Review

Data management issues in mobile ad hoc networks

By Takahiro HARA*¹,†

(Communicated by Makoto NAGAO, M.J.A.)

Abstract: Research on mobile ad hoc networks (MANETs) has become a hot research topic since the middle 1990’s. Over the first decade, most research focused on networking techniques, ignoring data management issues. We, however, realized early the importance of data management in MANETs, and have been conducting studies in this area for 15 years. In this review, we summarize some key technical issues related to data management in MANETs, and the studies we have done in addressing these issues, which include placement of data replicas, update management, and query processing with security management. The techniques proposed in our studies have been designed with deep considerations of MANET features including network partitioning, node participation/disappearance, limited network bandwidth, and energy efficiency. Our studies published in early 2000’s have developed a new research field as data management in MANETs. Also, our recent studies are expected to be significant guidelines of new research directions. We conclude the review by discussing some future directions for research.

Keywords: mobile ad hoc networks, data replication, update management, query processing

1 Introduction

1.1 History of MANET research. Mobile ad hoc networks (MANET) has its origin in Packet Radio Network, which was studied in the early 1970’s. Afterward, MANET has been a hot research topic since the 1990’s, in the computer science and IT research communities. While there are different definitions of a MANET, the most typical and well-known definition is a wireless network which is temporally constructed solely by mobile nodes with wireless communication capabilities. In a MANET, mobile nodes have a limited range of wireless communication, basically restricted to the reachable area (coverage) of radio signals. Therefore, two mobile nodes located beyond their own communication range communicate with each other via other intermediate nodes located between them, who relay their communication messages. Thus, in a MANET, communication between nodes is achieved in a multi-hop manner.

*¹ Department of Multimedia Engineering, Osaka University, Osaka, Japan.
† Correspondence should be addressed: T. Hara, Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, 1-5 Yonadaoka, Suita, Osaka 565-0871, Japan (e-mail: hara@ist.osaka-u.ac.jp).

Compared with other existing network infrastructures, such as the Internet and WiFi, MANETs have a significant advantage, in that they can be constructed without centralized network controls (i.e., fixed infrastructure). Thus, MANETs are expected to be useful in situations where a fixed network infrastructure is not available, such as in excavation work or military affairs, and also in rescue operations where network infrastructures are compromised. A MANET is also useful for car-to-car communication, for safe driving and other services, because it can provide real-time communication, and is more energy and cost efficient than infrastructured communication.

In a MANET, the movement of mobile nodes often causes nodal disconnection (i.e., two nodes that were within the communication range of each other move beyond this range and lose the ability to communicate directly with each other). Thus, a number of techniques have been developed to support communication between arbitrary pairs of nodes in a MANET, where the network topology is dynamically and continuously changing. Since mobile nodes are typically small battery-driven devices such as laptop computers, mobile phones, or sensor nodes, they are limited by resource constraints in terms of battery-life and communication channel. Therefore, almost
all the studies in the early stage of MANET research (in the 1990’s) sought networking techniques for achieving efficient communication between nodes.\textsuperscript{5–7} In particular, designing routing protocols to find efficient communication paths between source and destination nodes, and relay message packets along such paths, has been a central focus of research for many years.

Figure 1 shows an example in which the source node $S$ tries to find an efficient communication path to the destination node $D$ (for example, to access a data item held by $D$), where a line between a pair of nodes indicates that there is a radio communication link between them (\textit{i.e.}, they are within the communication range of each other). In this example, the routing protocol basically finds the shortest path to the destination node, indicated here by the solid arrow, rather than, for example, that indicated by the dashed arrow.

1.2 New research direction: data management in MANETs.

1.2.1 Motivation. Thanks to the networking techniques developed by the existing studies, MANETs have become available for practical use in some situations such as military affairs\textsuperscript{8,9} and disaster sites.\textsuperscript{10,11} Thus, these studies have clearly been of value. However, the networking techniques are neither ideal nor sufficient, primarily because they only seek efficient communication between two nodes connected to each other via a one-hop (direct connection) or multi-hop path, and thus are ineffective (cannot do anything) if the two nodes have no communication path between them (\textit{i.e.}, the network is partitioned).

In most MANET application environments, such as rescue operations, users not only employ direct communication, such as voice communication or video chat, but also often share information (\textit{e.g.}, sensor data to monitor environmental situations, or information on the progress of collaborative work). And this fact directed us in a new research direction, toward MANET data management. While the system cannot control the mobility of users, \textit{i.e.}, it cannot avoid network partitioning, the system can control data operations, for example, through data replication and replica placement, \textit{i.e.}, it can manage the data. Therefore, even when the network is partitioned, we can keep high performance of information sharing in the MANET, by controlling data operations effectively. More specifically, if we replicate, on another node, a data item held by a particular node (data owner), we can continuously access the data item, even when the data owner is not accessible (\textit{i.e.}, the owner is disconnected from the network or network partitioning occurs). This notion prompted our initial address of data management issues in MANETs.

Figure 2 shows an example. Here in a MANET shown in Fig. 1, the seven nodes on the left move in an opposite direction from the seven nodes on the right, and the network is partitioned. Now, let us assume that the source node $S$ in Fig. 2 wishes to communicate with the destination node $D$, as $S$ seeks to access $D$’s held data item. In this situation, no routing protocol can solve the network partitioning problem, and thus node $S$ simply cannot access $D$’s held data item. However, if this data item is replicated (\textit{i.e.}, copied) on one of the seven nodes on the left side, before the network is partitioned, $S$ (and the other six nodes) still can access that data item. Here, we assume that at some timing before the network partitioning, a node decides to replicate the data held by $D$ according to some strategy, \textit{e.g.}, a strategy to decide data replication when finding a critical link which may cause network partitioning, and replicates data items beyond the found critical link.

Here, it should be noted that while the MANET research has the history of more than 40 years, its application domains have been still limited to, for example, military affairs and disaster sites. One of the main reasons of this is that MANET is basically useful for situations where network infrastructures such as the Internet are not available, which are rare situations today, \textit{i.e.}, the usage of MANET is
naturally limited. However, in some application domains such as interpersonal communication in town and information sharing among vehicles for safety/autonomous driving, it has recently been required that applications should be achieved with less overhead on the Internet. Thus, MANET has been attracting much attention again as a key technology for off-loading and edge computing in such application domains.

However, there are still some remaining technical issues until MANETs are widely used in such application domains, and we believe that the most serious issue is lack of practical middleware including data management techniques. In other words, achieving just high communication performance cannot achieve satisfactory performance for information sharing (e.g., data availability, data access latency/throughput, scalability, and security/dependability) which the applications require. In this review, we aim to enlighten such technical issues from the data management perspective.

1.2.2 Application examples. As aforementioned, information sharing based on data management techniques in MANETs is highly beneficial to a variety of real-world applications. Actually, existing MANET studies including recent ones have assumed a variety of application domains. We present typical applications as below.

- Rescue operations at a disaster site12–14
- Real-time information sharing among vehicles15
- Interpersonal communication in town16
- PANs (Personal Area Networks)17 and BANs (Body Area Networks)18
- Mobile sensor networks19,20
- Information sharing in conferences and classrooms21,22
- Military affairs23,24
- Smart home25

Among the above application examples, we pick up the first three ones, and discuss them in detail.

A. Rescue operations at a disaster site. At a disaster site, information is often the most critical element in efficient and effective rescue operations, such as rescue planning, rescuer assignment, and resource allocation. If the information on structural damage, injured persons (injured level and location), the status of rescuer activities, etc., is effectively shared among rescuers and management staffs, over a wide range and in real-time, the efficiency and effectiveness of the rescue operations that depend on such information will be significantly improved. Unfortunately, however, information sharing in real-world rescue operations is still far from adequate, with much critical information not being fully utilized. Instead, in most cases, rescuers must work without sufficient information (sometimes with no real-time information at all), and rescue leaders can obtain such information only after a delay of a few hours or even half a day.

Therefore, if we could develop mechanisms enabling real-time information sharing based on MANET data-management techniques, various kinds of critical information could be shared among rescuers and other staffs in a real-time, efficient (e.g., less energy consumption), reliable, and secure manner, resulting in significant improvements in rescue operations.

B. Real-time information sharing among vehicles. In-vehicle systems, such as car navigation systems, typically involve high-end mobile devices which offer various kinds of services to users. Recently, more advanced ITS applications, enabling safe-driving support and autonomous cars, have attracted much attention. In such applications, real-time information sharing among vehicles is essential. For example, for both safe-driving support and autonomous cars, driving information on nearby vehicles, and environmental data on traffic jams, accidents, road conditions, etc., must be shared in a real-time, efficient, and accurate manner.

In traditional ITS systems, such information is shared by means of infrastructure such as the Internet and fixed devices on roads (e.g., Japanese Vehicle Information and Communication System (VICS)). However, since such information includes various types of sensor data, and its data volume is typically large, infrastructure-based approaches suffer from delays and unnecessary communication traffic, which may not satisfy the system’s real-time requirements. In addition to this problem, the data traffic generated by such ITS systems, as well as other mobile and IoT (Internet of Things)/M2M (Machine-to-Machine) applications, make massive demands on typically limited network capacity.

MANET-based information sharing is expected to solve these problems,15 since here, wireless communication among nearby nodes involves very short delays, and does not inject any data traffic into an infrastructure network (e.g., the Internet).

C. Interpersonal communication in town. Similarly to the ITS applications described above, mobile applications on smart phones typically generate large data traffic. Among these, location-based services (LBSs), such as nearby restaurant recommendations
and shop advertisements, as well as social networking services (SNSs), have become very popular; and recently, location-based social network services (LBSNs), which integrate these two popular services, have become widely available. MANET-based communication is highly effective for offloading data traffic from the Internet in such LBSNs; and thus, like corresponding ITS applications, MANET-based applications can achieve short delays and efficient traffic offloading.

1.2.3 Technical issues. It should be noted that data replication in MANETs is not, in itself, an ideal solution. In fact, it presents two new and serious challenges needing addressing, and two additional challenges.

A. Replica placement. First, due to several factors, such as limited storage space and network bandwidth, the number of replicas (copies) allocatable to each node is basically limited. Therefore, effective replica allocation is crucial to system performance; for example, in terms of data availability (data access request success rate), network traffic, and response time.

B. Update management. Second, since data items are generally updated by the data owners or other nodes, we must effectively manage different versions of data copies, in order to ensure consistency of data access; that is, all forms of data access (read operations) must read a valid version of the target items.

While data access consistency has been actively studied in the database and distributed system communities, MANETs have significant properties which make data access consistency difficult to achieve, including dynamic change in network topology, difficulty in recognizing the global view, and limited resources such as network bandwidth, storage, and computational power. Therefore, we need to develop new mechanisms for preserving data access consistency in MANETs.

C. Query processing. In addition to the problems posed by data replication, finding (locating) data items of interest (i.e., query processing) is also a significant issue, because this directly affects the performance of the system or application. Here, there are several different types of queries. The most naive involves accessing data items by specifying their data identifiers. In this type of query, it is important to efficiently transmit the query message to the target data item’s owner node (i.e., location management of data items). There are also more complex types of query, which specify certain query conditions that define the data items sought by the query-issuing node. \(k\)-nearest neighbor (\(k\)NN) searches\(^{27}\) and top-\(k\) searches\(^{28}\) are typical examples of such complex queries.

\(k\)NN and top-\(k\) searches have been thoroughly studied in the database communities, where the main focus is how to reduce the computation overhead to find the search results from a massive volume of data. On the other hand, these two types of queries are also useful in multi-hop wireless networks such as wireless sensor networks (WSNs) and MANETs, where the main focus is different from that in databases, i.e., how to reduce communication overhead rather than computation overhead. This is because in WSNs and MANETs, the volume of data processed in the networks is generally not as huge as that in infrastructured networks (e.g., data centers), i.e., finding the result from the data is not so computational costly, but data traffic should be the key bottleneck because of limited resources such as network bandwidth and battery.

Here, \(k\)NN and top-\(k\) searches in WSNs have been recently well studied.\(^{29}-^{32}\) These existing studies proposed infrastructure-free query processing methods, which are efficient in retrieving search results with low message overhead (traffic). In these methods, nodes must exchange messages including node information (e.g., one-hop beacon messages including the location of the sender), in order to recognize their neighbors for query processing.

On the other hand, query processing in MANETs has not yet been thoroughly investigated. This is because MANETs possess notable characteristics, such as limitations on network bandwidth, and dynamic topology change due to the movement of mobile nodes. More specifically, due to high mobility of nodes in MANETs, it is impractical to frequently exchange beacon messages to accurately know changing locations of neighboring nodes. Therefore, existing techniques proposed in WSNs cannot be directly applied to MANETs. Addressing query processing in MANETs is a significant issue from both an academic and social perspective.

D. Security management. In complex types of queries, such as top-\(k\) searches, in-network processing techniques (e.g., data aggregation) are used for efficient query processing. In such situations, security is a significant issue in order to accurately process queries. For this aim, existing security techniques in MANETs are applicable.\(^{33}\) However, in addition, new types of security issues arise for query processing in MANETs.
For example, in top-k query processing, malicious nodes may elude the query processing protocol, and replace top-ranked data items with less valuable items, which we call a data replacement attack. This kind of attack is difficult to detect because the malicious nodes can still attack, even though they strictly follow the underlying network protocol (i.e., we cannot detect it with existing security techniques); and this kind of attack can significantly damage the system, because it can selectively remove top-ranked data items which should be included in the final top-k result.

Here, data replacement attacks can take place only in in-network query processing in multi-hop networks such as MANETs, WSNs, and peer-to-peer (P2P) networks, where relaying nodes aggregate the intermediate query results for efficient query processing. Among MANETs, WSNs, and P2P networks, data replacement attacks are very serious only in MANETs, because in the other two types of networks, the network topology does not change very frequently, and thus, malicious behaviors of adversary nodes are easily detected by the neighbors. On the other hand, in a MANET, since the network topology (i.e., neighbors) frequently changes, some new approaches are needed to handle data replacement attacks with low overhead and latency with limited resources such as network bandwidth.

E. Summary. Table 1 summarizes the MANET data management issues described above, and our pioneering studies with regard to each of these issues. In the following sections, we will discuss some typical research achievements in this regard. The techniques proposed in our studies have been designed with deep considerations of MANET features such as network partitioning, node participation/disappearance, limited network bandwidth, and energy efficiency. Our studies published in early 2000’s have developed a new research field as data management in MANETs. Also, our recent studies are expected to be significant guidelines which show new research directions.

Table 1. Data management issues in MANETs

| Issue | Description | Our work |
|-------|-------------|----------|
| (1)   | Replica placement | How to effectively place replicas of data objects | 35–39 |
| (2)   | Update management | How to manage versions of replicas and how to keep consistency of data operations | 40–42 |
| (3)   | Query processing | How to efficiently retrieve data items of interest | 43–47 |
| (4)   | Security management | How to achieve secure query processing | 34, 48, 49 |
| (1) + (2) | — | — | 37, 50, 51 |
| (1) + (3) | — | — | 45, 47, 52 |

1.3 Contributions of this review paper.

1.3.1 Contributions. The survey papers in Refs. 53, 54 well presented and categorized early studies on data replication in MANETs. These survey papers recognized data replication in MANETs as a new research topic, listed up fundamental technical issues, and presented a number of typical studies on this topic. Specifically, these categorized the existing studies based on various fundamental technical issues including:

1. Decentralized (or centralized)
2. Dealing with node disappearance and network partitioning (or not)
3. Considering stability of wireless links (or not)
4. Considering data update (or not, i.e., read-only)
5. Addressing energy efficiency (or not)
6. Considering geographical locations of nodes and/or data items (or not)
7. Addressing data retrieval efficiency, i.e., data access latency (or not)

Starting from the paper in Ref. 35, we have published a number of papers addressing the above technical issues in the earliest stage. These papers were presented in the above survey papers using lots of space, as representative pioneering works. In this review, our works presented in section 2 and section 3.1 cover (address) most of the above technical issues. In addition, others’ studies presented in Refs. 53, 54 and most of very recent studies (published after those survey papers) also focused on those technical issues. For example, recent data/replica allocation schemes in MANETs aim to improve data availability and data access latency based on similar ideas as the traditional existing studies, but some new criteria such as node selfishness, information density, and social relationship are taken into account.

On the other hand, our works presented in and after section 3.2 were based on the papers published in and after 2009, which were not covered by the survey papers. These works firstly tackled new
technical issues, which were not addressed by anyone before us. Therefore, we believe that we have significantly contributed to the advancement of this research area (i.e., data management in MANETs). The main objective of this review is to publicize the new technical issues in MANETs to the research communities, and to encourage researchers and engineers to address the new issues, contributing to the technical advancements and application developments in MANETs.

1.3.2 Scope. It should be noted that research on data management and replication in MANET presented in this review strongly relates to researches in iMANET (a special type of MANET in which some nodes have a connection to a stable network infrastructure, i.e., the Internet)\(^{(63,64)}\) and DTN (Delay Tolerant Network)\(^{(65)}\) because all the three research areas assume wireless multi-hop networks and share some common technical issues such as resource constraints and dynamic change in network topology including network partitioning. Since all of them basically assume a MANET (or iMANET), all of them can be recognized as different types of research on data management in MANET.

Specifically, the iMANET and DTN research areas have different focuses compared with the MANET data management research. For example, the iMANET research mainly focuses on balancing response time (i.e., query delay) and communication cost because nodes can download data of interest from not only the Internet but also other nodes through wireless multi-hop communications.

The DTN research mainly focuses efficient data distribution (i.e., push-based data access) where there is a trade-off between the quickness and communication cost, while data replication in MANETs mainly focuses pull-based data access. The main advantage of the DTN-based approach is that it can achieve more efficient data distribution (e.g., emergent information dissemination), and thus, useful in some disaster situations. On the other hand, the MANET-based approach is more effective for general purposes (not only data dissemination) where various data operations such as real-time data retrieval and advanced query processing (e.g., top-\(k\) and \(k\)NN queries) are performed. Therefore, some recent works have focused on integrating MANET and DTN approaches\(^{(66)}\).

In this review, in order to make our scope clear, we just focus on data management issues for pull-based data access in pure MANETs where we do not assume any nodes connecting to the Internet.

2 Replica placement

The first issue is how to effectively and efficiently allocate replicas to nodes in MANETs, which was the first MANET data management challenge addressed by our research group\(^{(35)}\). In this section, we summarize the three approaches to MANET replica placement, which we proposed in Ref. 35.

2.1 Three approaches proposed in Ref. 35. 2.1.1 Preliminary. We assume a general MANET model where nodes move freely and a pair of nodes can directly communicate with each other through a wireless radio link when these are within the communication range of each other. Thus, the network topology dynamically changes. In the network, if two nodes have a multi-hop link between them, these nodes can also communicate with each other through intermediate nodes which exist between them and relay communication packets.

In this subsection, for simplicity, we assume an environment where the original of each data item is held by a particular node, and is not updated. Each mobile node has limited storage (or memory) space in which to replicate original data items held by others.

When a mobile node issues an access request for a given data item, the request is successful if either (i) the mobile node holds the original/replica of the data item, or (ii) at least one mobile node, which is connected to the requesting node through a one-hop or multi-hop link, holds the original/replica. Thus, the requesting node first confirms whether it holds the original/replica of the target data item; and if it does, the request succeeds on the spot. If it does not, the node sends a request message for the target data item, to connected mobile nodes. If it receives a reply from some other node(s) holding the original/replica of the target data item, the request is also successful. Otherwise, the request fails. Note that in this paper, mobile nodes connected to each other by one-hop or multi-hop wireless links are simply called connected mobile nodes.

In this subsection, we assume the following system model and notations.

- The set of all mobile nodes in the system is denoted by \(M = \{M_1, M_2, \cdots, M_m\}\), where \(m\) is the total number of mobile nodes, and \(M_j\) \((1 \leq j \leq m)\) is a node identifier. Each mobile node moves freely.
- Each pair of nodes can directly communicate with each other when these are within the communication range of each other. In addition, if two nodes have a multi-hop link between
them, these also can communicate with each other.

Due to node mobility, the network topology dynamically changes, and network partitioning may occur.

- Each mobile node, $M_i$, has a storage space of $C_i$ data items for replica allocation, in addition to the space for the original data items held by the node.

- The frequency of access to each data item by mobile nodes is known, and does not change. This assumption is not always realistic, but in some situations (applications), the speed of access frequency change is slow enough to assume that access frequencies to data items do not change in a short period. Actually, this assumption has been often made in existing studies in the database and distributed system communities.\textsuperscript{56,67} In a real environment, the access frequency can usually be known by maintaining logs of access requests at each node. Many existing studies adopted such a log-based access frequency estimation\textsuperscript{68,69} and proved that it works well. Also, in real systems (e.g., GlassFish application server), this technique is often used for load estimation and other purposes.

- Data is handled in the form of data items, which are collections of data. The set of all data items is denoted by $D = \{D_1, D_2, \cdots, D_n\}$, 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, and each original data item is held by a particular mobile node.

- The data items are not updated. There are many real situations where data items are not updated or are updated with enough low frequency so that we can assume that they are not updated, e.g., user generated documents such as SNS posts and web contents, and results of assigned tasks in a collaborative work. Many existing studies\textsuperscript{56,62} made this assumption. Of course, there are also many applications where data items are updated. We have also addressed the issue of data management in MANETs in such situations (see section 2.3, section 3, and Refs. 37, 40–42, 50, 51).

2.1.2 Three replica placement methods. In terms of data access request success rate, the replica placement problem is a form of resource allocation problem in a distributed networked system, which is well known to be NP-hard. In our study, the replica placement problem is considered as a problem in determining which data items are to be replicated at each mobile node (each of which has limited storage space), in order to maximize the data access request success rate.

In order to determine the optimal assignments among all possible combinations of replica allocation, we must analytically determine the combination which produces the highest data access request success rate. The computational complexity here is very high, and this calculation must be performed every time the network topology changes due to mobile node migration, which is impractical in real-world situations. For these reasons, we took the following heuristic approach in Ref. 35:

- Replicas are periodically relocated with a specific interval (relocation period).

- During each relocation period, replica allocation is determined based on the access frequency to each data item by each mobile node, and (optionally) the network topology at the time. Based on this approach, we proposed three replica placement methods, which differ in the emphasis put on access frequency and network topology.\textsuperscript{35}

1. SAF (Static Access Frequency) method: Only the access frequency to each data item by each node is taken into account.

2. DAFN (Dynamic Access Frequency and Neighborhood) method: The access frequency to each data item and the neighborhood of mobile nodes are taken into account.

3. DCG (Dynamic Connectivity based Grouping) method: The access frequency to each data item and the entire network topology are taken into account.

A. Static Access Frequency (SAF) method. In the SAF method, each mobile node $M_i$ allocates replicas of $C_i$ data items, in descending order of its own access frequency to the data items. That is, each node allocates replicas in a selfish manner, and does not take into account which data items are replicated by other nodes.

Now, let us suppose that six mobile nodes ($M_1, \cdots, M_6$) are present and $M_i$ ($i = 1, \cdots, 6$) holds $D_i$ as an original copy. The access frequency to each data item by each mobile node is shown in Table 2. Figure 3 shows the result of executing the SAF method. In this figure, a straight line denotes a wireless link, a gray rectangle denotes an original data item, and a white rectangle denotes an allocated replica.
In the SAF method, mobile nodes do not need to exchange information with each other for replica placement. Moreover, replica relocation basically does not occur after replica placement is completed. As a result, this method allocates replicas with low overhead and low traffic. On the other hand, since each mobile node allocates replicas based solely on its own access frequencies to data items, mobile nodes with the same access characteristics allocate the same replicas; and the resultant volume of replica duplication leads to a low data access request success rate, especially when many mobile nodes have similar access characteristics.

B. Dynamic Access Frequency and Neighborhood (DAFN) method. To solve this problem with the SAF method, the DAFN method eliminates the replica duplication among neighboring mobile nodes. Here, each node still basically pursues its own ends, but at the same time at least partially collaborates with others toward the global end.

Initially, this method determines replica allocation in the same manner as the SAF method. However, if there is replica duplication of a data item by two neighboring mobile nodes, the node with the lower access frequency to the data item changes the replica to another replica (a replica of the data item with the next highest access frequency). Since the neighboring status changes as mobile nodes move, the DAFN method is executed during each relocation period. The order of pairs for eliminating replica duplication is determined based on the order of visiting nodes, using a breadth-first search in the MANET, which begins with the mobile node with the lowest node identifier suffix, and ends when all the connected mobile nodes have been traversed.

Figure 4 shows an example of executing the DAFN method in the environment described by Table 2 and Fig. 3. In Fig. 4, a dark gray rectangle denotes a replica allocated to eliminate replica duplication. While six types of replica were allocated network-wide in the SAF method, seven are allocated in the DAFN method; and by eliminating replicas duplicated by neighboring nodes, the data access request success rate is expected to be higher than in the SAF method.

However, the DAFN method does not completely eliminate replica duplication among neighboring nodes, because it only executes the elimination process by scanning the network once, based on the breadth-first search. Thus, in Fig. 4, we can see duplication of replica $D_7$ by $M_4$ and $M_5$, and of $D_4$ by $M_1$ and $M_6$. And here, both the overhead and traffic are higher than in the SAF method, because mobile nodes exchange information and relocate replicas during each relocation period.

C. Dynamic Connectivity based Grouping (DCG) method. The DCG method shares replicas among larger groups of mobile nodes than the DAFN method, which shares replicas only among neighboring nodes. In this method, then, nodes behave with the greatest attention to the global end.

In order to share replicas effectively, each group should be stable (i.e., the group is not easily partitioned due to changes in network topology). To this end, the DCG method creates groups of mobile nodes that are biconnected components in the given network. A biconnected component denotes the maximum partial subgraph which is connected...

| Data | Mobile node | $M_1$ | $M_2$ | $M_3$ | $M_4$ | $M_5$ | $M_6$ |
|------|-------------|-------|-------|-------|-------|-------|-------|
| $D_1$ | 0.50        | 0.25  | 0.30  | 0.35  | 0.25  | 0.20  |
| $D_2$ | 0.45        | 0.50  | 0.40  | 0.40  | 0.40  | 0.45  |
| $D_3$ | 0.35        | 0.45  | 0.50  | 0.25  | 0.45  | 0.35  |
| $D_4$ | 0.30        | 0.15  | 0.10  | 0.60  | 0.10  | 0.25  |
| $D_5$ | 0.50        | 0.20  | 0.15  | 0.25  | 0.70  | 0.20  |
| $D_6$ | 0.05        | 0.35  | 0.45  | 0.20  | 0.25  | 0.60  |
| $D_7$ | 0.40        | 0.25  | 0.30  | 0.30  | 0.35  | 0.20  |
| $D_8$ | 0.20        | 0.30  | 0.20  | 0.25  | 0.30  | 0.20  |
| $D_9$ | 0.20        | 0.20  | 0.20  | 0.25  | 0.25  | 0.20  |
| $D_{10}$ | 0.15       | 0.10  | 0.05  | 0.15  | 0.15  | 0.10  |
of a performance study,\textsuperscript{35}) which was done through a simulation. In the simulation, 40 mobile nodes exist in a size $50 \times 50$ (corresponding to about $500$[m] $\times 500$[m]) flatland, each of which holds an original data item and has a storage to replicate 10 data items. These nodes initially locate at random positions and move according to the random walk model (which randomly determines the movement speed and direction at every time step). Due to the limitation of space, we omit the detail of the simulation setting (see the detail in Ref. 35).

Figure 6\textsuperscript{(a)} shows the performances of the three methods when varying the radio communication range of each node, where 1 corresponds to about 10[m]. The performance metrics are data accessibility (Fig. 6(a)) and traffic (Fig. 6(b)). The data accessibility is defined as the ratio of the number of successful data requests to the total number of data access requests issued during the simulation time. The traffic is defined as the total hop-count of data transmission for allocating/relocating replicas.

Figure 6(a) shows that as the radio communication range gets longer, the data accessibility increases in every method. When the communication range is very long, every method also gives almost the same data accessibility. This is because most mobile nodes are connected to each other, and thus mobile nodes can access original data items in most cases. The DCG method gives the highest data accessibility and the DAFN method gives the next highest.

From Fig. 6(b), as the radio communication range gets longer, the traffic caused by the DAFN method and the DCG method also gets larger at first, but it gets smaller from a certain point. In most situations, the DCG method produces the highest
When the radio communication range is very small, the traffic produced by these two methods is small. This is because the number of mobile nodes connected to each other is small, and thus replica relocation does not produce large traffic. When the radio communication range is very long, the DCG method produces smaller traffic than the DAFN method. This is because in the DCG method, the number of mobile nodes in a group is very large (40 in most cases) and thus replica relocation rarely occurs.

From the simulation result, we can confirm that the DCG method gives the highest accessibility, while the SAF method produces the lowest traffic. The DAFN method shows the balanced performance (i.e., the second best for both data accessibility and traffic) where the radio communication range is not very long (i.e., the number of neighbors of each node is not very high). Therefore, in a real environment, a proper method among the three methods should be chosen based on the system requirement, i.e., how much data accessibility and traffic are critical. For example, when either network bandwidth or node battery is very limited, the SAF method should be the best. On the other hand, when both the network bandwidth and the battery are rich and/or the data accessibility is critical, the DCG method should be the best.

2.2 Extensions. After our first paper on replica placement,\textsuperscript{35} we extended the methods proposed there, in several respects, including remaining batteries life and mutual dependency among data items. In Ref. 36, we took into account the fact that in many applications there are dependencies among data items; that is, multiple data items are often accessed at the same time. For example, at a disaster site, two data items, relating respectively to structural damage information and the progress of rescuer activity at the same location, are often accessed simultaneously. Thus, in the extended methods, we replaced the data access frequency to each data item, with the access frequency to pairs of data items accessed simultaneously.

In order to prolong the lifetime of MANETs, in Refs. 38, 39, we took into account the remaining battery life of mobile nodes, both when allocating data items to mobile nodes, and when determining the data items (replicas) to be accessed for a given data access request. By doing so, the proposed methods can prevent mobile nodes from exhausting their batteries, which is very important in MANET environments, because such battery exhaustion has a negative impact, both in terms of data availability (i.e., data items held by such nodes cannot be accessed) and network connectivity (i.e., these nodes cannot relay communication packets, which may cause network partitioning).

2.3 Replica placement considering data update. The above studies basically assumed a simple environment where data updates do not occur. We extended the methods proposed in Ref. 35 to take data updates into account when determining data replication.\textsuperscript{37,50,51}

The common idea among these studies\textsuperscript{37,50,51} is that the remaining time until each data item is updated next is taken into account when deciding data items replicated. More specifically, this remaining time is defined as the profit to replicate the data item as well as the data access frequency. In Ref. 50, we assumed the simplest situation that each data item is periodically updated (i.e., the remaining time until the next update is easily calculated from the update interval and the latest update time). In

![Fig. 6. Performance comparison among SAF, DAFN, and DCG.\textsuperscript{35}](image-url)
3 Update management

In Section 2 (except for Section 2.3), for the purpose of simplicity, we assumed an ideal environment where data items are not updated. However, in many real-world applications, data updates occur; and in MANETs, due mainly to nodal movement, disappearance (disconnection) of nodes and network partitioning frequently occur, and thus it is quite difficult to consistently and completely update all the versions of the replicas. And if non-updated versions of replicas exist, data access (read) operations will read old replicas, which is invalid in most applications. Reading old replicas is wasteful in terms of system features (e.g., consuming network resources and mobile node batteries), and requires the re-conduct of (valid) read operations on up-to-date replicas. Moreover, in the case of many real-world applications, reading old data items is simply unacceptable, as it may have a seriously negative impact on real-world activities; for example, rescue operations based on out-of-date information may be significantly impaired.

Therefore, the second issue is how to efficiently manage data updates in MANETs. There are basically two types of approach to update management in traditional database systems, which are also applicable in MANETs: optimistic and pessimistic.

In an optimistic approach, when a mobile node issues an access (read) request for a data item, but is not connected to the node holding the original copy, it tentatively accesses one of the replicas held by mobile nodes to which it is connected. The validity of the tentative access (i.e., whether it accesses an up-to-date replica) is then confirmed after the request-issuing node connects to the original item holder. As this form of update management often results in read operations that access stale (old) replicas, which may require rollbacks of the performed operations, it is typically difficult, if not impossible, to apply it in a MANET, where update (write) operations can be issued by any node. Therefore, in our research, we assume that optimistic approaches are used only in MANETs where the node holding the original copy issues write operations. In such an environment, there are basically two technical challenges: invalidation of old replicas, and dissemination of the latest (up-to-date) data items.

In a pessimistic approach, all access (read) operations must be confirmed to be consistent (valid) on the spot, during the operation period. However, it is very difficult to achieve this form of update management in MANETs, because of the frequency of nodal disconnections; and thus, further advances are needed in this area.

In this section, we summarize our studies on update management based on the two approaches above. Here we assume that all data items are updated at inconstant intervals; and after a given data item is updated, all its replicas basically become invalid, meaning that read operations are valid if and only if they are performed on up-to-date (i.e., valid) data items (replicas).

3.1 Optimistic approaches. Since, in optimistic approaches, the tentative read operations on replicas may be invalid, there are several performance metrics.

1. Read operations success rate (Success rate): As in the problem of replica placement, the read operation success rate is of course the most significant performance metric.

2. Rate of dirty read operations (Dirty-read rate): Since read operations on invalid (old) replicas (i.e., dirty reads) are not desirable, as many applications are aborted thereby, the rate of dirty read operations is also a significant metric. Even if the read operation success rate is high, some applications cannot accept too high dirty-read rate. Furthermore, increasing the success rate is generally accompanied by an increase in the dirty-read rate (i.e., there is a trade-off).

3. Delay until the validation of tentative read operations (Delay): Tentative read operations on replicas are only later validated, and thus the read-operation issuer incurs some delay until the validation. For most applications, a shorter delay is desirable.

4. Extra traffic (Traffic): To improve the results of the performance metrics above, a number of useful techniques have been developed, including invalidation of old replicas, and dissemination of the latest (up-to-date) data items. However, such approaches incur extra overhead in terms of communication traffic; and large
volumes of traffic make significant consumption
demands on network bandwidth and energy,
which are not desirable in MANETs.

In the following, we summarize our studies on
optimistic approaches to update management, which
aim to improve the values in the above metrics.
Again, in optimistic approaches, we assume that only
the node holding the original copy issues write oper-
ations (i.e., updates data items). We also present the
result of a performance study to show the e-
tiveness of our approaches. Due to the limitation of space,
we only show it for updated-data dissemination.

3.1.1 Invalidation of old replicas. In order
to reduce the dirty-read rate and delay, with low traffic
overhead, invalidating old replicas is often used in
traditional distributed systems and mobile systems.
In Ref. 41, we proposed two old-replica invalidation
methods, which involve the broadcast of a message
(invalidation report) containing the time stamp of
the latest version of the original copy, in order to
invalidate old replicas. We called these the Update
Broadcast (UB) and Connection Rebroadcast (CR)
methods. Here, we assume that each mobile node
manages a table in which information on the time
stamps of all the data items in the entire MANET is
recorded. A time stamp is the latest update time of
the corresponding data item, known by the mobile
node, which may differ from the actual latest update
time. This table is called the time stamp table.

By broadcasting an invalidation report more
frequently, we can reduce the dirty-read rate and
delay more effectively, but the traffic increases. The
two methods, UB and CR, adopt different strategies
on how to optimize the combination of dirty-read
rate, delay, and traffic. We outline the two methods
below. It should be noted that the old-replica
invalidation methods have no impact on the first
metric (i.e., success rate), because the methods do
not increase the number of valid replicas, but simply
invalidate old replicas.

A. Update Broadcast (UB) method. In the UB
method, a mobile node holding an original copy
broadcasts an invalidation report to connected mobile
nodes each time it updates the data item. The inval-
dication report includes the following information:

- the update identifier,
- the update time (time stamp).

If a mobile node that receives the invalidation
report holds a replica of the corresponding data item,
the node updates the entry. It also updates its own
time stamp table.

B. Connection Rebroadcast (CR) method. In the
CR method, similarly to the UB method, a mobile
node holding an original copy broadcasts an invalid-
dation report to connected mobile nodes each time it
updates the data item. But in addition, whenever two
mobile nodes are newly connected with each other,
they re-broadcast the invalidation reports they have
previously received. Figure 7 shows an example of
executing the CR method.

More specifically, the two newly connected
mobile nodes share their respective time stamp
tables, and each compares the respective entries for
each data item. If the time stamp for a data item in
the received time stamp table is more recent (i.e., the
other node has more recent information on that data
item), the node updates the entry.

Then, each of the two nodes re-broadcasts, to
the previously connected mobile nodes (before the
current new connection), invalidation reports for
those data items whose received time stamps were
greater (i.e., newer) than its own. Mobile nodes that
receive these invalidation reports discard their
replicas in the same manner as in the UB method.

Thus, in the CR method, connected mobile
nodes can maintain the same time stamp tables,
because invalidation reports are re-broadcast when-
ever two mobile nodes are newly connected. More-
over, invalidation reports are disseminated among
a larger number of mobile nodes than in the UB
method, and old replicas are effectively discarded
even if the replica owners are not connected to the
mobile nodes holding the original copies. Thus, this
method can further reduce the number of dirty-read
operations, in comparison with the UB method. However, when the network topology frequently changes, the traffic caused by sending invalidation reports is much higher in the CR method, due to the frequent broadcasts of the reports.

C. Data access. When a node issuing a read operation (access request) connects with the node holding the original copy of the target data item, it performs the operation on the original copy. If, on the other hand, it connects with a replica holder, it performs a tentative operation on the replica with the latest time stamp; and when it later connects with the mobile node holding the original copy, the success (or failure) of the tentative data read is confirmed.

3.1.2 Updated-data dissemination. As aforementioned, the old-replica invalidation methods presented above can reduce the dirty-read rate and delay, with low traffic overhead, but cannot increase the success rate, because they simply invalidate old replicas. Thus, in Refs. 42, 71, we proposed a number of updated-data dissemination methods, to increase the success rate by efficiently refreshing old replicas. Updated data items are disseminated after completing the old-replica invalidation procedures proposed in Ref. 41. The basic idea is very simple, with updated data items as well as invalidation reports being disseminated.

To this end, our proposed methods follow similar strategies to the old-replica invalidation methods described in Ref. 41, regarding data dissemination (i.e., disseminating data after data updates and new connections), with one important difference: here we proposed two alternate approaches to data dissemination after new connections, because data items are typically much larger than invalidation reports, and this may lead to unacceptable traffic volume if we simply broadcast the updated data items to all connected mobile nodes. Let us summarize the methods proposed in Refs. 42, 71.

A. Dissemination on Update (DU) method. In the DU method, old-replica invalidation is first performed, as in the UB method. But here, each mobile node that discards one of its replicas refreshes it by requesting the updated (up-to-date) data item from the mobile node holding the original copy.

This produces just low traffic, because the dissemination of an updated data item is performed only when the data item is updated. However, since only mobile nodes connected to the original copy holder can receive the latest version, the success rate is not appreciably improved.

B. Dissemination on Connection (DC) method. In the DC method, the procedure followed when the original copy holder updates its own original copy is the same as in the DU method. But in addition, whenever two mobile nodes newly connect with each other, replica invalidation is performed as in the CR method, and these two nodes then disseminate the updated data items. And here, as aforementioned, we proposed two variations, which differ in terms of the range over which updated data items are disseminated.

DC/OO (DC/One-to-One) method: In the DC/OO method, only the two newly connected mobile nodes share updated data items with each other, after they have broadcast their invalidation reports.

DC/GG (DC/Group-to-Group) method: In the DC/GG method, two groups of mobile nodes, which were previously connected to the two newly connected mobile nodes, disseminate their own updated data items after broadcasting their invalidation reports (i.e., the dissemination range is the same as that of the invalidation reports). Obviously, this method generates much more traffic than the DC/OO method, but can refresh much more old replicas, and thus can further increase the success rate.

C. Performance study. We briefly present the result of a simulation study. In the simulation, 40 mobile nodes exist in a size 500[m] × 500[m] flatland, each of which holds an original data item. These nodes initially locate at random positions and move according to the random waypoint model (in which each node randomly determines a destination position and moves toward it with a randomly determined speed, and repeats this behavior). Due to the limitation of space, we omit the detail of the simulation setting (see the detail in Ref. 71).

Figure 8 shows the performances of the proposed methods when varying the average update period (U), where we assumed that updates of each data item occur at intervals based on an exponential distribution with mean U [s]. For the purpose of comparison, the performances when the flooding of invalidation reports and the dissemination of updated data items are not performed are shown as “NO.”

The performance metrics are data accessibility (Fig. 8(a)), rate of accessing invalid replicas (Fig. 8(b)), and traffic (Fig. 6(c)). The data accessibility (i.e., success rate) is same as that in section 2.1.3. The rate of accessing invalid replicas (i.e., dirty-read rate) is defined as the ratio of the
number of invalid data requests to old replicas to the total number of data access requests issued during the simulation time. The traffic is defined as the total data volume for transmitting updated data items and control packets.

Figure 8(a) shows that the data accessibility in the DC/OO and DC/GG methods is higher than that in the DU method. The DC/GG method provides the highest data accessibility, since updated data items are disseminated to all mobile nodes that were originally connected to the two newly connected mobile nodes. As the average update period increases, in all of the proposed methods, the data accessibility improves because the replicas held by each mobile node are valid for a longer time.

Figure 8(b) shows that the rates of accessing invalid replicas in the DC/OO and DC/GG methods are lower than that in the DU method. This is because the DC/OO and DC/GG methods can invalidate more old replicas. The DC/OO method gives lower rate of accessing invalid replicas than the DC/GG method. This is because the DC/GG method not only allocates more valid replicas, but also allocates more invalid replicas than the DC/OO method.

Figure 8(c) shows that the DC/GG method produces the highest traffic and the DC/OO method produces the next. This result is obvious because the DC method disseminates updated data items more frequently than the DU method, and the DC/GG method disseminates them to wider ranges than the DC/OO method.

In summary, the simulation result shows that the DC method reduces the rate of accessing invalid replicas, but produces higher traffic than the DU method. The result also shows that the DC/GG method gives the highest data accessibility, but it also gives the higher rate of accessing invalid replicas than the DC/OO method and the highest traffic. In real environments, the most appropriate method should be chosen among the proposed methods according to the update frequencies of data items and system requirements.

3.2 Pessimistic approaches. As aforementioned, some applications require that the validity of data operations (read and write) must be confirmed immediately (i.e., tentative operations are not acceptable). Therefore, in a pessimistic approach, the validity (consistency) of each performed data operation is confirmed on the spot, during the operation period. Meanwhile, strict global consistency of data operations on replicas is not desirable in many applications, as it is too costly and difficult to achieve. Thus, new consistency maintenance, based on local conditions such as location and time, must be investigated. In Ref. 40, we attempted to classify different consistency levels, according to specific application requirements, and provide protocols to achieve these levels.

Here, we briefly describe these consistency levels and protocols. First, we outline the system model assumed, and then describe the proposed approaches.

3.2.1 System model. In Ref. 40, we assumed that the entire area in which MANET mobile nodes are present is divided into several regions (e.g., grid-based square regions). This assumption is based on
the fact that it is usually difficult to centrally maintain consistency over the entire network. We also assumed that the MANET consist of two kinds of mobile nodes: proxies and peers, where proxies are specially designated peers who manage other peers in a specific MANET region, and every node (including proxies) knows all the proxies in the network.

3.2.2 Global Consistency (GC). Data operation consistency is required over the entire MANET. A good example is a situation in which the members of a rescue service are divided into several groups, each of which is responsible for a certain region, and information on the progress of the tasks assigned to each group is shared over the entire network. To achieve GC, we adopt dynamic quorums similar to Ref. 72. The consistency is hierarchically managed at two levels: among the peers in a given region, and among proxies. First, the quorum size for write operations, \(|QW|\), and for read operations, \(|QR|\), over the entire MANET, are determined where the condition, \(|QW| + |QR| > l\), is satisfied, and where \(l\) is the total number of regions (proxies) in the entire MANET. In addition, in each region, \(R_i\) \((i = 1, \ldots, l)\), the quorum size for write operations, \(|QLW_i|\), and for read operations, \(|QLR_i|\), in the region are determined where the condition, \(|QLW_i| + |QLR_i| > P_i\), is satisfied, and where \(P_i\) is the total number of peers in the region.

The node which issues an operation (write or read) first sends a request message to the proxy of its region, which we call the coordinator. Then, the coordinator attempts to set the necessary number of local locks, \(|QW|\) and \(|QR|\), for replicas held by peers in its region of responsibility. If it succeeds, the global lock is set in the region. At the same time, the coordinator successively forwards the request message to other proxies, until it successfully sets the requisite number of global locks, \(|QW|\) and \(|QR|\), with each proxy that receives the request attempting to set the necessary number of local locks \((i.e.,\) set the global lock). Finally, if the coordinator succeeds in setting the necessary numbers, \(|QW|\) and \(|QLW|\) of global and local locks, among proxies and peers, the write (read) operation is performed on the replicas for which the global and local read (write) locks have been set. In read operations, the operation is performed on the most recent version of a given replica, among those with locks. In write operations, the operation is performed on all the replicas with locks. Based on the above-mentioned conditions, \(|QW| + |QR| > l\) and \(|QLW| + |QLR| > P_i\), the consistency of data operations can be maintained among both proxies and peers in each region; that is, a peer which issues a read operation can always read the most recent version of a given replica.

Figure 9 shows an example of executing this GC protocol, where a gray node denotes the proxy in each region \((R_1, \ldots, R_6)\) and arrows denote message flows among proxies. Let us assume that \(|QW| = 6, |QR| = 4\) \((i.e.,\) \(|QW| + |QR| = 10 (> 9))\), \(|QLR| = [P_i]\), and \(|QLW| = P_i - |QLR| + 1\). In this example, since the proxy in the region where the operation (write)-issuing node (the right upper node in \(R_i\)) locates successfully sets \(|QW|\) global locks with six regions \(R_1, \ldots, R_6\), where the proxy in each region successfully sets \(|QLW|\) local locks. Therefore, in this example, the write operation is successfully performed.

3.2.3 Local Consistency (LC). Data operation consistency is required only in each region of interest, and this consistency level weakens the strictness of consistency from a spatial perspective. An example of an application environment requiring LC is a situation in which the members of a rescue service share damage information, such as the number of injured persons and damaged buildings, which consists of distinct data items reflecting the varying extent of the damage. This information is used locally by the leaders in each region, to decide on resource allocation and task schedules in their respective groups.

To achieve LC, consistency is maintained only among peers in each region, in a manner similar to GC.

3.2.4 Time-based Consistency (TC). In TC, replicas are valid even if their versions are different but a predetermined time (validity period \(T\)) has not yet passed since they were last updated.

![Fig. 9. An example of executing the GC protocol.](image-url)
consistency level weakens the strictness of consistency from a temporal perspective.

In TC, typically, read and write operations are performed locally on the operation issuing peers. A read operation is performed if the operation issuing peer holds a valid replica. Otherwise, the node searches for a mobile node that holds a valid replica.

3.2.5 Peer-based Consistency (PC). Here, data operation consistency is required only in each peer. Thus, this consistency level further weakens the strictness of consistency, in comparison with LC, and is the weakest from a spatial perspective.

In PC, read and write operations are performed locally on the operation issuing peers.

3.2.6 Performance study. We briefly present the result of a simulation study. In the simulation, 240 mobile nodes exist in a size X x 4X/3[m] flatland (a region is a rectangle of X/3 x X/3), each of which holds an original data item and has an unlimited storage (i.e., it can replicate all data items). These nodes initially locate at random positions in their assigned region (20 nodes are assigned to each region), and move according to the random waypoint model within the region. Due to the limitation of space, we omit the detail of the simulation setting (see the detail in Ref. 40).

Figure 10 shows the performances of the four consistency management protocols (GC, LC, TC, and PC) when varying the area size (X) [m]. The performance metrics are success ratio (Fig. 10(a) and (b)), message traffic (Fig. 10(c) and (d)), and data traffic (Fig. 10(e) and (f)). We measured these metrics for both read and write operations. The success ratio is the same as data accessibility in section 2.1.3. The message traffic is defined as the average of the total hop-count for message exchanges to process a read/write operation excluding transmissions of data items. The data traffic is defined as the average of the total hop-count to transmit a data item to perform a (successful) read/write operation.

From Fig. 10(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 the connectivity among mobile nodes becomes lower. We can see an interesting fact that when the area size is larger than 450, the success ratio in GC suddenly gets lower but in LC remains high. This fact shows that even when the connectivity among mobile nodes is still high in each region, the connectivity among proxies becomes low, i.e., we should not choose unnecessarily strong consistency level if we wish to achieve high success ratio.

The success ratio of write operations in TC and those of write and read operations in PC are always 1 because every peer can perform operations 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 nodes cannot access valid replicas held by connected mobile nodes with high probability.

From Fig. 10(c) and (d), the message traffic of write and read operations in GC and that of write operations in TC first get higher and then get lower from a certain point (X = 450) as the area size gets larger. This is because the traffic firstly increases due to the increase of hop-count for communication, and then decreases due to the decrease of the number of neighbors (i.e., the network becomes sparse). Of the four protocols, GC produces the highest message traffic, and LC produces much lower than GC and TC (for write operations). Obviously, the message traffic of write operations in TC and those of write and read operations in PC are always 0.

From Fig. 10(e) and (f), the data traffic is much lower than the message traffic for both write and read operations, while the values do not represent actual traffic because the data size is not considered. If the sizes of messages and data items are given, it is determined which one is dominant. The data traffic of GC is much higher than LC for both write and read operations. The data traffic for read operations in TC is much higher than other protocols, because there are fewer valid replicas in TC, and request-issuing peers have to obtain valid replicas from far away peers.

In summary, we can confirm from the simulation result that all the four consistency management methods have quite different performance. Of course, the consistency level should be chosen based on the system requirement. The simulation result also suggests us that we should not choose unnecessarily strong consistency level because higher consistency level may cause significant degradation of success ratio and increase of traffic.

4 Query processing

As noted in Section 1.2.3, the process whereby data items of interest are found (i.e., query processing) is also a significant issue, because it directly affects the performance of the system or application. Here, focusing on complex types of queries, which specify certain query conditions that define the data items sought by the query issuing node, we describe our studies on top-k searches and k-nearest neighbor
\( kNN \) searches, which are typical examples of such queries. We also present the result of a performance study to show the effectiveness of our approaches. Due to the limitation of space, we only show it for \( kNN \) query processing.

4.1 Top-\( k \) searches. As it is important to efficiently acquire only necessary data items in MANETs, top-\( k \) queries offer a promising approach. In a top-\( k \) query, data items are ordered according to their score, calculated based on a specific set of attribute values using some scoring function, and the query-issuing node acquires the data items with the \( k \) highest scores. In Ref. 43, we proposed a message-processing method for top-\( k \) queries, which guaran-
tees an accurate query result \((i.e.,\) acquiring the data items with the \(k\) highest scores in the entire MANET\), while reducing the traffic as much as possible.

The basic means to achieve this consists in attaching a small but critical piece of information to each query message, which effectively narrows down the data item candidates to those included in the final top-\(k\) result. To this end, each mobile node roughly identifies data items with the \(k\) highest scores, anddesignates some of these scores as Standard Scores \((SS)\); then, as mobile nodes transmit query and reply messages, they reduce the number of candidates included in the top-\(k\) result, by referring to these SSs. Furthermore, if a mobile node detects the disconnection of a necessary radio link during the transmission of a reply message, it searches for an alternate path along which to transmit the reply message to the query-issuing node.

4.1.1 Determining the Standard Scores \((SS)\).
Each SS, \(B(i,j)(1 \leq j \leq N)\), calculated by mobile node \(M_b\), guarantees that some of the query-issuing node acquires more than \(\frac{k}{N}(1 \leq j \leq N)\) data items.

4.1.2 Query processing. The procedures for the query-issuing node, \(M_p\), and for the mobile nodes receiving a query message, are briefly explained below. First, \(M_p\) specifies the number of requested data items, \(k\), and the query conditions. Then, \(M_p\) calculates the scores of its data items based on the query conditions, using a scoring function, and initializes its SSs as follows:

\[
B(p,j) = S(p, k \cdot \frac{j}{N}) (1 \leq j \leq N). \tag{1}
\]

Here, \(S(i, h)\) denotes the \(h\)-th highest score among the scores calculated by \(M_b\). Specifically, the SSs of \(M_p\) are the every \(\frac{k}{N}\)-th scores calculated by \(M_p\).

Then, \(M_p\) transmits a query message, with the attached SSs, to its neighboring mobile nodes. Each mobile node that receives the query message updates the SSs based on the scores of its retained data items, and forwards the revised query message to its neighbors. The reply messages are sent back to \(M_p\) along the same routes through which the query was disseminated. Based on the information in the reply message, each mobile node that relays the reply sets its own threshold, ensuring that this is equal to or greater than the \(k\)-th highest score in the network, and sends to \(M_p\) its retained data items with scores equal to or greater than the threshold (Fig. 11).

In this way, mobile nodes can reduce the number of data item candidates included in the top-\(k\) result, which helps to reduce the traffic required for query processing.

4.1.3 Top-\(k\) query processing with replica placement. After Ref. 43, we have studied more efficient approaches for top-\(k\) query processing in MANETs. For example, in Ref. 47, we proposed some new top-\(k\) query processing methods which estimate the distribution of scores in the entire MANET and more precisely predict the \(k\)-th highest score than the method in Ref. 43 which roughly calculates the lower-bound of the \(k\)-th score using Standard Scores. For estimation of score distribution, these methods utilize either a histogram of scores or an approximation to a regular distribution based on the scores of data items obtained during query processing. In Ref. 47, we assumed an environment where data items are replicated to improve the query processing performance. Therefore, the proposed methods also take replication into account in query processing. Specifically, these methods try to avoid duplicate transmissions of replicas of same data items, and also minimize the length of paths along which the data items are replied. By doing so, we can reduce both the traffic and delay for query processing.

In Ref. 52, we addressed the data replication technique which is specially designed for top-\(k\) query processing, and proposed a simple replication scheme to avoid heavy replica duplication among nodes, which can occur if we simply replicate data items based on the scores. To this end, our scheme not only takes into account scores of data items, but also adopts a randomized approach to achieve diversity of replicas in the entire MANET.

4.2 \(k\)-Nearest neighbor search. Since MANETs are generally constructed of collaborating mobile users who are geographically distributed in the working area, location based queries such as that finding users near a specific location and that finding
data associated with a specific location (e.g., sensor data observed at the target location) are often used. However, there have been no studies addressing such queries in MANETs so far. Therefore, since 2010, we have addressed the issues of k-nearest neighbor (kNN) search in MANETs. kNN query is the most typical location based query, which retrieves the kNNs from a query designated location.

In Ref. 44, we addressed the issue of searching kNN nodes in MANETs. A naive approach for searching kNN nodes is that flooding the entire MANET with a query message and receiving a reply from each query receiving node, which includes the location of the node. Obviously, this produces too much unnecessary traffic, resulting in not only consuming a large amount of energy but also reducing the accuracy of the query result due to packet losses caused by the heavy traffic. Therefore, in Ref. 44, we proposed kNN query processing methods for reducing traffic and maintaining high accuracy of the query result in MANETs. These methods are based on the following key policy.

- Only mobile nodes that locate near the query specified location (i.e., ideally, only kNNs) participate in the query processing as much as possible.

To this end, in the proposed methods, the query-issuing node first forwards a kNN query using georouting to the nearest node from the location specified by the query (query point). Then, the nearest node from the query point forwards the query to other nodes close to the query point, and each node receiving the query replies with the information on itself. A possible way to achieve this is that every node continuously recognizes the locations of neighbor nodes by exchanging beacons (in other words, hello or heart-beat messages). However, in MANETs, since mobile nodes move freely, exchanging beacons frequently to precisely recognize neighbors’ locations produces unacceptably too much traffic. Thus, our methods were designed as beacon-less approaches. How to achieve this is explained later in this subsection.

4.2.1 Proposed methods. To achieve searching only nodes close to the query point, we proposed two approaches: the Explosion (EXP) method and the Spiral (SPI) method. In the EXP method, the nearest node from the query point broadcasts the query to nodes within a specific circular region, and each node receiving the query replies with information on itself (Fig. 12(a)). The circular region is determined based on the density of nodes in the MANET. If the node density around the query point is significantly different from that in the entire MANET, the EXP method cannot collect the kNN (low estimation) or collects the information on an unnecessarily large number of nodes (high estimation).

In the SPI method, the nearest node from the query point forwards the query to other nodes in a spiral manner, and the node that collects a satisfactory kNN result transmits the result to the query-issuing node (Fig. 12(b)). Thus, this method does not require to specify a circular region. To achieve this, the entire area is dynamically partitioned into a set of hexagonal cells whose size is determined based on the communication range of the mobile nodes (so that the information on nodes in a cell can be obtained through on-hop communication), with the query point at the center of a hexagonal cell.

In both methods, a designated node aggregates the information in the received replies, and thus unnecessary information is not sent in reply through a long path to the query-issuing node. In these ways, unnecessary transmissions of queries and replies can be reduced. Compared between the EXP and SPI methods, the EXP method generally achieves shorter query processing delay, while the SPI method generally achieves lower traffic in query processing.

4.2.2 Idea for making our methods beacon-less. The main idea of making the above methods beacon-less (i.e., processing queries without knowing neighbors locations) is that each node that received a query effectively sets the waiting time until replying to the query, based on the locations of the query-issuing node, query-relaying node, and query receiver. For example, in the EXP method, after broadcasting a query within the circular region, it is achieved that query replies start to be sent from farther nodes to closer nodes by setting the waiting time $RD$ as follows.
Fig. 13. Performance comparison (500 nodes).44)

\[ RD = \text{Max\_delay} \cdot \left( \frac{\alpha - d}{\alpha} \right) \]  

where Max\_delay is a positive constant specifying the maximum waiting time before sending a reply, \( \alpha \) is the radius of the searching range, and \( d \) is the distance between the source node’s location and the foot of the receiving node’s perpendicular to the line from the source node to the query point.

4.2.3 Performance study. We briefly present the result of a simulation study.44) In the simulation, 500 mobile nodes exist in a size 1000[m] × 1000[m] flatland. These nodes initially locate at random positions and move according to the random waypoint model. Due to the limitation of space, we omit the detail of the simulation setting (see the detail in Ref. 44).

Figure 13 shows the performances of the EXP and SPI methods when varying the number of requested \( k \)NNs (\( k \)). For a purpose of comparison, we also show the performances of the naive method and the DIKNN method as well as beacon-based versions of our methods. The naive method does not use beacons, but the query-issuing node first floods a query within the range, which is determined in the same way as the EXP method, and then each node receiving the query individually replies to the query-issuing node. The DIKNN method is the state-of-the-art of KNN query processing in WSNs, which is based on beacons. In the EXP and SPI methods using beacons, the node behavior is basically the same as in the beacon-less EXP and SPI methods, however messages are transmitted (unicast) based on the neighbor information obtained by beacons.

The performance metrics are traffic, response time, and accuracy of query result. The traffic is defined as the average total volume of query messages and replies exchanged in processing a query. The response time is defined as the average time from the transmission of a query message by the query-issuing node, to the reception of the \( k \)NN result. The accuracy of query result is defined as the average of the weighted ratio (MAP value73) of the number of \( k \)NNs whose information is included in the \( k \)NN result acquired by the query-issuing node, to the requested number of \( k \)NNs, \( k \).

From Fig. 13(a), as \( k \) increases, the traffic increases in all methods, because both the search area for processing a \( k \)NN query and the data volume of the reply increase. Our proposed methods generate far less traffic than the methods using beacons, as the periodical beacon exchanges involved in the latter cause a great deal of traffic. Our proposed methods also generate far less traffic than the naive method, because in our methods, replies are sent back to the query-issuing node in a more efficient manner. The EXP method produces more traffic than the SPI method, because in this simulation the estimated \( k \)NN circle is set large enough for safety, and thus many non-\( k \)NN nodes reply. In the SPI method, the traffic depends on the number of laps required for collecting the information on \( k \)NNs, and thus the traffic increases in a stepwise manner as \( k \) increases.

From Fig. 13(b), the response time in our proposed methods is greater than in the methods using beacons. In our methods, a node sets a waiting time before transmitting a message (which is a disadvantage of not using beacons), and thus the response time is increased. In the EXP method in particular, every node must wait for calculated waiting time before sending back a reply, which results in an increase in response time. In the naive method, such waiting times do not occur; however, in this method retransmissions of replies often occur, due to packet losses caused by the increased traffic. Overall, the response time in the naive method is roughly similar to that of the SPI method.
From Fig. 13(c), the accuracy of query result is very high (nearly 1) in our methods, in contrast to the lower accuracy of query result in the DIKNN and naive methods. This is because, in the latter, packet losses often occur due to individual replies from a large number of nodes.

In summary, our proposed beacon-less $k$NN query processing methods which were specially designed for MANETs perform significantly better than the naive method and the state-of-the-art approach for WSNs. Of our proposed methods, the SPI method achieves high performance when the node density is high (500 nodes). However, although not presented in this review, the result in Ref. 44 showed that the performance of the SPI method significantly degrades when the node density is low. Thus, the EXP method also has an advantage that it can stably achieve both reduction in traffic and high accuracy of the query result.

4.2.4 $k$NN query processing with replica placement. We extended the EXP method to retrieve location-dependent data items which are associated with some locations. Since a location-dependent data item is held by a particular mobile node, which also keeps moving, i.e., the data item’s current location is generally different from its associated location, some new techniques are needed for searching $k$NN data items by using the EXP method. Therefore, the method proposed in Ref. 45 extended the original EXP method as follows.

- The data items are managed to locate near their associated locations. Specifically, when the mobile node holding the original copy of a data item moves beyond $d$ from the data item’s associated location, the node passes it to the node closest to this location. By doing so, we can apply the EXP method by extending the radius of the circular region by $d$.
- Since data items can be replicated, which is effective to improve the search performance, each mobile node replicates $C$ data items whose associate locations are closer to itself, i.e., the replica maintenance is needed as the node moves. Specifically, each node relocate replicas every time when it moves distance $d$.

5 Security management

In our studies described in the above sections, we do not assume the presence of malicious nodes. If malicious nodes are present, the accuracy of query processing is expected to decrease. In Refs. 34, 48, 49, we defined a new type of attack for top-$k$ query, called data replacement attack (DRA), in which malicious nodes attempt to replace necessary data items with unnecessary data items, and proposed some novel techniques against DRA. Here, in top-$k$ query processing, a query-issuing node does not know the global top-$k$ result beforehand. Therefore, even if a malicious node has performed a DRA, the query-issuing node considers all the received data items with the $k$ highest scores to be the global top-$k$ result, rendering the DRA effectively undetectable, i.e., DRA is a stronger attack than other traditional forms of attack.

In this section, we present our approaches against DRAs, and also present the result of a performance study to show the effectiveness of our approaches.

5.1 Top-$k$ query processing and malicious node identification. When malicious nodes performing DRAs are present, we need to make the following actions to make the entire MANET robust against DRAs:

- keeping high accuracy of the query result,
- identifying malicious nodes.

The first action is needed to reduce the impact of DRAs from the application perspective. The second action is needed to remove the malicious nodes and keep the MANET healthy from the system perspective.

In Ref. 34, we proposed a top-$k$ query processing method (first action) against DRAs and a local malicious node identification method (second action). Moreover, in Ref. 49, we proposed a global malicious node identification method (second action) in which normal nodes which have detected malicious nodes share the information on the malicious nodes to more widely detect malicious nodes. Here, for the purpose of simplicity, we assumed a naive manner for top-$k$ query processing where the query-issuing node first broadcasts a query over the entire MANET and each nodes receiving the query sends back a reply with data items with the $k$ highest scores (local top-$k$ result) among its own data items and all data items received from its child nodes on the query propagation routes, i.e., simple data aggregation is performed.

5.1.1 Top-$k$ query processing. The idea for keeping high accuracy of the query result is simple but effective: each node receiving a query replies with data items with the $k$ highest scores along multiple (two) routes. By doing so, even if a malicious node is present along a query reply path, the query-issuing node can acquire the top-$k$ result. Our experimental results confirmed us that even if more than two
malicious nodes are present, just replying along two routes works well in most cases.

5.1.2 Local malicious node identification. To enable DRA and malicious node detection, we make each reply message include information on the route along which the message is forwarded. By doing so, the query-issuing node can know which data items should be sent back along the route, in other words, it can recognize a DRA. When detecting a DRA, the query-issuing node narrows down the malicious node candidates from the information attached in the received reply messages, and inquires with non-candidate nodes (neighbors of the candidates) information on the data items sent by these candidates, allowing it to identify the malicious nodes.

5.1.3 Global malicious node identification. Through simulation experiments, we confirmed that when there are multiple malicious nodes in a MANET, it is difficult for the method proposed in Ref. 34 to identify all the malicious nodes in a single query. This is partly because nodes, in this method, are more likely to only identify malicious nodes near their own location than those farther away. Thus, in order to rapidly identify a greater number of malicious nodes, a global malicious node identification method proposed in Ref. 49 makes nodes share information about identified malicious nodes with other nodes.

In this method, after receiving a predetermined number of queries, each node divides all nodes into some groups based on the similarity of the information on identified malicious nodes which was sent from them. Then, the node performs malicious node identification separately with each group, based on the information in the group, and comprehensively makes the final judgment of malicious nodes based on the identification results of all the groups. In this method, even if malicious nodes claim some normal nodes as malicious (which we call false notification attack (FNA)), there is a decisive difference in the nature of the information possessed by normal and malicious nodes concerning the identified malicious nodes, and therefore, the malicious nodes can be easily identified.

5.2 Signature-based top-k query processing against DRAs. By the method in Ref. 49, each node can identify a large number of malicious nodes more quickly than the local identification method in Ref. 34. However, the global malicious node identification is performed only after each node receives a predetermined number of queries, which still needs relatively long time to identify all malicious nodes. In addition, it sometimes happens that some normal nodes are determined as malicious by mistake.

To solve these problems and identify all malicious nodes more quickly, we proposed a signature-based top-k query processing method in Ref. 48. In this method, each node receiving a query message sends back a reply message which contains the local top-k result (i.e., tentative top-k data items) attached with encrypted information about the reply forwarding route (i.e., the list of nodes along which the reply message has been sent) and the sent data items (i.e., the list of data items newly added by this node and that deleted from the previous local top-k data items) as the digital signatures. By doing so, the query-issuing node can know the data items sent by each node in the MANET, and thereby can identify malicious nodes using the received signatures. After identifying the malicious nodes, the query-issuing node floods the MANET with a notification message including the signatures in which the identified malicious nodes replaced higher-score data items to their own lower-score items.

Figure 14 illustrates an example of forwarding reply messages with digital signatures, then detecting a DRA and identifying a malicious node, where SIGM denotes the digital signature by mobile node Mi. Here, we assume that malicious node M2 replaces the 88-score data item sent by M4 with its own 65-score data item. The query-issuing node, M1, examines the reply message from M2, and it is clear from SIGM that M2 has replaced the 88-score data item with its own low-score item. M1 thus detects a DRA and identifies M2 as a malicious node.

5.3 Performance study. We briefly present the result of a simulation study. In the simulation, 50 mobile nodes exist in a size 500[m] × 500[m] flatland, each of which holds 50 data items whose scores are randomly set. These nodes initially locate at random positions and move according to the random waypoint model. Due to the limitation of space, we omit the detail of the simulation setting (see the detail in Ref. 48).

Figures 15 and 16 show the performances of our signature-based method in Ref. 48 (denoted by MP-SIG), our non-signature-based method in Ref. 49 (denoted by MP-noSIG), a modified version of the method in Ref. 48 (denoted by SP-SIG), in which each node replies to only its parent (i.e., single-path-based method; SP), and the naive method (denoted by SIG-noSIG), which neither uses multi-path reply nor detects DRAs and malicious nodes.
In Fig. 15, we measured the number of queries which was necessary for detecting all malicious nodes when varying the number of malicious nodes (the number of requested top-\(k\) data items (\(k\)) was fixed as 30). Since the naive method (SP-noSIG) cannot detect malicious nodes, we omit the result. In Fig. 16, we fixed the number of malicious nodes as 5, and varied the number of requested top-\(k\) data items (\(k\)), and measured the three performance metrics; accuracy of the query result (the average ratio of the number of top-\(k\) data items included in the acquired top-\(k\) result, to \(k\)), traffic for top-\(k\) query processing (the average of the total traffic volume required for processing a top-\(k\) query), and traffic for notification (the average of the total traffic volume required for making notification of the identified malicious nodes).

Figure 15 shows that the signature-based method (MP-SIG) in Ref. 48 can detect all malicious nodes significantly faster than the non-signature-based method (MP-noSIG) in Ref. 49, which confirms us the effectiveness of using the signatures.

From Fig. 16(a), the accuracy of the query result in MP-SIG and MP-noSIG is greater than in SP-SIG and SP-noSIG, which shows the effectiveness of our multiple replies against DRAs. However, as \(k\) increases, the accuracy of the query result decreases in MP-SIG and MP-noSIG, because the chances of packet losses increase with the increase in the size of replies. In particular, MP-SIG achieves lower accuracy than MP-noSIG, and the difference between MP-SIG and MP-noSIG increases as \(k\) increases, because the size of the signatures in MP-SIG significantly increases, and thus packet losses more often occur.

From Fig. 16(b), in all methods, as \(k\) increases, the traffic required for top-\(k\) query processing increases because of the increase in the reply message size. From Fig. 16(c), as \(k\) increases, the traffic for notification increases in MP-SIG and SP-SIG because the size of the notification message increases with the increase in the size of the signatures. On the other hand, in MP-noSIG, even if \(k\) increases, the traffic required for notification is very small, as nodes send notification messages with only the information about identified malicious nodes (i.e., without signatures).

In summary, the simulation result shows the trade-off between quickness of malicious node detection (achieved by our signature-based method in Ref. 48) and high accuracy/low traffic (achieved by our non-signature-based method in Ref. 49). In real situations, we should carefully choose an appropriate method based on the system requirements (i.e., which performance metric is more critical in the system).
6 Concluding remarks

In this paper, we outlined our studies on data management in MANETs. In this section, as a summary of this paper, we first summarize the academic and social contributions of the studies. Then, we discuss some future directions of data management research in MANETs.

6.1 Contributions of our studies.

6.1.1 Academic contributions. A number of the studies summarized above have been acknowledged as pioneering works, which together have established a new field of research, on data management in MANETs. The study described in Ref. 35, in fact, was the first to address data replication techniques in MANETs; and following its publication, a lot of similar studies have been done, with most citing this study in their published reports.

After that initial study in Ref. 35, as summarized here, we addressed a variety of research issues related to data management in MANETs; and in most cases, published the first related papers in the research community.

Several survey papers on data management in MANETs have recently been published in leading journals and conferences, such as the VLDB Journal[44] and the IEEE Communications Surveys & Tutorials;[53] and in these surveys, our studies were evaluated highly as pioneering work.

Thus, our studies have contributed significantly to the advancement of academic research in this area.

6.1.2 Social contributions. Our studies also have significant social value, because technical advancement in MANET data management will significantly improve data availability and data access performance in important applications, such as those described in Section 1.2.2. In particular, the techniques proposed in our studies will significantly contribute to applications involving mobile-user collaboration, such as rescue operations at disaster sites and real-time information sharing among vehicles for safe/autonomous driving.

6.2 Future directions. As aforementioned, the techniques proposed in our studies have significant contributions from both the academic and social perspectives. However, to deal with more advanced MANET applications, further efforts are needed. In the last part of this paper here, we discuss some future directions. While of course, performance improvement of each of the techniques proposed by us is needed, we omit such discussion, but focus on other research directions.

6.2.1 Mobile crowdsensing/crowdsourcing. Recently, there has been a new trend of mobile crowdsensing/crowdsourcing in which a crowd of ordinary people having a mobile device are strongly involved in application missions. In mobile crowdsensing, mobile devices with sensing capabilities act as sensor nodes and help sensing operations. In mobile crowdsourcing, mobile users act as workers to conduct some tasks which are part of a large mission.

In MANETs, there are many applications in which mobile crowdsensing and crowdsourcing are useful, e.g., rescue operations. Therefore, addressing issues for mobile crowdsensing/crowdsourcing in MANETs will be an interesting and significant research direction, such as task assignment, task scheduling, resource allocation, and incentive mechanisms.

6.2.2 Integration with clouds. The data management techniques in MANETs are particularly useful in situations where no network infrastructures such as the Internet are available. However, even if the Internet is available, MANET technologies are still very useful, for example, for enlarging the coverage of the Internet and for off-loading of the Internet and server overhead, which have become hot research topics.
On the other hand, as MANET applications become more complex and advanced, it is easily expected that computation overhead to meet the application requirements significantly increases and exceeds the computation capability of mobile nodes. In such a situation, if the Internet connection is available with some MANET nodes (i.e., iMANET environments), it will be promising that the computation is off-loaded to clouds (and edge computers) in the Internet. Therefore, addressing issues for seamlessly integrating operations in MANETs and clouds will be challenging, such as task assignment/scheduling, consistency management, and fault recovery.

6.2.3 Real deployment. Now we are facing the period to deploy real data management applications in MANETs and verify the effectiveness of existing techniques in practical use. Rescue operations at a disaster site are typical and the most significant target especially in Japan.

References

1) Baker, D.J., Wieselthier, J. and Ephremides, A. (1982) A distributed algorithm for scheduling the activation of links in a self-organizing, mobile, radio network. Proc. IEEE ICC’82, 2F6.1–2F6.5.
2) Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y.C. and Jetcheva, J. (1992) A performance comparison of multi-hop wireless ad hoc network routing protocols. Proc. Mobicom’98, 159–164.
3) Kahn, R.E. (1975) The organization of computer resources into a packet radio network. Proc. National Computer Conference, 177–186.
4) Johnson, D.B. (1994) Routing in ad hoc networks of mobile hosts. Proc. IEEE Workshop on Mobile Computing Systems and Applications, 158–163.
5) Pearlman, M.R. and Haas, Z.J. (1999) Determining the optimal configuration for the zone routing protocol. IEEE J. Sel. Areas Commun. 17, 1395–1414.
6) Perkins, C.E. and Bhagwat, P. (1994) Highly dynamic destination-sequence distance-vector routing (DSDV) for mobile computers. Proc. ACM SIGCOMM’94, 234–244.
7) Perkins, C.E. and Royer, E.M. (1999) Ad hoc on demand distance vector routing. Proc. IEEE Workshop on Mobile Computing Systems and Applications, 90–100.
8) ASA (ALT) Public Affairs (2011) Army networking radios improve communications at tactical edge. U.S. Army.
9) Tactical Scalable MANET (TSM), TrellisWare Technologies, https://www.trellisware.com/manet-products/tsm/.
10) Anjum, S.S., Noor, R.M. and Anisi, M.H. (2016) Review on MANET based communication for search and rescue operations. Wirel. Pers. Commun.
11) Bluetronix, http://www.bluetronix.net/.
12) Alkunate, R., Ochoa, S.F., Pena-Mora, F. and Nussbaum, M. (2006) Robust mobile ad hoc space for collaboration to support disaster relief efforts involving critical physical infrastructure. J. Comput. Civ. Eng. 20.
13) Ochoa, S.F. and Santos, R. (2015) Human-centric wireless sensor networks to improve information availability during urban search and rescue activities. Inf. Fusion 22, 71–84.
14) Sakano, T., Kotabe, S., Komukai, T., Kumagai, T., Shimizu, Y., Takahara, A., Ngo, T., Fadhilah, Z.M., Nishiyama, H. and Kato, N. (2016) Bringing movable and deployable networks to disaster areas: development and field test of MDRU. IEEE Netw. 30, 86–91.
15) Willke, T.L., Tientrakool, P. and Mazencnuk, N.F. (2009) A survey of inter-vehicule communication protocols and their applications. IEEE Commun. Surv. Tutor. 11 (2), 3–10.
16) Zhang, D., Zhang, D., Xiong, H., Hsu, C.H. and Vasilakos, A.V. (2014) BASA: building mobile Ad-Hoc social networks on top of android. IEEE Netw. 28 (1), 4–9.
17) Wu, S.L. and Tseng, Y.C. (2007) Wireless Ad Hoc Networking: Personal-Area, Local-area, and the Sensory-Area Networks, Auerbach Publications.
18) Ibehara, C.S., Biagioni, E. and Crosby, M.E. (2007) Ad-hoc wireless body area network for augmented cognition sensors. Proc. Int’l Conf. on Foundations of Augmented Cognition (FAC 2007), 38–46.
19) Lambrou, T.P. and Panayiotou, C.G. (2009) A survey on routing techniques supporting mobility in sensor networks, Proc. Int’l Conf. on Mobile Ad-hoc and Sensor Networks (MSN 2009), 78–85.
20) Zhu, C., Shu, L., Hara, T., Wang, L., Nishio, S. and Yang, L.T. (2014) A survey on communication and data management issues in mobile sensor networks. Wirel. Commun. Mob. Comput. 14, 19–36.
21) Hui, P., Chaintreau, A., Scott, J., Gass, R., Crowcroft, J. and Diot, C. (2005) Pocket switched networks and human mobility in conference environments. Proc. ACM SIGCOMM Workshop on Delay-Tolerant Networking, 244–251.
22) Yau, S.S., Gupta, S.K.S., Gupta, E.K.S., Karim, F., Ahamed, S.L., Wang, Y. and Wang, B. (2003) Smart classroom: Enhancing collaborative learning using pervasive computing technology, Proc. ASEE Annual Conf. and Expo., 13633–13642.
23) Barrere, L., Chaumette, S. and Turbert, J. (2006) A tactical active information sharing system for military MANETs, Proc. IEEE Military Communications Conf. (MILCOM 2006), 3621–3627.
24) Burbank, J.L., Chimeno, P.F., Haberman, B.K. and Kasch, W.T. (2006) Key challenges of military tactical networking and the elusive promise of MANET technology. IEEE Commun. Mag. 44 (11), 39–45.
25) Mohamad, O.A., Hameed, R.T. and Taps, N. (2015) Smart home system based on comparative analysis among AODV and DSDV protocols in MANET, Proc. Int’l Conf. on System Theory,
28) Ilyas, I.F., Beskales, G. and Soliman, M.A. (2008) A survey of top-k query processing techniques in relational database systems. ACM Comput. Surv. 40 (4), article 11.
29) Fu, T.-Y., Peng, W.-C. and Lee, W.-C. (2010) Parallelizing itinerary-based KNN query processing in wireless sensor networks. IEEE Trans. Knowl. Data Eng. 22, 711–729.
30) Silberstein, A., Munagala, K. and Yang, J. (2006) Energy-efficient monitoring of extreme values in sensor networks. Proc. ACM SIGMOD Int’l Conf. on Management of Data (SIGMOD 2006), 169–180.
31) Wu, M., Xu, J., Tang, X. and Lee, W.-C. (2006) Top-k monitoring in wireless sensor networks. IEEE Trans. Knowl. Data Eng. 19, 962–976.
32) Wu, S.-H., Chuang, K.-T., Chen, C.-M. and Chen, M.-S. (2008) Toward the optimal itinerary-based KNN query processing in mobile sensor networks. IEEE Trans. Knowl. Data Eng. 20, 1655–1668.
33) Kannhavong, B., Hidehisa Nakayama, H., Nemoto, Y., Kato, N. and Jamalipour, A. (2007) A survey of routing attacks in mobile ad hoc networks. IEEE Wirel. Commun. 14 (5), 85–91.
34) Tsuda, T., Komai, Y., Sasaki, Y., Hara, T. and Nishio, S. (2014) Top-k query processing and malicious node identification against data replacement attacks in MANETs. Proc. Int’l Conf. on Mobile Data Management (MDM 2014), 279–288.
35) Hara, T. (2001) Effective replica allocation in ad hoc networks for improving data accessibility. Proc. IEEE INFOCOM 2001, 1568–1576.
36) Hara, T., Murakami, N. and Nishio, S. (2004) Replica allocation for correlated data items in ad-hoc sensor networks. ACM SIGMOD Rec. 33, 38–43.
37) Hara, T. and Madria, S.K. (2006) Data replication for improving data accessibility in ad hoc networks. IEEE Trans. Mobile Comput. 5, 1515–1532.
38) Shimo- hara, M., Hayashi, H., Hara, T. and Nishio, S. (2006) Replica allocation considering power consumption in mobile ad hoc networks. Proc. Int’l Workshop on Pervasive Wireless Networking (PWN 2006), 463–467.
39) Shimo- hara, M., Hara, T. and Nishio, S. (2007) Data replication considering power consumption in ad hoc networks, Proc. Int’l Conf. on Mobile Data Management (MDM 2007), 118–125.
40) Hara, T. and Madria, S.K. (2009) Consistency management strategies for data replication in mobile ad hoc networks. IEEE Trans. Mobile Comput. 8, 950–967.
41) Hayashi, H., Hara, T. and Nishio, S. (2003) Cache invalidation for updated data in ad hoc networks, Proc. Int’l Conf. on Cooperative Information Systems (CoopIS 2003), 516–533.
42) Hayashi, H., Hara, T. and Nishio, S. (2005) Updated data dissemination methods for updating old replicas in ad hoc networks. Pers. Ubiquitous Comput. 9, 273–283.
43) Hagihara, R., Shimo- hara, M., Hara, T. and Nishio, S. (2009) A message processing method for top-k query for traffic reduction in ad hoc networks, Proc. Int’l Conf. on Mobile Data Management (MDM 2009), 11–20.
44) Komai, Y., Sasaki, Y., Hara, T. and Nishio, S. (2014) KNN query processing methods in mobile ad hoc networks. IEEE Trans. Mobile Comput. 13, 1090–1103.
45) Komai, Y., Sakai, Y., Hara, T. and Nishio, S. (2015) K nearest neighbor search for location-dependent sensor data in MANETs. IEEE Access 3, 942–954.
46) Komai, Y., Hara, T. and Nishio, S. (2015) Processing convex hull queries in MANETs. Proc. Int’l Conf. on Mobile Data Management (MDM 2015), 64–73.
47) Sasaki, Y., Hara, T. and Nishio, S. (2014) Top-k query processing for replicated data in mobile peer to peer networks. J. Syst. Softw. 92, 45–58.
48) Tsuda, T., Komai, Y., Hara, T. and Nishio, S. (2015) Signature-based top-k query processing against data replacement attacks in MANETs. Proc. IEEE Int’l Symp. on Reliable Distributed Systems (IEEE SRDS 2015), 130–139.
49) Tsuda, T., Komai, Y., Hara, T. and Nishio, S. (2016) Top-k query processing and malicious node identification based on node grouping in MANETs. IEEE Access 4, 993–1007.
50) Hara, T. (2003) Replica allocation methods in ad hoc networks with data update. ACM-Kluwer Journal on Mobile Networks and Applications 8, 343–354.
51) Hayashi, H., Hara, T. and Nishio, S. (2004) Replica allocation considering data update intervals in ad hoc networks, Proc. IFIP/IEEE Int’l Conf. on Mobile and Wireless Communication Networks (MWCN 2004), 131–142.
52) Hara, T., Hagihara, R. and Nishio, S. (2010) Data replication for top-k query processing in mobile wireless sensor networks, Proc. IEEE Int’l Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC 2010), 115–122.
53) Derhab, A. and Badache, N. (2009) Data replication protocols for mobile ad-hoc networks: A survey and taxonomy. IEEE Commun. Surv. Tutor. 11 (2), 33–51.
54) Padmanabhan, P., Gruenzwald, L., Vallur, A. and Atiquzzaman, M. (2008) A survey of data replication techniques for mobile ad hoc network databases. Vldb J. 17, 1143–1164.
55) Chen, K. and Shen, H. (2015) Maximizing P2P file access availability in mobile ad hoc networks through replication for efficient file sharing. IEEE Trans. Comput. 64, 1029–1042.
56) Choi, J.-H., Shim, K.-S., Lee, S.K. and Wu, K.-L. (2012) Handling selfishness in replica allocation over a mobile ad hoc network. IEEE Trans. Mobile Comput. 11, 278–291.
57) Fiore, M., Casetti, C. and Chiasserini, C.-F. (2011) Caching strategies based on information density estimation in wireless ad hoc networks. IEEE Trans. Vehicular Technol. 60, 2194–2208.
58) Liu, Y., Han, Y., Yang, Z. and Wu, H. (2015) Efficient data query in intermittently-connected mobile ad hoc social networks. IEEE Trans. Parallel Distrib. Syst. 26, 1301–1312.
59) Ting, I.-W. and Chang, Y.-K. (2013) Improved group-based cooperative caching scheme for mobile ad hoc networks. J. Parallel Distrib. Comput. 73, 595–607.
60) Wu, W., Cao, J. and Fan, X. (2013) Design and performance evaluation of overhearing-aided data caching in wireless ad hoc networks. IEEE Trans. Parallel Distrib. Syst. 24, 450–463.
61) Xia, F., Ahmed, A.M., Yang, L.T., Ma, J. and Rodrigues, J.J.P.C. (2014) Exploiting social relationship to enable efficient replica allocation in ad-hoc social networks. IEEE Trans. Parallel Distrib. Syst. 25, 3167–3176.
62) Zhang, Y., Yin, L., Zhao, J. and Cao, G. (2012) Balancing the trade-offs between query delay and data availability in MANETs. IEEE Trans. Parallel Distrib. Syst. 23, 643–650.
63) Lim, S., Lee, W.-C., Cao, G. and Das, C.R. (2006) A novel caching scheme for improving Internet-based mobile ad hoc networks performance. Ad Hoc Netw. 4, 225–239.
64) Lim, S., Lee, W.-C., Cao, G. and Das, C.R. (2004) Performance comparison of cache invalidation strategies for Internet-based mobile ad hoc networks. Proc. of Int’l Conf. on Mobile Ad-hoc and Sensor Systems (MASS 2004), 104–113.
65) Tomell, S.M., Calafate, C.T., Cano, J.-C. and Manzoni, P. (2015) DTN protocols for vehicular networks: An application oriented overview. IEEE Commun. Surv. Tutor. 17, 868–887.
66) Nishiyama, H., Ito, M. and Kato, N. (2014) Relay-by-smartphone: realizing multihop device-to-device communications. IEEE Commun. Mag. 52 (4), 56–65.
67) Zaman, S. and Grosu, D. (2011) A distributed algorithm for the replica placement problem. IEEE Trans. Parallel Distrib. Syst. 22, 1455–1468.
68) Chen, K., Shah, S.H. and Nahrstedt, K. (2002) Cross-layer design for data accessibility in mobile ad hoc networks. Wirel. Pers. Commun. 21, 49–76.
69) Huang, Y., Sistla, P. and Wolfson, O. (1994) Data replication for mobile computers. Proc. ACM SIGMOD Int’l Conf. on Management of Data (SIGMOD 1994), 13–24.
70) Aho, A.V., Hopcroft, J.E. and Ullman, J.D. (1974) The Design and Analysis of Computer Algorithms, Addison-Wesley.
71) Hayashi, H., Hara, T. and Nishio, S. (2004) Updated data dissemination in ad hoc networks, Proc. of Int’l Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS 2004), 29–43.
72) Luo, J., Hubaux, J.P. and Eugster, P. (2003) PAN: Providing reliable storage in mobile ad hoc networks with probabilistic quorum systems, Proc. ACM MobiHoc’03, 1–12.
73) Manning, C.D., Raghavan, P. and Schutze, H. (2008) Introduction to Information Retrieval, Cambridge University Press.

Profile

Takahiro Hara received the B.E., M.E., and Dr.E. degrees in Information Systems Engineering from Osaka University, Osaka, Japan, in 1995, 1997, and 2000, respectively. Currently, he is a Distinguished Professor of the Department of Multimedia Engineering, Osaka University. He has published more than 430 journal and conference papers in the areas of databases, mobile computing, peer-to-peer systems, WWW, and wireless networking. He served as general chair of IEEE SRDS 2014 and Mobiquitous 2016, and program chair of IEEE MDM’06/10, IEEE AINA’09/14, and IEEE SRDS’12. His research interests include distributed databases, peer-to-peer systems, mobile networks, and mobile computing systems. He received more than 70 research awards including JSPS prize from Japan Society for the Promotion of Science. He is an ACM distinguished scientist and a senior member of IEEE and a member of three other learned societies.