Resource-Efficient Policies for Information Transfer in a Mobile Environment

Craig E. Wills, Matthew T. Murray and Ravindran Thangarajah*

Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609

WPI-CS-TR-96-3

Abstract

Mobile computing environments are traditionally resource-constrained with mobile computers typically characterized by less network bandwidth and power availability than stationary computers. Ideally, users should be able to experience both mobility and performance without sacrificing one for another. Therefore, there is a need for information retrieval policies that are resource efficient.

In this work we propose policies for doing resource-efficient information transfer. The basis of the proposed policies is that information users tend to have different levels of interest for components within an information element. The new policies we propose use a user profile in filtering out the "useless" or "uninteresting" information components, to minimize the amount of information transferred and processed. The work focuses on an implementation for the transfer of network news to minimize wireless bandwidth and mobile computer power use. The implementation was done through modifications to the xrn newsreader and Network News Transfer Protocol (NNTP) for a filter process to transfer individual article components (such as headers or screens of the article body) to the mobile host.

The transfer policies are evaluated individually and in combination with respect to network bandwidth and the number of requests needed for the newsreader to obtain components of a article. Behavior of the policies with respect to information transfer latency is also evaluated. The results from testing show that for a variety of users and article sizes, user profile-based

*Current address is Fore Computing Systems, Pittsburgh, PA.

component-level filtering policies yield better overall performance than conventional transfer methods.

KEYWORDS: mobile computing, resource efficiency, information quality

1 Introduction

The popularity of mobile computing is steadily increasing as the performance of portable computers continues to rise, and the availability of wireless communication technology increases. However, with the advantage of portability comes the fact that mobile computing is inherently a resource-constrained environment. Users must sacrifice network bandwidth and storage capacity, as well as power availability for mobility. Therefore, it is important to make the most effective use of all of the system resources.

Researchers in mobile computing area predict that information retrieval will be one of the driving applications of mobile computing [6, 9]. Other applications such as personal communication through e-mail and reliance on servers in the network for executables combine with information retrieval to make network access a dominant aspect of mobile computing.

The network bandwidth scarcity that characterizes mobile computing makes efficient network bandwidth utilization all the more important. Another potentially useful aspect that arises from bandwidth conservation is that it leads to reduced amounts of information transfer to the mobile hosts and hence more effective utilization of the limited storage capabilities of a mobile host. Another important property of wireless data transmission and reception adds a power consumption dimension to the above problem. Wireless communication is characterized by asymmetric power consumption for message transmissions and receptions. Transmission of wireless messages cost more in terms of power than reception of wireless messages [5, 9]. Thus any attempt to effectively utilize the bandwidth should not perform poorly with respect to power consumption as a result of an inordinate number of transmissions from the mobile host. Reducing additional wireless transmissions from the mobile host also aids in improving the response time of the system to the user who is browsing information. Though it is not clear at this moment as to which of the two concerns is more serious, a solution to the resource utilization problem should try to minimize both bandwidth use and power consumption.

In our work we have examined information transfer policies that make effective use of the limited wireless bandwidth while conserving as much power as possible by reducing wireless transmissions from the mobile host. Our original work tested the effectiveness of these policies through simulation using network news as a representative information domain [11]. This work successfully demonstrated that policies employing a user profile-based based approach for filtering the components of information elements sent to a mobile host could be more effective than traditional policies. We have followed up this work with an implementation of these policies by modifying the *xrn* newsreader and enhancing the Network News Transfer Protocol (NNTP) to test these policies with actual users [10].

The remainder of this paper describes related work in mobile computing and information filtering. We then describe the specific problem we focused on in the context of other work on information filtering and describe the information transfer policies we examined. The results from testing implementations of these policies and their significance is presented followed by conclusions drawn from this work and directions for future study.

2 Related Work

The problem of effectively utilizing the resources of mobile computers, especially network bandwidth and power, has received attention from the research community. In the design of the Wit system, applications were structured to reveal information on which data are necessary and important, which data will be needed immediately and what alternatives can be done if the data are unavailable under loss of network connectivity [12]. Researchers at Rutgers University have developed a protocol to conserve power by allowing a mobile host to remain in an energy efficient doze mode [5].

Work at MIT has focussed on producing a WWW browser that does caching and prefetching to hide the latency of wireless networks by reducing the burstiness of the network access traffic [6]. They also hint at using a proxy server running on the base station to monitor access patterns and use this information in prefetching frequently accessed data. The approach of monitoring access patterns to obtain knowledge on what to prefetch bears some similarity to our approach, but it was not the main focus of their work.

In their work on modifying Unix for use in a nomadic environment, the nomadic systems group in Sun Microsystems has developed a mail application to make more efficient use of the wireless bandwidth by reducing the amount of unused information that is transferred [2]. The work at Sun indexes mail messages server based on attributes of the message and then the complete mail item is fetched from the mail server. This approach uses component-level information in the filtering operation, but does not do component-level filtering.

Research work on mobile computing at the Columbia University has examined inter-process communication mechanisms for the mobile environments [1]. The work classifies the components of an interprocess message hierarchically according to the degree of importance to help distinguish vital data from incidental data. Follow-up work has been to translate this approach into a generalized architecture for distributed systems [13]. This work is similar to our's, but it does not specify how the components of a message are classified into an importance hierarchy.

3 Approach

The fundamental approach of this work is to filter the components of an information sent to a user at a mobile host in order to make more efficient use of the available resources. An information element is the smallest information unit—for example an article, a file or a document. Information components are parts of an information element that describe aspects or attributes of the concept that the information element represents. Depending on the concept they represent, information elements from a universe can be grouped into sub-universes. For example, the sub-universe of network news contains news articles that appear on different topics and are distributed to a large number of sites that are on the Internet. Depending on the sub-universe, it can be further broken down into smaller sub-domains. For example, the network news sub-universe can be broken down into different newsgroups that carry discussions on specific topics.

Information filtering can occur at a number of different levels as shown in Figure 1. At the highest level, users select particular information sub-universes of interest to them. Further filtering can be done until at the lowest level of the filtering hierarchy we obtain information elements that qualified in the filtering process at each level of the hierarchy. Filtering of information at each level is a matching of the user's interests called the user profile, with the attributes of the elements of the particular domain that is being considered at that level. Most information filtering schemes use a user profile as a guide in identifying information that does or does not match user preferences [3, 8].

As Figure 1 illustrates, conventional information filtering is used to identify those information elements most likely of interest to a user. However, our work has looked at a different type of filtering—at the component level. This approach is motivated by the observation that users are not equally interested in all the components of an information element, particularly elements that are composed of a large number of components. Our hypothesis is that if user preferences with respect to components of an information element can be gathered, then they can be used to filter out components and transmit just the important components, thereby increasing the effective utilization of the resources.

Component-level filtering does not have any restriction on the information sub-universes over which it can be applied. For reasons of feasibility, and to help be focused amidst the myriad of information sub-universes that exist, we choose network news because it is a popular information sub-universe and is a canonical

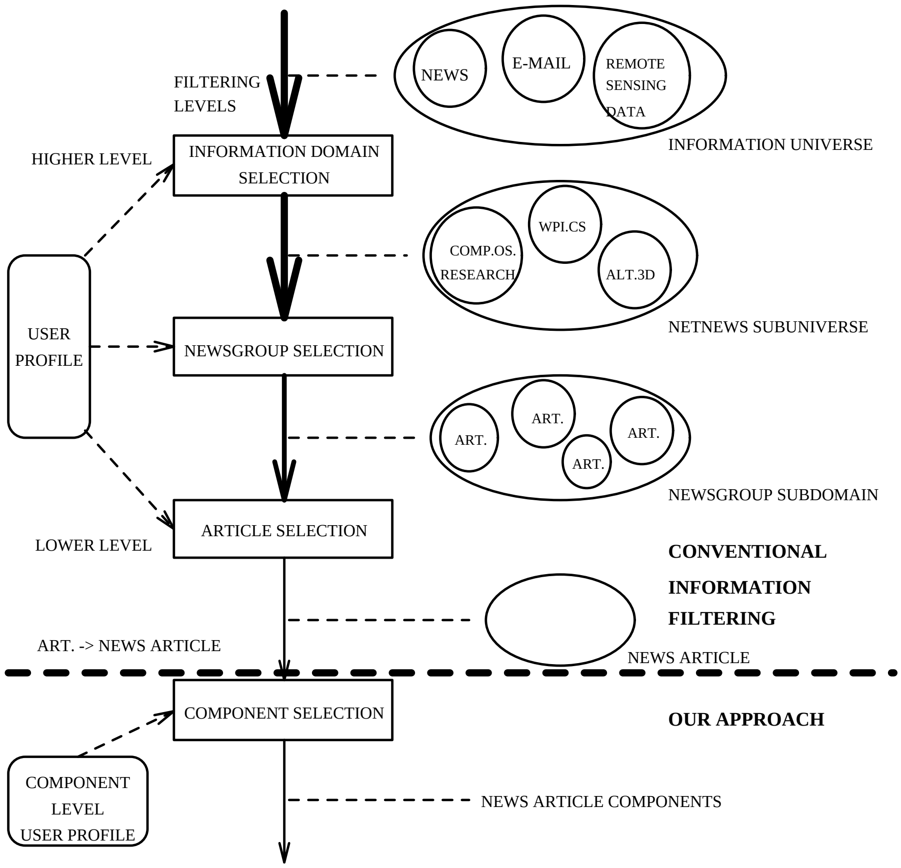

Figure 1: Hierarchical Filtering of Information

application for the mobile computing environment. Our specific aim then was to propose schemes for resource-efficient transfer of network news between the fixed network and the mobile host. Though the study focused on network news, the principles behind the techniques developed are general enough to be applied in any information browsing environment.

Network news is an Internet-wide bulletin board system for discussion on a variety of topics, organized into newsgroups. Typically a central news server for a site serves as the repository for news articles posted on different newsgroups. Client programs executing on behalf of the users interact with this news server to perform operations such as reading articles, which is the focus of our work. The interaction between a news client and server is governed by an application layer level protocol called the Network News Transfer Protocol (NNTP) [7]. News articles, though unrestricted in contents and size, are required to have a well defined structure [4]. A news article is divisible into a header and a body (or contents) portion. The header portion contains a number of header fields, with each header field on a separate line of text. Only a few header fields are required, but many others exist and news applications are free to create new header fields. After the headers comes the body of the article. The body of the article is an unrestricted text stream. Though the body of the article is unrestricted in size, it is normally displayed screen-by-screen by newsreaders where the size of a screen is dependent upon on the newsreader. In our work we used a screen size of 30 lines, which is the default size used by our *xrn* newsreader.

This organization and use of news articles naturally causes different pieces of an article to have different qualities of information for users in terms of their interest in the piece. Due to the typical sequential, screen-by-screen display of news article body within a newsreader, portions of it which lie within the first screen are read more frequently than those within the following screens. If we think of each screen (in our case 30 lines) as a component of the article then initial components within the body of the article have a higher quality. Similarly, each header of an article is a component and each of these components has a different level of importance. Some fields such as the subject or the author have higher priority, while optional headers often have a low priority.

4 Information Transfer Policies

The goal of our work is to develop policies that try to both minimize the amount of data transferred from a news server at a base station to a newsreader on a mobile host and also minimize the number of requests for information made by the mobile host to the base host. The approach we used was to investigate information transfer policies that exploited the different qualities of information we identified for components within an information element. The environment we used to test these policies is shown in Figure 2.

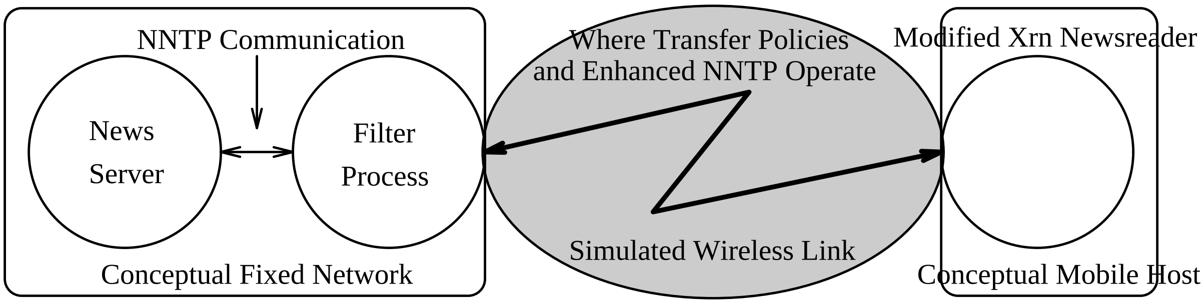

Figure 2: Scenario Where the Policies Operate

Figure 2 shows that a news filter server process was created, which communicates with the news server via NNTP. These two servers constitute the base host for our study. The newsreader, a version of *xrn* modified to handle all policies communications with the filter process using an enhanced version of NNTP. *Xrn* displays one line summaries (subject, author, number of lines) for each article in a newsgroup, from which a user can select articles to retrieve and read. Our enhancements allow *xrn* to request specific screen components and headers within a news article. In contrast, the standard NNTP protocol allows three retrieval options:

only the headers, only the body or both the headers and body of an article. As the figure shows, the policies were tested by simulating actual wireless transmission rates.

The specific policies within this environment are described in the following. The policies themselves only are concerned with transfer of components from the body of the article. In terms of headers, each policy transfers only the set of headers preferred by a user as part of his/her profile. Potential resource savings based on transferring only user preferred headers is addressed in the Results section.

4.1 Transfer All Policy

The Transfer All Policy serves as the first of two baseline policies used to evaluate the remaining policies. This policy behaves almost in the same fashion as conventional news transfer. When a newsreader requests to retrieve an article, this policy results in the transmission of the entire article, with the exception of article headers not preferred by the user (consistent with the handling of headers by all policies). No other information filtering takes place.

4.2 Minimal Transfer Policy

The Minimal Transfer Policy serves as the second baseline policy used in this work. The filter process transmits only specifically requested article screens. The mobile client transmits a separate request to the filter for each article screen desired. As opposed to the Transfer All Policy, this policy guarantees that a user retrieves only desired article screens. Thus this policy is optimal with respect to use of bandwidth, but yields the greatest number of requests.

4.3 Two Screen Cutoff Policy

This policy is similar to the Minimal Transfer Policy, but initially transfers two screens of information when a user requests to read an article. Otherwise it is identical to the previous policy in making additional requests.

4.4 Static User Profile Policy

The Static User Profile Policy utilizes a user profile to assign importance to each component and a cutoff value to determine the initial transmission of desired article components. Implementation of this and the remaining policies assume that the mobile host client sends the user profile, cutoff value and list of preferred headers upon initial connection with the filter server.

The user profile assumes that articles are read in a serial manner beginning with the first screen. While this assumption is not necessary for our work it simplifies the implementation. A user profile is a set up tuples. A sample profile is $< 1, 47\% >, < 2, 62\% >, < 3, 67\% >, < 4, 83\% >$ and $< 5, 100\% >$. These numbers mean that 47% of the time the user does not read beyond the first screen when viewing an article, 62% beyond the second screen and so on. In this example, the user never reads beyond five screens. The profile could be set by the user, but in our testing is calculated based upon the first ten articles examined by a user. Note that the profile does not account for the fact that articles are of different sizes.

The cutoff value controls the number of initial screens that are transferred to the newsreader. A cutoff of 70% would cause an initial transfer of three screens with the sample user profile. After the initial transmission for this policy, the mobile user may request additional components of the current article, at which time the filter transmits only the next screen of that article.

4.5 Dynamic User Profile Policy

This policy is identical to the Static User Profile Policy in that it also transmits the initial number of components based on the user profile and cutoff value, and it transmits only one component at a time for each subsequent request for more of the current article. The difference with this policy is that here the user profile is periodically updated to reflect recent article component usage behavior (number of components actually used per article) for that user. This policy does not rely on the same static user profile for the duration of an entire news reading session, but gradually changes the profile based on most current usage patterns.

4.6 Static Eager Remaining Send Policy

The Static Eager Remaining Send Policy functions just as the Static User Profile Policy for the determination of the initial number of article screens to transmit to the mobile client. However, the Static Eager Remaining Send Policy transmits all remaining screens of the current article upon a second request from the mobile client for more of that article. The expectation is that the article is of interest to the user and is likely to be read in its entirety.

5 Results

In the following we highlight results of testing these policies with varied users and articles. [10] contains more complete results. Although the specific testing environments were different, these results are consistent in outcome with those obtained in our previous simulation work [11]. The results shown are for two scenarios: typical users, typical articles and typical users, large articles. In the first case we employed sample users to read news articles of their choice under a modified version of *xrn*, which traced their news reading actions. We then automatically tested each of the transfer policies using the user trace to obtain results for each policy for that user. The user profile for a user was determined based upon the first ten articles with the succeeding articles read used for evaluating policy performance.

A small problem with this scenario was that the news articles read were relatively small (86% were 1-3 screens) partially because the articles themselves are small and partially because the size of our "screen" was 30 lines, which is larger than one would expect in a mobile environment. Hence, the other scenario examines the case where large articles (59% greater than 10 screens) were read. The results for the scenarios in terms of bandwidth efficiency, request efficiency and a combination of the two are given below. We also examine other results from savings for header components and network latency.

5.1 Bandwidth Efficiency

The first measure used to determine the effectiveness of each transfer policy and cutoff value for minimizing wireless bandwidth waste is the bandwidth efficiency. The approach used in this work was to express the bandwidth efficiency with respect to transmission of only desired information. This efficiency measure is expressed as the ratio of the amount of desired information transmitted to the total amount of information transmitted and is determined as follows:

> Bandwidth Efficiency =
> (amount of desired information transmitted / total information transmitted) * 100

Figure 3 shows the bandwidth efficiency results for the policies with different cutoff values in the typical user, typical articles scenario. All bandwidth efficiency results shown are measured in terms of bytes rather than screens.

As shown, the Minimal Transfer Policy always yields an efficiency of 100% because it only transmits articles actually read by the user. At the other extreme, the Transfer All Policy is always the least efficient by transmitting the entire article whether it is all read or not. However, it has an efficiency of 79% in this scenario because articles are relatively small in terms of the number of screens. The other policies lie in between these two extremes. Note that only the profile-based policies change in performance for different cutoff values.

To better examine the differences between these policies we also examined a scenario using large articles (most containing more than ten screens). These results

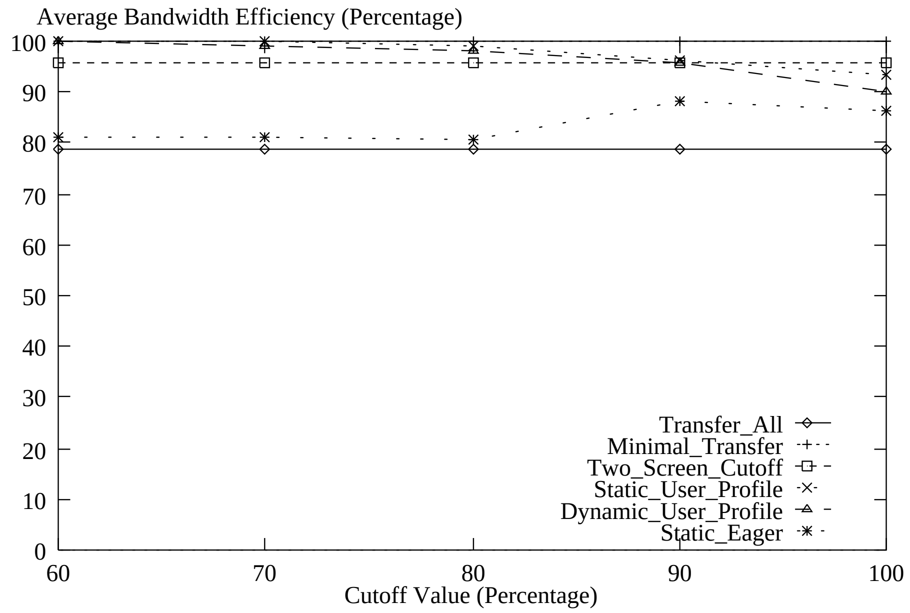

Figure 3: Average Bandwidth Efficiency (Typical User, Typical Articles)

are shown in Figure 4. As shown, the Transfer All Policy now has a bandwidth efficiency of only 30%. The profile-based policies perform well (except for Static Eager) under low cutoff values, but less well under higher cutoff values as they more closely resemble the Transfer All Policy.

5.2 Request Efficiency

The request efficiency, used to measure the power usage, is defined as the ratio of the difference between the number of requests necessary when using the Minimal Transfer Policy and number of requests necessary when using the current policy, to the number of requests necessary when using the Minimal Transfer Policy:

Request Efficiency =
((# requests for Minimal Policy - # requests for current policy) /
requests for Minimal Policy) * 100

Figure 5 shows the request efficiency results for the policies and different cutoff values in the typical user, typical articles scenario. In contrast to bandwidth

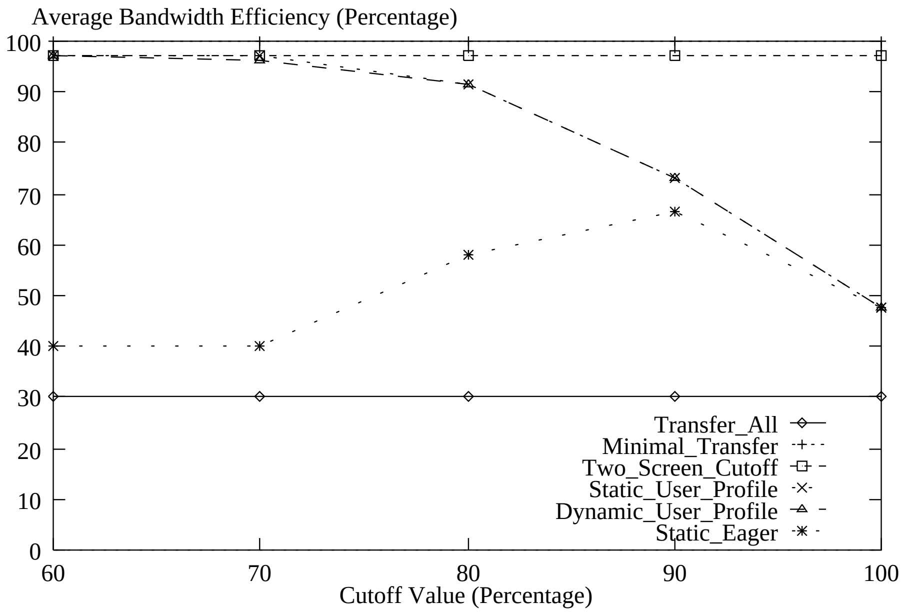

Figure 4: Average Bandwidth Efficiency (Typical User, Large Articles)

efficiency, the Minimal Policy always has a zero request efficiency and the Transfer All Policy the best (it always requires just one request for an article). Similar to the bandwidth efficiency results for this scenario, the range of results is relatively small. Again this result is because the typical articles contain relatively few screens of information.

Figure 6 shows the request efficiencies for the policies on large articles. Here, the Minimal Transfer Policy performs at over 70% efficiency and the Static User Profile Policy also performs consistently well.

5.3 Relative Merit Factor

The results of each of the respective efficiency measures is as would be expected. However, it is desirable to have a means for measuring the overall performance of the policies with respect to both bandwidth and request efficiency. Therefore a merit factor was defined, which is a weighted value based on the parameter α. This parameter takes on values between zero and one. It is defined as

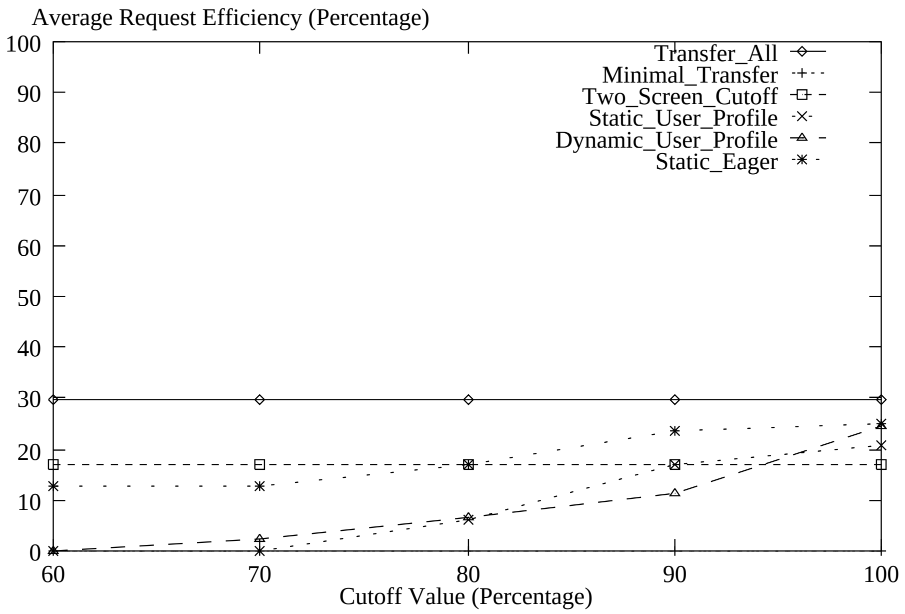

Figure 5: Average Request Efficiency (Typical User, Typical Articles)

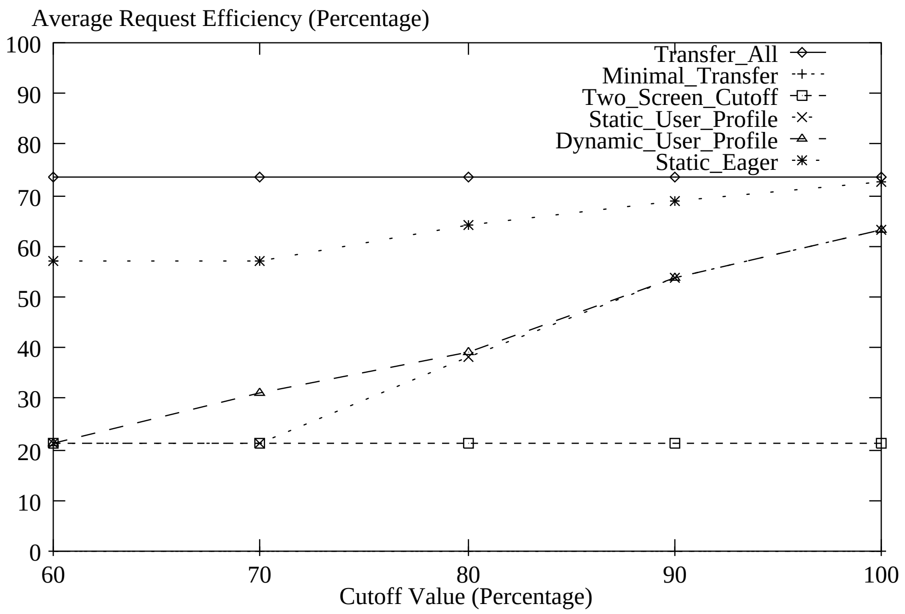

Figure 6: Average Request Efficiency (Typical User, Large Articles)

Merit Factor =
α * Relative Bandwidth Efficiency + $(1 - \alpha)$ * Relative Request Efficiency

The relative efficiency measures represent the relative performance of each policy with respect to the Minimal Transfer Policy (always one extreme of performance) and the Transfer All Policy (always the other extreme of performance). The value of α allows different weights depending on whether bandwidth or request efficiency is more important. In the case where $\alpha = 0.5$ the two concerns are of equal weight. As shown in Figures 7 and 8, a value of $\alpha = 0.5$ causes both the Minimal Transfer and Transfer All Policies to have a merit factor of 0.5. More importantly, the figures show that the profile-based policies (except for the Static Eager Policy) are superior to either of the base policies. At cutoff values of 80 or 90% the profile-based policies show the best performance, particularly for large articles.

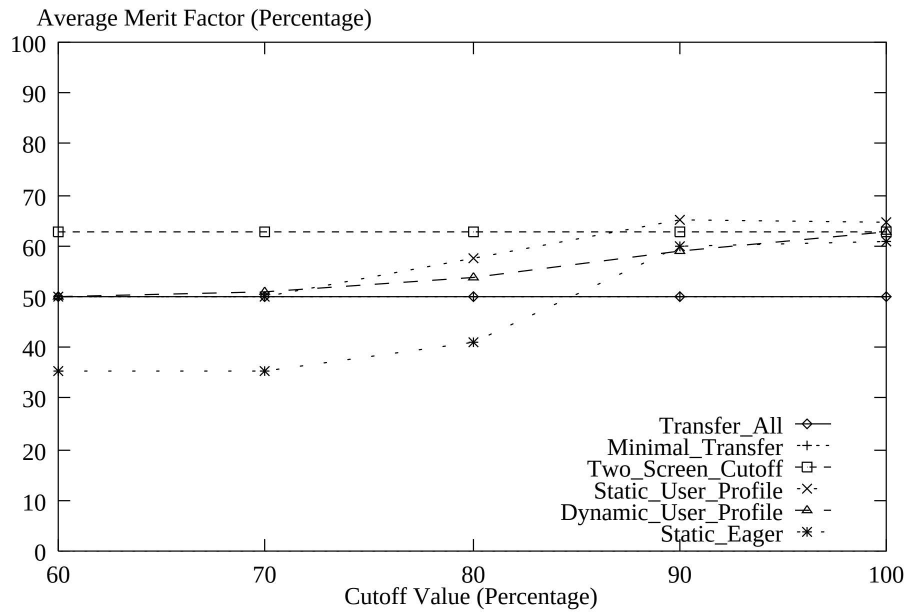

Figure 7: Average Merit Factor for $\alpha = 0.5$ (Typical User, Typical Articles)

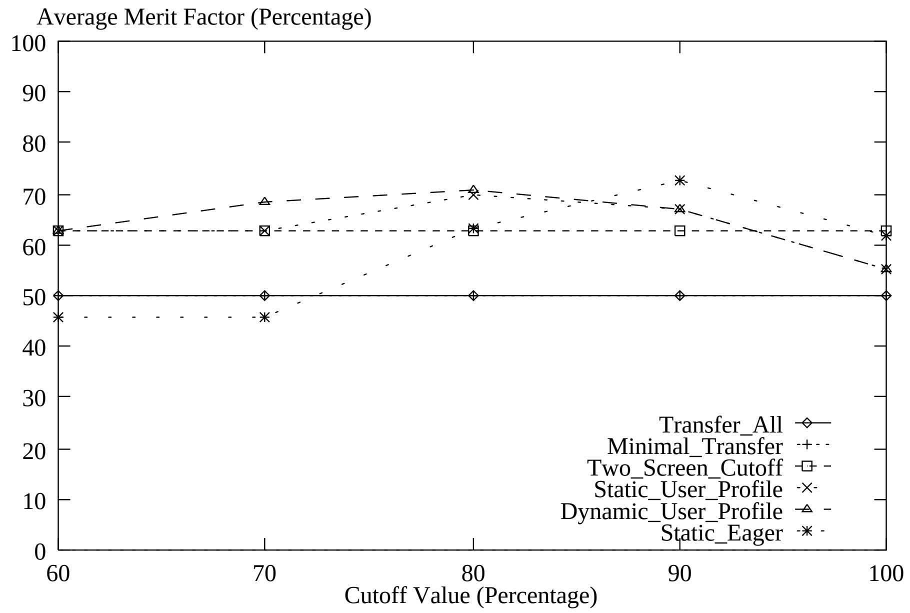

Figure 8: Average Merit Factor for $\alpha = 0.5$ (Typical User, Large Articles)

5.4 Header Components

Another potential improvement in bandwidth efficiency is to enhance NNTP to allow only preferred article headers to be transferred to the newsreader rather than all headers. The capability for a user to define preferred headers was implemented as part of our enhanced *xrn* newsreader. In most cases, users defined a small number of headers (e.g. From, Subject and Lines headers) leading to savings of 80-90% in terms of the number of header bytes and lines transferred. In terms of the overall article this savings corresponds to 12-16 lines—about a half screen.

5.5 Latency

In other testing, we allowed users to use the various policies to read articles over a simulated wireless network with a bandwidth of 14.4Kbps. The total delay experienced by users reading an entire article was minimized with the Transfer All Policy, but this policy caused many complaints in our subjective testing due to the long delay in initially loading any article more than a few screens. In contrast, when the same users used the Minimal Transfer Policy they were satisfied with the initial transfer time and did not see an appreciable change in the time to obtain additional screen components.

6 Summary

Overall, the results of this work demonstrate that user profile-based information transfer policies can out-perform conventional policies in making more efficient use of resources in a mobile computing environment. In this specific research we have used network news as a representative information domain and have shown the resource savings that can be obtained for relatively simple policies acting on selective transfer of header and screen components.

The results of this work also point to many directions of future work. Within the network news domain we plan to examine profiles that take into account the fact that articles are of different sizes. We have proposed generating not a single user profile, but a set of profiles—one for each article size [10]. Another direction is to exploit the redundancy among information elements. It is common for the contents of a news article to be included in subsequent articles. With a component cache on the mobile host, this redundancy could be exploited.

The idea of different component qualities within an information element can also be explored in other domains. Electronic mail is a similar domain as network news. The retrieval of Web documents is also a ripe area for application of this work. Finally, it is important to better explore how this idea of component-based filtering relates to other work on semantic-based filtering. Certainly semantic-based filtering can eliminate uninteresting information elements, but even within selected elements there is variable quality of the components.

References

[1] Andrew Athan and Daniel Duchamp. Agent-mediated message passing for

constrained environments. In *Proceedings of Usenix Symposium on Mobile and Location-Independent Computing*, pages 103 – 106. USENIX, August 1993.

[2] Michael Bender, Alexander Davidson, Clark Dong, Steven Drach, Anthony Glenning, Karl Jacob, Jack Jia, James Kempf, Nachiappan Periakaruppan, Gale Snow, and Becky Wong. Unix for nomads: Making Unix support mobile computing. In *Proceedings of Usenix Symposium on Mobile and Location-Independent Computing*, pages 53 – 68. USENIX, August 1993.

[3] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, Dec. 1992.

[4] M. Horton and R. Adams. Standard for interchange of USENET messages. IETF RFC 1036, 1987.

[5] Tomasz Imielinski, Monish Gupta, and Sarma Peyyeti. Energy efficient data filtering and communications in mobile wireless computing. In *Proceedings of the Second USENIX Symposium on Mobile and Location-Independent Computing*, pages 109–119. USENIX Association, April 1995.

[6] M. Frans Kaashoek, Tom Pinckney, and Joshua A. Tauber. Dynamic documents: Mobile wireless access to the WWW. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 1994.

[7] Brian Kantor and Phil Lapsley. Network news transfer protocol: A proposed standard for the stream-based transmission of news. IETF RFC 977, 1986.

[8] Shoshana Loeb. Architecting personalized delivery of multimedia information. *Communications of the ACM*, 35(12):39–48, Dec. 1992.

[9] Brian Marsh, Fred Douglis, and Ramon Caceres. Systems issues in mobile computing. Technical Report MITL-TR-50-93, Matsushita Information Technology Laboratory, 1993.

[10] Matthew T. Murray. Implementation of user profile-based component-level information filtering for a mobile environment. Master's thesis, Worcester Polytechnic Institute, May 1996.

[11] Ravindran Thangarajah. Resource efficient policies for information transfer in a mobile environment. Master's thesis, Worcester Polytechnic Institute, May 1995.

[12] Terri Watson. Application design for wireless computing. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, Dec. 1994.

[13] Bruce Zenel and Dan Duchamp. Intelligent communication filtering for limited bandwidth environments. In *Proceedings of the Fifth Workshop on Hot Topics in Operating Systems*, pages 28–34, May 1995.