Scaling the World-Wide Web

Ghaleb Abdulla Marc Abrams
Edward A. Fox

Virginia Polytechnic Institute and State University
Department of Computer Science
Blacksburg, VA 24061-0106
+1-540-231-6931
{abdulla,abrams,fox}@vt.edu

September 11, 1996

Introduction

Will the World Wide Web (WWW) survive to the year 2001? The ever increasing demand by growing numbers of Web users for bandwidth-intensive media types is already outstripping the world-wide Internet capacity to deliver documents. Limited trans-atlantic network bandwidth today reduces European Web users accessing United States documents to the position of an elephant drinking through a straw.

Will the future be one in which smooth functioning of the Web depends on bandwidth conservation (voluntary or enforced), or is there a network architecture solution?

The Web has changed the fundamental dynamics of network usage. A URL broadcast on a popular television program (such as during a commercial in the Superbowl game in the United States) or in a historic event (such as the Shoemacker-Levy comet encounter with Jupiter) can produce a sudden spike in demand for to a particular server. This phenomenon was termed the "flash crowd" by Jakob Nielsen, when winter sports fans tried to access the latest results of the 1994 Winter Olympics in Lillehammer that were posted on the Web by the Norwegian Oslonett.

Scalability

Scalability can be defined as the ability to increase the size of the problem domain with a small or negligible increase in the solution's time and space complexity. Scalability of the World Wide Web architecture is the ability to increase the number of servers, clients, users, data types, data size, and the ability to handle servers in widely spread geographic locations, with minimum change in the quality of service. In the following sections we will try through a simple analytical model to assess whether the Web is scalable or not. We start by identifying techniques that improve scalability other than the brute force solutions of increasing network bandwidth and server throughput. These techniques are: use of caching on the server side (e.g., [1]), on the client side (e.g., caches built into Web browsers), and in the network (known as "Proxy caching") (e.g., [2]), and finally data compression.

Caching

One impediment to scalability is use of the wrong protocol for document delivery. For example, the aforementioned flash crowd phenomena consumes Internet bandwidth and server capacity because HTTP delivers a separate document copy to many readers. However, the convenience of Web browsers has made misuse of HTTP inevitable. In many cases, caching can alleviate the bandwidth consumption.

The Web design, although inherently unscalable, can be helped by widespread migration of document copies from servers to points closer to users. Migration can follow a distribution model, in which servers control where document copies are stored prior to client requests, or a cache model, in which copies automatically migrate in response to user requests. Distribution or replication is popular with commercial users that want to protect material with copyrights, and will only trust certain sites to keep copies of such material, rather than relinguishing control over who has a copy of the material.

Cache placement

Caching can be implemented in three places: at servers, in the network itself, and at clients. Caching on the server side is implemented by replicating the file system and the HTTP server and connecting the replicated servers with a high speed network [1]. This is similar to server mirroring, where the data on the server is copied into several other servers to reduce the load on the network and the original server; however in server mirroring the servers are not placed in one location; they can be separated by long distances. Server caching is used mainly to reduce the load on the server and enhance its response time and throughput.

Caching on the client side will reflect the user(s) interests. The cache contents will change according to user(s) access pattern and the cache size. Caching on the network side will reflect the access pattern of a group of users who share the cache. The effectiveness of the network cache can increase by placing it where we know that a group of users have a high degree of locality, and by implementing multiple proxy [3] or hierarchical proxy caching. Multiple proxy caching is when many clients share many caches and a cache that misses can query other caches. In two-level caching we have several network caches connected to another network cache with a larger cache size. If the document is not found in the first cache level, the second level caching is accessed, and if the document is not in the second cache level then it should be accessed from the source.

How caching helps scalability

Caching will not provide a magical solution for the scalability problem but it will help in solving the problem. Combined with other techniques, such as data compression, server replication, and better networks we might be able to provide a usable Web architecture for the year 2001. The model of section 3 will demonstrates how caching can decrease network load and support more clients.

Obstacles to Caching

First, caching only works with static and infrequently changing documents, whose number is declining due to the trend among commercial content providers toward dynamically generating Web pages from databases. Second, in HTTP 1.0 there is no reliable method to identify whether a Web document is cacheable (e.g., one cannot reliably distinguish a static text file from a script generated file). The current HTTP 1.1 draft provides a response message pragma to request no caching to address this problem. Third, there is no accepted method in the Web of keeping cached copies consistent. Finally, copyright laws could legislate caching proxies out of existence, unless they incorporate a pay-per-view method.

Network bandwidth

Network bandwidth connection speeds range from a 2400b/sec modem connection to Gb/sec or faster. Most small companies are connected to the Internet either by an ISDN (64 kb/sec) or through a T1 link (1.5 Mb/sec). A typical enterprise will be connected either by a single T1 or multiple T1 links. Some are connected through a T3 link (45 Mb/sec) and some are connected to the internet with an FDDI or ATM connection. LANs are 10 Mb/sec.

To study scalability we have to consider all these options and the number of clients connected to the network. In the future, clients from home will not use modems with current speeds to watch real time video. New kinds of home connection should appear to support the huge data transfere rate that is needed to play real time video. These can be provided either by cable companies or by telephone companies. For our model we will start with a T1 connection as a connection to the enterprise, we picked this number for two reasons. First Virginia Tech is connected to the internet with a T1 link. Second, our model is a simple model and we will not try to explore all the options available for network connectivity. Instead we will leave this for a further study to construct a detailed model that will consider all the other options.

Data compression

Data compression has seen limited use in the Web, but will continue to be an option to help scalability. Using networks with more bandwidth and using powerful servers will encourage WWW users to get eager for enhanced video and audio and again increasing the workload on both networks and servers. Data compression will help in slowing the network congestion problem. We can appreciate the value of compression by comparing the storage space needed for one minute of video before and after compression.

In the following section we will try to see how can we scale up the Web by using a combination of powerful servers, powerful networks, caching, and data compression.

A Simple WWW Model

In this section, we raise some fundamental questions about the ability of the Web to scale for future demands. Then we introduce a simple model of the WWW to help answer these questions. We will discuss the results obtained from the analysis of the model and come up with recommendations to achieve scalability. Finally we will list potential future research problems.

Questions We Want to Examine

We examine the following questions:

1. For a fixed number of clients in an enterprise, what is the bottleneck in client access of WWW documents? What can be done to alleviate the bottleneck?

2. For a fixed document request rate by a client to a popular Web server, how many Web clients can one server support? What can be done to increase the number of clients?

3. What happens when we start using video and audio heavily; can the scaled model withstand the new requirements imposed by the new media types?

The Model

A *client* is defined as an instance of a Web browser running on a computer. Our model is illustrated in Figure 1 and uses the following parameters:

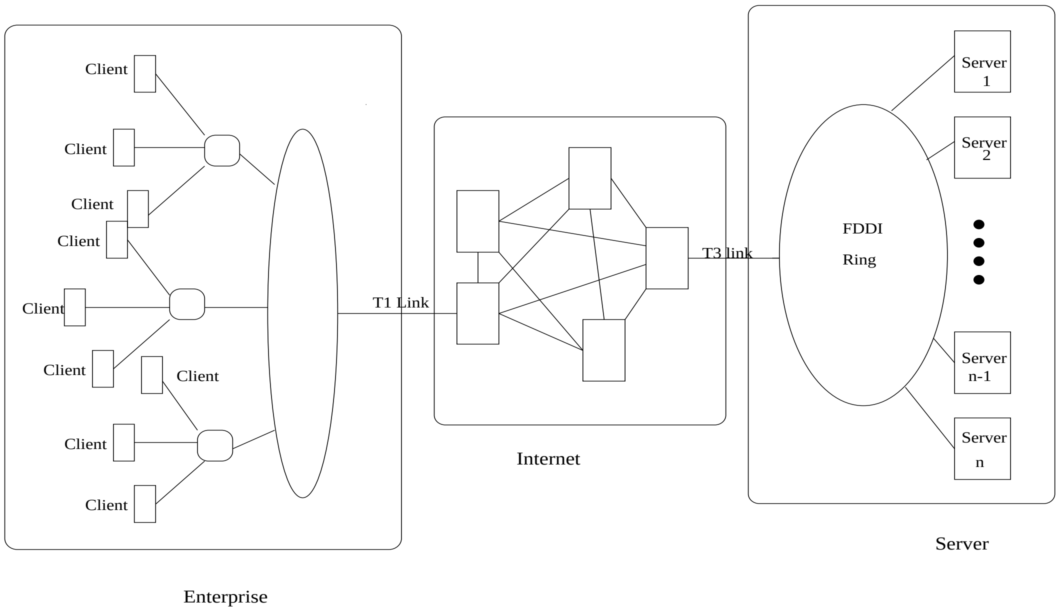

Figure 1: WWW Model.

n : total number of clients sharing one internet connection; in Figure 1, n is the clients inside the box labeled enterprise.

pv : percentage of transferred bytes that is video data.

pa : percentage of transferred bytes that is audio data.

pr : percentage of transferred bytes that are not and or audio(e.g., jpeg, gif, txt, html); $pr = 1 - pv - pa$.

U : utilization of a client (or fraction of time that a client is busy [i.e., a user is using a Web client]).

R : average requests/sec generated by a client when busy.

v_s : average video file transfer rate needed to achieve the real time playback.

a_s : average audio file transfer rate needed to achieve the real time playback.

r_s : average transfer rate for the rest of the files that makes accessing the Web acceptable for the rest of the users.

c_m : data compression ratio.

h_r : Proxy cache hit rate [4].

c_r : client cache hit rate.

C_t : traffic generated from one client.

I_c : the enterprise connection to the Internet.

The number of clients that can be supported in school with a T1 link can be easily computed by the following formula:

$$n = \frac{I_c - (\text{outgoing traffic})}{C_t}. \qquad (1)$$

For simplicity we will assume that the outgoing traffic equals zero; later we will change this assumption and see the effect of the outgoing traffic. The previous equation becomes

$$n = \frac{I_c}{C_t}. \tag{2}$$

We need to device a formula for C_t; it can be shown easily that C_t can be computed by the following expression,

$$C_t = U * R * [(pv * v_s) + (pa * a_s) + (pr * r_s)] * c_m. \tag{3}$$

Model Parameter Justification

Let us solve a hypothetical scenario, based on the numbers justified below. Then we vary these numbers and study the effect on scalability. We call this scenario because the parameter values we use are representative of values encountered in certain measurements, but are not necessarily representative of any real enterprise or organization. Table 1 lists the parameters and their values.

We choose the values for pv, pa, and ps based on the results obtained from log files we collected of and reported in [5]. The value of U, was calculated from the log files of a PC machine that is used by several students in a research lab by dividing the number of hours the machine was used on average by 24 hours. The value for R was obtained by tracing 10 different users' sessions and averaging them. In each session we counted the number of GETs and POSTs issued by a client; this number was then divided by the total session time to get the average requests/sec from the client.

The value for v_s was obtained by assuming that we want to play real time video over the network; we used QuickTime format with 160X120 resolution at 10 frames/sec accompanied with 8 bit 22kHz mono sound. The v_s value was computed by dividing the size of the QuickTime movie by its duration. We did this for five different movies and then we took the average of the obtained numbers. The transfer rate to playback a QuickTime movie with 160X120 resolution at 10 frames/sec in real time is much less than the rate reached when downloading a movie to play it back, given that the network is not heavily loaded. We study the effect of playing real time video since we expect it to be used heavily in the future also this transfer rate will provide a lower bound on retrieving video files since this is the slowest rate that is needed to view QuickTime video files with resol160X120 at 10 frames/sec.

The value for a_s was obtained in a similar way to v_s but we used Real Audio files. Real Audio files require very low transfer rate 8796 b/sec.

The value for r_s was chosen by noticing that most Web users are happy when they get a transfer rate of 5 KB/sec.

The value for the proxy cache hit rate was measured from our log files, ranging between 30%-60% [4]. Similar numbers have been reported in the literature [3]. In this paper we use the value 40% for the network cache hit rate. The value for the client hit rate cr also was measured by examining the caches on several clients in one of our labs, and the measurement did not look at the files that have been reused by a single client; instead we measured the percentage of the shared files in clients' disk caches between several Netscape clients in an under graduate lab over a month period. The measured value was 17%.

Model Analysis

Scenarios 1 and 2, Enterprise Side

Scenario 1 deals with network scalability. The scenario will answer the following types of questions: how many clients can we support in an enterprise (here we use a university campus, Virginia Tech, as an example)? How many clients can we support if we add caches? How many clients can we support if we use data compression? At what point is increasing network bandwidth the only way to increase the number of clients? Will the increase of

Table 1: Factors and their values for the base case.

Factors	pv	pa	pr	U	R	vs Bits/Sec	as Bits/Sec	rs Bits/Sec
Values	4%	18%	78%	.125	.124	468262	8796	40960

media quality (e.g. better sound, better video) decrease the number of supported clients? Is this also true if the percentage of using video and audio increases?

To answer these questions we will look at the part labeled "Enterprise" of Figure 1, which represents a simplified model of the campus network topology. Before we start the analysis we require the following assumptions: First, the load on the Internet backbone is light compared to its bandwidth; this assumption eliminates the internet as a potential bottleneck. Second, there is an infinite number of servers that can be accessed and that the load is distributed between the servers; this assumption eliminates the server as a bottleneck. Later we will see how can the server be a bottleneck and how can we scale it up. Third, we will use a compression ratio of 1:2, or $c_m = 0.5$ for all media types. Fourth, we will ignore the LAN connections inside the campus for two reasons. First, the LAN bandwidth ranges from 10Mb/sec to 155Mb/sec (Ethernet to ATM) which is higher than the campus connection to the internet (T1, 1.5Mb/sec). Second, the number of clients sharing a LAN segment is very small compared to the total number of clients.

We start the scenarios by computing the number of clients that can be supported for a base case; the values for the factors used to compute this base case are shown in Table 1.

From equations 2 and 3, the number of clients that can be supported in the base case is n =1942 clients. This number is reasonable for an enterprise such as Virginia Tech with its current population and Web usage behavior. Is this a reasonable number for the same enterprise 5 years from now? First let us see how can we use caching, compression, and the increase of network bandwidth to increase the number of supported clients.

Table 2: Factor level combination for one-at-a-time approach; underlined items indicate what is varied.

Factors				
Case Number	Client Caching	Proxy Caching	Compression	Network Connection T1/T3
0	No	No	No	T1
1	<u>Yes</u>	No	No	T1
2	Yes	<u>Yes</u>	No	T1
3	Yes	Yes	<u>Yes</u>	T1
4	Yes	Yes	Yes	<u>T3</u>

Scenario 1

Table 2 lists the factors and their combinations. To examine the effect of a factor, we will change it while keeping the other factors fixed and compute the number of clients that can be supported. This is a classic one-at-a-time experimental design approach. We will assume for simplicity that pa, U , R , and r_s will not change. We will also drop pr from the table since its value depends on pv and pa.

Table 3 shows how caching can increase the number of clients from 1942 in the base case to 2339 when we use client caching and to 3899 when we use a combination of client and proxy caching. Compression shows impressive results; upping the number of clients to 7798 if we manage to compress the data into half of its size we get 100% increase in the number of clients. The most impressive result is achieved when we use the 45Mb/sec since we are getting 28 times the number of previously supported clients. These numbers show that a combination of the previous techniques will allow supporting a huge number of clients.

The bad news is that these numbers are assuming all the data types did not increase in

Table 3: Number of clients that can be supported for each combination in Table 2.

Case	Number of Clients	% change over case 0
0	1942	-
1	2339	20%
2	3899	66%
3	7798	100%
4	233932	2800%

quality and percentage. The next scenario will show what happens when we increase data quality and utilization.

Scenario 2

In scenario 2 we will start from the results achieved in scenario 1. Assuming that we have the situation of case 4 in Tables 2 and 3, how does this number change by changing the factors shown in Table 4?

Table 5 list the cases and the resulting numbers of clients, plus the percentage of change. The percentage of change in this table reflects the percentage of decrease in the number of supported clients.

Table 5 shows that the effect of increasing the percentage of video data from 4% to 40% reduces the number of supported clients by 294%. The biggest effect was due to the video quality increase to 640X480, 30 frames/sec, when the number of clients decreased by 2254%. The increase in audio quality did not make a big difference for two reasons, first we assumed that the percentage of audio will remain 18%. Second, we only used multiple rate of the current Real Audio rate which is very small. Case 5 is worth looking at, for by the year 2001 we expect that the demand for digital video will be very high and the

Table 4: Factor level combination for one-at-a-time approach; underlined items indicate what is varied.

Factors				
Case	pv	vs	as	U
0	4%	160X120X10	Real Audio	.125
1	<u>40%</u>	160X120X10	Real Audio	.125
2	40%	<u>320X240X10</u>	Real Audio	.125
3	40%	<u>640X480X30</u>	Real Audio	.125
4	40%	640X480X30	<u>10XReal Audio</u>	.125
5	40%	640X480X30	10XReal Audio	<u>.36</u>

Table 5: Number of clients that can be supported and the percentage of change due to the cases in Table 4.

Case	Number of Clients	% change
0	233932	-
1	59322	294%
2	15919	272%
3	676	2254%
4	674	0.3%
5	234	63%

expected video quality will be either near the assumed value in the previous calculations or even better. This implies that in an enterprise such as Virginia Tech we will only be able to support 234 clients. We expect that the school will have more simultaneous active clients. This means that even with client caching, proxy caching, and compression we have a serious problem. In this case we might turn to the school Internet connection and instead of using a T3 with 45Mb/sec, use an ATM connection which will increase the number of clients , but only to 668 clients. 668 is still not many so we might think of Gb/sec network connection.

So far we assumed that the enterprise has only one type of traffic, that is traffic accessed by the clients. What if the enterprise have servers with data on them? For simplicity and to get an idea of the effect of such traffic let us assume that the outgoing traffic is the same as the incoming traffic. Then we have to divide the number of clients in half in Tables 3 and 5, which means that we will reach a bottleneck faster since the number of supported clients will be half of the numbers shown in Tables 3 and 5.

Scenario 3, Server Side

From the server side (see Figure 1) we are interested in the number of clients that a certain server can support. The following scenario will answer questions of the following type: how many clients can a contemporary workstation support at the same time? Can we use server caching to support more clients? When do we need to increase the server network bandwidth?

Contemporary workstations can get to a state where they cannot keep up with the request rate r_r or cannot fork the number of processes required to satisfy the arrived requests. This can happen to servers that host popular data, or servers that are used as network directories, for example SunSite or Yahoo. To alleviate this problem caches at the server side should be used. If the server is replicated m times, the request rate will drop on each server to r_r/m and the required number of processes to be forked to r_r/m. The designers of the network architecture can pick a workstation that can support a request rate= r_r and connect more than one workstation with a high speed network such as an FDDI ring as shown in Figure 1. This solution was adopted by NCSA [1] to scale up their popular server.

Scenario 3

Secenario 2 deals with server scalability; the factors *pv, pa, pr*, v_s, a_s, and r_s will be defined the same way as they were defined for the previous scenario except that these values represent the traffic going out from the server. However R will represent the rate of requests arriving at the server. This number might be very high compared to the client request rate since it represents the aggregation of requests by all clients. For a Web dedicated workstation, such as the SGI WebFORCE, the benchmarks show that it will sustain up to 96 requests/sec [6]. We will assume that the server will not do any thing more than retrieving and sending the required documents, and the server will not support querying or searching the data stored on the server since this will add extra load. We will try to eliminate the bottlenecks one after another and see how can we achieve the needed bandwidth to deliver video of 640x480 resolution at 20 frames/sec.

We will assume that all the requests that arrive at the server will result in a document transfer, in other words there is no proxy or client caching. We will include compression as a factor, and will assume that the compression ratio *cm* is 0.5 for all data types.

We will start by assuming that the server can support 96 requests/sec with no errors [6]. For this scenario we will use two cases from the previous scenarios as base cases. The first base case is the same base case for scenario 1, where we concluded that 1942 simultaneous clients can be supported. This case represent the currently measured values for the previous parameters. The second case will use the values shown in the last raw of Table 4 since

Table 6: The bottlenecks from increasing the number of clients in multiples of 10 for the first base case.

Clients	Server Connection T1/T3/FDDI	transfer Rate Mb/sec	Number of server Caches	Bottleneck
96	T1	.037	1	T1
960	T1	.37	1	server
9600	T1	3.7	10	T1 and server
96000	T3	37	100	server

this represents our upper bound on video resolution, audio quality, usage, and utilization. Using equation 3 for the first base case we get:

$C_t = .125 * .124 * [(.04 * 468262) + (.18 * 8796) + (.78 * 40960)] * .5 = 405$ b/sec per client. and for the second base case we get: $C_t = .36 * .124 * [(.4 * 7492176) + (.18 * 87960) + (.42 * 40960)] * .5 = 67628$ b/sec per client.

Starting from 96 clients accessing the workstation at the same time, Tables 6 and 7 show what happens when we increase the number of clients in multiples of 10.

Table 6 shows that with the current quality and demand of digital material, with a T3 link the server can support up to 96000 clients. The bottleneck in this configuration is the server; scaling it up requires 1000 server side caches. The price of such a caching system will be beyond the reach of most enterprises. This result shows that the server hardware will be a major future bottleneck. To solve this problem we should consider ideas such as data migration or replicating the server in other places where we anticipate lots of access activity. This solution will allow distributing the cost of the server replicas to other enterprises.

Table 7 shows that if servers are to support high quality real time video in the future they

Table 7: The bottlenecks from increasing the number of clients in multiples of 10.

Clients	Server Connection T1/T3/FDDI	transfer Rate Mb/sec	Number of server Caches	Bottleneck
96	T1	6.2	1	T1
960	T3	61.9	1	T3 and server
9600	FDDI	619	10	FDDI and server
96000	1Gb/sec	6190	100	network and server

have to have very high speed networks (greater than 1Gb/sec) and they should implement server caching.

Conclusions and Future Research

To achieve scalability we should use a combination of:

- Client caching,
- proxy caching,
- server caching,
- data compression,
- and increasing the network bandwidth.

The demand for high network bandwidth will increase and this will trigger more Web usage again introducing more demand for bandwidth. To be able to transfer real time video with high quality we need Gb/sec network connections especially to connect popular servers. Clients can live with 10Mb/sec or 1Mb/sec depending on how many clients will share the network connection.

The previous scenarios show that scalability of the Web is achievable, however it is not an easy or cheap process.

Future research should focus on constructing better and more detailed models. Modeling is an important concept in designing huge and expensive systems and it is of great help in designing infrastructure for the Web.

Traffic modeling is an important research area. WWW traffic is hard to characterize because the Web has so many user communities that differ in their use of the Web, we anticipate that it will require a number of different traffic models. An interesting problem is to check for self-similarity of the Web traffic in a specific user community and compare that with other studies for different communities [7]. Other interesting topics are, to study multi-level caching; will it be helpful? How and where can it be implemented? How can prefetching help? Can it reduce the server and network load? What is the best architecture that will allow efficient prefetching?

Efforts to log and monitor Web traffic should continue, since the WWW is changing rapidly and the assumptions that might work for today might not be valid tomorrow. It is very important to identify the factors that will shape the Web in the future, observe the rate of Web growth, observe users' behavior and see how behavior changes with the new technology.

We have been monitoring the WWW traffic for a year, and we have developed tools to log, analyse, and visualize collected traffic. Currently we are developing a comprehensive tool for WWW traffic monitoring and visualization [4]. We invite the other researchers to share their collected data and tools to help the WWW evolve to meet demands of the the future.

References

[1] T. T. Kwan, R. E. McGrath, and D. A. Reed. User access patterns to NCSA's World-wide Web server. Technical Report UIUCDCS-R-95-1934, Dept. of Comp. Sci., Univ. of IL, February 1995.

[2] A. Luotonen and K. Altis. World-Wide Web proxies. *Computer Networks and ISDN Systems*, 27(2), 1994. <URL: http://www1.cern.ch/PapersWWW94/luotonen.ps>.

[3] R. Malpani, J. Lorch, and D. Berger. Making world wide web caching servers cooperate. In *4th International World-wide Web Conference*, pages 107–117, Boston, December 1995.

[4] M. Abrams, S. Williams, G. Abdulla, S. Patel, R. Ribler, and E. A. Fox. Multimedia traffic analysis using Chitra95. In *Proc. ACM Multimedia '95*, pages 267–276, San Francisco, November 1995. ACM.

[5] S. Williams, M. Abrams, C. R. Standridge, G. Abdulla, , and E. A. Fox. Removal policies in network caches for world-wide web documents. In *submitted for publication*, February 1996. <URL: http://ei.cs.vt.edu/~succeed/96sigcomm/96sigcomm.html>.

[6] Michael Blakeley. Webstone performance analysis: Sun netra i20. <URL:http://www.sgi.com/Products/WebFORCE/WebStone/sun-ss20/sun-ss20.html>, December 1995. SGI.

[7] Mark E. Crovella and Azer Bestavros. Explaining world wide web traffic self-similarity. Technical Report TR-95-015, Boston University, October 1995.