The Multicast Bandwidth Advantage in Serving a Web Site

Yossi Azar[1], Meir Feder[2], Eyal Lubetzky[3], Doron Rajwan[4], and Nadav Shulman[5]

[1] Dept. of Computer Science, Tel-Aviv University, Tel-Aviv, 69978, Israel. azar@tau.ac.il
[2] Bandwiz, Israel and Department of Electrical Engineering - Systems, Tel Aviv University, Tel-Aviv, 69978, Israel. Meir@bandwiz.com
[3] Bandwiz, Israel. EyalL@bandwiz.com
[4] Bandwiz, Israel. Doron@bandwiz.com
[5] Bandwiz, Israel. Nadav@bandwiz.com

Abstract. Delivering popular web pages to the clients results in high bandwidth and high load on the web servers. A method to overcome this problem is to send these pages, requested by many users, via multicast. In this paper, we provide an analytic criterion to determine which pages to multicast, and analyze the overall saving factor as compared with a unicast delivery. The analysis is based on the well known observation that page popularity follows a Zipf-like distribution. Interestingly, we can obtain closed-form analytical expressions for the saving factor, that show the multicast advantage as a function of the site hit-rate, the allowed latency and the Zipf parameter.

1 Introduction

One of the largest problems in the web is to deliver the content efficiently from the site to the user. High load on the server and on the network leads to long delays or more extremely denial of services. Increasing the capacity for delivering the content results in a high cost of extra servers and extra bandwidth. Moreover, the capacity is planed to some value, though larger than the average load, but almost always cannot accommodate the peak load. This is specially correct for popular pages were the access pattern may be unpredictable and very unstable (e.g. the famous Starr report case).

There are several methods to try to overcome the problem. One is to use caches [13, 7, 1]. However, caches are not effective for frequently changing content or for long files (e.g video, audio). A different possibility that we consider in this paper is to use multicast [4, 8, 6], i.e., to deliver the content simultaneously to many (all) users via multicast dynamic tree. Obviously, one may also combine both caching and multicasting to further improve the solution.

At first, it may seem that multicast could be effective only if many users requests exactly the same content at exactly the same time, which can occur mainly in real time events. However, it is well known (see, e.g., [8]) that one can cyclicly transmit by multicast a page until all users requested the page in the multicast tree receive it. Note that each user needs to receive one cycle from the time that he joins the tree (which does not need to be a beginning of a new cycle) assuming that there are no faults. A more efficient methods that overcomes possible packet losses can be achieved by using erasure codes, e.g., [11, 3].

The multicast advantage is manifested by combining together overlap requests to a single transmission. This way the server load and bandwidth decrease dramatically since all overlapped users appear almost as a single user. Hence, the most attractive pages (files) to multicast are pages that are popular, i.e., have many hits per second, and pages that are large. Fortunately, the access pattern to pages of a site are far from being uniform. Any non-uniformity on the distribution of the access pattern to pages enhances the advantage of using multicast since it results in more popular, hence higher concurrency, pages. It has been observed [2, 10, 9] that indeed the access pattern for pages in a site is highly non-uniform and obeys a Zipf-like distribution with α parameter that is in the range of $1.4 - 1.6$. With this distribution, a fixed number of pages account for almost all requests for pages (say 95%). As in many other events, Zipf distribution occurs naturally, and so we assume that this is the request pattern in order to obtain quantitive expressions for the multicast advantage. We will present the results in terms of the Zipf parameter α and note that even for the pure Zipf distribution, i.e. for parameter $\alpha = 1$, and furthermore even for Zipf-like distribution with $\alpha < 1$, a small number of pages (maybe not as small as for $\alpha > 1$) still account for most of the requests. Since a Zipf-like distribution has a heavy tail, assuming such a distribution on the access pattern is one of the weakest possible assumptions in terms of the advantage of multicast.

It is worthwhile to mention that the popular pages may change over time. An appropriate system that keeps track of the access pattern can easily maintain the list of the hot pages. Hence, such a system can decide which pages to multicast at each point in time according to the estimated parameters of the access rate and the size of the pages.

We next discuss the results of this paper. We start, in section 2, by an analysis of a site in which all the information regarding the access pattern and file distribution is given. The analysis is based on a criterion we derive, that determines which pages to multicast. This criterion assumes that the page access rate is given, or estimated, and it also depends on the allowable delay to receive the page, which in turn, determines the bandwidth in which the page is multicasted. The major result of our paper appears in section 3, and contains a set of analytical expression for the gain in bandwidth (and server load) in serving a *typical site* by selective multicast (i.e., multicast of hot pages) as compared with the standard unicast serving. For the typical site we assume that the access pattern follows a Zipf-like distribution with some parameter α. The overall saving bandwiz factor achieved depends on the access rate to the site and the latency that we allow for pages. Section 4 extends the analysis to a site with various typical file groups. The paper is summarized in section 5.

2 Analysis for a Given Site

We make the following notations

- n the number of pages in the site.
- p_i probability of requesting page i for $1 \leq i \leq n$ given that a page was requested from the site.
- S_i is the size of page i, in bits, for $1 \leq i \leq n$.
- λ the average access rate in hits per unit time, to the site. We note that $\lambda = N\lambda_0$ where N is the size of the population accessing the site and λ_0 is the average access rate of a person from the population to the site.

As a step toward an analysis for a typical site we make an analysis for a given site with the probably unrealistic assumption that all the above parameters (n, p_i, S_i, λ) are known. In this section we first compute the minimal required bandwidth to serve this site by unicast. We then consider serving the site by selective multicast, where we first determine which pages worth multicasting and then compute the resulting bandwidth. By that we estimate the gain in serving this site by multicast. Note that we assume that the site is planned to have the ability of serving all requests and not to drop/block some of them.

2.1 Serving by unicast

Using the above notation the amount of bits per unit time generated on the average in serving the page i is $\lambda p_i S_i$. Consider now

$$B_u = \sum_{i=1}^{n} \lambda p_i S_i = \lambda \sum_{i=1}^{n} p_i S_i \ .$$

This formula is the information theoretic lower bound on the required bandwidth for serving all the pages by unicast, since the total average number of bits requested per unit time must by equal (on the average) to the total number of bits transmitted. Note that the lower bound is independent of the transmission rate of the pages. Moreover, the above formula stands for the minimum possible bandwidth in the ideal case where we can store the requests in a queue and output continuously exactly the same number of bits per time without any bound on the latency encountered for delivering the files. The actual bandwidth required by any practical system to support all requests (in particular, with bounded latency) needs to be higher than this. Nevertheless, we essentially demonstrate the multicast bandwidth advantage by showing that multicast requires less (sometimes much less) bandwidth than this information theoretic bound.

2.2 Serving by selective multicast

In serving a file i by multicast, a carousel transmission (or better, a coded stream using, e.g., Bandwiz block-to-stream code [11]) of the file is transmitted at some particular bandwidth w_i and all requests for the file are handled by receiving from this multicast transmission. The bandwidth advantage in serving a file this way comes from the fact that the file is served at the fixed bandwidth w_i and this bandwidth allocation is sufficient no matter how many requests the file has during its transmission. In unicast, on the other hand, each request requires an additional bandwidth allocation.

One may conclude that multicast can lead to an unbounded saving compared with unicast, simply by allocating a small bandwidth w_i to serve the file i. But there is a price for that. The latency in receiving the file, whose size is S_i will become large. A reasonable multicast bandwidth allocation is such that the desired latency L_i is guaranteed. Note that the information theoretic lower bound computed for unicast was independent of the latency we allow to deliver any file (although the realistic bandwidth, higher than that, does depend on it as discussed above). Thus, as the allowed latency is larger, the multicast advantage is larger.

In view of this discussion, we assume that the bounds on the latencies for the various files are imposed on the system. We use the following definitions:

- Let L_i be the latency we allow for delivering page i using multicast.
- Thus, $w_i = S_i/L_i$ is the rate that we chose to transmit page i.

We note that the value of w_i and L_i are functions of the typical capability of the receivers and network conditions. For example, w_i should not be larger than the typical modem rate if typical receivers access the site through a modem. This implies that L_i cannot be small for large files. Also for small files it does not pay to have small L_i since creating the connection from the receiver to the site would dominate the delay. Hence we conclude that L_i is never very small and may be required to be reasonably large. As will be seen, the larger the L_I, the better is the multicast advantage.

Out of the bandwidth allocated to unicast, the portion of the minimal bandwidth required to transmit the file i is $\lambda p_i S_i$ (which is the amount of bits per unit time requested of this file). Thus, in using multicast, we reduce the bandwidth to all the pages in which

$$\lambda p_i S_i > w_i$$

and in this case we replace $\lambda p_i S_i$ by the bandwidth by w_i. The above formula, which provides the criterion for transmitting the file by multicast, is equivalent to

$$\lambda p_i L_i > 1 \ .$$

Hence we conclude that the total bandwidth required by the selective multicast is

$$B_m = \sum_{i|\lambda p_i L_i > 1} w_i + \sum_{i|\lambda p_i L_i \leq 1} \lambda p_i S_i \ .$$

3 Analysis for a Typical Site

We consider a site, where the various pages can be partitioned into typical groups. In each group the pages are of similar characteristics, i.e. approximately the same size and same required latency for delivery to the user. For example, one group can be text HTML files, another group can be pages with images and yet another group can be audio, or video files. We first consider one such group of pages. It is well known and has been consistently observed that the access pattern to the pages in a group is not uniform. In fact, the advantage of multicast improves as the distribution becomes less uniform since one needs to multicast less pages to deliver the same fraction of the traffic. We make one of the weakest possible assumptions on that distribution, i.e., a family of heavy tail distributions on the access pattern. If the distribution is more skewed then the saving by using multicast increases.

Assumption. Among a group of pages with the same latency the popularity of pages is distributed according to Zipf-like distribution with some parameter $\alpha > 0$. Specifically, the probability of the i'th most popular page is proportional to $1/i^\alpha$ or equal to $\frac{1}{C(\alpha)i^\alpha}$ where $C(\alpha) = \sum_{i=1}^{n} \frac{1}{i^\alpha}$.

The above assumption is crucial for our analysis. The typical parameter α which is usually observed for a typical site is in the range $1.4 - 1.6$. In the sequel we will use the following approximation $\sum_{i=a+1}^{b} \frac{1}{i^\alpha} \approx \int_a^b \frac{1}{x^\alpha} dx$ or $\sum_{i=a}^{b} \frac{1}{i^\alpha} \approx \frac{1}{a^\alpha} + \int_a^b \frac{1}{x^\alpha} dx$. In particular $\sum_{i=1}^{n} \frac{1}{i^\alpha} \approx 1 + \int_1^n \frac{1}{x^\alpha} dx$.

Now, we are ready to continue the analysis. First we consider unicast. We can approximate the expression

$$B_u = \lambda \sum_{i=1}^{n} p_i S_i$$

by

$$B_u = \lambda E(S)$$

where $E(S)$ is the expected size of a random page in the group.

Using the Zipf-like distribution we can evaluate the total bandwidth required by multicast. Recall that it is worthwhile to multicast a page if $\lambda p_i L > 1$ (L is fixed for all pages in the group) and we should multicast the most popular pages regardless of their size. Let k be the number of such pages that are worth to multicast. Then k is the largest integer that satisfies $\lambda p_k L > 1$ or

$$\frac{1}{C(\alpha)k^\alpha} = p_k \geq \frac{1}{\lambda L} .$$

Following the above formula there are three different cases that we need to analyze according the to value of the smallest k that satisfies the above formula:

- No need to multicast any page. This is the case where the access rate is small and the required latency is so short that it is not worthwhile to multicast even the most popular page (smallest $k \leq 1$). That corresponds to $\lambda L \leq C(\alpha)$.

- Multicast all pages. Here the access rates are high or the number of pages is relatively small such that it is worthwhile to multicast all pages ($k \geq n$). Here all pages are popular which corresponds to $\lambda L \geq C(\alpha) n^{\alpha}$.
- Multicast popular pages. This is the typical case where $1 < k < n$ and we multicast only the popular pages according to our metric. This corresponds to $C(\alpha) < \lambda L < C(\alpha) n^{\alpha}$.

Clearly, in the first case multicast saves nothing. Later we discuss the saving when we multicast all pages. We begin, then, with the interesting case where $1 < k < n$, i.e., the case of multicasting only the popular pages.

3.1 Multicasting the popular pages

In this case we get $k = \left\lfloor \left(\frac{\lambda L}{C(\alpha)}\right)^{1/\alpha} \right\rfloor$ where $1 \leq k \leq n$.

If we plug it into the formula of the total bandwidth of the multicast (i.e. multicast the first k pages and unicast the rest) we get

$$B_m = \sum_{i=1}^{k} S_i / L + \sum_{i=k+1}^{n} \lambda p_i S_i \ .$$

Since the pages in a group have similar characteristics in terms of size and required latency we can approximate the above by the following

$$\begin{aligned} B_m &\approx \frac{E(S)}{L} \left(\frac{\lambda L}{C}\right)^{1/\alpha} + \frac{E(S)\lambda}{C} \int_{(\frac{\lambda L}{C})^{1/\alpha}}^{n} \frac{1}{x^{\alpha}} dx \\ &= \frac{E(S)\lambda}{C} \left(\left(\frac{\lambda L}{C}\right)^{1/\alpha - 1} + \int_{(\frac{\lambda L}{C})^{1/\alpha}}^{n} \frac{1}{x^{\alpha}} dx \right) \end{aligned}$$

where we drop the integer value and we set

$$C = C(\alpha) = 1 + \int_{1}^{n} \frac{1}{x^{\alpha}} dx \ .$$

Next we separate between the case $\alpha = 1$ and the case $\alpha \neq 1$. For the case $\alpha \neq 1$ we also consider asymptotic behavior.

the case $\alpha = 1$. Clearly

$$C = 1 + \int_{1}^{n} \frac{dx}{x} = 1 + \ln n - \ln 1 = \ln en$$

and

$$\int_{\frac{\lambda L}{C}}^{n} \frac{dx}{x} = \ln n - \ln \frac{\lambda L}{C} = \ln \frac{nC}{\lambda L} = \ln \frac{n \ln en}{\lambda L} \ .$$

Hence for the range of the typical case i.e., $\ln en < \lambda L < n \ln en$, we have

$$B_m \approx \frac{E(S)\lambda}{\ln en}\left(1 + \ln\frac{n\ln en}{\lambda L}\right) = E(S)\lambda\left(\frac{\ln n + 1 + \ln\frac{\ln en}{\lambda L}}{\ln en}\right)$$
$$= E(S)\lambda\left(\frac{\ln en - \ln\frac{\lambda L}{\ln en}}{\ln en}\right) = E(S)\lambda\left(1 - \frac{\ln\frac{\lambda L}{\ln en}}{\ln en}\right) .$$

If we compare it to standard unicast, the saving factor is

$$R = \frac{1}{1 - \frac{\ln\frac{\lambda L}{\ln en}}{\ln en}} .$$

Examples of the savings can be seen in Table 1. Here λ is given in hits per second for the site (i.e. total rate for all pages), L is given is seconds (4 seconds for html page, 20 seconds for page with pictures and 300 seconds for audio or video clip) and n is the number of pages of the site. Plots of R appear in Figure 2 as a function of λ (and also for various α's, see also below).

λ	L	n	**saving,** $\alpha = 1$
200	20	10^4	2.41
200	4	10^3	2.40
20	300	10^3	6.19

Fig. 1. Examples of the saving factor for $\alpha = 1$

the case $\alpha \neq 1$. In this case

$$C = 1 + \int_1^n \frac{dx}{x^\alpha} = 1 + \frac{n^{1-\alpha} - 1}{1-\alpha} = \frac{n^{1-\alpha} - \alpha}{1-\alpha}$$

and

$$\int_{(\frac{\lambda L}{C})^{1/\alpha}}^{n} \frac{dx}{x^\alpha} = \frac{n^{1-\alpha} - (\frac{\lambda L}{C})^{\frac{1-\alpha}{\alpha}}}{1-\alpha} .$$

Hence for the range

$$\frac{n^{1-\alpha} - \alpha}{1-\alpha} < \lambda L < \frac{n^\alpha(n^{1-\alpha} - \alpha)}{1-\alpha}$$

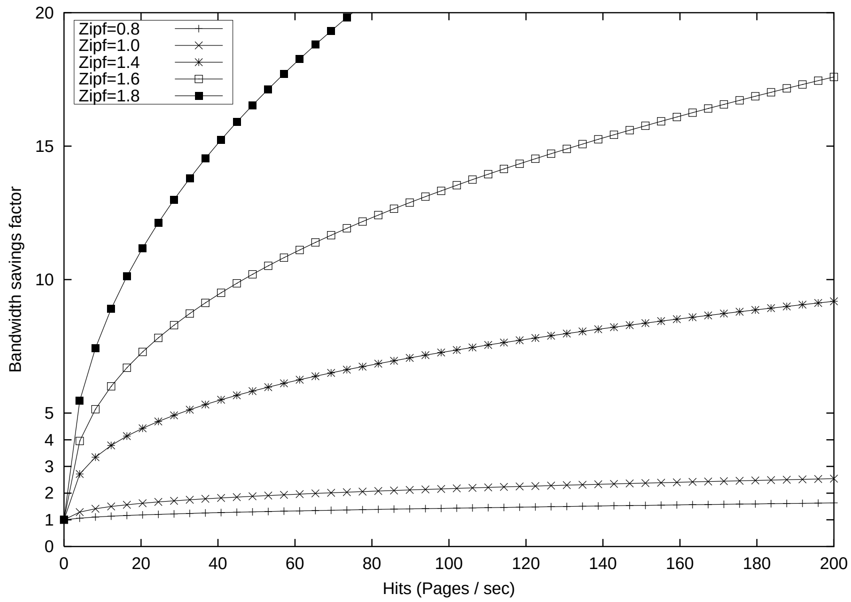

Fig. 2. The saving factor (relative to unicast) of the bandwidth (load) of a server for multicast with Zipf-like distribution for various values of the parameter α as a function of the number of hits per second. The number of pages is $10,000$ and the latency is 25 seconds.

we have

$$
\begin{aligned}
B_m &\approx \frac{E(S)\lambda}{C}\left(\left(\frac{\lambda L}{C}\right)^{1/\alpha-1} + \frac{n^{1-\alpha} - (\frac{\lambda L}{C})^{1/\alpha-1}}{1-\alpha}\right) \\
&= \frac{E(S)\lambda}{C}\left(\left(\frac{\lambda L}{C}\right)^{1/\alpha-1}(1-\frac{1}{1-\alpha}) + \frac{n^{1-\alpha}}{1-\alpha}\right) \\
&= \frac{E(S)\lambda}{(1-\alpha)C}\left(-\alpha\left(\frac{\lambda L}{C}\right)^{1/\alpha-1} + n^{1-\alpha}\right) \\
&= \frac{E(S)\lambda}{n^{1-\alpha}-\alpha}\left(-\alpha\left(\frac{\lambda L(1-\alpha)}{n^{1-\alpha}-\alpha}\right)^{1/\alpha-1} + n^{1-\alpha}\right) \ .
\end{aligned}
$$

We conclude that the saving factor compared with unicast is

$$
R = \frac{n^{1-\alpha}-\alpha}{n^{1-\alpha} - \alpha\left(\frac{\lambda L(1-\alpha)}{n^{1-\alpha}-\alpha}\right)^{1/\alpha-1}} \ .
$$

Again, plots of R as a function of λ and various α's appear in Figure 2.

asymptotic expression - $\alpha > 1$. It is interesting to consider the asymptotic behavior of the saving factor for a site, as the number of pages grows. It is not hard to show that the saving function is monotone non increasing with the number of pages. Moreover, for the case $\alpha > 1$, it turns out that the saving factor approaches to a limit which is bounded away from 1. Hence, to bound the saving factor for any number of pages we can assume that the number of pages n approaching infinity. The saving factor R in the asymptotic case, which as will be seen has a simpler expression (independent of n), is a lower bound on the saving factor for any n (i.e. we save at least that much). This is very useful since the number of pages in a site is usually large and continuously growing.

For evaluating the asymptotic behavior we approximate the expression for R by replacing $n^{1-\alpha}$ with zero. Then for the range $\frac{\alpha}{\alpha-1} < \lambda L$ we have

$$\begin{aligned} B_m &\approx \frac{E(S)\lambda}{-\alpha}\left(-\alpha\left(\frac{\lambda L(1-\alpha)}{-\alpha}\right)^{1/\alpha-1}\right) \\ &= E(S)\lambda\left(\lambda L(1-1/\alpha)\right)^{1/\alpha-1} \ . \end{aligned}$$

Hence the saving factor relative to unicast is

$$R = \left(\lambda L(1-1/\alpha)\right)^{1-1/\alpha}$$

and it is independent of n.

The saving factor of the total bandwidth for a site (including both unicast pages and multicast pages) yields by multicasting the relevant pages can be found in Figure 3 for $\alpha = 1.4$, $\alpha = 1.6$ and $\alpha = 1.8$ for few examples.

λ	L	**saving,** $\alpha = 1.4$	**saving,** $\alpha = 1.6$	**saving,** $\alpha = 1.8$
200	20	7.48	15.25	27.82
200	4	4.72	8.49	13.60
20	300	8.39	18.07	33.31

Fig. 3. Examples of the saving factor for $\alpha = 1.4$, $\alpha = 1.6$ and $\alpha = 1.8$

asymptotic expression - $\alpha < 1$. Now assume that $\alpha < 1$. For the asymptotic behavior we can approximate the expression by assuming that $n^{1-\alpha}$ is relatively large compare to α (i.e n is relatively large). Then for the approximate range

$$\frac{n^{1-\alpha}}{1-\alpha} < \lambda L < \frac{n}{1-\alpha}$$

we have

$$\begin{aligned} B_m &\approx \frac{E(S)\lambda}{n^{1-\alpha}} \left(-\alpha \left(\frac{\lambda L(1-\alpha)}{n^{1-\alpha}} \right)^{1/\alpha-1} + n^{1-\alpha} \right) \\ &= E(S)\lambda \left(1 - \frac{\alpha}{n^{1-\alpha}} \left(\frac{\lambda L(1-\alpha)}{n^{1-\alpha}} \right)^{1/\alpha-1} \right) \\ &= E(S)\lambda \left(1 - \alpha \left(\lambda L(1-\alpha)/n \right)^{1/\alpha-1} \right) . \end{aligned}$$

Hence the saving factor is

$$R = \frac{1}{1 - \alpha \left(\lambda L(1-\alpha)/n \right)^{1/\alpha-1}}$$

relative to unicast. This expression depends on n (as n goes to infinity, the saving factor goes to 1, i.e., no saving) but it is a simpler expression than above.

3.2 Multicast all pages

Here we multicast all pages i.e., $k = n$ which corresponds to the range $\lambda L \geq C(\alpha)n^{\alpha}$. We have $B_m = \sum_{i=1}^{n} S_i/L = E(S)n/L$. If we compare it to unicast, we get that the saving factor is

$$R = \frac{\lambda L}{n} .$$

It is worthwhile to note that the above saving factor holds for all values of α. The range for achieving this saving factor is $\lambda L \geq n \ln en$ for $\alpha = 1$ and $\lambda L \geq \frac{n^{\alpha}(n^{1-\alpha}-\alpha)}{1-\alpha}$ for $\alpha \neq 1$. The range for $\alpha \neq 1$ can be approximated by the range $\lambda L \geq \frac{\alpha n^{\alpha}}{1-\alpha}$ for $\alpha > 1$ and $\lambda L \geq \frac{n}{1-\alpha}$ for $\alpha < 1$.

It is also worthwhile to mention that the case $\alpha = 0$ (i.e. uniform distribution) always falls in the extreme case or the low traffic. That is if $\lambda L > n$ it is worth while to multicast all pages and otherwise it is not worthwhile to multicast any page.

3.3 Properties of the saving function

We list the following useful observations:

- The saving function is continuous monotone non-decreasing as a function of λL for any given α and n in the admissible range. This can be easily proved by considering the saving function directly.
- The saving function is continuous monotone non-increasing as a function of n for any given α and λL in the admissible range. This can be easily proved for $\alpha = 1$. For $\alpha \neq 1$ this can be proved by showing that the saving function is monotone in $n^{\alpha} - \alpha$ which is monotone in n.
- The saving function seems to be continuous monotone non-decreasing as a function of α (also at $\alpha = 1$) for any given n and λL in the admissible range.

4 A Site with Various Groups

In this section we assume that not all pages have a similar size and latency. We partition the files into r groups where in each group the files are of approximately the same size and latency. For group i we denote by $f_u^j(E_j(S), \lambda_j))$ the average bandwidth required for group j using the standard unicast serving and by $f_m^j(E_j(S), \lambda_j, L_j)$ the average bandwidth required for group j using multicast. Recall that we do not limit the number of pages that we multicast and hence, the decision if to multicast a page does not conflict with the decisions to multicast other pages. Hence the overall bandwidth is superposition of the bandwidth of the individual groups. Thus, we have that the total bandwidth used in unicast is

$$\sum_{j=1}^{r} f_u^j(E_j(S), \lambda_j))$$

where $f_u^j(E_j(S), \lambda_j)) = \lambda_j E_j(S)$. The total bandwidth for multicast serving is

$$\sum_{j=1}^{r} f_m^j(E_j(S), \lambda_j L_j)$$

where for group j of the extreme case

$$f_m^j(E_j(S), \lambda_j, L_j) = E_j(S) n / L_j$$

and for group j of the typical case with $\alpha = 1$

$$f_m^j(E_j(S), \lambda_j, L_j) = E_j(S)\lambda_j \left(1 - \frac{\ln \frac{\lambda L_j}{\ln en_j}}{\ln en_j}\right)$$

where for $\alpha \neq 1$

$$f_m^j(E_j(S), \lambda_j, L_j) = \frac{E_j(S)\lambda_j}{n_j^{1-\alpha_j} - \alpha_j} \left(-\alpha_j \left(\frac{\lambda_j L_j (1-\alpha_j)}{n_j^{1-\alpha_j} - \alpha_j}\right)^{1/\alpha_j - 1} + n_j^{1-\alpha_j}\right) .$$

5 Summary

Our main contribution in this paper is the analytical analysis of the saving factor that can be achieved by using multicast versus using unicast in serving a a typical site. The analysis assumes the Zipf-like distribution for the access pattern for the pages in the site. We note that for the most interesting case where the parameter α of the Zipf-like distribution is larger than 1 the saving factor is almost independent of the number of pages (i.e the site may contain a huge number of pages). We also note that a crucial parameter in determining the saving factor is the product between λ and L which is the access rate for a group and the maximum latency we are allowed to deliver the files. We have also designed a simple criterion for a given site to decide in advance (or dynamically while collecting the information on the access pattern for the site) which pages to multicast and which pages to continue to transmit with the standard unicast.

We note that the saving factor can be further improved, if we further consider the peak behavior and not the average behavior of the requests. In this case the requirement for unicast bandwidth grow, while the requirement for multicast is stable. We can change somewhat the criterion of which pages to multicast - instead of comparing the average required rate for sending a page in unicast to its multicast bandwidth, we compare the instantaneous demand. The exact analysis in this case requires assumptions regarding the stochastic access pattern. Recent studies show that requests are not coming as, say, a Poisson process, but have a self-similar heavy tail distribution (see e.g. [12, 5]). Thus, this analysis can be complicated. Still, an approximation for the true saving can be obtained by using the results derived here, and choosing for λ a higher value, that will reflect the peak demand instead of the average access rate.

References

[1] J. Angel. Caching in with content delivery. *NPN: New Public Network Magazine, http://www.networkmagazine.com*, 2000.

[2] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and zipf-like distributions: Evidence and implications. In *Infocom*, 1999.

[3] John W. Byers, Michael Luby, Michael Mitzenmacher, and Ashutosh Rege. A digital fountain approach to reliable distribution of bulk data. In *SIGCOMM*, pages 56–67, 1998.

[4] R. J. Clark and M. H. Ammar. Providing scalable Web services using multicast communication. *Computer Networks and ISDN Systems*, 29(7):841–858, 1997.

[5] M. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1997.

[6] D. Dolev, O. Mokryn, Y. Shavitt, and I. Sukhov. An integrated architecture for the scalable delivery of semi-dynamic web content. Technical report, Computer Science, Hebrew University, 2000.

[7] A. Dornan. Farming out the web servers. *NPN: New Public Network Magazine, http://www.networkmagazine.com*, 2000.

[8] Z. Fei, K. Almeroth, and M. Ammar. Scalable delivery of web pages using cyclic-best-effort (udp) multicast. In *Infocom*, 1998.

[9] P. Krishnan, D. Raz, and Y. Shavitt. The cache location problem. In *IEEE/ACM Transaction on Networking (ToN)*, 2000.

[10] V. Padmanabhan and L. Qiu. The content and access dynamics of a busy web site: Findings and implications. In *ACM SIGCOMM'00*, 2000.

[11] Bandwiz White Paper. http://www.bandwiz.com/solu_library.htm.

[12] V. Paxson and S. Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[13] A. Wolman, G. M. Voelker, N. Sharma, N. Cardwell, A. R. Karlin, and H. M. Levy. On the scale and performance of cooperative web proxy caching. In *Symposium on Operating Systems Principles*, pages 16–31, 1999.