Energy Efficient Algorithm for Disconnected Write Operations in Mobile Web Environments*

Jong-Mu Choi, Jin-Seok Choi, Jai-Hoon Kim, and Young-Bae Ko

College of Information and Communication,
Ajou University, Republic of Korea
{jmc, shinechoi79}@dmc.ajou.ac.kr, {jaikim, youngko}@ajou.ac.kr

Abstract. Because wireless links are subject to disturbance and failures, supporting disconnected operations is important issues in mobile computing research. Many algorithms have been proposed for supporting the efficient disconnected operations to continue performing task with a low communication cost and are focused on how the mobile client-server model can be adapted to the dynamic wireless environments. However, energy efficient algorithms are also important in mobile computing environments since mobile devices are powered by battery. In this paper, we propose energy efficient algorithm for disconnected write operations in mobile Web environments and develop analytical models with the goal of saving energy-consumption.

1 Introduction

Mobile computing and wireless networks are fast growing technologies. Since the mobile computation on the wireless networks is subject to disturbance and failures, it is essential to support disconnected operations in mobile computing. When a wireless link is expected to disconnect, process and/or data can be transferred to a mobile host (MH) to continue its operations during the disconnection. The MH is in the *hoarding* state before entering disconnection. In the hoarding state, data expected to be used during the disconnection is moved into the MH. When a wireless link is disconnected, the MH enters the *disconnected* state. During the disconnected state, the MH can continue operation using local data only including the hoarding data prefetched during the hoarding state. Data modification at the MH is logged on a stable storage. If the MH requests data not available locally in the disconnected state, then the MH enters a *reintegration* state. The MH can't execute further instructions in the reintegration state. These three states can be used to support disconnected operations [1,2,3].

* This work is supported by grant for Upcoming Research Program of the Korea Research Foundation(KRF-2002-003-D00262), the Ministry of Education of Korea(Brain 21 Project Supervised by Korea Research Foundation), and University Fundamental Research Program supported by Ministry of Information & Communication in Republic of Korea(2001-103-3).

H. Kosch, L. Böszörményi, H. Hellwagner (Eds.): Euro-Par 2003, LNCS 2790, pp. 661–668, 2003.

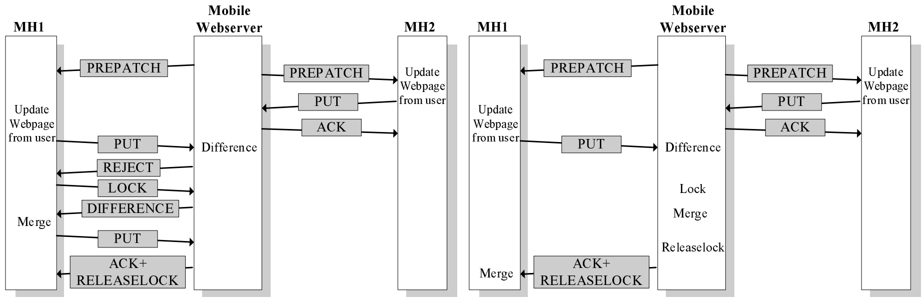
In case of mobile Web environments, a number of algorithms supporting disconnected operations have been proposed in the past years [4,5,6]. These algorithms assume read-only property in Web browsing during disconnections. To support Web page write operation, a few different approaches have been proposed [7,8]. However, all of these algorithms supporting disconnected operations on mobile Web are mostly focused on how to support disconnected operations for mobile Web environments and to optimize their algorithms in the aspect of computation and communication cost (mostly they use computing and communication time as a cost factor). Because these algorithms do not consider energy consumption in their algorithms, it can cause energy-shortage since mobile devices are powered by battery. Thus, in mobile environments, energy efficiency in the MHs (mobile hosts) are stressed as another important considering factor in the disconnected operations. To solve this problem and to further improve the energy performance of existing disconnected operations, we propose a new energy efficient algorithm for disconnected write operations in mobile Web environments. Using our proposed algorithm to reduce energy consumption, the number of messages transferred in the MHs are reduced and work load of disconnected operations for mobile Web service is performance on the Web server instead of MH. Web server does not suffer from energy-shortage since plenty of energy can be constantly supplied.

2 Related Works

In this section, we briefly summarize some of existing disconnected operations in mobile Web environments. We motivate our works by pointing out energy consumption problem not considered in these algorithms.

Many algorithms have been proposed to support disconnected operations efficiently [1,2,3]. These are based on cache management to support disconnection states. Files to be used during the disconnected state are prefetched at the MH (mobile host). In hoarding state, users can prefetch the files manually or system prefetches the files automatically by utilizing file access history.

Recently, algorithms to support disconnected operations for mobile Web environments are proposed and these are based on read-only property in Web files [4,5,6]. These algorithms assume that Web data is not changed or added in the MH. Reference [4] describes the eNetworks Web Express systems which uses paired proxies to optimize communication in low-bandwidth networks and to enable Web browsing in which both disconnected and asynchronous operations are supported. For optimizing the Web prefetching scheme, there are various algorithms. Among those algorithms, Reference [5] introduces efficient access probabilities and prefetch thresholds to reduce access delay, and prepare for disconnection. The prefetching scenario discussed in [5] is adopted to verify the performance improvement of our algorithm. To support intermittent or weak connections ARTour Web Express [6] permits updates to be gradually and asynchronously submitted to a server and all requests are served when connectivity is re-established.



(a) Disconnected write protocol [8]   (b) Proposed disconnected write operation

Fig. 1. Disconnected wirte operation in [8] and proposed algorithm

More recently, reference [8] proposed disconnected write operations for wireless Web access in mobile client-server environments. Their algorithm is depicted in Fig. 1(a). According to their scenario, two mobile hosts (MH1 and MH2) cache the Web page from mobile Web Server beforehand, then MH1 enters the disconnected state. First, two MH users modify the cached Web pages during the disconnected state of MH1. After MH1 is reconnected to mobile Web server, MH1 sends PUT request message which contains the differences between the modified Web page and the original Web page, and the version number of original Web page. Now, two different cases (**Case 1** and **Case 2**) can be occurred by Web server[8].

Case 1 (No update conflict): The Web page in the mobile Web server has not been updated by other MH (e.q., MH1). Thus, the Web server sends an ACK message to MH2 to notify the acceptance of the PUT request.

Case 2 (Update conflict): The Web page in the server has been modified by other MH (e.q., MH2). It can be detected by comparing the version numbers.

In Case 2, the mobile Web server sends the REJECT message to MH1. When MH1 receives the REJECT message, it needs to follow the special operations listed below to synchronize with the Web server[8].

1. MH1 sends the LOCK message to mobile Web server to lock the Web page.
2. Mobile Web server generates DIFFERENCES message which contains information about difference between prefetched data to MH1 and updated data from MH2.
3. On receiving DIFFERENCES message from mobile Web server, MH1 performs *merging* algorithm to update the Web page.
4. After updating the Web page, MH1 resends PUT request message which contains information about updated Web page.
5. Finally, mobile Web server releases locking on their Web page then it replies ACK message to MH1.

The above algorithm, a mobile host oriented approach, may result in generating too many messages and wasting the energy consumption in mobile host due to exchanging many messages and performing merge algorithm. We think that these problems occurred because the existing disconnected operations algorithms such as reference [8] do not consider the energy consumption as a cost metrics in mobile computing. Thus, we attempt to reduce the number of message transfer and the amount of computation in mobile host by shifting work load to mobile Web server, which is free from energy limitation and has enough computation power.

3 Proposed Disconnected Write Operation

Our proposed algorithm for disconnected write operations are depicted in Fig. 1(b). The operations of MH2 are same as in Fig. 1(a). The Web page in the mobile Web server is not updated by other MH until receiving PUT request message. However our proposed algorithm operates differently when the Web page in the server has been modified by other MH. In this situation, mobile Web server does not send REJECT message to MH1 but voluntarily merges and updates its own Web page using update information from PUT request message sent by MH1. When the version information on receiving PUT message from MH1 and on server are different, the mobile Web server operates as follows:

1. Lock the Web page to prevent the updating operation from other MH.
2. Find difference between the data prefetched to MH1 and updated from MH2.
3. Merge the difference from the previous step and update data according to PUT request message.
4. Release the locked Web page.
5. Send the ACK and DIFFERENCE, which contains merging data, to MH1.

On receiving DIFFERENCE message, MH1 can update its prefetched data. By doing so, MH1 synchronize the content of date with mobile Web server. Transferring more work load to mobile Web server, our proposed algorithm can dramatically reduce the number of message needed for disconnected write operations. Mobile Web server may be somewhat over loaded and consumes more energy than the previous algorithms as shown in Fig. 1(a). However, as Web server consists of high performance computer and enough system resources such as CPU, memory, and storages, it can endures the work loads. By supplying eternal power, energy problems can also be resolved since mobile Web servers are electrically connected to power supply construction. However, in some applications, merge algorithm can't be performed in Web server (e.g., mobile users need to decide their choice when their update operation is conflicted on the Web server). We think that mobile user can cancel the merge operation in the Web server, and send PUT message again after receiving PREFETCH message.

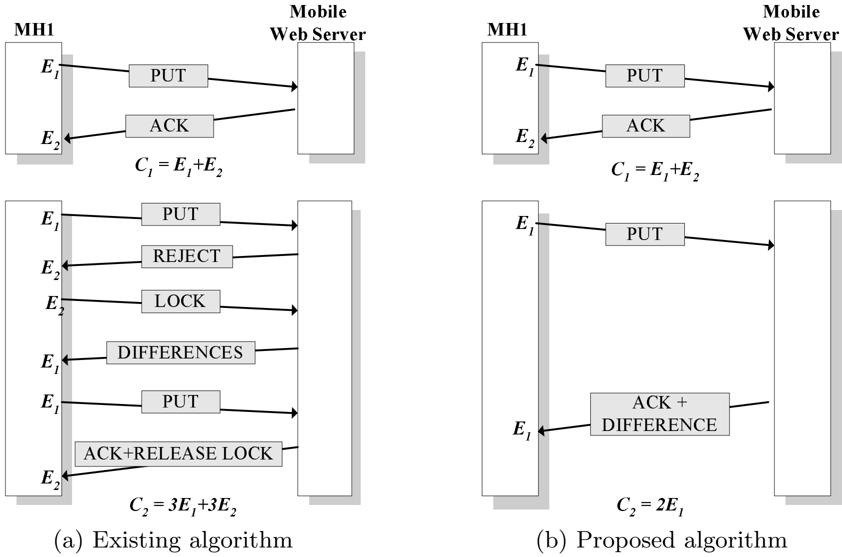


Fig. 2. Energy const model for existing algorithm and proposed algorithm

4 Performance Evaluation

The objective of our proposed algorithm is to minimize the energy consumption during disconnected write operations in mobile Web environments. Thus we use the energy consumption as a performance metrics. Proposed algorithm is compared to the existing disconnected write operations[8] for mobile Web environment by means of energy consumption. Let the energy cost on the MH be E, which include the energy to transmit the message as well as receive the message. According to the first-order radio model in [9],

$$E = E_{TX} + E_{RX} = N_{TX} \times \epsilon_{TX} + N_{RX} \times \epsilon_{RX} \quad (1)$$

where N_{TX} and N_{RX} is the number of bits in message for transmissions and receiving, respectively, and ϵ_{TX} and ϵ_{RX} is the energy consumed to transmit and receive a bit via a wireless link, respectively. If ϵ_{TX} and ϵ_{RX} are assumed to be the same, these two parameters can be denoted as ϵ, We also assume two energy cost parameters, E_1 and E_2, which represent the energy cost for exchanging message of different size. These two parameters can be classified whether it is update message or control message. Namely, E_1 is the energy cost for message transfer which contains update data such as PUT/DIFFERENCE and E_2 is the energy cost for control message such as ACK/REJECT/LOCK. These two parameter can be represented by $E_1 = p_m s_1 \epsilon$ and $E_2 = s_2 \epsilon$. where p_m is the average fraction of Web page being modified and s_1 and s_2 is the average size of Web page and control message, respectively.

Finally, we assume some parameters related to update behavior. The Web page in server can be updated by any MH. In this case, the update rate by any MH for Web page is denoted as λ, and update rate by a specific MH itself is denoted as λ_m. If MH1 updates data during a disconnected state, MH1 sends PUT request message after the end of disconnected state. Assume that other MHs also update the Web page in the mobile Web server during the disconnected state of MH1 (L), then the next PUT request message from MH1 may be denied. Namely, the **Case 2** in section 2 is occurred. To model this case, probability that an update request is denied by the mobile Web server (r), is given by $r = 1 - e^{-(\lambda-\lambda_m)L}$ where $\lambda - \lambda_m$ is the rate of update by other MHs and L is duration of disconnected state. We assume that the arrival of Web page update at the system is governed by the Poisson process. We represent energy cost as C_1 and C_2, for **Case 1** and **Case 2**, respectively. Thus, by applying the above equations, the average energy consumption involved in update propagation from the MH to the Web server as shown in Fig. 1(b) can be modeled following Eq.(2).

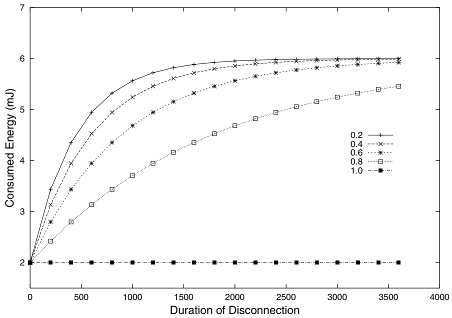
$$C_e = r \times C_2 + (1 - r) \times C_1 \tag{2}$$

Fig. 2 shows the energy cost model for existing algorithm and our proposed algorithm. We evaluate only the aspect of the MH since mobile Web server does not concern the energy limitation. As shown in Fig. 2, we can compute the four different energy costs classified in Table 1. In the Case 1, energy cost is the same for both algorithms, $C_1 = E_1 + E_2$ (E_1 for PUT and E_2 for ACK message). However, in the Case 2, two algorithms require different energy cost as two algorithms operate differently. Existing algorithm requires $3E_1$ for three control messages and $3E_2$ for two PUT messages and one DIFFERENCES message. Proposed algorithm requires $2E_2$ for PUT message and ACK+DIFFERENCE message. By applying these four energy costs to Eq.(2), we can obtain the two equations, $C_e^{EA} = 3r(E_1+E_2)+(1-r)(E_1+E_2)$ and $C_e^{PA} = 2rE_1+(1-r)(E_1+E_2)$, for each energy consumption behavior of existing algorithm and proposed algorithm, respectively.

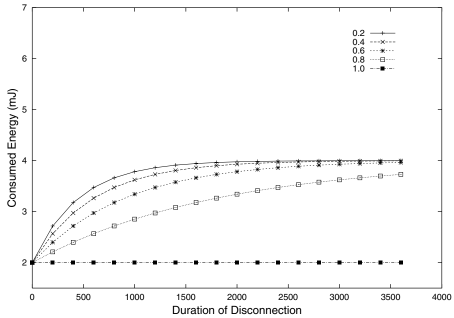
Table 1. Energy cost for different cases

	Existing algorithm	**Proposed algorithm**
Case 1: No update conflict (C_1)	$E_1 + E_2$	$E_1 + E_2$
Case 2: Update conflict (C_2)	$3E_1 + 3E_2$	$2E_1$

We have analyzed under following conditions (similar to [8] and [9]). Initially, the average rate of updating the Web page by a MH is 8/3600 and any MH is 10/3600, respectively – i.e., update process have been occurred 8 and 10 times per hour for each. We fix ϵ (radio electronics) And p_{Mm} (average fraction of the Web page modified) is 0.1. Thus, s_2 can be $0.1 \times s_1$. For different performances analysis, we varied λ_m/λ (ratio between rates of updating the Web page by a MH and any MH), ϵ, and $p_m \times s_1$ (size of updated page). Varying these three values intends to observe the effect of these parameters on energy consumption when the disconnected write operations are executed in the mobile Web environment. The value of λ_m/λ is varied from 0.2 to 1 in our first analysis. The value of ϵ is varied from 0.2 to 1 mJ/KB. And at the last analysis, we also vary $p_m \times s_1$ from 5 KB to 25 KB.

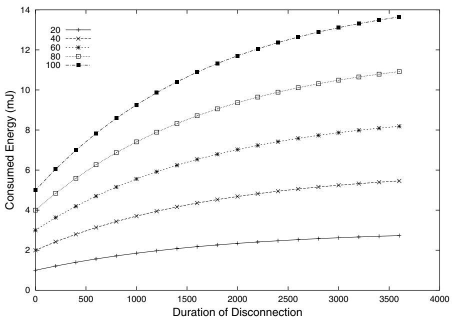


(a) Existing algorithm

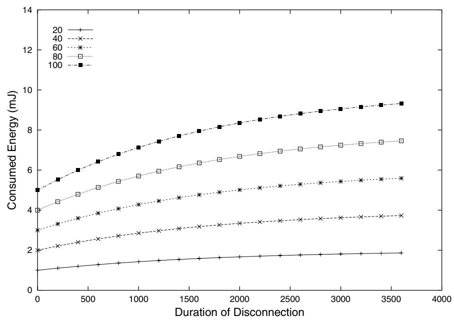


(b) Proposed algorithm

Fig. 3. Energy consumption varying λ_m/λ



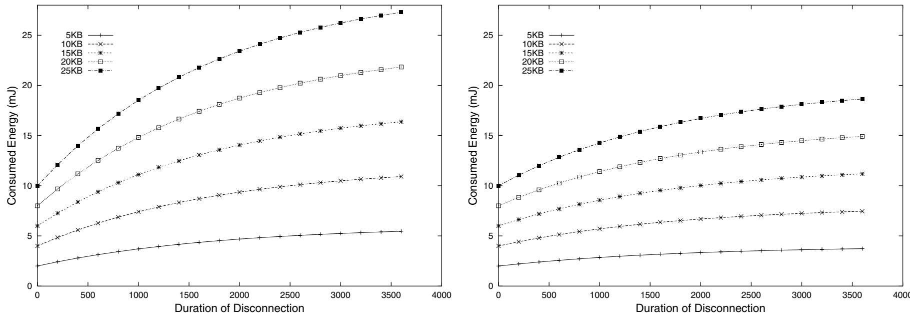
(a) Existing algorithm



(b) Proposed algorithm

Fig. 4. Energy consumption varying ϵ

The effect of varying the values λ_m/λ, ϵ, and $p_m \times s_1$ are shown in Fig. 3, 4, and 5, repectively. As these parameters are increased, the energy cost begins to increase for both algorithms. However, proposed algorithm requires a less energy (up to 30 %) than existing algorithm. The reason is that proposed algorithm reduce the number of message for disconnected operations by giving more work load to the mobile Webserver when update conflict occurs.



(a) Existing algorithm (b) Proposed algorithm

Fig. 5. Energy consumption varying $p_m \times s_1$

5 Conclusion

Many approaches have been proposed to support disconnected operation, these previous algorithms are focused on minimizing the response time. However, energy efficiency is also important in mobile Web environment since mobile host are powered by battery. In this paper, we proposed disconnected write operations for mobile Web environments to save energy consumption by reducing the exchange of unnecessary messages and transferring the work load to Web server. To verify excellence of proposed algorithm, we have developed simple analytical model to compare our algorithm with existing algorithm. The result of performance comparison shows that proposed algorithm can save the energy consumption up to 32 % over the existing algorithm.

References

1. E. Pitoura: Data Management for Mobile Computing. *Kluwer Publishers,* (1998).
2. S. Jeong and J.-H. Kim: Modeling and Performance Analysis of Disconnected Operation for Mobile Computing. *1st Annual ICCIS,* Oct. (2001).
3. J. Jing, A. Helal, and A. Elmagarmid: Client-Server Computing in Mobile Environments. *ACM Computing Surveys,* Vol. 31, No. 2, Jun. (1999).
4. R. Floyd, and B. Housel: Mobile Web Access Using eNetworks Web Express. *IEEE Personal Commn.,* Vol. 5, No. 5, Oct. (1998).
5. Z. Jiang and L. Kleinrock: Web Prefetching in a Mobile Environment. *IEEE Personal Commn.,* Vol. 5, No. 5, Oct. (1998).
6. H. Chang, et al.: Web Browsing in a Wireless Environments: Disconnected and Asynchronous Operation in ARTour Web Express. *ACM Mobicom '97,* Sep. (1997).
7. M. S. Mazer and C. L. Brooks: Writing the Web While Disconnected. *IEEE Personal Commn.,* Vol. 5, No. 5, Oct. (1998).
8. I.-R. Chen, N. A. Phan, and I.-L. Yen: Algorithms for Supporting Disconnected Write Operation for Wireless Web Access in Mobile Client-Server Environments. *IEEE Transaction on Mobile Computing,* Vol. 1, No. 1, Jan. (2002).
9. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan: Energy-efficient Communication Protocols for Wireless Microsensor Networks. *HICSS,* Jan. (2000).