



Missouri University of Science and Technology  
Scholars' Mine

---

Computer Science Faculty Research & Creative Works

Computer Science

---

01 Jan 2005

## Consistency Management Among Replicas in Peer-to-Peer Mobile Ad Hoc Networks

Sanjay Kumar Madria

Missouri University of Science and Technology, [madrias@mst.edu](mailto:madrias@mst.edu)

Takahiro Hara

Follow this and additional works at: [https://scholarsmine.mst.edu/comsci\\_facwork](https://scholarsmine.mst.edu/comsci_facwork)

 Part of the [Computer Sciences Commons](#)

---

### Recommended Citation

S. K. Madria and T. Hara, "Consistency Management Among Replicas in Peer-to-Peer Mobile Ad Hoc Networks," *Proceedings of the 24th IEEE Symposium on Reliable Distributed Systems (SRDS'05)*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2005.

The definitive version is available at <https://doi.org/10.1109/RELDIS.2005.12>

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](mailto:scholarsmine@mst.edu).

# Consistency Management among Replicas in Peer-to-Peer Mobile Ad Hoc Networks

Takahiro Hara  
Dept. of Multimedia Engineering  
Osaka University  
hara@ist.osaka-u.ac.jp

Sanjay K. Madria  
Department of Computer Science  
University of Missouri-Rolla  
madrias@umr.edu

## Abstract

Recent advances in wireless communication along with Peer-to-peer (P2P) paradigm have led to increasing interest in P2P mobile ad hoc networks. In this paper, we assume an environment where each mobile peer accesses data items held by other peers which are connected by a mobile ad hoc network. Since peers' mobility causes frequent network partitions, replicas of a data item may be inconsistent due to write operations performed by mobile peers. In such an environment, the global consistency of data items is not desirable by many applications. Thus, new consistency maintenance based on local conditions such as location and time need to be investigated. This paper attempts to classify different consistency levels according to requirements from applications and provides protocols to realize them. We report simulation results to investigate the characteristics of these consistency protocols in a P2P wireless ad hoc network environment and their relationship with the quorum sizes.

## 1 Introduction

With the advancement in wireless communication and miniaturization of computers, mobile computing environment is becoming a more common platform. In such environments, mobile users carry portable computers or personal digital assistants with them when they move from a place to another. This has led to a new concept called the *mobile ad hoc networks (MANETs)*, where two or more mobile hosts can form a temporary network when they are together, without the need of any existing network infrastructure or centralized administration [1]. In MANETs, mobile hosts act as routers themselves, keeping information on routes to reach other mobile hosts, and help in forwarding data packets sent from a mobile host to another. At present, MANETs are actively used in military applications, rescue services, and sensor networks but applications in industries such as manufacturing are expected in the near future.

Most conventional research works on MANETs, which have been done in various research projects such as IETF

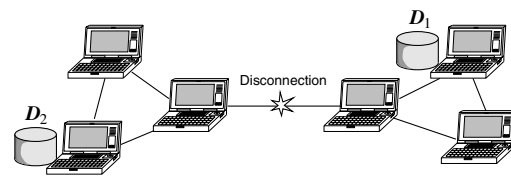


Figure 1. Network partition and data access.

(Internet Engineering Task Force), have proposed routing protocols to support communications among mobile hosts connected to each other by one-hop/multihop links [14, 21]. Such routing protocols are useful for applications in which mobile hosts directly communicate with each other, e.g., video conferencing systems. However, in MANETs, there are also many applications in which mobile hosts access data held by other mobile hosts. That is, mobile hosts in MANETs construct a *peer-to-peer (P2P)* system and share the data. We term this as P2P MANETs in this paper. A good example is when a research project team constructs an MANET and the team members refer to data obtained by other members for scientific computing.

In MANETs, as mobile hosts move freely, disconnections often occur. This causes data in two separated networks to become inaccessible to each other. 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. Preventing the deterioration of data accessibility at the point of network partition is a very significant issue in P2P MANETs [8, 15]. To improve data accessibility, data replication is the most promising solution in such an environment.

Based on this idea, we have designed effective data replication techniques in P2P MANETs in our previous papers [8–11]. In [9, 10], we have proposed replication methods in an environment where data items are updated. Furthermore, we have proposed replica invalidation methods to efficiently invalidate old replicas [12].

In [9, 10], we assume that replicas of a data item become invalid after the host holding the original updates it and the consistency among replicas is kept in the entire net-

work. However, since mobile hosts frequently disappear from the network and network partitions frequently occur in a MANET, this strict consistency management heavily deteriorates the data availability. Moreover, many applications in P2P MANETs do not require such a strict consistency. For instance, consider a situation where members of a rescue service team that constructs an ad hoc network in the disaster area are divided into several groups each of which is responsible of a certain region and the members in each group share the information on their progress, i.e., the information is replicated at mobile hosts to deal with possible network partitions. In this situation, the consistency among replicas must be strictly kept in the same group and is not required among replicas in different groups since the information sharing in a different group is only for reference.

In this paper, we discuss different possible consistency conditions among replicas in P2P MANETs. First, we classify consistency levels according to application requirements. Next, we propose protocols to realize them. We also report simulation results based on experiments to evaluate the behavior of schemes proposed under different consistency conditions.

The remainder of this paper is organized as follows. In section 2, we introduce some related work. In section 3, we explain the system model. We classify consistency levels in P2P MANETs and describe protocols to realize them in sections 4 and 5. In section 6, we show the simulation results. Finally we conclude this paper in section 7.

## 2 Related Work

In P2P systems, there are several conventional works that address the consistency management among replicas. In [17], the authors discuss consistency management in P2P networks by broadcasting invalidation reports using a hybrid push/pull approach. In [5], the authors proposed an update propagation method to replicas based on the gossiping messaging. These approaches are more optimistic in nature, whereas in our approach, the strict consistency conditions at various levels are maintained by broadcasting lock requests on quorums.

In mobile database environments (but not MANET), several consistency management strategies that consider peer disappearance have been proposed [13, 19, 22, 23]. Most of these strategies assume an environment where mobile hosts access databases at sites in a fixed network, and replicates/caches data on the mobile hosts because wireless communication is more expensive than wired communication. They address the issue of keeping consistency between original data and its replicas or cached data with low communication costs. These strategies assume only one-hop wireless communication, and thus, they are different from our approach which assumes multi-hop communication in MANETs. Since network partitions frequently occur, different consistency management strategies are required in P2P MANETs. In [23] the authors proposed a formal theory for maintaining temporal and semantic based conditions

in terms of broadcast transactions. The idea of maintaining temporal consistency in a group is similar to our TC (Time-based Consistency) described later. However, the focus in [23] was more on time-based transaction consistency. Their idea can be integrated with our TC in the sense that we can use transaction's temporal consistency condition within a group of peers.

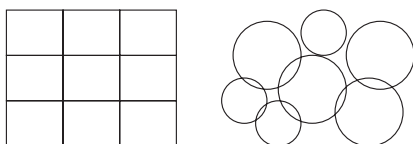
Recently, data replication is becoming more popular and significant topic of research in MANET [4, 6, 25]. Several methods have proposed for preserving consistency among replicas in MANETs [15, 18, 24]. In [15], the authors proposed methods by which replicas are allocated to a fixed number of mobile hosts that act as servers and keep the consistency among the replicas. In there, the consistency is maintained by employing a strategy based on the *quorum system* that has been proposed for distributed databases [2]. In [18], the authors extended the methods proposed in [15] by applying *probabilistic quorum system* [20] and *gossip-based message routing* [7]. Their methods are considered similar to ours because consistency among replicas is maintained based on the quorum system. However, in [15] and [18], the authors did not assume the strict consistency but aimed to keep the consistency in the entire network. Therefore, the locality in P2P MANETs described in this paper, was not taken into account.

In [24], the authors defined two different consistency levels, *local observation consistency* and *global observation consistency*. Global observation consistency is equivalent to GC (Global Consistency) considered in this paper. Local observation consistency is almost equivalent to PC (Peer-based Consistency), except that it requires replicas to eventually converge to the most recent version. In [24], only two different consistency levels are defined, whereas in here we define seven levels. Moreover, the authors tried to keep consistency based on an optimistic manner, i.e., transactions are tentatively committed and the consistency is checked afterward by using serializability graphs. Such an optimistic approach may not work well in MANETs because it will cause a large number of aborts and rollbacks of transactions due to conflicts of data operations performed in partitioned networks.

## 3 System Model

In this paper, we assume an environment where each mobile host accesses data items held by other mobile hosts in a P2P MANET and each mobile host allocates replicas of the data items on its memory space. We also assume that the area in which mobile hosts can move around is divided into several regions and the consistency among replicas is managed in each region. This assumption is due to the fact that it is usually difficult to centrally keep the consistency in the entire network. Details of the system model are as follows:

- Each mobile host (peer) knows its current location by using some devices such as GPS, and moves around in



**Figure 2. Examples of regions.**

the given area. Peers communicate with others using wireless communication to construct a P2P MANET.

- The P2P MANET consists of two kinds of mobile hosts; proxies and peers. A proxy is a specially designated peer who manages other peers in a specific region in the P2P MANET. A proxy has limited movement and does not go out of its region, whereas other peers move randomly. Each proxy knows all other proxies in the entire network. Each peer also knows all the proxies. In the example of a rescue service team mentioned above, the leader of each group is a proxy, which manages the work progress in the group and issues instructions to its members.
- We do not restrict to any particular architecture design for regions; it may be shared-nothing rectangles or arbitrary circles as in Figure 2. Proxies may not be within direct communication range of their neighboring proxies, i.e., they may not directly communicate with their neighbors. In this case, communication packets are forwarded from a source proxy to a destination proxy via other peers or proxies that exist between the source and the destination. Even so, when the network is partitioned, proxies cannot communicate with each other.

The set of all regions in the entire network is denoted by  $R = \{R_1, R_2, \dots, R_l\}$ , where  $l$  is the total number of regions and  $R_i (i = 1, \dots, l)$  is the region identifier.

- When a peer moves into a new region, the peer notifies the proxy in the region of its entrance into the region by registering its peer identifier. When the peer cannot make this registration, i.e., it does not connect to the proxy, the peer is considered not in the region even if it does exist in the region.

A peer also tries to notify the proxy in the region that the peer previously exists of its exit from the region by sending its peer identifier. When the peer cannot do this, the peer is considered in the region even if it is out of the region. Therefore, if the peer succeeds in the registration but fails the exit notification, the peer is considered in the two regions at the same time.

- Each proxy periodically checks members (peers) in its region by issuing a membership query and collecting replies from the members. The proxy considers a peer to be absent in its region if it does not receive the reply from the peer in a predetermined time period to successive membership queries.

This process can allow the proxy not to miss the exit of a peer that has failed to notify the proxy. However, this also may cause a peer to be considered absent from the region if it has not connected to the proxy.

- Each data item is updated (write operations) in different time intervals by mobile hosts.
- Each peer performs data accesses (read operations) to original data items and replicas held by itself and others.
- Every peer and proxy has unlimited memory space for creating replicas and thus it replicates all data items in the entire network. This is for the purpose of simplicity as in [15] and [18]. When data items of small volume are shared, such as location information and statistical information of collaborative works, this assumption is reasonable.

## 4 Classification of Consistency Levels

As noted earlier that in the P2P MANET environment, it is very difficult and in some cases not desirable to apply traditional consistency management protocols since peers often disappear from the network causing partitions. Therefore, different consistency management strategies are required in P2P MANETs. Moreover, there are many kinds of applications possible in P2P MANETs such as information sharing by a rescue service team and parallel and distributed processing of sensing data in sensor networks, there cannot be one universal optimal strategy for consistency management. Thus, we propose various consistency levels among replicas in P2P MANETs. Based on applications, peers can select the desired level of consistency while accessing the data items. We classify them as follows:

### 4.1 Global Consistency (GC)

The consistency among replicas is required in the entire network. This is equivalent to traditional notion of global consistency. An example of an application that requires this consistency level is a situation in which statistics of data obtained by sensors in the entire sensor network are calculated in parallel at multiple peers. This level of consistency provides very strict consistency. Providing such a global consistency requires many hops of message passing and therefore, it is an expensive proposition. In P2P MANETs, such level of consistency is hard to achieve and many applications don't desire global consistency of the data items.

### 4.2 Local Consistency (LC)

The consistency among replicas is required only in each region of interest. An example of an application that requires this consistency level is a situation in which members of a rescue service team are divided into several groups with each group has a charge of a certain region and the members collaboratively share the group work by referring to the work progress of other members. Another simple example includes finding the price of gas in a nearby location or the cheapest hotel in the current region.

### 4.3 Hopcount-based Consistency (HC)

This is a derivative version of LC. The consistency among replicas is required among peers that are connected by wireless links of equal or less than a predetermined hopcount. The simple example is when a user is only interested in finding information of interest with few hops only.

### 4.4 Time-based Consistency (TC)

Our intuition is based on the fact that replicas are consistent even if their versions are different but has not passed a predetermined time (the valid period) since they have been updated last. There are applications such as weather maps, etc., where updates arrive periodically and application only needs to know a consistent value in a certain period.

### 4.5 Peer-based Consistency (PC)

The consistency among replicas is required only in each peer. An example of an application that requires this consistency level is a situation in which a mobile user refers to the information on shops recommended by other users and determines which shops to enter for deep discounts. In this case, behaviors of the user (data operations) do not affect other users' behavior.

### 4.6 Hybrid Consistency (HBC)

A hybrid consistency level that is a combination of few of the above consistency levels is also possible. For example, let us suppose a situation in which a research project team engaged in the investigation of digging is divided into several groups each of which has charge of a certain region and statistics of sensing data obtained by the members are periodically calculated in each region. In this case, a hybrid consistency of LC and TC is required.

### 4.7 Application-based Consistency (AC)

Many kinds of application exist and each of them may require its own consistency level. In AC, assuming that different kinds of applications requiring different consistency levels exist, all the required consistency levels are satisfied at once.

## 5 Consistency Management

### 5.1 Global Consistency (GC)

To realize GC, many approaches such as those based on the two phase locking protocol and the serializability graph can be considered. However, considering characteristics of P2P MANETs such as frequent peer disappearances and network partitions, a quorum based system seems to be the best similar to the approaches in [15, 18].

Based on this idea, we employ a quorum system based on dynamic quorums similar to [18]. The consistency among replicas is hierarchically managed at two levels; among peers in a region and among proxies. More specifically, read and write operations to replicas are performed as follows:

First of all, the quorum size for write operation,  $|QW|$ , and that for read operation,  $|QR|$ , in the entire network are determined where the condition,  $|QW| + |QR| > l$ , is satisfied. Here,  $l$  is the total number of regions (proxies) in the entire network described in section 3. Moreover, in each region  $R_i$  ( $i = 1, \dots, l$ ), the quorum size for write operation,  $|QLW_i|$ , and that for read operation,  $|QLR_i|$ , in the region are determined where the condition,  $|QLW_i| + |QLR_i| > P_i$ , is satisfied. Here,  $P_i$  is the total number of peers in the region.

When a read(write) operation is issued by a peer in region  $R_i$ , first, the peer unicasts a request for the operation to the proxy that has a charge of the region. If the request cannot reach the proxy, i.e., the peer does not connect to the proxy, the peer tries to find another proxy in a different region by sequentially unicasting the request to proxies (from those in closer regions) until the request reaches a proxy. If the peer fails to find a proxy, the operation fails immediately. Otherwise, the proxy that received the request becomes the *coordinator* to perform the operation.

The coordinator tries to set *global read(write) locks* to arbitrary  $|QR|(|QW|)$  replicas held by proxies including itself (*The quorum system in the proxy network*). This is done by sequentially unicasting the global lock request to other proxies (from those in closer regions) until the total number of the global locks set to replicas reaches  $|QR|(|QW|)$ . Here, a proxy in region  $R_i$  that received the request tries to set *local read(write) locks* to arbitrary  $|QLR_i|(|QLW_i|)$  replicas held by itself and peers in its responsible region (*The quorum system in the region*). This is done by broadcasting the local lock request to peers in its responsible region. Each peer that received the request sets the local lock to its holding replica and notifies the proxy of the fact. If the proxy succeeds to set the necessary number of local locks, i.e., more than  $|QLR_i|(|QLW_i|)$  peers or the proxy replied the local lock request, the global read(write) lock is set to the replica that the proxy holds and the proxy notifies the coordinator of the fact. Otherwise, the proxy notifies the coordinator of that it fails to set the global lock.

If the coordinator succeeds to set the necessary number of global locks, it notifies the operation issuing peer of the fact and the read(write) operation is performed on the replicas that have been set the global and local read(write) locks. As for a read operation, the operation is done on a replica of the latest version among those with locks. As for a write operation, the operation is performed to all the replicas with locks. From the above mentioned conditions,  $|QW| + |QR| > l$  and  $|QLW_i| + |QLR_i| > P_i$ , the consistency among replicas can be kept among both proxies and peers in the region, i.e., a peer which issues a read operation can always read a replica of the latest version.

### 5.2 Local Consistency (LC)

Similar to GC, we employ a dynamic quorum system to realize LC. The consistency among replicas is managed only among peers in each region. Specifically, read and

write operations to replicas are performed as follows:

First of all, in each region  $R_i (i = 1, \dots, l)$ , the quorum size for write operation,  $|QLW_i|$ , and that for read operation,  $|QLR_i|$ , in the region are determined where the condition,  $|QLW_i| + |QLR_i| > P_i$ , is satisfied. Here,  $P_i$  is the total number of peers in the region.

When a read(write) operation is issued by a peer in region  $R_i$ , the peer tries to set read(write) locks to arbitrary  $|QLR_i|(|QLW_i|)$  replicas held by the proxy or other peers in the region in which the peer exist (*The quorum system in the region*). This is done by broadcasting the lock request to peers in the region. If it succeeds, i.e., more than  $|QLR_i|(|QLW_i|)$  peers or the proxy replied the request, the read(write) operation is performed to the replicas that have been set the read(write) locks. As for a read operation, the operation is done for a replica of the latest version among those with locks. As for a write operation, the operation is performed to all the replicas with locks. From the above mentioned conditions,  $|QLW_i| + |QLR_i| > P_i$ , the consistency among replicas can be kept among peers in the region.

Here, a performed write operation in a region is not necessary but better to be propagated to proxies and peers in other regions.

### 5.3 Hopcount-based Consistency (HC)

HC is a derivative version of LC. When a read(write) operation is issued by a peer, the peer multicasts the request to the proxy in the region and peers that are within a predetermined hop count,  $H$ . Then, the operation is performed to replicas held by the proxy and peers. As for a write operation, the operation is performed to all of them. This causes that multiple versions of replicas that were written by different peers exist within the predetermined hop count from a peer. Therefore, as for a read operation, the operation is performed for a replica of the latest version among them. That is, the consistent value is defined as the latest value.

A peer can pre-select hop count based on the application, network conditions, and the cost involved for each hop. Based on the history of such accesses, a peer can adjust the hop count. Other possibility is that a peer decides the hop count based on the application and QoS requested.

### 5.4 Time-based Consistency (TC)

In TC, basically, read and write operations are performed locally at the operation issuing peers.

When a read(write) operation is issued by a peer, the operation is performed to a replica held by the peer. Here, the replica held by the peer may have become invalid because it has passed longer than the valid period,  $T$ , since its last update. In this case, as for a read operation, the peer tries to find valid replicas held by the proxy and peers in the region. If it succeeds, the read operation is performed to one of the valid replicas. If it fails, the peer tries to find valid replicas held by proxies and peers in other regions. This is done by broadcasting the read request to all proxies and peers in the entire network. Then if it succeed, i.e., at least one proxy or

peer replied, the operation is performed to one of the valid replicas.

Here, as for a write operation, the operation is not necessary but better to be propagated to the proxy and peers in the region and also those in other regions. This helps many replicas to be valid(fresh) for long time.

### 5.5 Peer-based Consistency (PC)

Read and write operations are performed locally at the operation issuing peers. Here, as for a write operation, the operation is not necessary but better to be propagated to the proxy and peers in the region and also those in other regions.

### 5.6 Hybrid Consistency (HBC)

Since HBC is a combination of few of different consistency levels, it can be realized by a combination of the realization manners described above.

For example, when a peer issues a read(write) operation with respect to LC and TC, the peer tries to set read(write) locks to arbitrary  $|QLR_i|(|QLW_i|)$  replicas held by the proxy or other peers in the region in which the peer exist (*The quorum system in the region*), similar to LC. If it succeeds, i.e., more than  $|QLR_i|(|QLW_i|)$  peers or the proxy replied the request, the read(write) operation is performed to the replicas that have been set the read(write) locks. As for a read operation, the operation is done for a replica of the latest version among the replicas that are valid with respect to TC and have been set the locks. As for a write operation, the operation is performed to all the replicas with locks. From the above mentioned conditions,  $|QLW_i| + |QLR_i| > P_i$ , the consistency among replicas can be kept among peers in the region.

Here, available consistency levels for read operations depend on how write operations are performed, i.e., only same or lower consistency levels are available. For example, if write operations are performed to keep GC, all possible hybrid levels are available. If write operations are performed to keep LC and TC, only hybrid levels of LC and TC, TC and PC, and LC and PC are available.

### 5.7 Application-based Consistency (AC)

AC provides all consistency levels required from different applications. To realize AC, all write operations must be performed in the same way as GC. A read operations are performed in the same way as the required consistency level from each application.

### 5.8 Discussions

Table 1 shows the summary of how to perform read and write operations among proxies (*inter-region*) and among peers in the region (*intra-region*) for the proposed four consistency levels. In this table, we omit HC because HC is a derivative version of LC.

Here, in this table, 'QW' and 'QR' denote write and read operations based on the quorum system in the proxy network described in subsection 5.1, respectively. 'QLW' and

**Table 1. Consistency levels and write and read operations.**

	Write (intra-region)	Write (inter-region)	Read (intra-region)	Read (inter-region)
GC	QLW	QW	QLR	QR
LC	QLW	(propagate)	QLR	–
TC	L(propagate)	(propagate)	L/B	/B
PC	L(propagate)	(propagate)	L	–
HBC	arbitrary	arbitrary	arbitrary <sup>-</sup>	arbitrary <sup>-</sup>
AC	QLW	QW	A	A

‘QLR’ denote write and read operations based on the quorum system in the region described in subsections 5.1 and 5.2. ‘(propagate)’ denotes that a write operation is not necessary but should be propagated other peers in the region (in the case of intra-region) or proxies and peers in other regions (in the case of inter-region). ‘L’ denotes that an operation can be executed locally at the operation issuing peer. ‘B’ denotes that a read operation is performed by broadcasting a request for the operation. That is, in the case of intra-region (inter-region), a request for a read operation is broadcast to peers in (outside) the region. The symbol ‘/’ denotes that if its left side operation fails, its right side operation is performed. For example, ‘L/B’ denotes that if a read operation at the local peer fails, i.e., the peer does not hold the valid replica, the operation is performed by broadcasting a request for the operation. ‘arbitrary’ denotes that an operation can be performed in an arbitrary consistency level and ‘arbitrary<sup>-</sup>’ denotes that an operation can be performed in an arbitrary consistency level where it is same or lower than that of the performed write operations. ‘A’ denotes that a read operation is performed in the consistency level required from the application.

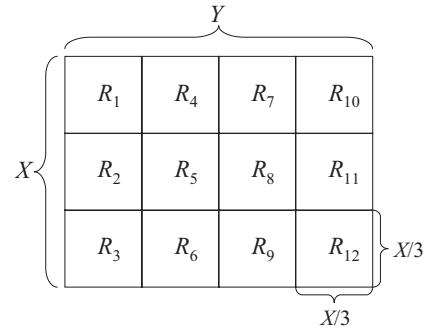
As shown in Table 1, different kinds of consistency levels are defined in P2P MANETs according to requirements from applications. The consistency level heavily affects the strictness of consistency and performance of entire system such as success ratio of database operations and network traffic. The system administrator should carefully choose an appropriate consistency level based on requirements from applications.

## 6 Simulation

In this section, we briefly show results of simulation experiments to investigate the characteristics of the proposed consistency levels.

### 6.1 Simulation model

Mobile hosts exist in an area of  $X \times Y$  [m<sup>2</sup>] which consists of 12 regions of  $X/3 \times X/3$ ,  $R = \{R_1, \dots, R_{12}\}$  (See Figure 3). Here, ratio  $X : Y$  is kept to 3:4, i.e.,  $Y$  is automatically determined if  $X$  is determined. The number of mobile hosts in the entire system is 120 ( $M = M_1, M_2, \dots, M_{120}$ ).  $M_i$  ( $i = 1, \dots, 12$ ) is the proxy of region  $R_i$ , and  $M_j$  ( $j = 13, \dots, 120$ ) is a peer that exists

**Figure 3. Simulation area.**

in region  $R_{(m \bmod 12)}$  if  $(m \bmod 12)$  is not 0 or  $R_{12}$  if  $(m \bmod 12)$  is 0. Each peer does not move beyond its assigned region. This assumption is for simplicity and helps us to properly investigate the characteristics of each consistency level because failures in registration and exit notification when peers move into another region do not occur. As a result, each region contains static ten mobile hosts.

Each peer moves according to the random waypoint model [3] in its assigned region. That is, each host remains stationary for a pause time,  $S$  [s]. Then, it selects a random destination in its assigned region and moves to the destination at a speed determined randomly between 0 and  $V$  [m/s]. After reaching the destination, it again stops for a pause time and repeats this behavior. The communication range of each mobile host is a circle with radius  $C$  [m]. The number of data items in the entire network is 500, ( $D = D_1, \dots, D_{500}$ ).  $D_j$  ( $j = 1, \dots, 500$ ) is held by a proxy or peer in region  $R_{(j \bmod 12)}$  as the original. Each proxy and peer creates replicas of all the 500 data items.

Read and write frequencies of each proxy and peer to each data items are 0.08 [1/s] and  $0.008 \times W$  [1/s], respectively. When  $W = 10$ , the write frequency is same as the read frequency. In TC, the valid period for read operation is set to  $T$  [s]. In GC and LC,  $|QLR_i|$  ( $i = 1, \dots, 12$ ) is set to  $QL$  for every  $i$  and  $|QLW_i|$  is set to  $|10 - QL + 1|$ . Here it should be noted that the number of mobile hosts in each region is 10. In GC,  $|QR|$  is set to  $Q$  and  $|QW|$  is set to  $|12 - Q + 1|$ . In LC, a write operation is not propagated to peers and proxies in other regions. In TC and PC, a write operation is not propagated to other peers and proxies.

Table 2 shows parameters and their values used in the

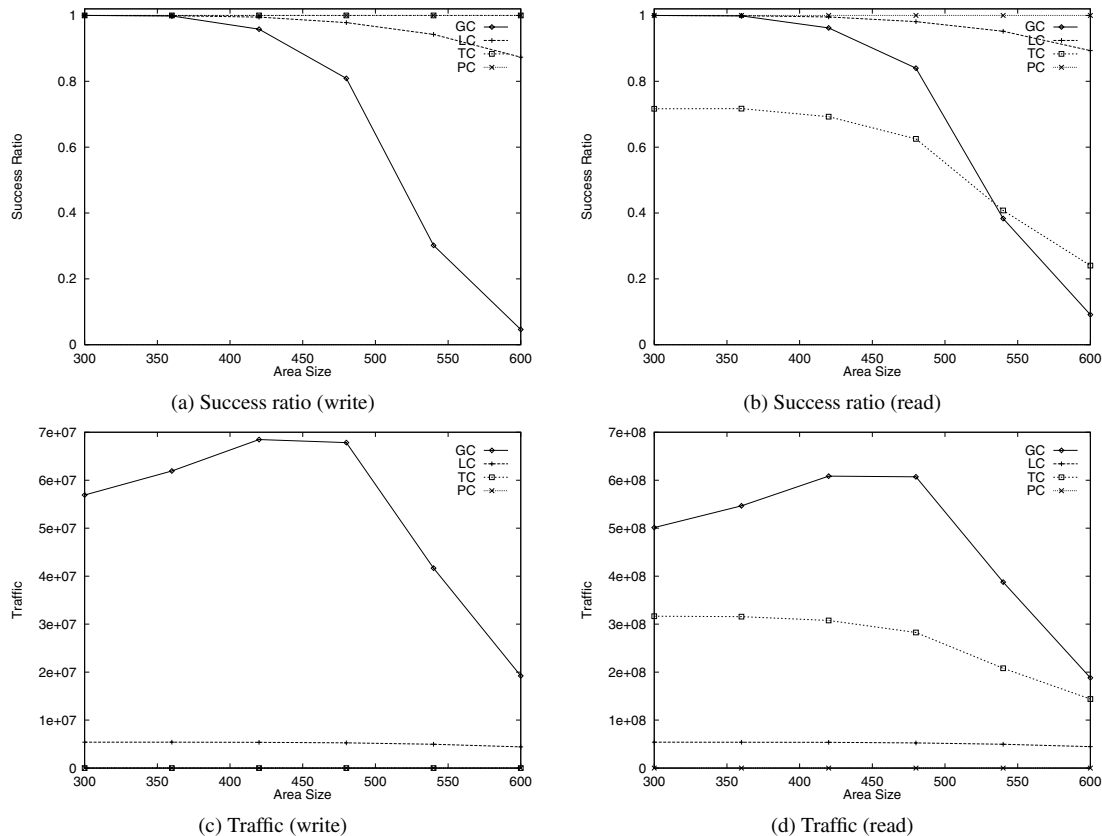


Figure 4. Effects of the area size.

Table 2. Parameter configuration.

Parameter	value
$X$	600 [m] (300~600)
$W$	1 (1~10)
$QL$	5 (2,5)
$Q$	6 (3,6)
$V$	10 [m/s]
$S$	0 [s]
$C$	70 [m]
$T$	10 [s]

simulation experiments. The parameters are basically fixed to constant values, but each of  $X$ ,  $W$ ,  $QL$ , and  $Q$  is changed in a range represented by the parenthetic values in each of the simulation experiments.

In the simulation experiments, we examine the success ratios of read and write operations and the total traffic for message exchanges of each of the four proposed consistency levels during 10,000 units of time. The success ratio is defined as the ratio of successful read/write operations to the number of all requests of read/write operations issued during the simulation time. The traffic is defined as the total hop count for message exchanges for read and write operations excluding transmissions of data items that are per-

formed during the simulation time. It should be noted that in the experiments, we only evaluate the four basic consistency level, GC, LC, PC, and TC, because the other three levels are derivation or hybrid versions of the basic ones. We assume that the network topology does not change during message exchanges for read and write operations in every consistency level. The probability that the network topology changes during the message exchange is so small that we can neglect its impact on performance, since message exchanges are usually done in less than one second. Thus, we include this assumption for the purpose of simplicity.

## 6.2 Effect of the area size

First, we examine the effects of the area size,  $X$ , on each of the four consistency levels. Figure 4 shows the simulation results. In all graphs, the horizontal axis indicates the area size,  $X$ . The vertical axis indicates success ratio in the cases of (a) and (b), and traffic in the cases of (c) and (d).

From Figure 4(a) and (b), the success ratios of both read and write operations in GC and LC get lower as the area size gets larger. This is because when the area size is large, the connectivity among mobile hosts becomes low, and thus, the proxy that received an operation request cannot set the necessary number of locks to replicas with high probability. The differences in success ratio between write and read



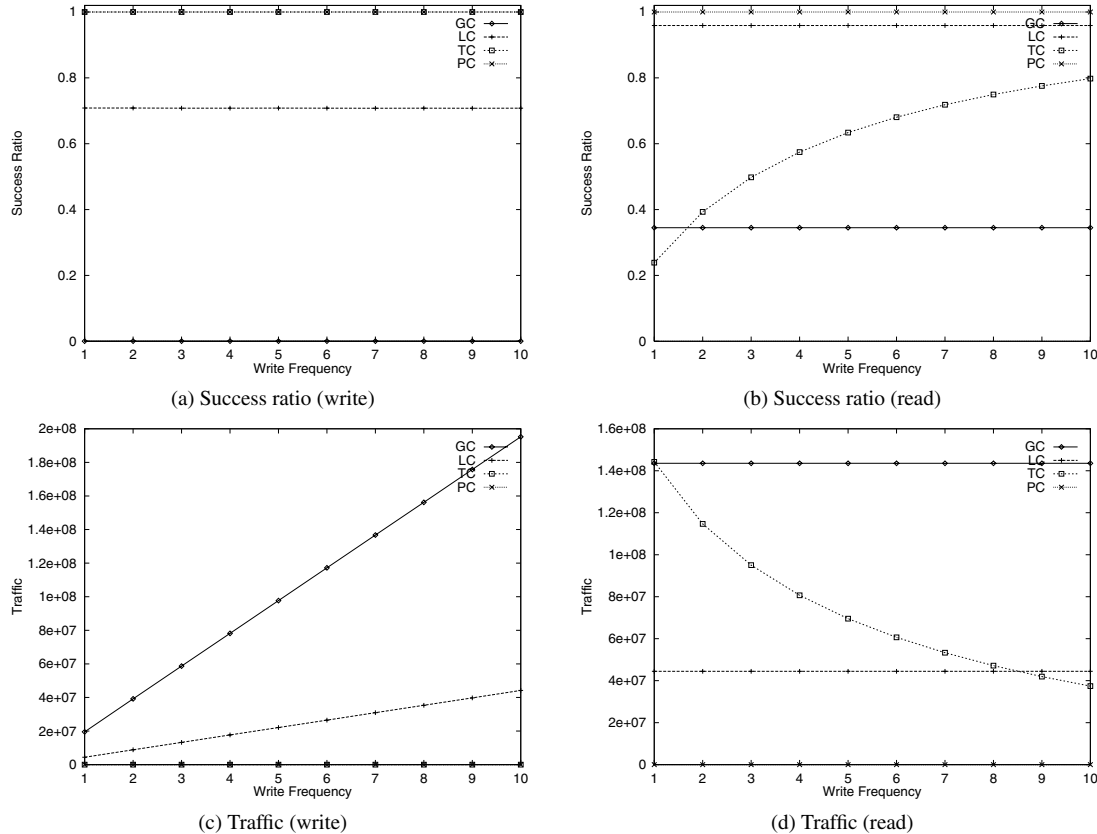


Figure 5. Effects of the write frequency.

operations are small in both GC and LC because the difference in quorum sizes between write and read operations are only 1. We can see an interesting fact that when the area size is larger than 420, the success ratio in GC suddenly gets lower but that in LC retains high. This fact shows that even when the connectivity among mobile hosts is still high in each region, the connectivity among proxies becomes low. This seems due to the employed mobility model, i.e., random waypoint model, in which peers tend to locate near the center of the region.

The success ratio of write operations in TC and those of write and read operations in PC is always 1 because these operations can be executed locally. The success ratio of read operations in TC gets lower as the area size gets larger. This is because when the connectivity is low, mobile hosts cannot access valid replicas held by connected mobile hosts with high probability.

From Figure 4(c) and (d), the traffics of write and read operations in LC and that of read operations in TC get lower as the area size gets larger. This is because the connectivity among mobile hosts get lower, and thus, the number of mobile hosts that receive and forward messages also becomes lower. This fact can be confirmed from the results in Figure 4(a) and (b) in which the success ratios in these cases gets lower. Since the connectivity among peers in the same

region is still high even when the area size is large, the decrease in traffic is slow in LC. The traffics of write and read operations in GC first gets higher and then gets lower from a certain point ( $X = 420$ ) as the area size gets larger. The reason why the traffic first gets higher is that the proxy that received a request of a write/read operation from a peer fails more times to find proxies that can set the necessary number of local locks in their responsible regions. The traffic of write operations in TC and those of write and read operations in PC are always 0 because these operations can be executed locally.

### 6.3 Effect of the write frequency

Next, we examine the effects of the write frequency,  $W$ , on each of the four consistency levels. Figure 5 shows the simulation results. In all graphs, the horizontal axis indicates the write frequency,  $W$ . The vertical axis indicates success ratio in the cases of (a) and (b), and traffic in the cases of (c) and (d).

From Figure 5(a) and (b), the success ratios of write and read operations in GC, LC, and PC and that of write operations in TC are not affected by the write frequency. Especially, the success ratios of write and read operations in PC and that of write operations in TC is always 1. The success ratio of read operations in TC gets higher as the write fre-

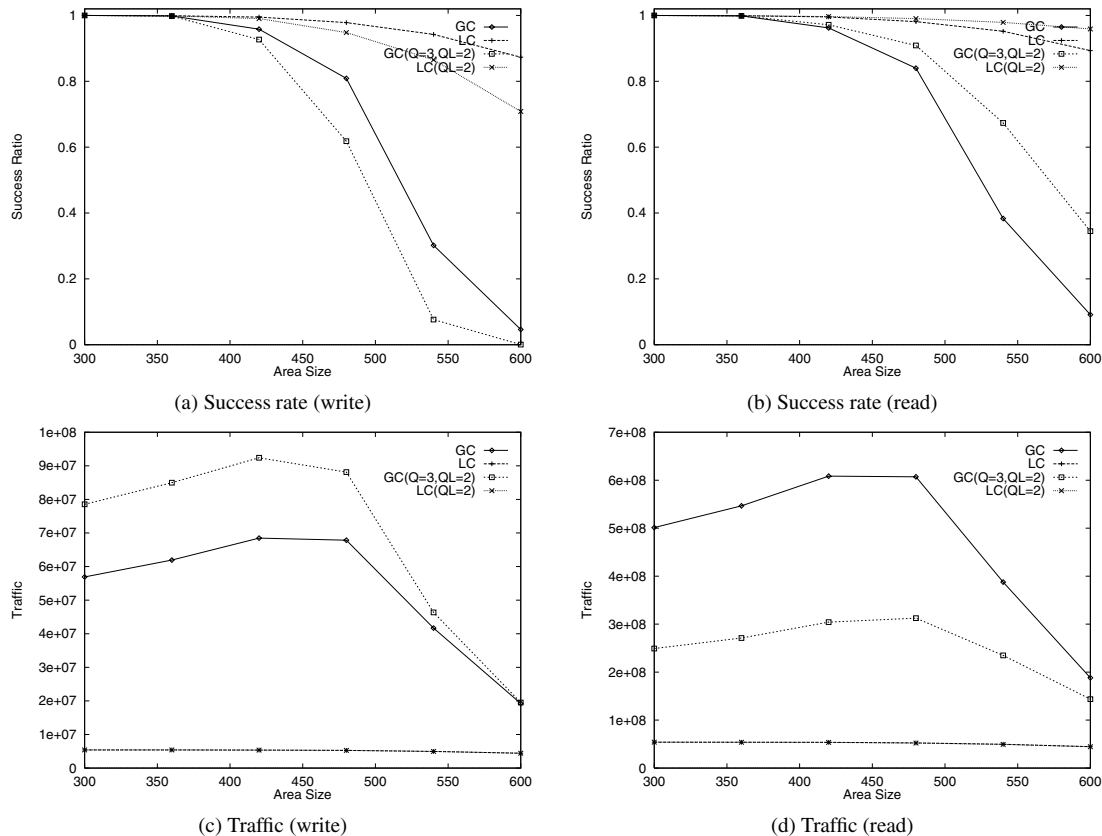


Figure 6. Effect of the quorum size.

quency gets higher. This is because mobile hosts can hold more recently updated replicas and they have more chances to access valid replicas that have been updated within the valid period  $T$ .

Figure 5(c) and (d) shows that the traffics of read operations in GC and LC are not affected by the write frequency. The traffics of write operations in GC and LC get proportionally get higher as the write frequency gets higher. These results are obvious from the characteristics of these consistency levels. The traffic of write operations in TC and those of write and read operations in PC are always 0. In TC, as the write frequency gets higher, the traffic of read operations gets lower. This is because a request issued mobile host and mobile hosts in the same region hold valid replicas with higher probability, i.e., the request is not broadcast in the entire network.

#### 6.4 Effect of the quorum sizes

Finally, we examine the effects of the quorum sizes,  $QL$  and  $Q$ , on GC and LC. In doing so, we perform the same experiments as those in section 6.2 by changing the quorum sizes as  $QL = 2$  and  $Q = 3$ . Figure 6 shows the simulation results. For comparison, the results of GC and LC in Figure 4 are also shown in these graphs. In all graphs, the horizontal axis indicates the area size,  $X$ . The vertical axis

indicates success ratio in the cases of (a) and (b), and traffic in the cases of (c) and (d).

From Figure 6(a) and (b), in both LC and GC the success ratios of read operations where  $QL = 2$  and  $Q = 3$  are higher than those where  $QL = 5$  and  $Q = 6$ , especially where the area size is large, while that of write operation is lower than that in Figure 4. Since the read frequency is ten times higher than the write frequency in this simulation environment, the total success ratio of database operations is improved by setting the size of read quorums,  $Q$  and  $QL$ , as smaller ones.

Figure 6(c) and (d) shows that the traffics of read and write operations in LC where  $QL = 2$  are almost the same as those where  $QL = 5$ . This is because in LC a request for a read or write operation is always broadcast in the region even when the quorum size is small. On the other hand, in GC the traffic of read operations is much (about half) lower than that where  $QL = 5$  and  $Q = 6$ , while that of write operations is (about 25~30%) higher. This is because in GC the proxy that received an operation request sequentially unicasts the request to other proxies until the number of successful locks at the proxies equals to the necessary quorum size. Since the read frequency is ten times higher than the write frequency in this simulation environment, the total traffic of database operations is drastically improved

by setting the size of read quorum,  $Q$ , as smaller one.

From the above discussions, it is shown that the quorum sizes for read and write operations must be carefully chosen according to the system requirement.

## 7 Conclusion

In this paper, we have discussed how to realize different types of consistency criteria in P2P MANETs. Since in P2P MANETs peers' disappearance causes frequent network partitions, therefore, it is very difficult and in some cases even not desirable to provide traditional strict consistency among replicas. Moreover, since there are many kinds of applications possible in P2P MANET environments, there cannot be one universal optimal strategy for consistency management. Thus, we have classified consistency levels according to applications demand, and then, have developed protocols to realize them.

We have done extensive simulations to investigate the behavior and features of our proposed consistency protocols. From these results, it is shown that even when the connectivity among peers in each region is high, the success ratio for write/read operations in GC (Global consistency) may degrade. It is also shown that the size of quorums affects the performance in GC and LC (Local Consistency). Moreover, in TC (Time-based Consistency), as the write frequency gets higher, the traffic of read operations gets lower whereas in GC and LC it gets higher.

As part of our future works, we plan to consider replica allocation methods that are appropriate for each of the proposed consistency levels.

## 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] D. Barbara and H. Garcia-Molina, "The reliability of vote mechanisms," *IEEE Trans. on Computers*, Vol.36, No.10, pp.1197-1208, 1987.
- [3] 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, 1998.
- [4] G. Cao, L. Yin, C.R. Das, "Cooperative cache-based data access in ad hoc networks," *IEEE Computer*, Vol.37, No.2, pp.32-39, 2004.
- [5] A. Datta, M. Hauswirth, and K. Aberer: "Updates in highly unreliable, replicated peer-to-peer systems," *Proc. ICDCS'03*, pp. 76-85, 2003.
- [6] 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.
- [7] Z.J. Haas, J.Y. Halpern, and L. Li, "Gossip-based ad hoc routing," *Proc. IEEE Infocom 2002*, pp.1707-1716, 2002.
- [8] T. Hara, "Effective replica allocation in ad hoc networks for improving data accessibility," *Proc. IEEE Infocom 2001*, pp.1568-1576, 2001.
- [9] T. Hara, "Replica allocation methods in ad hoc networks with data update," *ACM-Kluwer Journal on Mobile Networks and Applications*, Vol.8, No.4, pp.343-354, 2003.
- [10] T. Hara and S.K. Madria, "Dynamic data replication schemes for mobile ad-hoc network based on aperiodic updates," *Proc. DASFAA 2004*, pp.869-881, 2004.
- [11] 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.
- [12] H. Hayashi, T. Hara, and S. Nishio, "Cache invalidation for updated data in ad hoc networks," *Proc. Int'l Conf. on Cooperative Information Systems*, pp.516-535, 2003.
- [13] Y. Huang, P. Sistla, and O. Wolfson, "Data replication for mobile computer," *Proc. ACM SIGMOD'94*, pp.13-24, 1994.
- [14] D.B. Johnson, "Routing in ad hoc networks of mobile hosts," *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, pp.158-163, 1994.
- [15] G. Karumanchi, S. Muralidharan, and R. Prakash, "Information dissemination in partitionable mobile ad hoc networks," *Proc. SRDS'99*, pp.4-13, 1999.
- [16] R. Ladin, B. Liskov, L. Shriram, S. Ghemawat, "Providing high availability using lazy replication," *ACM Trans. on Computer Systems*, Vol.10, No.4, pp.360-391, 1992.
- [17] J. Lan, X. Liu, P. Shenoy, and K. Ramamritham, "Consistency maintenance in peer-to-peer file sharing networks," *Proc. IEEE Workshop on Internet Applications*, pp.90-94, 2003.
- [18] J. Luo, J.P. Hubaux, and P. Eugster, "PAN: Providing reliable storage in mobile ad hoc networks with probabilistic quorum systems," *Proc. ACM MobiHoc 2003*, pp.1-12, 2003.
- [19] S.K. Madria, "Timestamps to detect R-W conflicts in mobile computing," *Proc. Int'l. Workshop on Mobile Data Access (MDA'98)*, pp.242-253, Nov. 1998.
- [20] D. Malkhi, M.K. Reiter, and A. Wool, "Probabilistic quorum systems," *Information and Computation*, Vol.170, No.2, pp.184-206, 2001.
- [21] 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.
- [22] E. Pitoura and B. Bhargava, "Maintaining consistency of data in mobile distributed environments," *Proc. IEEE ICDCS'95*, pp.404-413, 1995.
- [23] E. Pitoura, P.K. Chrysanthis, and K. Ramamritham, "Characterizing the temporal and semantic coherency of broadcast-based data dissemination," *Proc. Int'l Conf. on Database Theory (ICDT'03)*, pp.410-424, 2003.
- [24] 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.
- [25] K. Wang and B. Li, "Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks," *Proc. IEEE Infocom'02*, Vol.2, pp.1089-1098, 2002.