depends on the relative position of the congestion and the placement of surrogate server, that is, on the relative values of L_1, L_2, and ξ (See Figure 6). The relative values of these variables gives rise to six cases for which the formulas for ρ_c can be evaluated. Once again, for the sake of brevity, we will omit the details of these computations.

Table 2 shows the possible acceleration factor Π for $C = 2$ and 10, and for different congestion lengths $L_2 - L_1$. Clearly, the acceleration Π lies roughly between 0.25 and 0.70.

4.2 Numerical Computation of Acceleration Factor on Randomly Generated Chain Graphs

In this subsection, we will examine the acceleration provided by an optimally placed surrogate server for randomly generated client population profiles and link-delay profiles. In this subsection, to determine the acceleration, we use numerical calculations as opposed to the analytical expressions used in the earlier subsections.

Let us assume that $U(v)$, the client population on vertexes of the chain graph, is distributed according to the density function $p_U(u)$, and let us assume that $D(e)$, delay along the edges of the chain graph, is distributed according to the density function $p_D(d)$. We assume that the client population on different vertexes and delays along different edges of a chain graph are independent of each other. In the following, we will describe results for two sets of client population profiles and networking link-delay profiles. We believe these profiles capture different extreme conditions that are likely to exist in a network.

In order to compute the range of acceleration factor for a given client population profile function $p_U(u)$ and a given link-delay profile function $p_D(d)$, we generate 100,000 random chain graphs according to the given densities. For each generated graph, we compute the optimum placement of surrogate server and the

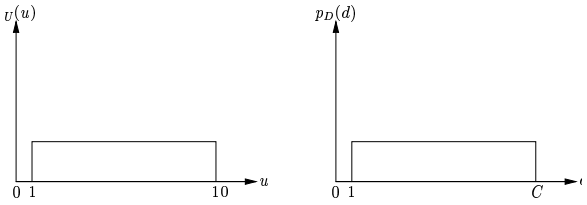

Figure 7: Client population and link-delay distributions for Case 1

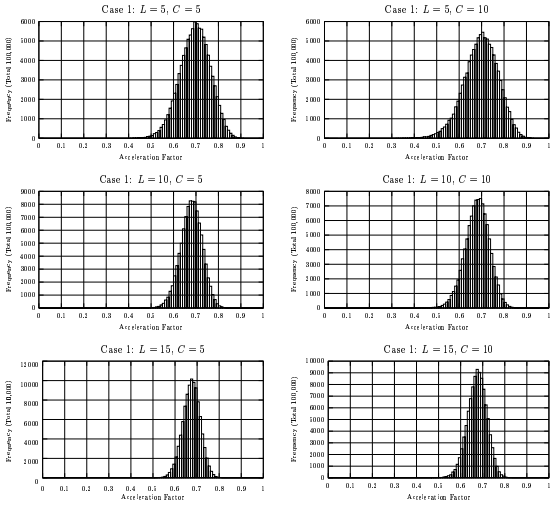

Figure 8: Distribution of Acceleration Factor for Different Lengths L and Peak Congestion C in Case 1

reduction in round-trip time provided by it, and plot the histogram of acceleration factor. The results are as follows:

Case 1. In this case, the client population on a vertex is chosen randomly from the uniform distribution $[1, 10]$ (see Figure 7), and the link delay $D(e)$ along an edge is chosen randomly according to a uniform distribution $[1, C]$. The resultant histogram of acceleration resulting from placing one surrogate server is shown in Figure 8 for different values of chain length L and peak congestion C. Clearly, as the chain length increases to 15, the acceleration factors cluster around 0.65. For shorter chains, the distribution of Π is more spread out—it goes from 0.9 to 0.4 for chain length 5. If we assume that in a typical network, all constituent chain graphs will not have either very high or very low value of acceleration, then it is reasonable to assume that under these distributions an optimistic value of the average acceleration factor is 0.65.

Case 2. In this case, both client population and link delay are either relatively small (with 85% probability) or relatively large (with 15% probability). The details of client population and link-delay distributions are shown in Figure 9. In this case for shorter chain lengths, the distribution of acceleration factor clearly is spread over a wider range than the range obtained in Case 1. However, the acceleration values are still centered around 0.65.

We have also considered other distributions for the generation of client population and link delays. In all cases, we found that the maximum acceleration factor provided by one optimally place surrogate server lies around 0.65.

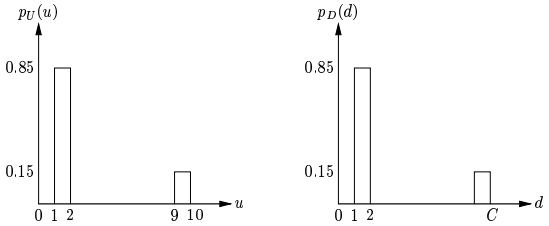

Figure 9: Client population and link-delay distributions for Case 2

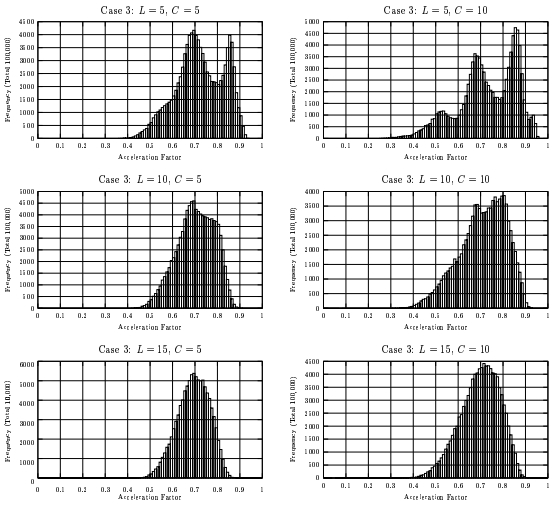

Figure 10: Distribution of Acceleration Factor for Different Lengths L and Peak Congestion C in Case 2

5 Discussion

The calculations herein show that for a wide range of client population and link-delay profiles, the acceleration factor Π lies between 0.25 and 0.70. The bounds derived in Section 3 imply that the acceleration factor provided by a content distribution network on a tree graph will also fall in the same range as long as the majority of its constituent chain graphs inherit no more than one surrogate server from the content distribution network.

At present, the Internet consists of a few thousand client clusters with the majority of clusters having almost identical load characteristics [9]. Assuming that the total number of surrogate servers is less than 50, the probability of a surrogate server being co-located with a client cluster is less than $50/2000 = 0.025$. This in turn implies that with probability more than 0.90, a chain graph of length 5 will have one or less surrogate server. Assuming that the client population profile and link-delay profile in this work are representative of that of the Internet, calculations in this paper show that an optimistic estimate of the acceleration factor is 0.65. A corresponding study of the response times of typical web sites shows that at present, server and client processing time constitutes a significant fraction of the overall response time. This fact combined with the fact that acceleration factor is likely to be less than or equal to 0.65 indicates that for short web transfers, merely bringing content closer to the clients may not provide noticeable improvement in the response time.

In addition to the acceleration of content delivery, a distribution network may also provide two additional functions: savings in bandwidth and load balancing among servers. Our study shows that the optimum placement of servers for these three goals is likely to be different. As a consequence, a content distribution network which aspires to provide all three functions (with a limited number of servers) may not optimize any of the three performance metrics.

We hope that our work here will stimulate an interest in evaluating whether there is a point beyond which increasing the number of surrogate servers in a content distribution network results in diminishing returns, and if so, determining the the optimum size of a content distribution network. Furthermore, we hope that this work will result in a more careful analysis of application offload architectures currently being designed in the industry.

6 Acknowledgment

The authors would like to thank Anees Shaikh, Kang-Won Lee, Arup Acharya, and Khalil Amiri for many useful discussions. A substantial amount of work in this paper was motivated by these discussions.

References

[1] D. Agrawal, J. Giles, and D. Verma, "On the performance of content distribution networks," in *SPECTS 2001*, July 2001, pp. 132–139.

[2] S. B. Calo, D. Verma, J. Giles, and D. Agrawal, "On the effectiveness of content distribution networks," unpublished manuscript, submitted to IEEE INFOCOM 2002.

[3] K.L. Johnson, J. F. Carr, M. S. Day, and M. Frans Kaashoek, "The measured performance of content distribution networks," in *5-th International Web Caching Workshop and Content Dilivery Workshop*, May 2000.

[4] P. Krishnan, D. Raz, and Y. Shavitt, "The cache location problem," *IEEE Transaction on Networking*, pp. 568 – 582, Oct. 2000.

[5] S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "On the placement of internet instrumentation," in *IEEE INFOCOM 2000*, Mar. 2000, pp. 295–304.

[6] B. Li, M. Golin, G. Italiano, and X. Deng, "On the placement of web proxies in the internet," in *IEEE INFOCOM 1999*, Mar. 1999.

[7] L. Qiu, V. Padmanabhan, and G. Voelker, "On the placement of web server replicas," in *IEEE INFOCOM 2001*, Apr. 2001.

[8] Anees Shaikh, Renu Tewari, and Mukesh Agrawal, "On the effectiveness of dns-based server selection," in *IEEE INFOCOM 2001*, Apr. 2001.

[9] Anees Shaikh, "private communication," Apr. 2001.