Ubiquitous Web Access: Collaborative Optimization and Dynamic Content Negotiation

Xavier Sanchez-Loro, Victoria Beltran, Jordi Casademont and Marisa Catalan
Wireless Networks Group – Telematics Department
Technical University of Catalonia
Barcelona, Spain
Mod. C3 Campus Nord, c/ Jordi Girona 1-3, 08034 Barcelona, Spain
{xsanchez,vbeltran,jordi.casademont,mcatalan}@entel.upc.edu

Abstract

Traditionally, cellular wide area networks like UMTS are used as Internet access networks for particular users but, in some cases, they can be employed to provide Internet access to other smaller networks as well. The main inconvenient is that cellular networks have not the same bandwidth than wired networks and therefore, the cellular channel becomes a network bottle-neck. To help to mitigate this situation and in order to improve the user's experience different optimization techniques exist, especially in web traffic. This paper studies first the existing synergies at HTTP layer between device capabilities expression, content negotiation, channel optimization and content adaptation. And secondly, it presents a system where HTTP requests transmission is optimized, showing a significant improvement in response time by means of HTTP header reduction over the cellular channel. In order to obtain a successful browsing experience, headers should be restored when reaching the Internet. This dynamic header reconstruction allows giving enriched and more expressive information about user's device and browser capabilities. Thus navigation speed and user's QoE can be enhanced by means of dynamic content negotiation in order to obtain adapted (and lighter) content and responses from web servers and adaptation proxies alike.

1. Introduction

Nowadays, connecting to the Internet can be done with several different access networks and in heterogeneous environments, with different types of devices, every one of them with very different software and hardware characteristics and capabilities (e.g. PCs, Personal Digital Assistants -PDAs, phones, sensors/actuators, etc.). Most of them have one or more network interfaces, usually radio, and can browse the World Wide Web (WWW), which is the current deployment platform for multiple applications on a wide spectrum of devices and surroundings. This heterogeneity requires a contextualization or profiling of the user's device [1-3] in order to provide tools for pervasive, ubiquitous and device independent web access - with applications and content adapted to the characteristics of the delivery context. Device characterization allows web servers and adaptation proxies in the delivery chain to adapt and personalize content and application behaviour to actual device capabilities.

On the other hand, cellular networks like Universal Mobile Telecommunications System (UMTS) provide a wide coverage area and can provide access to the Internet to different mesh, ad hoc or local area networks with no wired connection. Hence a UMTS operator, for example, can use mesh or WLAN networks as a means to increase indoor coverage. Likewise emergency networks or networks without support by a wired infrastructure can use cellular networks to access to the Internet. Figure 1 illustrates the system architecture, where the client access network (WLAN, ad hoc, mesh, etc) is connected to the Internet through one or more gateways (GWs) and a Proxy Server (PS).

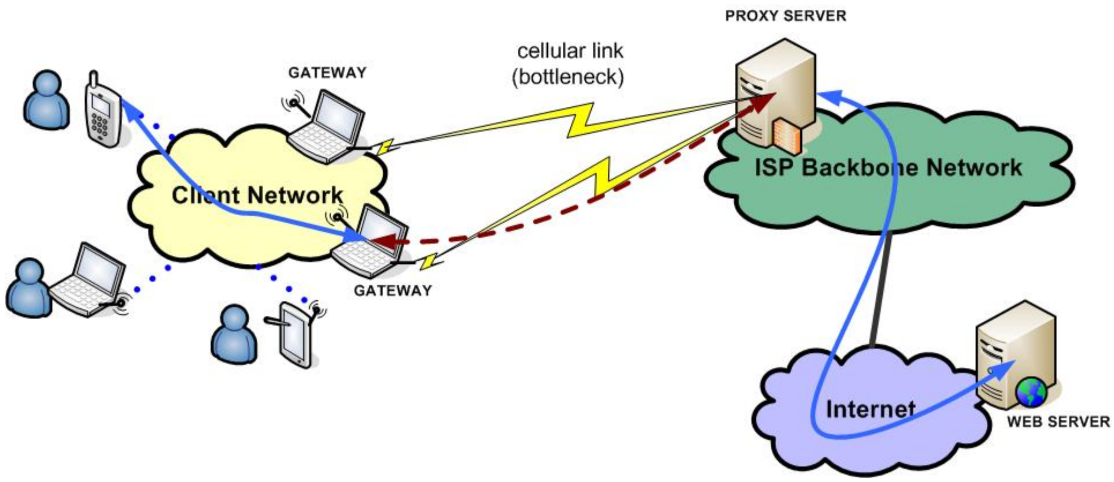

Figure. 1. System architecture

In these cases, the cellular link with the backbone network is usually an asymmetric link and with lesser capacity in terms of bandwidth and delay than the client access network (e.g. General Packet Radio Service -GPRS, High-Speed Downlink Packet Access -HSDPA, etc.), becoming a potential network bottleneck. For this reason it is necessary to apply optimization techniques to improve access to different Internet services, like web navigation. In addition, an enhancement on the uplink optimizes the overall behaviour and web page downloading time due the Stop&Wait behaviour of HTTP protocol and the asymmetry between uplink and downlink channels [5]. One technique is removing most static content negotiation headers in HTTP requests over the cellular link and later reconstructing them at the wired side of the network. Headers must be restored in order to provide web servers information about browser capabilities (content negotiation headers) and navigation information (e.g. cookies, referrer, host, etc.) in order to not degrade web navigation nor content negotiation and adaptation. If no modification of user's browser is wanted, GWs on client access network and the PS on the wired network (e.g. ISP backbone network) must work together to intercept and optimize requests. So GWs hijack HTTP related connections; removing most HTTP headers except on those indispensable for web browsing (e.g. Host, Cookies, etc) not inferable from browser and device capabilities. Then GWs forward requests to the PS and it regenerates headers values before reaching remote web servers. This topology present some opportunities to further enhance and optimize web navigation by means of toying with header values in order to take advantage of content adaptation capabilities of server and other proxies in the delivery chain.

On one hand, header reconstruction implies some kind of mechanism to detect current browser in use and how it identifies itself and negotiates content. This could be as simple as copying original header values sent by device's browser; but, with this necessity of browser capabilities profiling, we could take advantage of this opportunity to deploy a more sophisticated device capabilities detection and user profiling system. Such a system allows giving enriched and more expressive information about user's device and browser capabilities; enhancing navigation speed and user's QoE (Quality of Experience) by means of dynamic capabilities expression in order to obtain adapted (and lighter) content and responses from web servers and adaptation proxies alike.

On the other hand, this interception model allows PS to further toy with the content of HTTP headers sent by the browser to the remote server, adding its own vision of the delivery context. This way, the

PS can modify header and profile values according with its own privileged vision of the device capabilities, access network state and cellular link condition. Consequently, different regeneration policies can be applied with different purposes. In example, dynamic policies could be applied to change header values depending on available bandwidth of cellular link, forcing remote web servers to deliver lighter or heavier contents in order to mitigate the effect of network congestion and delay (i.e. plain-text, HTML, DHTML or ActionScript documents). Hence a web server with minimal intelligence can modify content to deliver in order to better fit delivery context characteristics.

1.1. Related & Previous work

Use of Application Layer Performance Enhancing Proxies (PEP) for addressing the challenges induced by the problematic characteristics of low bandwidth Wireless-Wide Area Networks links is a widely studied field and its efficiency has been demonstrated [5, 12]. There are several techniques for performance enhancing at HTTP layer like extended caching [15], pipelining, compression and object adaptation [14, 15], delta encoding [11], etc. Efficiency of link optimization at the HTTP layer by means of reducing number and size of request headers was demonstrated in [13-14].

In [13] HTTP protocol reduction is limited to omitting *Accept* header, ignoring other redundant content negotiation headers like *Accept-Charset*, *Accept-Language* and *Accept-Encoding* which would further optimize performance. Besides, the results shown for protocol reduction are not detailed enough. It gives an overall result for TCP and HTTP reduction techniques without differencing the contribution of each individual optimization mechanism. Besides, their approach for header restoring (copying original *Accept* header values) could be improved by using an accurate device capabilities profiling system like the one developed in [9] and expanded herein to work in a collaborative environment.

In contrast, the results shown in [14] for HTTP header reduction are obtained performing GET requests directly from a modified browser software, omitting all HTTP headers, consequently achieving the best optimization results, but this solution is not functional as web browsing is degraded with omission of state, content negotiation, connection management, cache and session related headers. However it takes into account the need for a profiling system to restore device capabilities, although it does not address how such a system could be built.

Furthermore, these proposals do not explore strategies for header restoring in order to enhance and ease content negotiation and adaptation. Herein we propose using intermediaries for transparent managing of HTTP protocol optimization and contextualization functions without modifying browser implementation or configuration. This protocol reduction with posterior header restoring does not exclude using other optimization techniques like the mentioned above to further enhance performance (i.e. delta encoding, compression, extended caching, pipelining, etc).

Regarding device capabilities expression, several proposals exist [6-8]; each of them designed for different environments with different purposes. All the proposed vocabularies and frameworks present certain deficiencies. This way in previous works [9] we developed a system for device capabilities detection with mechanisms for capabilities expression and content negotiation management and tested it as a HTTP proxy. This solution mitigates certain specification and implementation issues of current standard frameworks (mainly Composite Capabilities/Preferences Profile 2.0 –CC/PP and User Agent Profile 2.0 -UAProf). Besides it makes browsers with no tool for capabilities and preferences expression appear as if they were UAProf compliant. An extensive contextualization of the user's devices allows us to present to the Internet, via Wireless Profiled HTTP (W-HTTP) [7], a complete and accurate profile of the device of the user. This profile is expressed using UAProf vocabulary and is compatible with available parsers (i.e. JSR-188, DELI, etc). Therefore, we obtain a single profiling platform for different types of devices (PCs, PDAs, mobile phones, etc) with just one profile vocabulary. Hence, this paper aims to investigate existing synergies among device capabilities expression, delivery context characterization, content adaptation and application layer protocol optimization.

2. System Description

The system general architecture (figure 1) is composed by one gateway (GW) at one end of each link to be optimized working collaboratively with a central Proxy Server (PS) at the wired side of the network. The system is able to work with several GWs from the same client access network and the PS is also able to cope with multiple GWs from different client access networks. Figure 2 illustrates the distribution of modules among each kind of node.

Regarding channel characteristics, any link is susceptible to be optimized whenever request transmission time is the limiter instead of total optimizations processing time –including header reduction at the GW and later header restoring at the PS. It should be noted that enhancement in response time increases with greater asymmetry between uplink and downlink channels.

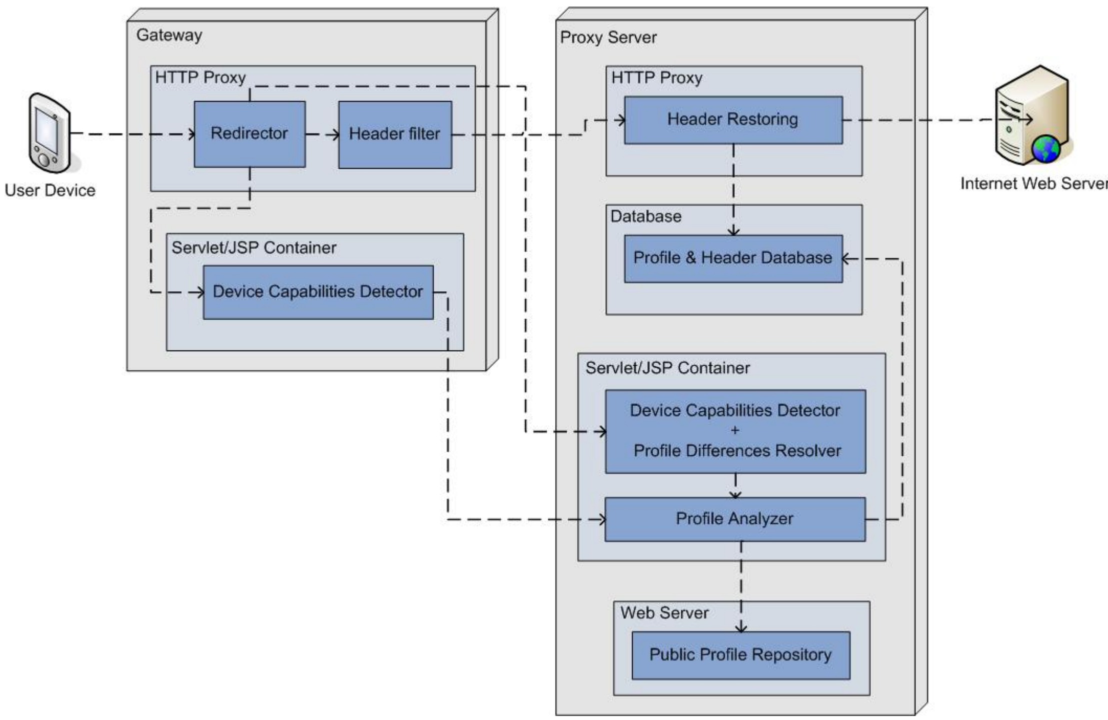

Figure. 2. Deployment diagram

2.1. Gateway modules

1. Redirector: this hijacks HTTP-related connections and manages optimization and detection triggering. If there is an active profile for this user and device, it forwards client's HTTP requests to the header filtering module. If no active profile exists, it triggers the detection service.
2. Header Filtering: HTTP optimizer. It manages the header reduction functions. All the content negotiation headers are removed except by a modified *User-Agent* (with a univocal user and device identifier) and the *Host* headers before the request is forwarded to the PS. Other headers related to management of state, management, cache and session (i.e.: cookies) are not removed.
3. Device Capabilities Detector: it manages the detection processes on this end of the link (from the access network up to the cellular link). The profile obtained is then sent to the PS which analyses and stores it in the repositories.

2.2. Proxy Server modules

1. Header Restoring: module that restores content negotiation HTTP headers before forwarding

requests to web servers on the Internet.

2. Profile&Header Database: internal database which stores the headers to restore for each device, classified by executed analysis policy.

3. Device Capabilities Detector + Profile Differences Resolver: it is in charge of managing detection processes on this end of the link (from the PS up to the Internet). The detection processes of those devices pointing to a profile hosted by a server on the Internet will be managed by this module instead of GW's detection module. The module is able to manage differences between results of the detection processes and content of profiles provided by manufacturers.

4. Profile Analyser: it analyses content of detected profiles and infers header content according to different profile analysis and header restoring policies.

5. Public Profile Repository: It serves device profiles in RDF/XML format (using UAProf as vocabulary) to web servers and other context consumers on the Internet.

3. Header restoring policies

Pervasive and ubiquitous Web access demands content adaptation for successful interaction with web applications and suitable rendering on any kind of device (i.e. PDAs, PCs, portable game consoles, etc.). These devices show very different hardware and software capabilities, even with different interaction modalities from traditional ones. Moreover this ubiquitous access mechanism involves different wireless interfaces and networks; some with serious restrictions in channel capacity, available bandwidth and latency affecting web interaction and content downloading. Hence content adaptation and personalization is critical to obtain a functional user experience in any kind of environment, regardless of mechanism access, and to accomplish the desired author principle of "write once, deliver it anywhere" [2].

W3C defines delivery context [2] as "a set of attributes that characterizes the capabilities of the access mechanism, the preferences of the user and other aspects of the context into which a web page is to be delivered". Following this idea, each entity along the delivery chain should add its own context vision to web requests, so that content adaptation could be performed by content servers and/or adaptation proxies. Ideally, adaptation should be performed or managed by content servers so author's transformation guidelines could be fulfilled without too much burden -enhancing QoE and adapted content fidelity in order to provide harmonized user experiences [16]. In order to reduce load on origin servers, adaptation functions may be distributed on behalf of content servers among proxies and surrogates. Using protocols like Internet Content Adaptation Protocol (ICAP) and Open Pluggable Edge Services (OPES) content servers can control the adaptation processes, requesting proxies for specific adaptation tasks whilst still meeting the quality criteria of the author by following his transformation guidelines .

However, a proxy not controlled by origin servers whether must infer and deduce document structural information and purpose along with author's intentions or the original document must be overloaded with metadata giving author's own content adaptation guidelines. Anyway, overhead increases significantly, performance decreases (extra metadata to interpret) and authors lose some degree of control over adaptation process. Lastly, client side adaptation is inefficient in data traffic volume; document overhead also increases and usually demands too many resources on client device.

Ideally, web content adaptation should be done based on accurate and complete information about current device capabilities, user preferences and network characteristics, dynamically expressed with HTTP content negotiation headers and profiling frameworks like CC/PP and UAProf. At the moment there are other solutions, like WURLF [4], which are based on identifying User Agent strings to extract device capabilities information from a profile database. This approach has several drawbacks or limitations like configuration changes and personalization made on user devices are not reflected in the static information stored in the database. Device information is also limited to a known and limited range of devices and it is difficult to maintain it up to date. Thus, the range of known devices is constrained mainly to a set of mobile phones, excluding other devices like PPC, portable game consoles, TabletPCs, etc. Furthermore, this approach converts a dynamic element, the user context, into a static element, leaving no option to adapt content dynamically to changes in network state and/or device capabilities/configuration. If adaptation functions are allocated in content providers' servers and/or in 3rd-party proxies along the delivery chain, these elements should be able to dynamically adapt content according to device capabilities. Capabilities expressed by the client device itself and complemented by intermediate proxies in such a way that all the entities in the delivery chain contribute with their own vision of the delivery context. Following this idea, the proposed PS/GW collaboration system contributes with a privileged insight into a segment of the delivery chain with severe restrictions in available bandwidth, capacity and latency. Thus, content can be adapted bearing in mind the restrictions imposed by the cellular link.

The need of restoring headers in order to allow a satisfactory browsing experience, with usual content negotiation and adaptation, brings an excellent opportunity to take advantage of synergy between device capabilities detection and application layer optimization in order to provide enhanced capabilities expression and dynamic content negotiation mechanisms. Hence content negotiation information is dynamically modified, adapting context expression to the current network and device characteristics in order to enhance the user's perceived QoE. Different user profile analysis and content negotiation header restoring policies can be designed with different enhancement philosophies and purposes. Thus the detected profile will be dynamically modified according to the restoring policy and it will be announced with W-HTTP. HTTP content negotiation headers will be also modified to reflect these changes.

On one hand, dynamic policies can modify header and profile values according to the state of the network in order to request lighter or heavier multimedia objects to mitigate the effect of network congestion on user browsing experience -i.e. if the network is congested users could prefer to see only text objects in order to speed up downloading time, thus the user profile would be modified to appear as if it only supported text objects and only MIME text objects would be announced in *Accept* HTTP header.

On the other hand, device capabilities information can be enriched in order to enhance content adaptation and personalization with purposes of easing device independence and/or context-awareness of applications. Also, as explained above, some web pages use straightforward adaptation based on *User Agent* string identification. In this case server-driven web content adaptation is constrained to a limited range of profiled devices. So, for small hand-held devices like phones, this range can be increased by means of cloaking *User Agent* strings in order to appear as another mobile device of similar characteristics; one profiled by the web application. Another possibility is reflecting on client requests available adaptation capabilities of intermediate proxies on cellular operator's network, e.g. announcing adaptable MIME types, changing profile values, etc. Therefore web servers with any intelligence can better adjust content to device capabilities. In example, Google serves different search pages for different devices like PCs, PDAs or mobile phones. Furthermore, for mobile phones it also acts as a non-transparent adaptation proxy using URL rewriting techniques, image adaptation, document repagination, eliminating embedded objects, etc. So, knowing this, we can toy with header values in order to obtain different responses according to user's needs -i.e. forcing Google to act as an adaptation proxy by means of cloaking user agent and device capabilities.

One of the implemented policies in the testbed aims to achieve device independence, providing the most detailed information on device capabilities in order to allow web servers and intermediate proxies along the delivery chain adapt content to device and browser capabilities. Thus, content negotiation header values are enriched with all the detected information about the capabilities of the device and W-HTTP *x-wap-profile* header is added referencing to the user's device enriched profile.

The other tested policy is designed to cloak device capabilities depending on available bandwidth in the cellular link. In case of bandwidth reduction, depending on user's navigation preferences, PS modify headers and profile values to cloak device capabilities, profile and User Agent as another device (with more restrictions on size and hardware and software capabilities) in order to request lighter content. A bandwidth estimation system based on Packet-Pair Probes [10] monitors the cellular link on behalf the PS and informs the PS when bandwidth is reduced considerably. This restoration policy was tested against Internet's most popular web page, Google's search page. It was chosen because Google's servers adapt its search page's document structure, style (CSS), dynamism, language, pagination, format and encoding depending on the announced capabilities. Furthermore for mobile phones, Google acts as an adaptation proxy reformatting web contents to fit mobile phone characteristics, using URL rewriting techniques to restructure document anchors and embedded objects links. Thus, we analyzed and tested Google's behaviour to see the degree of context-awareness and adaptability of one of most technologically up to date commercial web applications on the Internet.

3.1. Google Analysis

For the analysis (see table 1), we requested Google's search engine page with different types of devices and models (mainly PCs, PDAs and mobile phone ones); accessing with different browsers for each single hardware platform (i.e. Qtek 9000 was tested with IE MSIE 4.01, Mozilla Minimo 0.2 and Opera Mobile v 8.65) and examining differences between versions of the returning hyper-text document (i.e. format, size, scripting structure, charset, encoding, style, etc) and related multimedia objects (i.e. image size, format, quality and object size). We also searched for changes in application behaviour to see when Google acts as adaptation proxy. For the same model and browser, we also toyed with the existence and values of the different content-negotiation related HTTP headers (*User-Agent*, *Accept*, *Accept-Charset*, *Accept-Encoding* and *Accept-Language*). Furthermore, W-HTTP profile headers (*x-wap-profile* and *x-wap-profile-diff*) and profile values were also taken into account. The procedure followed was:

1. Complete requests (with all content negotiation headers): with *x-wap-profile* header (when needed) and without *x-wap-profile* header
2. Requests with just the *User-Agent* header (with and without other content negotiation headers): well-known *User-Agent* strings and modified, unusual or new *User-Agent* strings
3. *User-Agent* + W-HTTP *x-wap-profile* header (with and without the rest of negotiation headers): well-known *User-Agent* strings and modified, unusual or new *User-Agent* strings. Also, varying *x-wap-profile* to refer to a modified profile and adding *x-wap-profile-diff* to express differences with manufacturer profiles

When a *User-Agent* is not known, Google uses the appearance or not of *x-wap-profile* header and contents of Accept list to infer if it is a mobile device; but in this case it does not act as adaptation proxy. We also found it does not retrieve the referenced profile in the *x-wap-profile* header and ignores *x-wap-profile-diff*; therefore it does not analyze profile information. Consequently, we deduce the classification process is based mainly on User Agent identification against some kind of profile database.

Regarding adaptation process, Google serves different versions of its search engine page depending on capabilities announced by browsers as traditional HTTP content negotiation demands. It delivers two visually different search engine pages, one for mobile or hand-held devices and one for PCs; but each of these visually-equal documents shows considerable differences in behaviour and in its internal foundation depending on type of device. Hence, we discovered that Google classifies devices in three categories: PCs, hand-held devices and mobile phones.

In case of devices without restrictions (i.e. Personal Computers, laptops, UltraMobile PCs, etc), Google adapts its search page to fit to the announced capabilities. So it adapts content to announced encoding (none, gzip, deflate, etc); hyper-text format (i.e. xhtml, html, wml); style (i.e. CSS); scripting (i.e. use of different JavaScript/ECMA functions); image format (i.e. JPEG, GIF); human language and charset (i.e. ISO-8859-1, UTF-8).

In case of mobile and hand-held devices (i.e. PDAs, mobile phones, portable game consoles, etc), Google's search page presentation is also adapted to fit device rendering capabilities, (especially screen size); adapting size and format of images in case of searching for images. However, the scope of the search engine application is expanded for mobile phones, as Google's acts as an adaptation proxy on behalf of mobile devices. Web sites linked through the search results page are transparently proxied by Google using URL rewriting techniques. Thus, contents of the requested site are adapted to hardware and software capabilities of the mobile phone.

This adaptation of 3rd parties' web contents is done in a very straightforward manner, depending on static device capabilities information indexed by *User Agent*, like in WURLF-based applications. Furthermore, adaptation is not transparent to users, as adapted documents clearly state that they are the result of Google's adaptation engine. Users have limited control over the adaptation procedure: they can disable image display and request the original documents. Some detected actions taken by Google's adaptation engine are: HTML is translated into xHTML; documents are repaginated; Embedded objects are removed (even in case of supported MIME object types); and image are resized with changes in format and quality to reduce file size.

TABLE I. SIZE IN BYTES OF GOOGLE SEARCH ENGINE PAGE

Device/Browser	Default headers	User-Agent
Qtek 9000/MSIE 4.01	1875[1]	1875[1]
Qtek 9000/Minimo 0.2	1887[2]	1875[1]
UltraMobile PC (WinXP) /Mozilla 1.8.1	4075[1]	4075[1]
UltraMobile PC (WinXP) /IExplorer 7	4505[1]	4505[1]
UltraMobile PC (Debian 3.0) /Mozilla 1.8.1	4057[1]	4057[1]
UltraMobile PC (WinXP/ Debian 3.0) /Opera 9	3785	3795
Nokia N91/default	1895[2]	1875[1]
Nokia 6329/Opera 8.00	1895[2]	3795[1]
Nokia 6329/ Default[m]	1539[3]	1895
Fake Mobile Phone Browser	1539[3]	3795[1]

1-html 2-xhtml 3-wml m-modified Accept header (wml prioritized)

4. Optimization Results

Tests were executed with the following equipment: Gateway node is a Linux-based laptop (Debian 4.0 distribution with 2.6.18 kernel version) using an Intel® ipw3945 802.11a/b/g chips. The cellular link is a HSDPA link, approximately 1 Mbps downlink and 384 Kbps uplink. The client access Network is 802.11.b and the client node is a Windows XP-based UltraMobilePC (Mozilla 1.7.12). The PS node is a Linux OS Debian 4.0 distribution with 2.6.18 kernel version (PC with Intel® P4 3GHz dual core processor and 2 GB of RAM memory). Besides, in order to avoid the effect of public server congestion, we deployed a virtual web hosting system [15] at the laboratory network mirroring popular web pages. With this system, clients still access to these pages through the Internet but servers work load is under control. Virtual Servers have the same hardware and OS as PS.

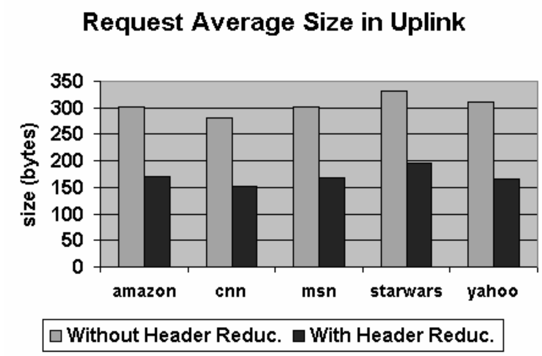

Figure 3. Request average size in uplink

Figure 3 shows the obtained reduction in requests size for different web pages. The differences in client's request size among the different pages depend on the implementation of the requested web application. This is due to the appearance of cookies and similar dynamic headers and not due to changes in content negotiation headers, which are static. The header reduction mostly reduces request size by removing these static headers. These headers obviously vary between different browsers and they even vary between two browsers of the same family and version, depending on the installed plug-ins.

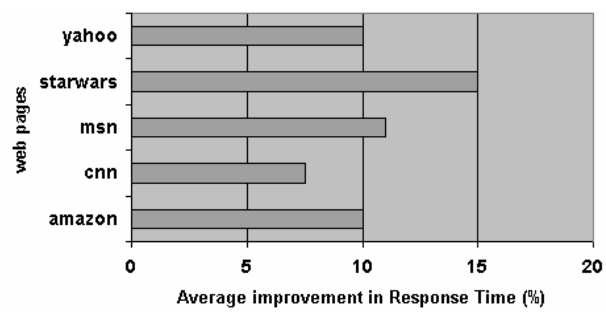

Figure 4. Average improvement in Response Time (%)

Figure 4 shows the average improvement in response time for the same pages. This improvement is about 10%. Differences between improvements in response times depends mainly on the number of objects, regardless their type. More objects to request, more requests to optimize and improvement increases. However, it is demonstrated [5] that the downloading order of the objects of a web page affects the overall response time, it is better to interleave small objects with large objects to keep the server transmitting instead of waiting. So differences in web document structure affect the downloading order and, thus, the size of consecutive objects also affects the response time in conjunction with the number of objects composing the web page.

5. Conclusion

The usage of optimization techniques based on request header reduction at HTTP layer brings an excellent opportunity to deploy sophisticated device characterization and contextualization solutions that allow not only speeding up web downloading but also enriching content negotiation in order to enhance adaptation; achieving a more ubiquitous web access and improving performance by downloading adapted (lighter) web pages. The GW/PS collaborative interception model allows PS to toy with the content of HTTP headers. Hence, this work has shown how PS can modify header and profile values according with its own privileged vision of the device capabilities, access network state and cellular link condition. Consequently, it has been discussed how to dynamically toy with content negotiation HTTP header values and profile content in order to modify web application behaviour and response in order to better adapt to user's device capabilities and delivery context characteristics. So that this work has explained how to take advantage of synergies among content negotiation, device capabilities expression, delivery context characterization, content adaptation and application layer protocol optimization; demonstrating how to build a collaborative system to improve performance, ubiquity and QoE in links with restrictions.

6. References

L. Suryanarayana and Johan Hjelm. "Profiles for the Situated Web". *ACM 1-58113-449-5/02/0005*. WWW 2002, May 7-11, 2002.

R. Gimson , R. Lewis and S. Sathish.. "Delivery Context Overview for Device Independence", *W3C Working Group Note*, 20 March 2006

C. Canali, M. Colajanni and R. Lancellotti. "Distribution of adaptation services for ubiquitous Web access driven by user profiles", *Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC'06)*

Luca Passani. "Wireless Universal Resource File (WURFL)". http://wurfl.sourceforge.net/

C. Gomez , M. Catalan, D. Viamonte, J. Paradells, A. Calveras "Web browsing optimization over 2.5G and 3G: end-to-end mechanisms vs. usage of performance enhancing proxies". Wireless Communications and Mobile Computing. 12/9/2006

Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 2.0 W3C Working Draft 30 April 2007. http://www.w3.org/TR/CCPP-struct-vocab2/

User Agent Profile 2.0 . Wireless Application Protocol WAP-248-UAPROF-20011020-a http://www.openmobilealliance.org/tech/affiliates/wap/wap-248-uaprof-20011020-a.pdf

T. Lemlouma. Architecture de Négociation et d'Adaptation de Services Multimédia dans des Environnements Hétérogènes.. PhD thesis, 9/5/2004. http://wam.inrialpes.fr/publications/2004/TheseLemlouma.pdf

X. Sanchez-Loro, J. Casademont, J.L. Ferrer, and J. Paradells. "A Proxy-based Solution for Device Capabilities Detection". Proceeding (558) Internet and Multimedia Systems and Applications - 2007. Chamonix, 14 March 2007

R. Prasad, C. Dovrolis, M. Murray and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools", IEEE Network Magazine, Nov/Dec 2003

J. Mogul, F. Douglis, A. Feldmann, Y. Goland, A. van Hoff and Marimba D. Hellerstein. "Delta encoding in HTTP". RFC 3229

J. Border, M. Kojo, J. Griner, G. Montenegro and Z. Shelby. "Performance enhancing proxies intended to mitigate link-related degradations". *RFC 3135*, June 2001.

B. C Housel, G. Samaras and D. B. Lindquist. "WebExpress: a client/intercept based system for optimizing Web browsing in a wireless environment", *Mob. Netw. Appl.*, 3 (1999), 419-431.

M. Catalan, C. Gomez, P. Plans, J. Paradells, A. Calveras, J. Rubio and D. Almodóvar. "Extending Wireless Mesh Networks over UMTS: A proxy-based approach". *Qshine'06*, August 7–9, 2006, University of Waterloo, Ontario, Canada.

R. Chakravorty, A. Clark, I. Pratt. "Optimizing web delivery overwireless links: design, implementation and experiences". IEEE. Journal on Selected Areas in Communications 2005; 23(2): 402.

K. Smith. "Device Independent Authoring Language (DIAL)" W3C Working Draft 27 July 2007. http://www.w3.org/TR/dial/

Authors

Xavier Sanchez Loro received his M.S. degree in Telecommunications Engineering from Technical University of Catalonia (UPC), Barcelona, Spain, in 2005. He is currently pursuing his PH.D. at the Department of Telematics Engineering, at the mentioned university. His research interests include device and service characterization and context acquisition in context-aware and ubiquitous computing.

Victoria Beltrán Martínez received a degree in computer engineering from the University of Murcia, Spain, 2005. She is currently a PhD candidate in the Department of Telematics Engineering of the Technical University of Catalonia (UPC), Spain. Her research interests include context management, context-aware applications and presence-based systems.

Jordi Casademont is an Assistant Professor at the Technical University of Catalonia (UPC) in Barcelona. He received his M.S. Degree in Telecommunications from the UPC in 1992 and the Ph.D. Degree in 1998. He is an active member of the Wireless Networks Group (WNG) at the Telematics Department of the UPC and has published papers in the fields of networking and medium access control. His main research interests are the design and evaluaation of MAC mechanisms and mesh networks. He is actively involved in national and European projects.

Marisa Catalan Cid received her M.S. degree in Telecommunications Engineering from Technical University of Catalonia (UPC), Barcelona, Spain, in 2003. In the same year she joined the Wireless Networks Group (WNG). She is currently pursuing her PH.D. at the Department of Telematics Engineering, at the mentioned university. Her research interests are the performance of wireless communications and the optimization of Internet protocols in 2.5G, 3G and wireless multi-hop networks.