University of Helsinki
Department of Computer Science
Series of Publications C, No. C-1996-28

Enhanced Services for World-Wide Web in Mobile WAN Environment

Mika Liljeberg, Heikki Helin, Markku Kojo, and Kimmo Raatikainen

Helsinki, April 1996

Report C-1996-28

University of Helsinki
Department of Computer Science
P. O. Box 26 (Teollisuuskatu 23)
FIN-00014 University of Helsinki, FINLAND

Enhanced Services for World-Wide Web in Mobile WAN Environment

Mika Liljeberg, Heikki Helin, Markku Kojo, and Kimmo Raatikainen

Department of Computer Science, University of Helsinki
Report C-1996-28
April 1996
12 pages

Abstract. **Abstract.** Currently, the World-Wide Web (WWW) is the best-known and most widely used distributed multimedia application. By utilising the data services of a cellular telephone system, for example the digital GSM system, WWW can be brought to nomadic users. The advantage of using an existing cellular network like the GSM is that it takes care of terminal mobility, including location updates and handovers. The characteristics of cellular telephone links are, however, quite different from those of wireline links. The basic characteristics of any cellular telephone link are narrow bandwidth, highly variable transmission delays, and sudden disconnections. Therefore, such a link creates problems for WWW and other multimedia applications designed to operate with fast and reliable connections. WWW uses the HyperText Transfer Protocol (HTTP), which in turn is based on TCP. These protocols exhibit a number of performance problems in a wide-area mobile networks. In this paper we describe Mowgli WWW, a WWW middleware implementation that demonstrates several techniques for solving these problems. We report our experiences with Mowgli WWW and give measurement results showing radically improved performance.

Key Words: World-Wide Web, mobility, wireless connections, cellular telephone

CR Classification: C.2.1,C.2.2,D.4.8,H4.3,H.5.1,I.7.2

Contents

1 Introduction

Recent developments in mobile communication and personal computer technology have created a new and challenging platform for mobile information processing. Today the application of cellular telephone networks, such as the Global System for Mobile Communications (GSM) [MP92, Rah93], for data transfer between portable personal computers and traditional fixed data communication networks already makes mobile computing possible. However, cellular telephone links have a low and variable throughput. They are also prone to sudden disconnections and are often expensive to use. These characteristics create problems for many existing applications.

Information retrieval is predicted to be one of the most important applications for future mobile workstations. The World-Wide Web (WWW) [BL+94] is a globally used distributed collaborative information system that has a rapidly increasing user community. Hence, there is real interest for bringing WWW to mobile workstations. However, the World-Wide Web suffers from a number of performance problems that become especially severe coupled with those in the wireless WAN environment. These problems need to be solved to make WWW a practical application on a weakly connected mobile workstation.

In [KRA94] we introduced the Mowgli[1] communication architecture designed to solve the problems inherent in the specific features of cellular telephones. The key idea in the Mowgli communication architecture is to split the end-to-end connection into two separate parts: one over the fixed network, the other over the wireless network. This enables each part of the connection to be treated in a way that is well suited to the underlying communication media. In wireless LAN environment, similar approaches have successfully been used to improve the performance of TCP-based communication (See [BB95, YB94, BSAK95]. However, our approach is more general and allows improvements at the higher levels of communication, because it also supports application level mediators that are transparent to clients and servers. We also take into account the specific characteristics of the low-bandwidth wireless links by replacing the TCP/IP with a special transport service which we call Mowgli Data Channel Service (MDCS) [KKLR96].

In [LAK+95] we introduced several ideas for improving the performance and usability of World-Wide Web browsing in wireless WAN environment. In this paper we describe a set of enhanced services supporting WWW, implemented as Mowgli WWW Agent, Mowgli WWW Proxy, and Mowgli Data Channel Service. Due to the agent-proxy approach of Mowgli neither WWW browsers nor servers need to be modified. The most important features of Mowgli WWW include: 1) more efficient protocols over the wireless medium, 2) intelligent reduction of transmitted data, 3) background transfers reducing the burstiness of traffic, and 4) disconnected-mode support in the form of versatile user control over caching and cellular call setup.

The rest of paper is organised as follows. In section 2 we discuss the problems WWW browsing meets in wireless environments. In Section 3 we outline the essential features of the Mowgli communication architecture. In Section 4 we discuss the central ideas behind Mowgli WWW. In Section 5 we give an overview of the implementation of Mowgli WWW, and in Section 6 we present the results of our performance measurements.

[1] Mowgli is an acronym of the project name *Mobile Office Workstations using GSM Links.*

2 WWW in Wide-Area Wireless Environments

The Hypertext Transfer Protocol (HTTP) used in the World-Wide Web suffers from severe performance problems in high-latency environments such as wide-area networks [PM95, LAK+95]. The problems arise, because the HTTP protocol requires an excessive number of round-trips between client and server: each retrieved hypertext document involves the creation and deletion of several TCP-connections, one for each embedded document object. The connections are typically very short in duration, which implies that all the overhead associated with creating a new connection effectively ruins the overall performance in high-latency environments. In particular, the three-way handshake and the TCP slow-start algorithm [Jac88] always employed at the start of every connection, separately for each direction, generates a lot of round-trips.

The performance problems become especially severe in wide-area wireless environments where the typically high latency is coupled with low bandwidth. For instance, in the data service provided by the pan-European GSM network [MP92] the observed round-trip times approach one second while the nominal bandwidth is only 9.6 kilobits per second [AKL+94]. The smaller TCP segment sizes typical in low-bandwidth environments aggravate the problem, because even small requests and responses bring out the full effect of the TCP slow-start algorithm at the start of each TCP connection. The performance is further degraded, because the retransmission and flow-control algorithms used in the TCP protocol itself perform quite badly in wireless environments, particularly when multiple connections over a constrained link interfere with each other [KKLR96].

The W3 Consortium is already working on a faster HTTP protocol [Con95]. Unfortunately, an efficient protocol alone will not make WWW a practical application in low-bandwidth environments. While WWW documents are generally fairly small, they can sometimes also be extremely large. In particular, many documents contain pictures and other attachments, the value of which is often low aside from aesthetic import. These attachments can take prohibitively long to transfer to the mobile node and the user has little control over what is transferred and what is not. In addition, wireless links suffer from sudden disconnections and are often expensive to use. Thus, new fault-tolerance features, improved control over the link usage and support for disconnected-mode operation is needed.

A further problem is that WWW browsing is typically a highly interactive activity: short periods of active data transfer are interleaved with long periods of inactivity when the user reads the latest retrieved document. Thus, WWW browsing over cellular telephone networks can be expensive recreation for both the end-user, who also has to pay for the idle periods, as well as the teleoperator, who has to buy the network capacity to support all the subscribers even though the hardware is really idle much of the time.

3 The Mowgli Communication Architecture

The Mowgli communication architecture is designed to increase the usability, reliability, and efficiency of client-server communication between *Mobile Nodes* and fixed hosts. The basic idea is to split a channel with end-to-end control into two parts by using a store-and-forward -type

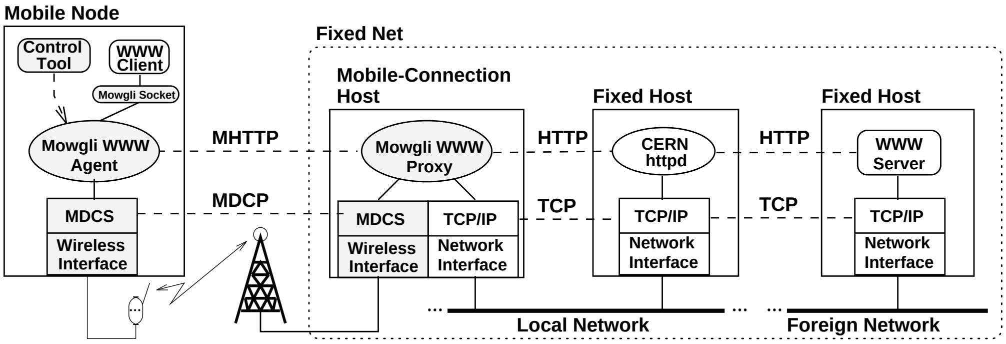

Figure 1: Overview of Mowgli WWW

interceptor. This interceptor, the *Mobile-Connection Host* (MCH), allows us to replace the client-server paradigm with a client-mediator-server paradigm and to implement two separate but flexibly cooperating data communication subsystems: one wireline oriented, the other wireless oriented (Figure 1).

In our architecture existing TCP/IP protocols are used between the MCH and fixed hosts so that the protocol software in the fixed net remains unmodified. However, reliable communication between the MCH and the mobile node is provided by a specialised transport service, the Mowgli Data Channel Service (MDCS). Two mediators, the *Mowgli Agent* and the *Mowgli Proxy*, which reside at each end of the wireless link, are used to provide functionality similar to that of TCP and UDP.

Applications on the MN connect to the agent through the Mowgli Socket Interface which is downwards compatible with the traditional BSD sockets. In their simplest form, the agent and the proxy serve as relays for the intervening traffic. However, by introducing specialised agents and proxies we enable the implementation of higher-level functionality: the agent and the proxy can optimise application-level protocols for the wireless environment by intercepting the appropriate messages.

The Mowgli Data Channel Service (MDCS) provides a flexible set of communication services to MDCS-clients: agents and proxies as well as future mobile node applications designed for the wireless environment. As a basic service, the MDCS offers bidirectional data channels for data transfer over the wireless link. Data channels can be opened as *stream channels* providing TCP-like functionality or as *message channels* if UDP-like datagram service is desired.

Each channel has independent flow control and its own attributes that control the behaviour of the channel. Data channels are multiplexed according to channel priorities. Priority-based scheduling allows the MDCS to provide reasonable response times to interactive applications even if there are bandwidth-intensive background transfers, for example file transfers, using up the link capacity.

Another convenient feature of the MDCS is improved fault-tolerance. The MDCS is designed to recover efficiently from unexpected temporary disconnections, which are quite common in cellular environments. The MDCS maintains the state information of the ongoing transmission over each channel so that an interrupted transmission can later be resumed. Each data channel has its own attributes that control the behavior of the channel.

4 Key Ideas in Mowgli WWW

The Mowgli WWW Agent and Proxy (Figure 1), located the ends of the wireless link, cooperate to fetch hypermedia documents from fixed servers to the mobile node. Both the agent and the proxy are completely invisible to WWW clients and WWW servers. They communicate with each other using the highly optimised Mowgli HTTP (MHTTP) protocol. Together, the agent and proxy optimise the data traffic over the wireless part of the network for maximum performance. By improving the performance of the bottle-neck resource we improve the performance of the whole system as a whole.

The Mowgli WWW Agent and Proxy establish a session binding when the user starts a browsing session. The binding is maintained until the user declares that the browsing session is over. The binding is maintained regardless of the current state of the wireless link, i.e. the Mowgli WWW Proxy retains awareness of the user even though his or her mobile workstation may at times be disconnected from the network. In this way the proxy can perform operations towards the fixed network on behalf of the mobile node even if it is momentarily unreachable. The persistent binding also avoids lengthy renegotiation when the mobile node again becomes available.

Mowgli WWW design is based on the following key principles:

1) Discourage burstiness on all levels. Bursty data traffic translates to low utilisation of the often expensive data link.
2) Minimise the volume of data transmitted over the wireless link. Volume of data translates directly to transfer time.
3) Allow operation without network connectivity.
4) Modifications to WWW browsers and WWW servers should not be required so that existing popular software could still be used.
5) Provide the user fine-grained control over how the wireless link is used.

The following sections describe in more detail, how these principles have been realised in Mowgli WWW.

4.1 Minimising Protocol Overhead

As described in section 2 protocol overhead occurs on two levels: on the transport level and on the application level. On the transport level the Mowgli architecture replaces the TCP protocol over the wireless part of the network with the Mowgli Data Channel Protocol (MDCP). MDCP is a light-weight protocol that has minimal packet headers and involves as few round-trips over the wireless link as possible. In particular, MDCP channels have an extremely low setup overhead and always transfer data at the maximum bandwidth available on the wireless link.

On the application level, Mowgli WWW replaces the normal HTTP protocol over the wireless part of the network with MHTTP. Unlike the human-readable HTTP the MHTTP is a binary encoded protocol. MHTTP protocol headers take up roughly one fifth of the space required by their equivalents in HTTP. This is not an insignificant improvement. However, the true power of the MHTTP protocol lies in that it allows the *predictive upload* of documents and document objects. This means that if the Mowgli WWW Proxy predicts a certain document object is going to be needed soon, it can upload the object to the mobile node without first receiving a specific request from the agent.

The proxy makes extensive use of the predictive upload facility by prefetching document objects embedded in WWW documents. When the proxy receives a request for a WWW document, it fetches the document from a server in the fixed network and forwards it to the client. While doing this, the proxy looks at the document and starts prefetching the embedded objects from the fixed server and uploading them to the mobile node with the assumption that they are going to be requested soon in any case. This technique significantly reduces the response times observed by the end-user when fetching typical WWW documents.

4.2 Reducing the Transfer Volume

Mowgli WWW provides three primary ways to reduce the transfer volume over the wireless link: data compression, caching, and intelligent filtering.

Mowgli WWW supports content-type specific as well as generic data compression. Content-type specific compression means that each particular document type (e.g. text, image data, audio data) can be assigned a different compression algorithm that performs best on that type of data. Generic compression is applied to all document objects that have not been assigned a content-type specific compression method. The Mowgli WWW Proxy compresses document objects while sending them over the wireless link. The Mowgli WWW Agent then decompresses them as necessary before forwarding them to the WWW browser. Data compression is not attempted on document objects that are already in compressed format or on small objects that are unlikely to benefit much from compression.

We chose the *splay-prefix* algorithm [Jon88] as the generic compression algorithm, because it is fast, simple and dynamic (i.e. it can compress a stream of data on a byte-by-byte basis) and thus well suited to data communication applications. Currently, the only supported content-type specific algorithm is GIF to JPEG conversion, which strictly is not a data compression algorithm but provides space and time savings of roughly a factor of five, depending on desired image quality. GIF to JPEG conversion gives very good results, because the JPEG format uses a lossy compression algorithm that sacrifices colour information in favour of smaller size. Because the display resolution and quality in mobile terminals is generally much worse than in desktop workstations, lossy compression does not incur as much penalty as one might think; significant compression can be achieved before the loss of image quality begins to disturb the user. However, in Mowgli WWW the user retains the opportunity to enable and disable compression methods at will and to adjust the loss of information in the compression.

Intelligent filtering is a more drastic form of lossy compression. One example of this is the choice, offered by many current WWW browsers, of downloading documents without images. In Mowgli WWW the user is given the opportunity to specify a maximum size for retrieved text documents as well as embedded images. This is a very useful feature in a low-bandwidth environment, because the size distribution of WWW documents is heavy-tailed [CB95], i.e. while most document objects are fairly small in size a small percentage of them are quite large. By specifying maximum sizes for document objects the user effectively "cuts the tail" of the distribution and gives the system an upper limit for how long a document retrieval may take.

When the Mowgli WWW Agent detects an incoming text document that is larger than the user-specified size limit, it notifies the user giving him or her the opportunity to abort the transfer immediately. While the user is deciding, the document is already being transferred and displayed. If it looks interesting the user may allow the transfer to proceed. Embedded images that exceed the size limit are not transferred to the mobile node. The Mowgli WWW Agent replaces them with a generic image to inform the user that the images were not received.

The Mowgli WWW Agent caches documents on the mobile node to avoid needlessly retrieving them over the slow link. While most WWW browsers also cache documents, the cache in the agent has persistence characteristics and advanced cache management features not generally available in browsers. These features are discussed in more detail in section 4.4.

4.3 Background Operations

Mowgli WWW implements a convenient *background transfer* feature. For instance, while still reading the current document, the user can instruct the Mowgli WWW Agent to start fetching interesting looking referred documents on the background. When the user has finished reading the current document, it is likely that the next one will already be present. Thus, the apparent wait times between documents become negligible. Background transfer is also very useful, when downloading files. File transfers can be started as background transfers, while the user goes on reading and finding more files to download.

To help manage multiple background transfers Mowgli WWW provides a *batch transfer queue*. Started background transfers go into the batch transfer queue, which proceeds to fetch them one by one. The user can look at the jobs in the queue, start and stop the queue at will, delete an individual job or move one up in the queue. The queue can also be started at a preset time, for instance at a time when the telephone calls are cheaper.

4.4 Disconnected-Mode Support

The uncertainty inherent in wireless communications raises the requirement that networking software must be prepared for occasional loss of connectivity. In the cellular telephone environment it may take minutes before a link can be successfully re-established after a disconnection. During this time period the end-user will want to go on working instead of having to wait for the return of connectivity. Thus, mobile software must be able to function in *disconnected-mode*. An even more compelling reason for supporting disconnected-mode operation is that the cost of maintaining a cellular telephone connection can be high.

The caching system in Mowgli WWW supports disconnected-mode in various ways. For instance, current WWW browsers typically check the validity of each document every time it is accessed. Since this type of validation requires a connection to the WWW server, it cannot be employed in disconnected-mode. In Mowgli WWW we chose to validate documents only when explicitly requested by the user. Thus, connectivity is not required to browse cached documents. The same

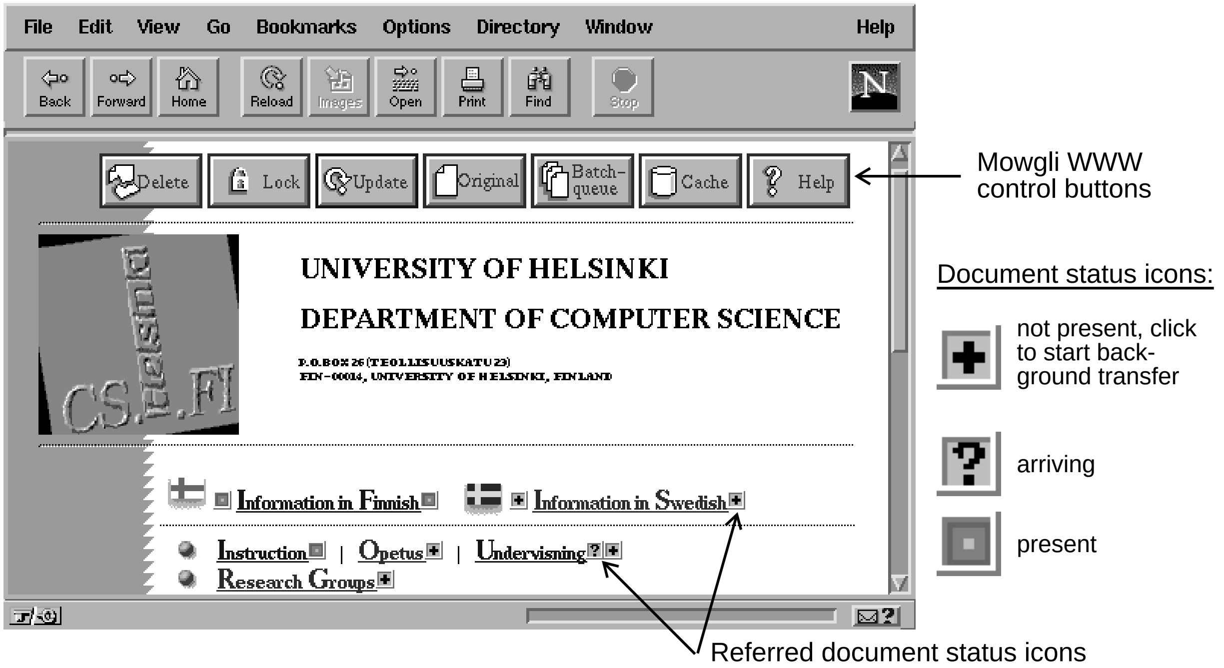

Figure 2: Mowgli WWW and Netscape

policy is also followed in connected-mode, because document validation over a high-latency link would still cause a noticeable delay.

Helping the user avoid documents that are not cached in the mobile node is an essential part of disconnected-mode support in Mowgli WWW. A user inadvertently trying to access such a document would cause a cache miss, which in turn would force the system to try to establish the wireless link and enter connected-mode. Mowgli WWW suffixes each hypertext link in a document with an indicator that tells the user if the referred document would have to be fetched from a server. In addition, Mowgli WWW offers the user access to cached documents directly through a cache inventory. In this way the user can stay safely within the bounds of the cache.

Transitions between connected-mode and disconnected-mode can be configured to occur automatically without user intervention, to require confirmation from the user, or to occur only when explicitly requested by the user. Link disconnection can also be based on a timeout; the link will be disconnected when it has been idle for a certain time period. It can later be automatically re-established when needed. Thus, the user has flexible control over how and when the wireless link is used.

The cache maintained by Mowgli WWW Agent is *persistent*, which means that it will be stored on disk and retained over multiple browsing sessions. Because mobile nodes are likely to run out of power, a mere memory cache would frequently get erased and all the important documents would need to be fetched again. The user can protect useful documents from automatic deletion so that they will not be removed when the cache manager makes space for a new document. The user can also delete unnecessary documents explicitly.

4.5 User Interface

One of the main principles in the Mowgli architecture is that existing applications should not need to be modified. Mowgli WWW has been designed in accordance with this principle. The user interface has been separated into a *Control Tool* that directs the overall operation of Mowgli WWW and a per-document user interface that contains settings and operations that apply to individual documents. The Control Tool is a separate program that is launched in conjunction with a conventional WWW browser. The per-document interface is implemented as set of WWW pages generated dynamically by the Mowgli WWW Agent. Thus, it can be used directly through the WWW browser.

The Control Tool allows the user to modify various configuration parameters. The parameters have been divided into five subsets. In the *document settings* subset the user can decide, if embedded images and/or background images should be fetched along with the document, set the size limits for document objects, etc. The *conversion settings* subset contains parameters related to data compression and presentation format conversions. The *link settings* subset controls when the wireless link can be automatically established and when it can be disconnected, when permission from the user is required, and how long to try the link setup before reporting failure. The *cache settings* subset contains parameters related to the local cache, e.g. maximum cache size. Finally, *batch queue settings* subset allows the user to start, stop or set the start time for the batch transfer queue.

The Mowgli WWW Agent modifies each document displayed to the user (Figure 2). A row of push-buttons is added to the top of the documents. The buttons let the user lock the currently viewed document in the cache, unlock it, delete it from the cache, update the document, view the document in its original form, go to the cache and batch transfer queue status pages, and to get help on using Mowgli WWW. In addition, the agent adds a small icon after each hypertext link in the document. A plus sign indicates that the document has not been fetched. The user can add the link to the batch transfer queue by clicking the sign with the mouse. A green square indicates that the document is present in the cache and can be viewed in disconnected-mode. A question mark indicates the document is currently being fetched. By clicking on the latter two icons the user gets more detailed information about the referred document.

The cache status page displays a list of all documents cached on the mobile node as well as cache usage statistics. The user can view and manipulate individual documents, e.g. delete, lock, or unlock documents. The batch transfer queue page displays all pending, incoming and finished batch transfers. The user can view the retrieved pages directly or go back to the parent page and continue from there.

5 Implementation Issues

The Mowgli WWW software consists of the following basic components (Figure 1): Mowgli WWW Agent, Mowgli WWW Proxy, Control Tool, and CERN httpd[2]. The agent, proxy and control

[2]CERN httpd is a free caching WWW proxy server implementation graciously made available by the W3 Consortium. The server can be downloaded via FTP from *ftp.w3.org:/pub/httpd/*.

tool have been built on top of the Mowgli Software Development Platform, including Mowgli Data Channel Service, Mowgli Event Library, and Mowgli Generic Cache.

The Mowgli HTTP protocol implementation relies heavily on the MDCS. Separate channels are allocated for different purposes. The control traffic between the Mowgli WWW Agent and Proxy (e.g. changes in user settings, requests to abort document fetches, cache synchronisation messages) goes through a special *control channel*. For each new document request the agent creates a new *document channel* over which it sends the request to the proxy. The proxy sends the response over the same document channel and then closes it. For each embedded document object (e.g. inline image), the proxy creates a new *embedded object channel* to the agent, sends the object and then closes the channel. The agent receives and processes the embedded objects as they arrive. In addition, background transfers have their own *batch transfer channels*.

While using multiple connections is anathema with TCP/IP, in the Mowgli system it is possible because the MDCS channels are extremely light-weight; creating a channel, writing a message and closing the channel incurs very little overhead. The advantages gained are simple program logic and the use of transfer priorities. In Mowgli WWW the control channel has the highest priority, then come the documents followed by document objects transferred on the foreground, and finally the batch transfer channels have the lowest priority. The priorities are pre-emptive; the highest priority channel with data to send always gets link capacity. The priority system ensures that the user always gets prompt response even though there are on-going background transfers.

Caching in Mowgli WWW is organised as follows. The Mowgli WWW Agent maintains a cache that is persistent over multiple browsing sessions. There is a separate Mowgli WWW Proxy for each mobile node. It does not cache documents, but maintains an index of the embedded document objects the agent has in its cache. The proxy uses the index to decide if a particular object needs to be prefetched and forwarded to the agent. The index is persistent over a single browsing session, including periods of disconnected-mode operation. Each Mowgli Proxy fetches documents through the CERN httpd, which serves as a site-wide cache. It also functions as a gateway to other information servers, e.g FTP, WAIS, Gopher. In Figure 1 CERN httpd is shown as an independent server. It might as easily be installed as a dedicated proxy on the Mobile-Connection Host.

Since the cache in a mobile node can change during disconnected mode (e.g. a user deletes documents) the cache index maintained by the WWW Proxy needs to be synchronised, when the mobile node comes back on-line. During the synchronisation phase the agent sends update messages to the proxy detailing the changes in cache state since the last connection. It may also be the case that the mobile node has been connected to a different MCH (and thus, a different proxy). The agent and proxy automatically detect this and perform a full cache synchronisation rebuilding the cache index in the proxy.

6 Performance Results

In the experiments, the mobile nodes connected to the fixed net through the GSM non-transparent data service[3] (bearer service with the RLP protocol) and PSTN with modems. The dialup servers

[3] In Finland GSM data service is commercially available and provided by Telecom Finland.

Table 1: Test page sizes (bytes)

Page	HTML	Images	Total
1	512	$2 * 384 + 1024 + 2048$	4352
2	2048	$4 * 384 + 2 * 2048 + 10240$	17920

Table 2: Measured Response Times

Response Times (seconds) for Test Page 1										
Test	MIN	5%	25%	MED	75%	95%	MAX	MEAN	IQR	STD DEV
Normal WWW/LAN	13.07	13.17	13.30	13.43	13.83	14.20	14.32	13.56	0.53	0.35
Mowgli WWW/LAN	6.18	6.19	6.21	6.26	6.35	6.93	7.45	6.38	0.15	0.28
Normal WWW/WAN	10.24	10.69	11.86	14.56	21.75	36.97	226.41	22.14	9.89	30.71
Mowgli WWW/WAN	6.59	6.61	7.19	9.43	13.85	28.27	111.50	13.99	6.67	15.46

Response Times (seconds) for Test Page 2										
Test	MIN	5%	25%	MED	75%	95%	MAX	MEAN	IQR	STD DEV
Normal WWW/LAN	30.76	30.89	31.16	31.28	31.66	33.03	33.43	31.53	0.49	0.65
Mowgli WWW/LAN	20.85	20.86	21.06	21.30	21.57	22.18	22.86	21.39	0.51	0.46
Normal WWW/WAN	26.35	28.39	36.53	41.29	58.10	143.79	160.72	55.45	21.58	33.56
Mowgli WWW/WAN	21.08	21.25	21.97	22.88	27.86	48.65	168.74	29.38	5.89	21.69

(Mowgli MCH and a PPP server) were located together at the University of Helsinki Computer Science Department in a private LAN, separated from the rest of the campus network by a router. In addition, a fixed host located at the University of Waterloo in Toronto, Canada, was accessed through the Internet.

We compared the performance of Mowgli HTTP (MHTTP) to traditional HTTP over TCP and PPP in two cases: with the WWW server located at the same local-area network with the dialup server, and with the WWW server located in a wide-area network (Internet). We conducted the measurements of HTTP and MHTTP simultaneously to achieve as comparable an environment as possible. The experiments were conducted in a normal office environment with good conditions on the GSM radio link. We timed the WAN experiments to a relatively uncrowded period to minimise the possible error sources within the Internet.

As workload we used two different WWW pages (see Table 1), which we believe to be fairly typical. Data compression and image conversions were not enabled as we wished to measure the performance of the protocol itself. For each test environment we conducted 100 WWW page retrievals while alternating between the two test pages. Our test client was configured to emulate the popular Netscape Navigator[4] WWW browser, using up to four concurrent TCP channels to fetch document objects.

The results of our measurements are summarised in Table 2. The table contains the means of transfer times together with the minima, medians, and maxima. We also give the quartiles and the

[4]Netscape and Netscape Navigator are trademarks of Netscape Communications Corporation.

interquartile range (IQR), which is the difference between the quartiles. Because the commonly reported mean and standard deviation are quite sensitive to exceptionally large observations, we prefer to draw our conclusions from the median and the IQR.

Table 2 shows that MHTTP outperforms HTTP in both the LAN and WAN cases. MHTTP retrieved our test pages roughly 1.5–2 times faster than HTTP. With the smaller test page, most of the improvement came from the elimination of connection setup overhead (TCP three-way handshake and initial slow-start). With the larger test page, the end-to-end error correction of TCP caused problems for HTTP, triggering additional TCP slow-starts in the WAN case. MHTTP fared better, because the packet loss within the fixed network does not disturb the transmission over the wireless link. MHTTP is more stable in this respect, as shown by the smaller IQR values.

7 Summary

We have described Mowgli WWW, a novel way to bring the World-Wide Web service to the mobile user. We have discussed the problems in using WWW over a slow and unreliable link and presented solutions that meet the needs of mobile users. Our solutions are aimed at better performance and better control of WWW browsing for the user. We have discussed our Mowgli WWW implementation and presented measurement results showing significant improvements in performance.

Although our agent-proxy approach makes it possible to exploit enhanced services in old applications, we expect that new clients aware of mobile communication will eventually come to the market. Although we have concentrated on the needs of the mobile user, similar problems also occur within wide-area networks. Our concepts and ideas are directly adaptable to the fixed network environment with equal benefit to the user.

Acknowledgements

This work was carried out as a part of the Mowgli research project supervised by Prof. M. Tienari and funded by Digital Equipment Corporation, Nokia Mobile Phones, Nokia Telecommunications, and Telecom Finland. The authors are thankful to Prof. J. Black at University of Waterloo, Toronto, Canada, for allowing us to use one of their machines in our WAN measurements. We are indebted to the rest of the Mowgli team, particularly Timo Alanko, Petteri Kaskenpalo, Jani Kiiskinen, and Heimo Laamanen, for their fruitful comments during the preparation of this paper. Finally, our heart felt thanks to the MWWWV95 student group on the Summer 1995 Software Engineering Course who did a great job of implementing Mowgli WWW.

References

[AKL+94] T. Alanko, M. Kojo, H. Laamanen, M. Liljeberg, M. Moilanen, and K. Raatikainen. Measured Performance of Data Transmission over Cellular Telephone Networks. *Computer Communications Review*, 24(5):24–44, October 1994.

[BB95] A. Bakre and B.R. Badrinath. I-TCP: Indirect TCP for Mobile Hosts. In *Proc. IEEE 15th International Conference on Distributed Computer Systems*, Vancouver, Canada, May 1995.

[BL+94] T. Berners-Lee et al. The World-Wide Web. *Communications of the ACM*, 37(8):76–82, August 1994.

[BSAK95] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz. Improving TCP/IP Performance over Wireless Networks. In *Proc. of the First ACM International Conf. on Mobile Computing and Networking (MobiCom '95)*, Berkeley, California, USA, November 1995.

[CB95] M. Crovella and A. Bestavros. Explaining World Wide Web Traffic Self-Similarity. Techical Report 95–15, Boston University Computer Science Deparment, August 1995. To appear in ACM SIGMETRICS'96, Philadelphia, USA.

[Con95] W3 Consortium. Hypertext Transfer Protocol - Next Generation. WWW page at <URL:http://www.w3.org/pub/WWW/Protocols/HTTP-NG/Overview.html>, 1995.

[Jac88] V. Jacobson. Congestion Avoidance and Control. In *Proc. ACM SIGCOMM '88 Symposium on Communications Architectures and Protocols*, pages 314–329, Stanford, Calif., August 1988.

[Jon88] D. W. Jones. Application of Splay Trees to Compressions. *Communications of the ACM*, 31(8), August 1988.

[KKLR96] J. Kiiskinen, M. Kojo, M. Liljeberg, and K. Raatikainen. Data Channel Service for Wireless Telephone Links. Report C-1996-1, Univ. of Helsinki, Dept. of Computer Science, Helsinki, Finland, January 1996. Accepted for presentation in the 2nd International Mobile Computing Conference, Hsinchu, Taiwan, ROC, March 25–27, 1996.

[KRA94] M. Kojo, K. Raatikainen, and T. Alanko. Connecting Mobile Workstations to the Internet over a Digital Cellular Telephone Network. In *Proc. Workshop on Mobile and Wireless Information Systems (MOBIDATA)*, Rutgers University, NJ, November 1994. Revised version published in Mobile Computing, pp. 253–270, Kluwer, 1996.

[LAK+95] M. Liljeberg, T. Alanko, M. Kojo, H. Laamanen, and K. Raatikainen. Optimizing World-Wide Web for Weakly-Connected Mobile Workstations: An Indirect Approach. In *Proc. 2nd International Workshop on Services in Distributed and Networked Environments (SDNE)*, pages 132–139, Whistler, Canada, June 1995.

[MP92] Michel Mouly and Marie-Bernadette Pautet. *The GSM System for Mobile Communications.* Mouly and Pautet, 1992.

[PM95] V. N. Padmanabhan and J. C. Mogul. Improving HTTP Latency. *Computer Networks and ISDN Systems*, 28(1 and 2):25–35, December 1995.

[Rah93] M. Rahnema. Overview of the GSM System and Protocol Architecture. *IEEE Communication Magazine*, 31(4):92–100, April 1993.

[YB94] R. Yavatkar and N. Bhagawat. Improving End-to-End Performance of TCP over Mobile Internetworks. In *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, California, December 1994.