Performance Evaluation of XHTML encoding and compression

Sathiamoorthy Manoharan

Department of Computer Science,
University of Auckland, Auckland,
New Zealand

Abstract. The wireless markup language (WML), used in WAP, is based on XML and therefore is quite verbose. Mobile devices do not typically have the processing power or memory to process such verbosity. Since all WML documents have to go through a mobile gateway, WAP 1.x specifies that the WML files are encoded to make them smaller and to process them faster on the mobile device.
WAP 2.0 includes support for a restricted version of XHTML, called the XHTML mobile profile. This is to make content development and management easier. XHTML, though as verbose as WML, is never encoded by the gateway. This is mainly because WAP 2.0 does not require the use of a gateway.
This paper investigates what performance gains are possible if XHTML were encoded and compressed. The experiments show that while there is a substantial performance gain using compression, gains due to encoding are comparatively small. The empirical results justify that XHTML encoding is not generally useful.

1 Introduction

XML-based languages are quite verbose. When used on a mobile or wireless network, such languages require considerable bandwidth. The wireless markup language (WML), used in WAP, uses a binary encoding to reduce this bandwidth requirement [1–3].

Newer mobile devices support XHTML, another XML-based language, but very closely related to HTML [4, 5]. This is to make content development and management easier. XHTML, though verbose just like WML, is never encoded by the gateway. This is due to the fact that WAP 2.0 does not require a WAP gateway [6]. Thus, when XHTML passes through an existing gateway, it is never encoded. Besides, the newer devices have more processing and memory power, so these devices, unlike their ancestors, are able to support compressed payload. Compression therefore can be used effectively to reduce bandwidth requirements. WAP 2.0 thus supports compressed HTTP payloads in responses [6].

The goal of this paper is to investigate what performance gains are possible if XHTML were encoded. We also employ several data compression schemes to compress the XHTML payload and measure the effectiveness of compression with and without encoding.

Even though WAP 2.0 does not require a gateway for standard HTTP pull transactions, it does require a gateway for WAP push transactions. Therefore, if XHTML encoding proves to be useful, the gateway could perform the encoding, if present.

The rest of the paper is organized as follows. Section 2 outlines the XHTML encoding used in the paper. Section 3 discusses the choice of workload and the figure of merit for the performance comparisons. Section 4 looks at the effectiveness of XHTML encoding in the light of experimental results. Section 5 examines the use of compression for XHTML and presents some experimental results. The final section concludes the paper with a summary.

2 Encoding XHTML

The WAP standards do not define an encoding for XHTML as it does for WML. We define a local standard for encoding based on WML encoding, and use this encoding as the basis for our investigation. Table 1 is a small selection of tags and their encodings, provided here as an example.

Table 1. A selection of XHTML tags and their encodings

Tag	Description	Encoding
<html>	Defines an HTML document	0x02
<body>	Defines the body element	0x03
<head>	Defines information about the document	0x04
<title>	Defines the document title	0x05
<meta>	Defines meta information	0x06
<a>	Defines an anchor	0x41
<link>	Defines a resource reference	0x42
<img>	Defines an image	0x43

Following the WML encoding standard, all the end tags (such as </html>, </head>, and) are encoded as 0x01. Similarly, tags with attributes will have 0x80 added to them while tags with content will have 0x40 added to them [3].

3 Workload and Figure of Merit

There are no standard corpora categorizing WAP documents. However, there are some research works which synthesize WAP load based on synthetic Internet loads [7, 8].

For our experiments, we picked several XHTML and HTML documents from mobile sites and, where the original documents were in HTML, translated them to XHTML. Most of these documents were small (about 2 KB on average), so we also adapted a set of XHTML documents that are medium and large (about 10 KB and 100 KB on average) from HTML documents on the Web. We did not use any document larger than 120 KB, since most of the mobile devices currently available have a per-document download limit (i.e. maximum PDU size) much smaller than this.

The figure of merit we use in the experiments is modelled on the time it takes to view a document once the user has sent the request for the document. This consists of the following entities:

1. Time to send the request
2. Time to receive the response
3. Time to decompress the payload in the response, if applicable
4. Time to render the payload

It is difficult to model the over-the-air data transfer time for sending the request and receiving the response, since it depends on packet size, fixed cost, etc. of the carrier (GSM, CDMA, etc). For the purpose of our experiments, we assume that the time to send the request is a negligible constant, and the time to receive the response is proportional to the size of the data. The time to decompress the payload, if compressed, is measurable at the receiver end.

Fig. 1. A simple WAP client implementation

The time to render the payload is modelled as the time to find all occurrences of a given XHTML tag (e.g. <p>) in the XML document. Note that finding the occurrence is not done through a simple string search, but by constructing the XML document tree, thus mimicking to some extent what a renderer would generally do. Rendering an encoded document is less time consuming than rendering an un-encoded one, mainly because there is no need to validate the XHTML document and parsing the document is also much simplified. Recall that the WAP gateway will validate the document while encoding; documents that do not conform to the DTD will not pass through the gateway. Un-encoded documents, however, are passed through the gateway without any validation checks.

The experiments are conducted using a custom WAP gateway (locally constructed to do XHTML encoding) and a custom WAP client. Time measured relative to the local CPU is normalized to reflect slower mobile CPUs. Figure 1 shows a simple WAP client implementation.

4 Effectiveness of XHTML Encoding

Our first experiment investigates the performance gains of XHTML encoding. Table 2 shows the average file size reduction for our three classes of XHTML documents. The compression ratio is the ratio of size of the original file and the size of the encoded or compressed file. A ratio of 1 or less means no benefit from encoding or compression. Higher numbers mean a better encoding or compression.

Table 2. Average compression ratio with encoding

File size	Compression ratio
Small	1.81
Medium	1.26
Large	1.28

Table 3 shows the measured time to view (as defined in Section 3) for these files.

Table 3. Average time to view with and without encoding

File size	Time to view (s)	
	Without encoding	With encoding
Small	2.37	1.02
Medium	25.79	18.81
Large	196.55	170.64

We note that encoding is less effective for large file sizes. The rendering time increases with large files, specifically for files without encoding, since XML validation required for these files and parsing is more time-consuming.

5 Use of Compression

Our next experiment investigates the use of compression in conjunction with encoding.

Early mobile devices were not powerful enough to decompress files due to the demand on CPU and battery power. However, newer devices have faster CPUs and higher power (enough to play Java games). Thus, device-level decompression is now practical. However, note that the extra processing demanded by decompression does require power, consuming battery when processing compressed documents.

We consider four compression schemes and compare compressed payloads against uncompressed ones. The compression schemes we tested are ZIP, GZIP, BZIP2, and LZSS [9]. The first three compressions are implemented using the SharpZip library [10]. Our comparisons included both encoded and un-encoded payloads.

Table 4. Average compression ratio without encoding

File size	Compression scheme			
	ZIP	GZIP	BZIP2	LZSS
Small	5.8	6.3	3.6	1.8
Medium	25	24.7	19.7	7.1
Large	97.6	93.3	48.6	8.3

Table 5. Average compression ratio with encoding

File size	Compression scheme			
	ZIP	GZIP	BZIP2	LZSS
Small	6.7	6.8	4.2	1.9
Medium	27.6	25.2	21.3	7.8
Large	98.3	95.4	51.1	9.7

Tables 4 and 5 show the average file size reduction for the four compression schemes; Table 4 is the results without encoding while Table 5 is the results with encoding. Note that the gain due to encoding is comparatively low.

Note that ZIP and GZIP perform better than BZIP2 and LZSS.

Table 6. Average time to view without encoding (s)

File size	Compression scheme			
	ZIP	GZIP	BZIP2	LZSS
Small	1.37	1.37	1.52	3.13
Medium	3.48	3.48	3.70	6.81
Large	11.91	11.91	12.47	17.35

Table 7. Average time to view with encoding (s)

File size	Compression scheme			
	ZIP	GZIP	BZIP2	LZSS
Small	1.16	1.16	1.40	2.72
Medium	3.20	3.21	3.63	5.79
Large	10.84	10.84	11.53	14.20

Tables 6 and 7 show the average time to view for the four compression schemes; Table 6 is the results without encoding while Table 7 is the results with encoding. Again, note that the gain due to encoding is quite small.

6 Summary and Conclusions

This paper investigated the possible performance gains of encoding and compressing XHTML documents used within the WAP application environment. The experiments show that while there is a substantial performance gain using compression, gains due to encoding are small in comparison to the gains due to compression. This validates that encoding XHTML files is not generally useful.

References

1. WAP Forum: Wireless Application Protocol Architecture Specification, WAP-210, July 2001.
2. WAP Forum: Wireless Markup Language Specification, WAP-191, February 2000.
3. WAP Forum: Binary XML Content Format Specification, WAP-191, July 2001.
4. Mark Baker et al, editors: XHTML Basic, W3C recommendation, December 2000. Available at http://www.w3.org/TR/xhtml-basic/
5. WAP Forum: XHTML Mobile Profile, WAP-277, October 2001.
6. WAP Forum: Wireless Application Protocol. WAP 2.0. Technical White Paper. January 2002.
7. Stuckmann, P., Finck, H., and Bahls, T.: A WAP Traffic Model and its Appliance for the Performance Analysis of WAP over GPRS, In Proceedings of the IEEE International Conference on Third Generation Wireless and Beyond (3Gwireless 01), San Francisco, USA, June 2001.

8. Stuckmann, P., Hoymann, C.: Dimensioning GPRS Networks for Coexisting Applications based on WAP and Conventional Internet Protocols. In Proceedings of the European Wireless Conference, Florence, Italy. February 2002.
9. Storer, J. A., and Szymanski, T. G.: Data Compression via Textual Substitution. Journal of the ACM, vol 19, no 4, (1982) 928–951.
10. SharpZip: Zip, GZip, BZip2 and Tar Implementation For .NET. Available at http://www.icsharpcode.net/Open Source/SharpZipLib/