7. Conclusion and Future Work

Based on the analysis done, text content-types affect network traffic the most (37%) and gzip gives the best compression algorithms to compress the text content-types. The system was implemented using these compression algorithms, and the performance was measured. The other major web content-types that affect the network traffic are images (29%). Decreasing quality factor of image/jpeg, leads to a compression while quality reduction is not noticeable. Text content-types achieve 80% of compression ratio while image/jpeg gives a 75% compression ratio on average.

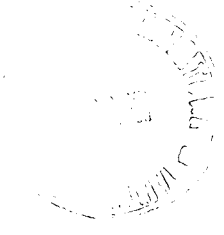
HTTP headers are text based and have a common set of tags in each. Applying gzip compression algorithms with a pre-set dictionary, gives a better compression on HTTP headers. It achieves 70% compression ratio on average.

Opening and closing a TCP connection takes a considerable length of time. Permanent connections were introduced in between the two proxies for increasing the performance of the system. Many HTTP connections are sent sequentially over one TCP connection. The improvement is greater in HTTP 1.0 connections.

The results, presented in this project, show that the inter-proxy data compression system gives good performance with the bandwidth utilization. When considering bandwidth usage, the new system uses less amount of the bandwidth, compared to old system, since compression is applied on major content-types such as Text Content-types, Image/jpeg, and HTTP headers.

Although the compression adds latency to the system, when we test the system on a low bandwidth link, we get better performance with the response time as shown in the Section 6.2.

From the results, it can be said that memory management is well-managed in the implementation.



7.1. Merits of the system

The System is independent of both client applications and web servers as proxies are located in between client applications and web servers. Client applications will only notice data coming faster than they had done earlier.

The HTTP compression is the most popular compression system for carrying web data over the network. HTTP compression system compresses data only in the HTTP Body but not in the HTTP Header. Compared to that system, our system gives HTTP header compression additionally. Also, it does not gain from image compression while we achieve lossy compression which leads to high compression ratios for images.

Hardware level compression is used mostly on network equipment today. But, this mechanism compresses data blindly as it uses the same technique (pattern replacement) for all data on the network. It does not consider the difference of nature of each content-type. A selective compression is done on the web data applying best possible algorithm for content-types in our system.

In addition to that, image-jpeg compression is provided by our system using quality reduction, which other existing systems have not provided.

TCP connections have to be made over limited bandwidth links through any of the existing systems, and data has to be sent over them. However, creating TCP connections is avoided by our system every moment and instead, it re-uses existing TCP connections.

7.2. Future Enhancements

As an optimization, a compressed file (jpg) may be uncompressed to obtain the Discrete Cosine Transform (DCT) coefficients and re-quantize them at a low quality factor. This improves performance as it eliminates the Inverse Discrete Cosine Transform (IDCT) and DCT operations in the previous method as well as the round-off errors associated with these operations [22], [23].

An extension of the system has been planned to handle GIF and PNG images, principally by reducing colour depth. Algorithms may also be found for other less-

common content-types such as Flash. Our implementation is easily expandable to handle these.

Some users may require the original image even if the quality reduction due to our lossy image compression is small. In that case, the system should have an option to bypass the compression modules and produce the original image. Provision of this option is planned in a future version.

Currently, the system is working on a single processor system. In order to distribute compression among several processors, a multi processor system can be introduced so that compression can be done in parallel, to get faster compression.

Today people mostly use HTTP 1.1, the newest HTTP version. The system was implemented to support only for HTTP 1.0. Extension of the system to support HTTP 1.1 is required.

References

[1] Sumith Gamage, Gihan Dias, *Dynamic Bandwidth Negotiation among Web Proxies*, ERU (Engineering Research Unit) Symposium 2004, Sri Lanka.

[2] Chamara Disanayake, Gihan Dias, C. R. de Silva, *A Market-Based approach to control Web bandwidth Usage*, APAN, Cairns, Australia, July 2004.

[3] Vishaka Nanayakkara, Sven Tafvelin, Gihan Dias, *Empirical Comparison of the Characteristics of Academic Network Traffic*, Technical Report No. 02-24, Department of Computer Engineering, Chalmers University of Technology, Sweden, November 2002.

[4] Douglas E. Comer, *Internetworking with TCP/IP, Volume I – Principles, Protocols and Architecture* (Second Edition – ISBN-81-203-0926-X)

[5] R. Fielding, J. Gettys, J.C. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, *Hypertext Transfer Protocol – HTTP/1.1*, RFC 2616, June 1999

[6] Zlib Manual and Documentations, January 2004, http://www.zlib.net/

[7] Independent JPEG Group, Various Documentations and Manuals, November 2003, http://www.ijg.org/

[8] Mark Nelson, *The Data Compression Book*, M&T Books, NY, 1991. ISBN 1-55851-214-4

[9] Huffman Algorithm, November 2003, http://datacompression.info/Huffman.shtml

[10] *Compression of Individual Sequences via Variable-Rate Coding* in IEEE Transactions on Information Theory (September 1978).

[11] Run Length Encoding, November 2003, http://datacompression.info/RLE.shtml

[12] JPEG, November 2003, http://datacompression.info/JPEG.shtml

[13] W. Richard Stevens, *UNIX Network Programming, Networking APIs: Sokets and XTI*, Pearson education, Inc, India, 2002. Volume 1
ISBN 81-7808-014-1

[14] WAN Compression FAQs, November 2003,
http://www.cisco.com/warp/public/116/wan_compression_faq.html

[15] FREED, N., BORENSTEIN, N., MOORE, K., KLENSIN, J., AND POSTEL, J. *Multipurpose Internet Mail Extensions* (MIME). RFC 2045.

[16] Squid Web Proxy Cache, September 2003, http://www.squid-cache.org

[17] Calamaris squid analyzing tool, September 2003,
http://cord.de/tools/squid/calamaris/Welcome.html

[18] ZLIB Manual, January 2004, http://www.zlib.net/manual.html

[19] GNU zip Documentations and Manuals, January 2004, http://www.gzip.org.

[20] The bzip2 and libbzip2 official home page, January 2004,
http://sources.redhat.com/bzip2/

[21] Official web site for LZOP Home page, January 2004, http://www.lzop.org

[22] H.H. Bauschke, C.H. Hamilton, M.S. Macklem, J.S. McMichael, and N.R. Swart, *Recompression of JPEG images by Requantization*, IEEE Transactions on Image Processing, July 2003.

[23] Syin Chan *Recompression of still images*, University of Kent, Computing Laboratory, University of Kent, Canterbury, UK, Technical Report 2-92, March 1992.