

4-3-2008

Updated Data Dissemination for Applications with Time Constraints in Mobile Ad Hoc Networks

Takahiro Hara

Sanjay Kumar Madria

Missouri University of Science and Technology, madrias@mst.edu

Follow this and additional works at: https://scholarsmine.mst.edu/comsci_facwork



Part of the [Computer Sciences Commons](#)

Recommended Citation

T. Hara and S. K. Madria, "Updated Data Dissemination for Applications with Time Constraints in Mobile Ad Hoc Networks," *Proceedings of the 22nd IEEE International Conference on Advanced Information Networking and Application*, Institute of Electrical and Electronics Engineers (IEEE), Apr 2008.

The definitive version is available at <https://doi.org/10.1109/AINA.2008.62>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Computer Science Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

Updated Data Dissemination for Applications with Time Constraints in Mobile Ad Hoc Networks

Takahiro Hara

Dept. of Multimedia Engineering
Grad. Sch. of Info. Science and Tech.
Osaka University, Japan
hara@ist.osaka-u.ac.jp

Sanjay Kumar Madria

Department of Computer Science
University of Missouri-Rolla, USA
madrias@umr.edu

Abstract

In our previous work, we proposed few updated data dissemination methods to refresh old replicas efficiently in mobile ad hoc networks. These methods disseminate updated data items every time when owners of original data items update the items or every time two mobile hosts are newly connected with each other and this causes heavy traffic in the entire network. In this paper, we assume applications that periodically execute read operations with strict deadlines to data items and propose few alternative updated data dissemination methods. These methods reduce the traffic for data dissemination while keeping a high success ratio for read operations.

1. Introduction

Recently, there has been increasing interest in *mobile ad hoc networks (MANETs)* that are constructed of only mobile hosts that play the role of a router [1, 12, 16]. Since mobile hosts move freely in a MANET, network partition frequently occurs. Mobile hosts in one of the two partitioned networks cannot access data items held by mobile hosts in the other network. Thus, data accessibility in MANETs is lower than that in conventional fixed networks. For example, in Figure 1, when disconnection happens between two hosts, data item D_1 becomes inaccessible to mobile hosts on the right side while data item D_2 becomes inaccessible to mobile hosts on the left side.

In MANETs, there are many applications in which mobile hosts access data held by other mobile hosts; a good example is when a research project team constructs a MANET and the team members refer to data obtained by other members in order to streamline work. Therefore, it is becoming increasingly important to improve data accessibility in MANETs [5, 13]. In [6, 8], we assumed MANETs wherein

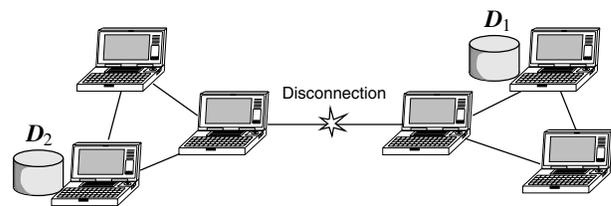


Figure 1. Network partition and data access.

data items are not updated, and proposed three replica allocation methods for improving data accessibility. These methods heuristically determine replica allocation based on the access frequency from each mobile host to each data item and the network topology at that moment.

In [7, 9, 10], we extended the three methods proposed in [6] to adapt to an environment where each data item is updated. In MANETs with data update, it is however not enough to consider only data accessibility because mobile hosts may access invalid replicas that have been updated and rollbacks occur as needed. Such invalid accesses consume the power of mobile hosts and this is a serious problem for mobile hosts that usually have poor resources. To solve this problem, in [11], we proposed two updated data dissemination methods in MANETs to refresh old replicas efficiently. However, the methods in [11] disseminate updated data items every time owners of original data items update the items or every time two mobile hosts are newly connected with each other. In many applications, such frequent disseminations of updated data items may not be acceptable because they cause heavy traffic in the entire network. Also, there are many applications that do not require frequent refreshment of replicas. A good example of such applications is that periodically executes read operations (data accesses) with deadlines, e.g., checking weather information or stock price or aggregating sensing data in sensor

networks. These applications need updated data only when they perform read operations.

In this paper, we assume applications that periodically execute read operations to data items which have strict deadlines and propose few alternative updated data dissemination methods. These methods reduce the traffic for data dissemination with keeping a high success ratio for read operations.

The remainder of the paper is organized as follows. In section 2, we show some conventional works related to our work. In section 3, we describe our assumed environment, then in section 4, propose few updated data dissemination methods. In section 5, we show the results of simulation experiments, and finally in section 6, we summarize this paper.

2. Related Works

A few studies in the research field of MANETs have been conducted to manage consistency among replicas[13, 17]. These methods keep the consistency among replicas in a pessimistic manner, i.e., read operations are performed only when the consistency can be guaranteed at the time, while our methods keep the consistency in an optimistic manner, i.e., read operations are tentatively performed to an arbitrary replica and the validity will be checked afterward.

Some studies on information dissemination in MANETs have been made [3, 14, 15]. In [3], the authors proposed an autonomous gossiping method to disseminate data items to users who are interested in these items. This approach is different from our methods that aim to efficiently update old replicas. In [14], the authors proposed an epidemic model for a simple information diffusion algorithm. However, they did not give a concrete algorithm for information dissemination. In [15], the authors introduced a distributed lookup service denoted as Passive Distributed Indexing (PDI) and proposed cache invalidation methods for reducing inconsistency among PDI index caches. This approach is different from ours because mobile hosts do not disseminate updated data items, and thus, data accessibility cannot be improved.

3. System Model

The system environment is assumed to be a MANET where mobile hosts access data items held by other mobile hosts. In this paper, mobile hosts connected to each other by one-hop/multihop links are simply called *connected mobile hosts*. In addition, we make the following assumptions:

- We assign a unique *host identifier* to each mobile host in the system. The set of all mobile hosts in the system is denoted by $M = \{M_1, M_2, \dots, M_m\}$, where m is

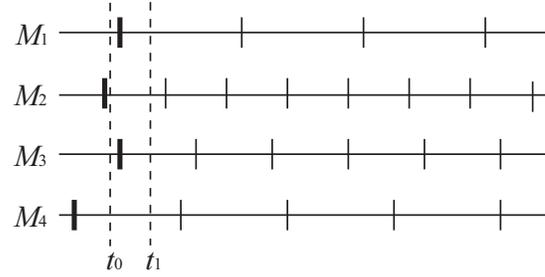


Figure 2. Data access schedules.

the total number of mobile hosts and M_j ($1 \leq j \leq m$) is a host identifier. Each mobile host moves freely.

- We assign a unique *data identifier* to each data item located in the system. The set of all data items is denoted by $D = \{D_1, D_2, \dots, D_n\}$, where n is the total number of data items and D_j ($1 \leq j \leq n$) is a data identifier. All data items are of the same size, $|D|$, and the original of each data item is held by a particular mobile host.
- Each data item is updated by the mobile host holding the original (primary copy) at irregular intervals. After a data item is updated, the replicas become invalid; that is, each data item is not partially updated and the update information is not represented by the difference from the previous version.
- Each mobile host holds a table in which the information on the latest update times (time stamps) of all data items in the entire network is recorded. This information table is called a *time stamp table*. This table incorporates the data identifier and the time stamp as the attributes.
- Time is synchronized among all mobile hosts by applying some conventional protocols such as [4].
- An application running at each mobile host periodically issues a data access (read) request with a deadline specified as the next operation issuing time. That is, applications with time constraints are assumed. Figure 2 shows a situation in which four mobile hosts, M_1, \dots, M_4 , issue periodic data access requests to a data item and the data access schedules (periods and timing) are different among the hosts.

An access request issued at time T_i succeeds if the request issuing host can read a valid replica before the next request issuing time T_{i+1} . Here, valid replicas are defined as those with time stamps between T_i and T_{i+1} or that with the time stamp t_s ($\leq T_i$) from which to T_i , no update occurred.

- We assume a rollback occurring at a mobile host does not affect data operations issued by other mobile hosts.
- Each mobile host has unlimited memory space and replicates all data items. This assumption is for the purpose of simplicity but appropriate in many applications in which the sizes of data items are enough smaller than the memory size. Good examples are the inter-vehicle information sharing (the memory size is very large) and the case in which small volume of data such as location information and sensing numerical data are shared.

Each mobile host knows which mobile hosts are the originals' owners.

- The owner of an original knows the data access schedules (data access periods and their timings) of other hosts that hold the replicas. In a real environment, several ways to accomplish this could be considered, e.g., each node registers its own schedule at the configuration phase.
- Messages and data are exchanged among mobile hosts by using an application-level routing protocol. The communication time to send data of volume $|D|$ from the sender M_i to the receiver M_j is expressed by the following formula:

$$C(|D|, M_i, M_j) = \sum_{(l,m) \in P} (|D|/B_{l,m} + H_{l,m}). \quad (1)$$

P denotes the set of wireless links involved in the shortest path between M_i and M_j , (l, m) denotes the link between mobile hosts M_l and M_m . $B_{l,m}$ and $H_{l,m}$ denote the network bandwidth and propagation delay of the link (l, m) .

It is also assumed that route (path) information to all connected mobile hosts is maintained at each mobile host by using a certain route maintenance strategy in MANETs. Each mobile host can directly unicast a packet to the destination host by specifying the path. Here, the maintained route information is not always correct because the update propagation of route information takes a little time after the network topology changes.

- For simplicity of discussion, we focus on a particular data item held by a mobile host as the original. Thus, a MANET consists of two kinds of mobile hosts; the original holder, which we call the *owner*, and replica holders.

4. Updated Data Dissemination Methods

In this section, we propose few alternative updated data dissemination methods for reducing the traffic in an environment where periodic data accesses with deadline occur. Since the original owner knows the data access schedules of all replica holders, the simplest way to disseminate the updated data item is that the owner unicasts the item to each mobile host just before the deadline of the data access request issued by the host. However, due to the following features of MANETs, this simplest way usually does not work well:

1. Unicasting to all replica holders causes very large traffic because each unicast requires multi-hop transmissions of the updated data in a MANET.
2. The owner may not connect with the mobile host at the time when the owner should send the updated data item to the host.
3. Even if the owner connects with the mobile host to which the updated data item should be sent, the host may not be able to meet the deadline due to the communication delay between the two hosts.

Since we assume application-level routing, when the owner unicasts the updated data item to a replica holder, other mobile hosts on the path between the two hosts also can receive the data item. In Figure 2, if all the four mobile hosts are on the same path from the owner, one unicast to the farthest host (suppose M_4) at time t_1 can meet the first data access requests issued by M_1 , M_2 , M_3 , and M_4 . Thus, the first problem mentioned above can be solved by determining an appropriate unicast or multicast schedules to disseminate updated data items.

To solve the second problem, a certain mechanism is needed that the disconnected mobile host can receive the updated data item from the owner or another host that holds the updated item. For the third problem, the owner should take the communication delay into account when it determines unicast or multicast schedules to disseminate the updated data item.

Based on these ideas, we propose four updated data dissemination methods.

4.1. Preliminary Setting

In every method, the owner revises the data access schedules of all connected mobile hosts to reflect the communication delays to the hosts. Specifically, the owner, M_o , shifts every operation issuing timing, $T_{i,k}$ ($k = 1, 2, \dots$), in the schedule of each mobile host, M_i , to $T_{i,k} - C(|D|, M_o, M_i)$ as shown in Figure 3. Here, $C(|D|, M_o, M_i)$ represents the

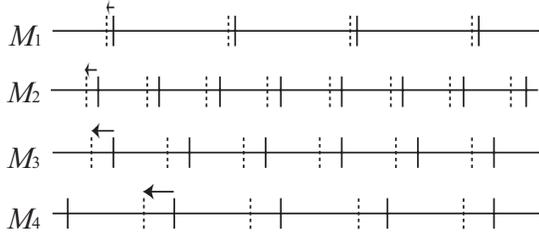


Figure 3. Shifting operation issuing timings.

communication delay to send the original from M_o to M_i . This estimation can be done with high accuracy because the owner precisely knows the path to the host.

Based on the revised data access schedules, the owner can meet the deadline of each data access request by sending the updated data item just before the deadline (the next request issuing time) comes as long as the networks topology does not change. In the following, “data access schedules” represent the revised ones instead of the original ones.

In the following of this section, we explain our proposed four updated dissemination methods.

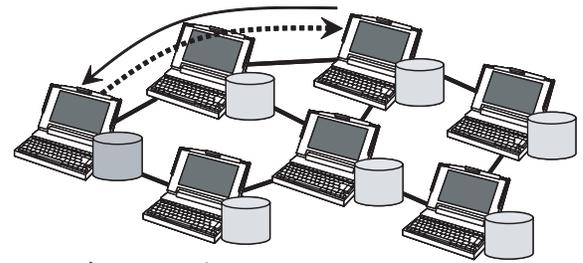
4.2. One-to-One (Pull)

Each mobile host that holds the replica sends a query to the owner to request the latest version. If the original has been updated since the host’s previous access, the owner sends back the latest version (Figure 4(a)). Otherwise, the owner sends the host a short message to let the host know that the original has not been updated. If the host holding the replica is not connected to the owner, it sends the request afterward when detecting the connection to the owner or another host holding the required version of the replica before the deadline.

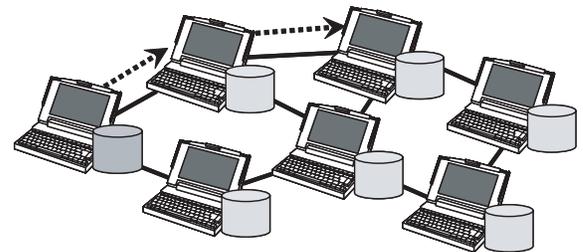
4.3. One-to-One (Push)

Based on the data access schedules, the owner unicasts the latest version to each host holding the replica before the deadline of the next read operation request (Figure 4(b)). If the intermediate nodes on the path have not completed the currently requested operation, they can get the latest version when relaying it. Here, if the original has not been updated since the host’s previous access, the owner sends a short message (validation report) to the host to let the host know the fact. A validation report includes the information on the data identifier and time stamps of the current and the previous versions of the corresponding data item. The detailed algorithm is as follows:

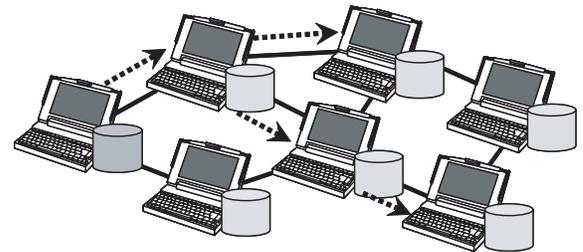
1. When the owner finishes the unicast of the data item of the latest version or a validation report, it checks the



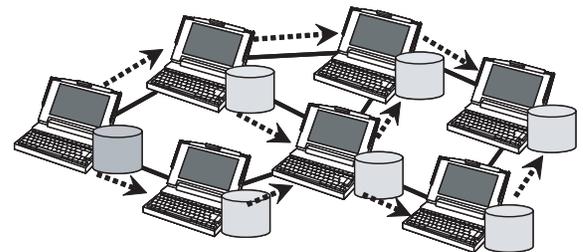
(a) one-to-one (pull)



(b) one-to-one (push)



(c) one-to- m (push)



(d) one-to-all (push)

Figure 4. Updated data dissemination methods.

data access schedules of all the connected mobile hosts for the next unicast and finds the mobile host (*destination host*) with the closest remaining time until the deadline of the next read operation.

2. The owner determines the timing when it unicasts the latest version to the destination host as the earliest time that can meet maximum number of operation requests issued by the intermediate nodes from the owner to the destination. Note that the later the unicast timing is, the more intermediate nodes that completed the latest operations issue the next operation requests and can meet the requests by the unicast. For example, suppose that the data access schedules in Figure 2 are the revised ones considering the communication delays, unicasting the latest version at t_0 to M_4 can meet only two requests issued by M_2 and M_4 while that at t_1 meet four requests issued by all the hosts. Thus, the unicast timing should be enough late to maximize the number of operation requests that can be met by the unicast. Choosing the “earliest time” among such timings tolerates errors in estimation on the communication delay.
3. The owner unicasts the latest version to the destination host at the time determined in step 2. If the original has not been updated since the host’s previous access, the owner unicasts a validation report.
4. If the destination host received the latest version, it replaces its holding replica with the received one and performs the read operation. At the same time, the intermediate nodes on the path from the owner to the destination also replace their replicas with the latest version and perform the read operations if they have a pending request. If the validation report is unicast, the operation is performed to the replica held by the destination host. As for each intermediate node, if the current time stamp specified in the report is equal to the version of the replica held by the node or the previous time stamp specified in the report is within the duration between the current request’s issued time and the deadline, the node performs the operation to its holding replica. The destination host sends an acknowledgment back to the owner. If a path from the owner to the destination host is broken, the node at the end of the broken path sends an acknowledgment back to the owner.
5. The owner records the fact that the destination host (or the host at the end of the broken path) and the intermediate nodes performed the read operations within the deadlines.

If the destination host is not connected with the owner or the unicast to the host fails due to a sudden topology change,

the owner re-sends the latest version afterward when it finds a reconnection with the host. This procedure is also performed in one-to- m . To further improve the operation success ratio, another solution to propagate the latest version is described in subsection 4.6.

4.4. One-to- m (Push)

Based on the data access schedules, the owner multicasts the latest version to mobile hosts holding the replica before the deadlines of their next read operations (Figure 4(c)). If the intermediate nodes on paths in the multicast tree have not completed the currently requesting operation, they can get the latest version when relaying it. The detailed algorithm is as follows:

1. When the owner finishes the multicast of the data item of the latest version or a validation report, it checks the data access schedules of all the connected mobile hosts for the next multicast and finds the mobile host (*core host*) with the closest remaining time until the deadline of the next read operation.
2. If the original has been updated since the core host’s previous access, the owner determines the timing when it multicasts the latest version to the core host (and other hosts in the multicast tree) as the earliest time that can meet maximum number of operation requests issued by all the connected hosts. This process is same as that in step 2 of one-to-one (push) except for considering all the connected hosts.
3. The owner creates the multicast tree consisting of mobile hosts that can perform read operations to the latest version sent by the owner. This is accomplished by creating a tree consisting of shortest paths to the hosts from the graph of connected mobile hosts, where the root is the owner.
4. The owner multicasts the latest version to the hosts in the multicast tree determined in step 3 at the time determined in step 2. If the original has not been updated since the core host’s previous access, the owner unicasts a validation report to only the core host.
5. The mobile hosts that received the latest version or a validation report perform the read operations in the same way as one-to-one (push). Each leaf node in the multicast tree sends an acknowledgment back to the owner. If a path in the multicast tree is broken, the node at the end of the broken path sends an acknowledgment back to the owner.
6. The owner records the fact that the mobile hosts that sent back acknowledgments performed the read operations within the deadlines.

4.5. One-to-All (Push)

Base on the data access schedules, the owner floods the entire network with the latest versions before the closest deadline among those of next read operations issued by replica holders (Figure 4(d)). The detailed algorithm is as follows:

1. When the owner finishes the flooding of the data item of the latest version or a validation report, it checks the data access schedules of all the connected hosts for the next flooding and finds the mobile host (core host) with the closest remaining time until the deadline of the next read operation.
2. If the original has not been updated since the core host's previous access, the owner unicasts a validation report instead of the latest version to only the core host. Otherwise, it floods the entire network with the latest version at the time t_o before the core host's deadline. t_o is a predetermined time interval to cope with errors in estimation of communication delays.
3. The mobile hosts that received the latest version or a validation report perform the read operations in the same way as one-to-one (push).
4. The owner records the fact that the connected mobile hosts performed the read operations within the deadlines. Here, note that this process is performed without acknowledgments and thus some hosts that were recorded to complete the operations may not actually performed them.

4.6. Re-dissemination of the Latest Version

As described in section 4.3, although the re-sending process in one-to-one (push) and one-to- m (push) improves the request success ratio for some degree, it cannot help if the host is not connected with the owner. To solve this problem, in one-to-one (push), one-to- m (push), and one-to-all (push), the latest versions and validation reports are re-disseminated when two disconnected groups of mobile hosts are newly connected. In doing so, mobile hosts that received or relayed the latest version buffer it for a certain period for mobile hosts that are not connected to the owner. Since buffering the latest version consumes the storage, to avoid buffering useless data for a long time, the owner while sending attaches the information on the last deadline that the sending latest version meets. By doing so, when time has passed beyond the specified last deadline, each mobile host that buffers the latest version can discard it from its storage. The detailed algorithms is as follows:

1. When a mobile host is newly connected with another mobile host, i.e., two disconnected groups of mobile hosts are newly connected, it will soon get the new route information on the mobile hosts in the newly connected group based on the applied route maintenance strategy.

If the host buffers the latest version, the host (*disseminator*) floods a *re-dissemination query packet* with the newly connected group. This packet includes the host identifier of the sender and the data identifier and the time stamp (version) of the replica.

2. If a mobile host that received the re-dissemination query packet has issued a read operation request to the replica corresponding to the data identifier specified in the received packet, it compares the time stamp with the request issued time and the deadline of the issued operation. If the time stamp is between the request issued time and the deadline, the host (*requester*) sends a *re-dissemination request packet* back to the disseminator. Otherwise, it discards the received packet.
3. If the disseminator received the re-dissemination request packet, it sends the replica of the latest version to the requester.
4. The requester that received the replica from the disseminator performs the read operation on the replica.

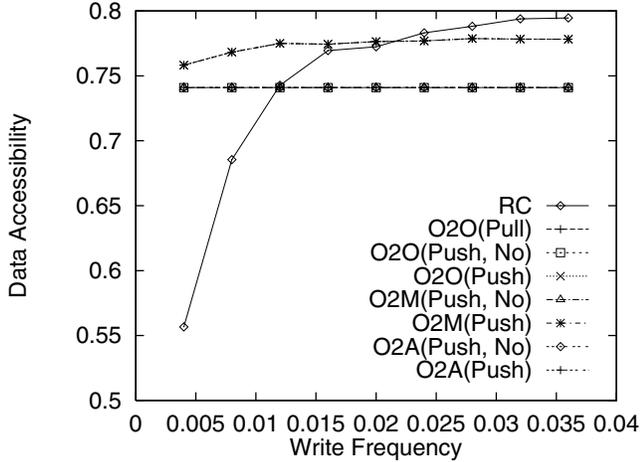
5. Simulation Experiments

In this section, we present simulation results to evaluate the performance of the proposed methods.

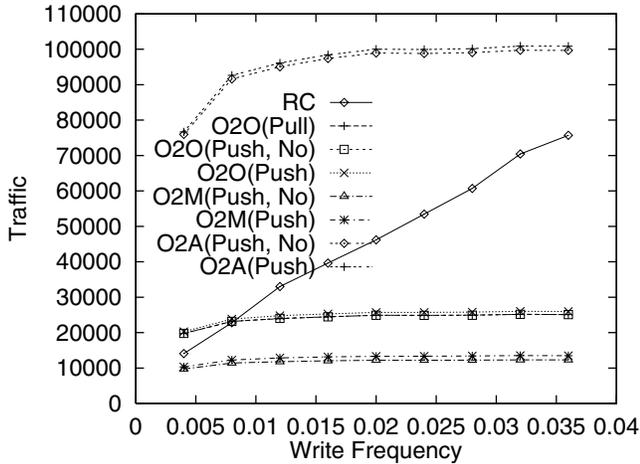
5.1. Simulation Model

The number of mobile hosts in the entire network is 40 ($M = M_1, \dots, M_{40}$), and they exist in a size 500 [m] \times 500 [m] flatland. Each mobile host moves according to the random waypoint model[2]. Specifically, each host randomly determines a destination in the flatland and moves toward the destination at a velocity randomly determined from 0 to 1 [m/s]. When the host arrives at the destination, it determines a next destination and moves toward that destination without pausing. The radio communication range of each mobile host is a circle with the radius 70 [m].

As mentioned in section 3, we focus on a particular data item, i.e., there is only one type of data item in the entire network. M_1 holds the data item as the original and updates (writes) it at intervals based on an exponential distribution with mean $1/W$ [s] (W is the write frequency). Every mobile host creates the replica of the data item. The request



(a) Data accessibility



(b) Traffic

Figure 5. Effect of write frequency.

issuing interval of mobile host M_i is determined at the beginning of the simulation based on a uniform distribution in the range from 200 [s] to 400 [s].

In the simulation experiments, we randomly determine the initial position of each mobile host and evaluate the *data accessibility* and the *traffic* of each of our proposed methods during 100,000 units of time. Here, the data accessibility is defined as the ratio of the number of successful read requests to the number of all read requests issued during the simulation period. The traffic is defined as the total hop count of data transmissions for disseminating updated data items that are performed during the simulation period. Here in the traffic, we neglect control messages for disseminating updated data because their sizes are much smaller than the sizes of data items.

5.2. Simulation result

We examine the effects of the write frequency W on each of the proposed methods. Figures 5 show the simulation result. In both graphs, the horizontal axis indicates the write frequency W . The vertical axes indicate the data accessibility and the traffic, respectively. In both graphs, for the purpose of comparison of the performance with the method proposed in [11] it is shown as “RC.” In the RC method, the updated data item is disseminated in the entire network every time the owner updates the item and every time two mobile hosts in different network partitions are newly connected with each other. “O2O(Pull),” “O2O(Push),” “O2M(Push),” “O2A(Push),” denote the one-to-one (pull), one-to-one (push), one-to- m (push), and one-to-all (push) methods, respectively. “No” in parentheses denotes the case without re-dissemination of the latest version described in section 4.6.

From Figure 5(a), four methods, O2O(Pull), O2O(Push), O2M(Push), and O2A(Push) give almost the same data accessibility. It is also shown that re-dissemination of the latest version improves the data accessibility. The RC method is heavily affected by the write frequency, where the higher the write frequency is, the higher the data accessibility is. This is because in the RC method, when the write frequency is low, the owner has a low chance to disseminate the latest version. Comparing our proposed methods with the RC method, our methods give much higher data accessibility when the write frequency is low, and give almost the same data accessibility when the write frequency is high.

Figure 5(b) shows that of our proposed methods, O2M(Push) produces the lowest traffic for data dissemination, and O2O(Push) follows. O2A produces much higher traffic than the RC method and the other two methods. The RC method is linearly affected by the write frequency. The difference in traffic between two cases with and without re-dissemination of the latest version is little. This shows the effectiveness of re-dissemination of the latest version, i.e., it improves the data accessibility while the increase in traffic is little. O2M(Push) produces much lower traffic than the RC method when the write frequency is high. Consequently, we can confirm that our proposed methods, especially O2M(Push), reduces the traffic for disseminating replicas of the latest version while maintaining high data accessibility.

6. Conclusions

In this paper, we assumed applications that periodically execute read operations to data items which have strict deadlines and proposed few alternative updated data dissemination methods. These methods reduce the traffic for data dissemination with keeping high data accessibility for

read operations. The simulation result showed the effectiveness of our approach.

As part of our future work, we plan to further evaluate our proposed methods in various environments where communication delays and failures exist. In addition, results obtained here can be used for designing data caching schemes for applications with time constraints.

Acknowledgments

The author would like to express his sincere appreciation to Professor Shojiro Nishio of Osaka University for invaluable comments on this work.

This research was partially supported by Grant-in-Aid for Scientific Research (A)(17200006) of the Ministry of Education, Culture, Sports, Science and Technology, Japan, and by International Communications Foundation.

References

- [1] D.J. Baker, J. Wieselthier, and A. Ephremides, "A distributed algorithm for scheduling the activation of links in a self-organizing, mobile, radio network," *Proc. IEEE ICC'82*, pp.2F6.1–2F6.5, 1982.
- [2] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva, "A performance comparison of multi-Hop wireless ad hoc network routing protocols," *Proc. Mobicom'98*, pp.159–164, 1992.
- [3] A. Datta, S. Quarteroni, and K. Aberer, "Autonomous gossiping: A self-organizing epidemic algorithm for selective information dissemination in wireless mobile ad-hoc networks," *Proc. of Int'l Conf. on Semantics of a Networked World (ICSNW'04)*, pp. 126–143, 2004.
- [4] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *Proc. OSDI'02*, pp.147–163, 2002.
- [5] L.D. Fife and L. Gruenwald, "Research issues for data communication in mobile ad-hoc network database systems," *SIGMOD Record*, Vol.32, No.2, pp.42–47, 2003.
- [6] T. Hara, "Effective replica allocation in ad hoc networks for improving data accessibility," *Proc. IEEE Infocom'01*, pp. 1568–1576, 2001.
- [7] T. Hara, "Replica allocation methods in ad hoc networks with data update," *Mobile Networks and Applications (MONET)*, vol. 8, no. 4, pp. 343–354, 2003.
- [8] T. Hara, N. Murakami, and S. Nishio, "Replica Allocation for Correlated Data Items in Ad-Hoc Sensor Networks," *ACM SIGMOD Record*, vol. 33, no. 1, pp. 38–43, 2004.
- [9] T. Hara and S.K Madria: "Data replication for improving data accessibility in ad hoc networks," *IEEE Transactions on Mobile Computing*, Vol.5, No.11, pp.1515–1532, 2006.
- [10] H. Hayashi, T. Hara, and S. Nishio, "Replica allocation considering data update intervals in ad hoc networks", *Proc. of IFIP/IEEE Int'l Conf. on Mobile and Wireless Communication Networks (MWCN 2004)*, pp. 131–142, 2004.
- [11] H. Hayashi, T. Hara, and S. Nishio, "Updated data dissemination methods for updating old replicas in ad hoc networks," *Personal and Ubiquitous Computing Journal*, vol. 9, no. 5, pp. 273–283, 2005.
- [12] D.B. Johnson, "Routing in ad hoc networks of mobile hosts," *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, pp.158–163, 1994.
- [13] G. Karumanchi, S. Muralidharan, and R. Prakash, "Information dissemination in partitionable mobile ad hoc networks," *Proc. SRDS'99*, pp. 4–13, 1999.
- [14] A. Khelil, C. Becker, J. Tian, and K. Rothermel, "An epidemic model for information diffusion in MANETs," *Proc. of Int'l Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'02)*, pp. 54–60, 2002.
- [15] C. Lindemann, and O.P. Waldhorst, "Consistency mechanisms for a distributed lookup service supporting mobile applications," *Proc. MobiDE'03*, pp. 61–68, 2003.
- [16] C.E. Perkins and E.M. Royer, "Ad hoc on demand distance vector routing," *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, pp.90–100, 1999.
- [17] K. Rothermel, C. Becker, and J. Hahner, "Consistent update diffusion in mobile ad hoc networks," *Technical Report 2002/04, Computer Science Department, University of Stuttgart*, 2002.