Network Traffic Reduction by Hypertext Compression

BEN CHOI & NAKUL BHARADE

Computer Science, College of Engineering and Science
Louisiana Tech University, Ruston, LA 71272, USA

pro@BenChoi.org

Abstract. This paper proposes a three-step process to reduce the size of hypertext documents for reducing the network traffic on the Internet. With the explosive growth in the number of users on the Internet, the bandwidth demand has grown tremendously. Before the improvement of network infrastructures can meet the demand, this paper addresses the problem by reducing the size of HTML documents that currently occupies majority of the network traffic on the Internet. The proposed process consists of three steps: (1) shrinking HTML documents by removing data that will not alter the appearance and the function, (2) encoding HTML documents by representing most frequent used tags and words with single bytes, and (3) compressing HTML documents by using standard compression utilities. Our test results show that the proposed process achieves a reduction ratio of 81% on average. We also developed a browser plug-in program that provides transparent access for the users when viewing the reduced-size HTML documents. With the plug-in program, the users will not see any differences when viewing the reduced-sized HTML documents comparing to the original ones. They will, however, experience a reduction in retrieval latency on average of 15%.

Keywords: network traffic, retrieval latency, hypertext, compression, and shrinking

1. Introduction

In recent years with the explosive growth in the number of users on the Internet, the bandwidth demand on the Internet has grown tremendously. HTML and various other files are transferred across the network mainly from servers to client computers. Most existing network infrastructures cannot meet the bandwidth demand. This in turn increases the retrieval latency experienced by Internet users. Hence, to reduce this bandwidth impact and to reduce the retrieval latency there is a need to improve the network infrastructures or to reduce the network traffic. Before the improvement of network infrastructures can meet the demand, this paper addresses the problem by reducing the size of HTML documents that currently occupies majority of the network traffic on the Internet, which is likely to change in near future with increasing use of audio and video streaming.

Ben Choi and Nakul Bharade, "Network Traffic Reduction by Hypertext Compression," International Conference on Internet Computing, pp. 877-882, 2002.

BenChoi.info

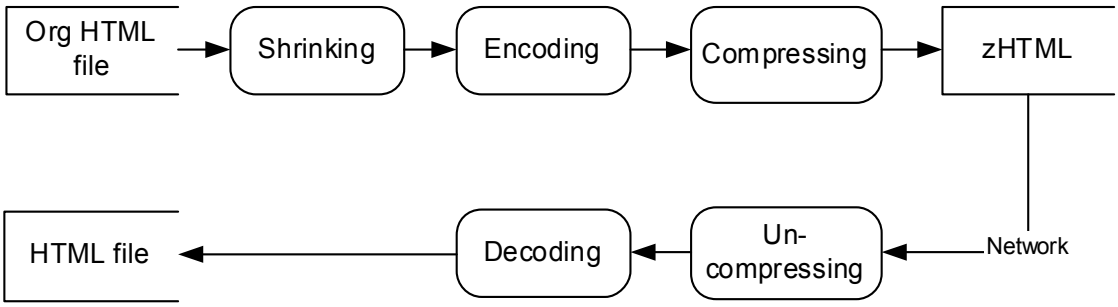

Figure 1. Complete process to reduce HTML network traffic

Several related research address this problem in transfer protocol level. For example, Barford and Crovella [1] and Heidemann and Obraczka [2] evaluated the performance of hypertext transfer protocols. Mao et al [3] optimized network transfer by cluster-based online monitoring system. Spring and Wetherall [4] eliminating redundant network traffic using protocol-independent technique, while Muller [5] improved multimedia transfer performance over TCP/IP.

Several other related research address this problem by using HTML shrinking alone. For example, some of the popular HTML shrinking utilities are: WebOverDrive [6], Xcition [7], and HTML (un)Compress [8]. HTML (un)Compress utility, for example, shows an average shrinking ratio, (original size – reduced size) / (original size) * 100, of 11% on 48 different HTML files selected at random. Xcition and WebOverDrive were also tested on the 48 HTML files and the results showed that they achieved a shrinking ratio of at most 25%.

We proposed to reduce HTML network traffic by a three-step process: (1) shrinking HTML documents by removing data that will not alter the appearance and the function, (2) encoding HTML documents by representing most frequent used tags and words with single bytes, and (3) compressing HTML documents by using standard compression utilities. We also developed a browser plug-in program that provides transparent access for the users when viewing the reduced-size HTML documents. With the plug-in program, the users will not see any differences when viewing the reduced-sized HTML documents comparing to the original ones.

2. Our Proposed Process

Our proposed process to reduce HTML network traffic consists of a three-step size-reduction and a two-step HTML recovery as shown in Figure 1. Original HTML files are shrunk, encoded, and compressed to form reduced-size HTML files called zHTML that are stored on the server. Client computers access the zHTML files through Web browsers. If the Web browser does not have a plug-in

to handle the zHTML file, a plug-in will be downloaded and installed in the client computer. The plug-in will automatically un-compress and decode a zHTML file to from a HTML file that will then be displayed in the browser.

Shrinking

In this step, the data that will not alter the appearance and the function of the HTML are removed. Such data include: extra spaces and tabs, carriage returns, META tags, DOCTYPE tags, comments, application specific data that is used to convert HTML back to the original creating application file format (e.g. Microsoft Words add about 17KB of data when creating HTML).

Encoding

In this step, frequent used tags and words of HTML are encoded using single bytes. A list of most frequent used HTML tags and words are first chosen based on a statistically sampling HTML files. Each of those chosen tags and words are replaced by one single ASCII character that has value above 128. Some special ASCII characters such as copyright character are kept unchanged. The result is a mapping table that maps a tag or a word to a special character. This techniques, however, is not applicable to Unicode.

Compressing

In this step, standard compression utilities are used to compress the

File Name	File Size	File Size After Using Our Algorithm	File Size After Using ZIP	Percent of Reduction Using our Algorithm	Percent of Compression Using ZIP
Boeing.html	11.8	2.91	3.72	75.33	68.47
Computersciences.html	13.8	3.10	3.40	77.53	75.36
Dictionary.html	7.54	2.21	2.97	70.69	60.61
Gigabit-ethernet.html	5.81	1.12	1.38	80.72	76.24
Hp.html	44.2	6.48	7.94	85.39	73.21
Redcross.html	37.3	6.68	7.94	82.09	73.21
Reuters.html	50.8	6.65	9.43	86.90	81.43
Washingtonpost.html	65.7	7.92	12.4	87.94	81.18
Aboutustoshiba.html	13.0	2.34	2.75	82.00	78.84
Brassring.html	33.5	4.32	5.33	87.10	84.08
Ececmu.html	21.5	3.32	3.85	84.55	82.09
Ge.html	74.0	5.43	14.4	92.66	80.54
Jobs.html	6.25	1.28	1.57	79.52	74.88
Mitresearch.html	3.68	1.25	1.43	66.0	61.14
Motorola.html	10.7	2.11	2.57	80.28	75.98
Msnbc.html	23.1	6.42	6.93	72.20	70.00

Table 1. Samples of comparing our size-reduction process with ZIP utility

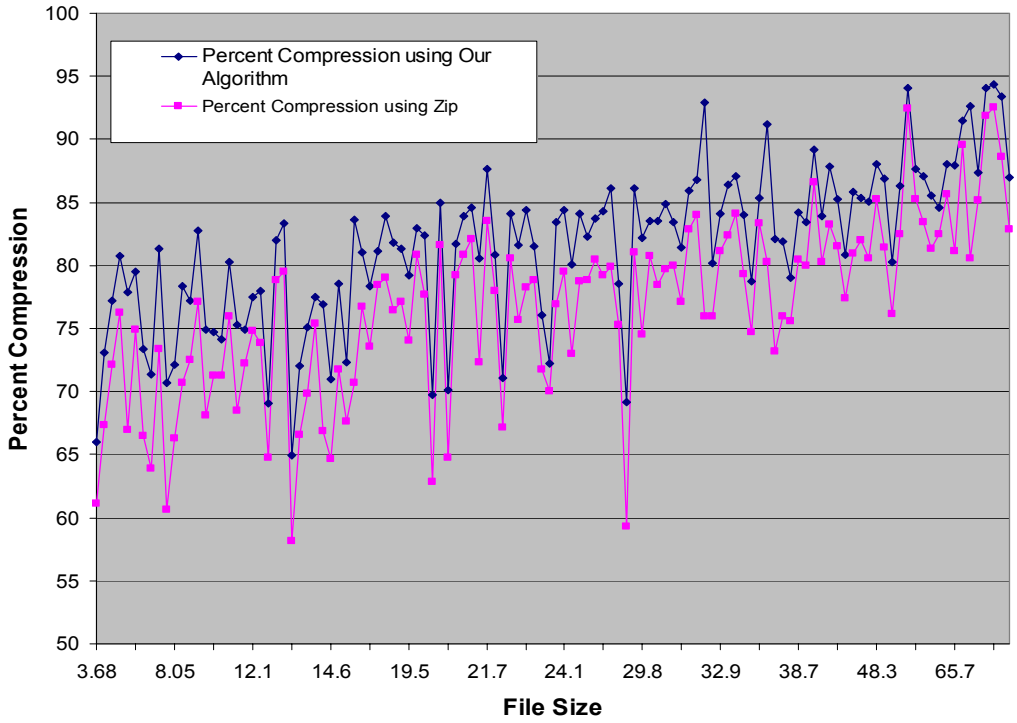

Figure 2. Ratio of size reduction

encoded-HTML file. We choose to use ZIP compression program. The file tag of the resulting compressed file is changed to zHTML.

Un-compressing

In this step, ZIP utilities are used to un-compress zHTML file. The result is an encoded-HTML file.

Decoding

In this step, the encoded-HTML file is translated back to HTML file. A copy of the mapping table used during the encoding step is used here to map special characters back to HTML tags or words.

3. Our browser plug-in

To be able to decode the encoded-HTML, the client computer must have the decoding algorithm. The decoding algorithm is implemented in our browser plug-in that also includes algorithm to un-compress the zHTML file. Currently our plug-in is developed using Microsoft Foundation Classes and only supports Windows platform.

One of the essential requirements for our project is that our strategy to reduce network traffic and retrieve latency

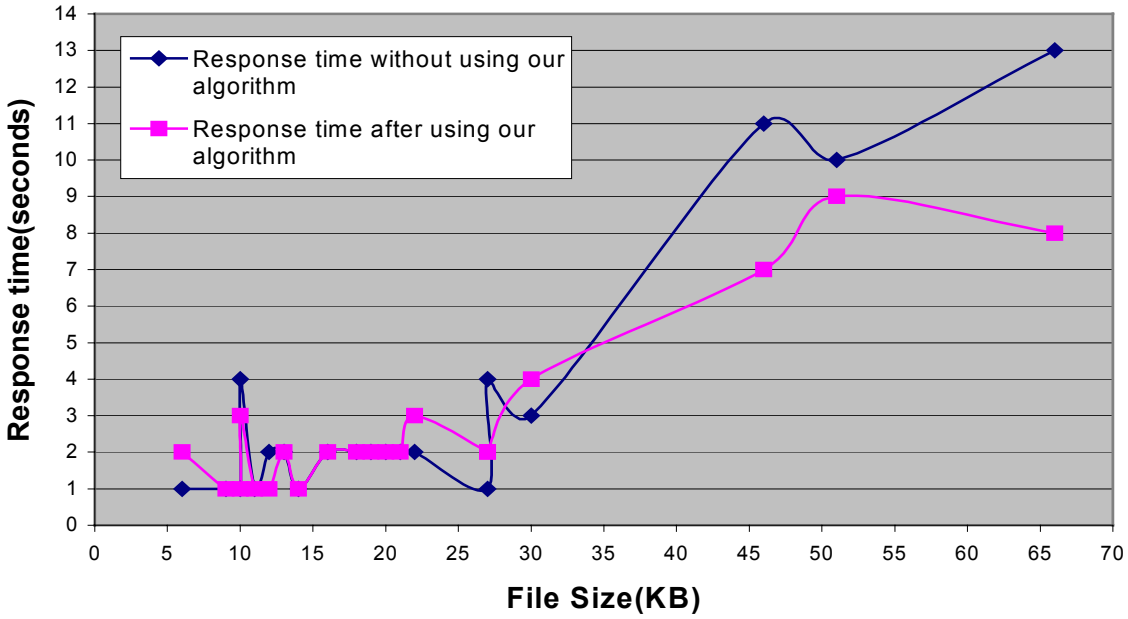

Figure 3. Response time

should not cause any incontinence to the users. We achieve this objective by developing a plug-in for browsers. If the required plug-in is not already installed in the client computer, it will be downloaded and installed as needed. The plug-in provides transparent access for the users when viewing zHTML file. The users will not see any difference in appearance or in function when viewing zHTML file comparing to the original HTML file.

4. Test Results

The proposed HTML size-reduction process was evaluated in two stages. First, we compare the size-reduction process to generate zHTML file with ZIP compression utility that can achieve one of the highest compression ratios among the existing compression utilities. Second, we evaluate the retrieval latency with our zHTML file comparing to original HTML file.

Table shows samples of files used for evaluating size-reduction process. The percentage of reduction is computed by (file size – file size after reduction) / (file size) * 100, and similarly for computing percentage of compression. The test results for testing 109 HTML files are shown on Figure 2. On average, our process achieves a size-reduction ratio of 81% comparing to using ZIP utility along that achieves a compression ratio of 76% on average.

Figure 3 shows the results for evaluating retrieval latency. In this experiment, we consider the retrieval latency as the time interval between sending a request of a HTML file to the completion of

displaying the file on a browser. Our test results show that using our proposed process and our plug-in achieves an average retrieval latency of 2.85 seconds comparing to 3.35 seconds in case of not using our approach. This amounts to a reduction of 15% on the retrieval latency on average under our test cases.

5. Conclusion

We proposed a process to reduce network traffic and retrieval latency by shrinking, encoding, and compressing HTML file. Although the process is simple and straight forward, the results are promising that are a reduction ratio of 81% for network traffic and a reduction ratio of 15% for retrieval latency. Our browser plug-in helps to maintain transparency access for Internet users. Since the improvement of network infrastructures are slow to keep up with the explosive growth of Internet usage, future research in network traffic reduction and in retrieval latency reduction will remain essential.

References

[1] Paul Barford and Mark Crovella, "A performance evaluation of hyper text transfer protocols", *ACM SIGMETRICS Performance Evaluation Review*, (Proceedings of the international conference on Measurement and modeling of computer systems), Volume 27 Issue 1, May 1999

[2] John Heidemann, Katia Obraczka, and Joe Touch, "Modeling the performance of HTTP over several transport protocols" *IEEE/ACM Transactions on Networking* (TON), Volume 5, Issue 5, October 1997.

[3] Yun Mao, Kang Chen, Dongsheng Wang, and Weimin Zheng, "Web Performance Optimization: Cluster-based online monitoring system of web traffic" in *Proceeding of the Third International Workshop on Web Information and Data Management*, November 2001.

[4] Neil T. Spring and David Wetherall, "A protocol-independent technique for eliminating redundant network traffic", *ACM SIGCOMM Computer Communication Review*, (Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication), Volume 30, Issue 4, August 2000.

[5] Nathan J. Muller, "Improving and managing multimedia performance over TCP–IP nets", *International Journal of Network Management*, Volume 8, Issue 6, December 1998.

[6] WebOverDrive, http://www.programfiles.com/index.asp?ID=2612

[7] Xcition, http://psyral.com/products/xcition.html

[8] HTML (un)compress, http://www.groundlayerz.com/apps/htmlcomp.shtml