Maintaining Data Consistency in Mobile Ad-hoc Networks based on Self-stabilizing Maximal Independent Set

By Nguyen Dang Thanh

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Associate Professor Xavier Défago

September, 2011

Maintaining Data Consistency in Mobile Ad-hoc Networks based on Self-stabilizing Maximal Independent Set

By Nguyen Dang Thanh (0910213)

A thesis submitted to
School of Information Science,
Japan Advanced Institute of Science and Technology,
in partial fulfillment of the requirements
for the degree of
Master of Information Science
Graduate Program in Information Science

Written under the direction of
Associate Professor Xavier Défago

and approved by
Associate Professor Xavier Défago
Professor Mizuhito Ogawa
Associate Professor Azman Osman Lim
Associate Professor Yasushi Inoguchi

August, 2011 (Submitted)

Acknowledgements

Firstly and foremost, during my time in JAIST, I have been very honorable to be a member of Dependable Distributed Systems Group (DDSG). I specially thank Professor Xavier Défago for being my supervisor. As my supervisor, Professor Xavier Défago is an erudite teacher who taught me a lot of things not only the knowledge of distributed systems but also the research methodology, developing the new idea, defining and figuring out the position of the research. Without him and his comments, I cannot finish my research.

Secondly, I would like to thank Dr. Francois Bonnet. His valuable comments made this thesis clearer and well-proved.

Next, I also would like to thank Hiroyuki Hiranuma who supports me a lots in the role of my tutor, without him, I cannot make acquainted with the life in JAIST quickly and focus on studying.

Finally, I would like to say many thanks to FIVE-JAIST Program. Without the support from the program I would not come to Japan to study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Mobile Ad-hoc Networks

Mobile Ad-hoc Networks (MANETs) are autonomous ad-hoc wireless networking systems consisting of dynamic and independent nodes moving in the deployed area and changing network connectivity, which is substantially researched in Networking area in recent years. Unlike the cellular wireless network which has centralized control based on manual or fixed infrastructure, MANETs are dynamic distributed systems and do not have centralized control. In MANETs, two nodes are connected and can communicate directly when they are within the transmission ranges of each other or via intermediate mobile nodes when they are out of their transmission ranges. These communication links are also changed constantly because of the movement of nodes in the network. Moreover, in MANETs, a node has the roles of both a mobile agent and a router. It means that it has to perform communication routing by itself instead of requesting route from a station like cellular network. Mobile nodes communicate to each other by one of three classes of communication services: *unicast, multicast, broadcast,* and *geocast*. The definition of each class of communication services is mentioned in the System Model section.

All above features make MANETs be a dynamic wireless distributed system. Because of the flexibility and dynamic attribute, MANETs are easily adapted others environments. For example, in the real world, MANETs are widely applied in various fields such as military, environment monitoring, vehicular communication, earthquake sensing or anywhere the infrastructure is not easily deployed and manually managed.

However, the mobility and the wireless communication of MANETs also cause a huge problems. Firstly, the mobility makes the network topology change constantly so there is no fixed routing path for every mobile nodes in MANETs. Secondly, the wireless communication between mobile nodes is message passing based on broadcasting radio signal. The broadcast protocol contains the problem of collisions between concurrent signals as messages. Consequently, the communication of MANETs is unreliable and unstable. While, the communication protocol is the fundamentals of any distributed system and the background of many other services such as data query, data collection, or data replication.

1.2 Data consistency

Among the services mentioned above, the data replication is an old technique to improve the system availability in the general distributed systems. In MANETs, mobile nodes are deployed in a large area and the mobility of nodes causes the disconnections frequently. Consequently, the entire network is usually partitioned. Hence, there is a demand of replicating data in different nodes to prevent the data loss, to improve the data availability and to reduce the time latency of a data query. The replicated data must be shared and updated frequently by mobile nodes. Data replication is also called as shared data memory or shared register in different modern works.

The major problem of the shared register is the replicas updating. When the shared data is changed by any node, every replicas of data must be updated eventually to maintain the data consistency. What inconsistencies are permissible and how the consistency can be maintained is the central mechanism of a shared register. There are different data consistency levels provided by the register according to the particular purpose of the application.

Since the characteristics of MANETs, there are huge challenges in designing a shared register and solving the consistency problem. In MANETs, the shared data must be duplicated and updated in different replicas by passing message. So, the unreliable communication and the collision problem directly affect to the replication scheme and the data consistency management. Some obvious examples are packet loss due to the collision or the dynamic communication makes propagating messages arrive to destination in unpredictable orders.

Therefore, a register of MANETs has to consist of not only a consistency management but also an effective communication protocol support to propagate the replicated data. There are several approaches to design an effective communication for MANETs, but all of them based on a dynamic structure of the network or use an extra information of neighbors. Considering MANETs under the viewpoint of a communication graph, a fundamentals of Graph Theory which is Maximal Independent Set (MIS) suggests several improvements to design a register. First of all, MIS proposes a cluster structure which supports to transmit messages through the network without collision. Moreover, the existence of MIS members as cluster heads benefits to manage the data consistency. The next section gives a brief introduction of MIS.

1.3 Maximal Independent Set

Given a graph $G(V, E)$, where V is a set of vertices and E is a edges connect any two vertices together. A set I is called an Independent Set if for any two vertices belonging to I, there is no any edge connecting them together. According to the definition of the Independent Set, *Maximal Independent Set* (MIS) is introduced as an Independent Set that is not a subset of any other Independent Set. A set S is called a MIS if every edge of the graph has at least one endpoint that is not in S and every vertex that is not in S connects to at least one vertex in S.

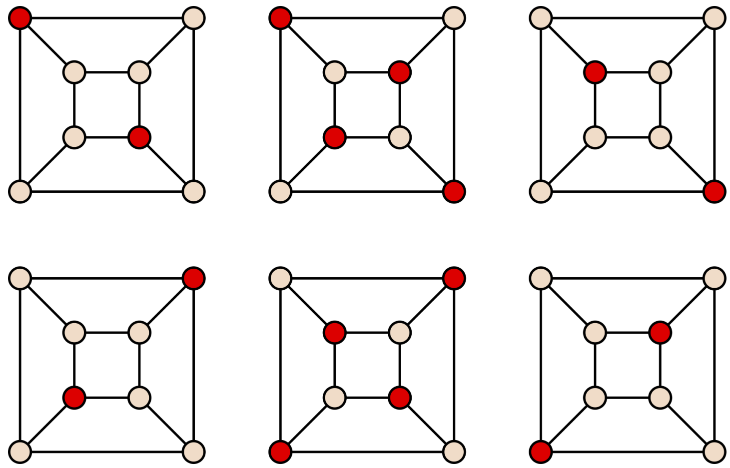

Figure 1.1: Maximal Independent Set

The Figure 1.1 represents 6 Maximal Independent Sets (MISs) in a graph. In this illustration, all red vertices are separated in 2-hop distance, hence a set of red vertices in each sub-image form a MIS of $G(V, E)$. A graph can have many different Maximal Independent Sets (MISs). This work mainly uses the Maximal Independent Set as a virtual backbone of the network.

There are several algorithms (both centralized and distributed) to find a MIS in a Wireless Ad-hoc Networks, the most significant result is a distributed construction proposed by P.J. Wan [1]. However, not many works achieve good results in a dynamic distributed system like MANETs. While the information of the MIS structure provides many advantages for other services of MANETs such as communication services and data management. Consequently, it is necessary to study the solution to maintain MIS structure for the dynamic distributed systems.

1.4 Contributions

Motivated by interests of Maximal Independent Set (MIS) and the Data Consistency problem, in this work we design a set of algorithms which have following achievements for MANETs .

- A distributed dynamic mechanism to define and to maintain the structure of MIS of MANETs which is self-stabilizing and has the message complexity is $O(n)$, where n is number of mobile nodes.
- Based on the MIS structure, we design a message passing mechanism called MIS-based propagation which reduces the occurrence of collision in the network communication.

- Using the MIS-based propagation and the MIS-structure, to design a regular register called MIS-based register. MIS-based register satisfies the causal consistency [1] model.

Firstly, our proposed algorithms to maintain the structure of MIS in MANETs has the network complexity is $O(n)$. That is significantly improve the result in [1]. Moreover, it is proved that self-stabilizing, which is a very important feature when applying to MANETs due to its mobility. This efficient protocol to maintain the information of the MIS structure is not only use in the data consistency management but also can be used in various of other applications such as creating *Connected Dominating Set* or designing a selective *Flooding* scheme.

Secondly, based on MIS-based propagation with collision preventing, we introduce a regular register called MIS-based register for multi-hop MANETs. MIS-based register is the cluster-based register in which the cluster is defined based on the dynamic MIS. The member u of MIS is considered as the cluster head. A cluster head u and its one-hop neighbors create a cluster. The proposed MIS-based register ensures the causal consistency model. Which is a significant enhancement when compared to the reference algorithms.

[1] Causal consistency is a model where the order of write operations is not necessary identical over the mobile nodes. Only the write operations which are causally related are seen by different nodes in the same oder

Chapter 2

Background

2.1 System model

In our work, we consider Mobile Ad-hoc Networks (MANETs) consisting of mobile nodes with full mobile ability. MANETs are deployed in finite areas which is consider as a 2D plane. In the simulation, the deployed area of MANETs is a square bordered area.

Every mobile node has a specific IP address assigned in the installation phase of MANETs. The identity (ID) of each node is inferred from the IP address value. Hence, each mobile node has a unique ID.

We assume that all nodes have the same transmission range r. Two nodes u and v form a communication link whenever the Euclidean distance between them is less than or equal to the transmission range ($\|u - v\| \leq r$). The network topology is defined as follow:

Definition 2.1. Given a network N, the topology of N is a communication graph $G(V, E)$, where V is the set of all nodes and E is the set of all edges which correspond to the communication links between two nodes in N.

In the rest of this work, r is normalized to 1 to be considered as a unit. The underlying graph is called unit disk graph. Moreover, communication links between mobile nodes are changed frequently due to the mobility during the working time of the network and nodes can be broken, leave or joins to MANETs. Consequently, the communication graph G changes. The dynamic G is define in definition below:

Definition 2.2. Given a MANET N, topology of N at a particular working time t is a communication graph $G_t(V_t, E_t)$. Where V_t is the set of all nodes and E_t is the set of edges which correspond to communication links between any two nodes in MANETs at the working time t.

Message passing service of MANETs is a set of communication protocols defining the way of communication between mobile nodes. There are four classes of routing protocols in MANETs: unicast, multicast, broadcast and geocast. In this thesis, it is not necessary for a mobile node to have the geographic information, so we do not consider the geocast. The next part gives an overview of unicast, broadcast and multicast in MANETs.

Furthermore, MANETs are studied here as asynchronous distributed system, which has no synchronization protocol in network time and message passing service. So, there are collisions and interferences in message passing service. Under unit disk graph, we have the definition of interference model as follow.

Definition 2.3. Mobile nodes cannot send and receive data simultaneously. In the interference model, each node has a transmission range r and an interference range $r_I \geq r$. A receiver v of a link uv is interfered by another sender p of a link pq if $\|p - v\| \leq r_I$. Here, $r_I = r = 1$.

Under all definitions above, the communication of MANETs based on a set of communication primitives which are consequently described in the next part.

2.1.1 Communication primitives

Broadcast

Broadcast is one of fundamental of MANETs. As far as the broadcast is concerned, there are two common concepts: *one-hop broadcast*, *fully broadcast*.

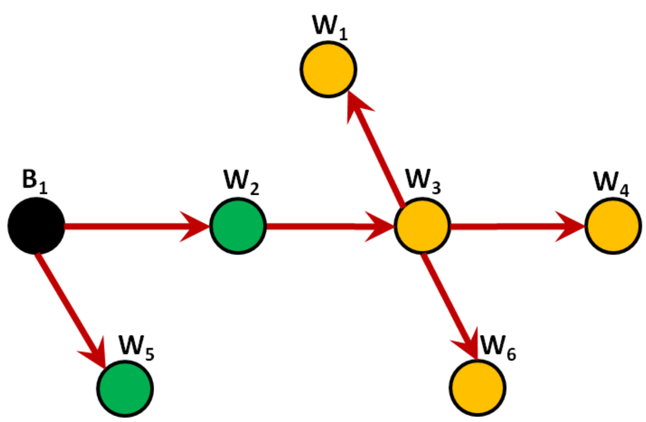

Figure 2.1: Broadcast

One-hop broadcast ensures that a packet is received correctly by all recipients within the transmission range of the sender. Actually, all routing protocols such as unicast, multicast and broadcast always use *one-hop broadcast* to transmit packets. The difference between protocols is only the address of destination which is set in the transmitted packet. According to IEEE 802.11 standard, when a node wants to perform *one-hop broadcast*, it sets the destination's address of transmitted packet to 255.255.255.255.

In Figure 2.1, only neighbors W_2 and W_5 (green ones) receive transmitting message from B_1 when it performs a *one-hop broadcast*. The other nodes in network (yellow ones) only receive transmitting message from B_1 when it executes fully broadcast mentioned below.

In the **fully broadcast**, when a recipient receives a one-hop broadcast packet, it retransmits this packet to the other nodes of the network. As mentioned before, this protocol causes the broadcast storm problem when a recipient receives broadcast packets from two senders simultaneously. In this study, the terminology *broadcast* is used to indicate the one-hop broadcast. Moreover, in this work broadcast has additional assumption below:

Assumption 2.1. When there is not any interfered packet exchanged concurrently, a packet is transferred to destination successfully. That means the success rate of non-interference packet delivery at mobile node is 100%.

Assumption 2.2. At the time of MANETs are deployed, every mobile node knows all ID of its neighbors in 1-hop distance.

Assumption 2.2 only requires that a node knows its neighbors at the time of network deployment. When MANETs work, and the network topology change, the neighbors of a node can be change. Consequently, there must be a update scheme if every node wants to keep neighbors' information.

Unicast

Unicast is a reliable service in which a packet is transmitted from a single node to another node that is identified by a unique address in MANETs. The unicast message is transmitted from a source node by broadcasting a packet with the specific IP address of destination node. If source and destination nodes are in connected in the transmission range, a packet is sent directly from the source node to the destination node. In the other case, a sending the packet is passed to the destination node hop-by-hop via intermediate nodes. In the second case, message is necessary to propagated with a routing protocol.

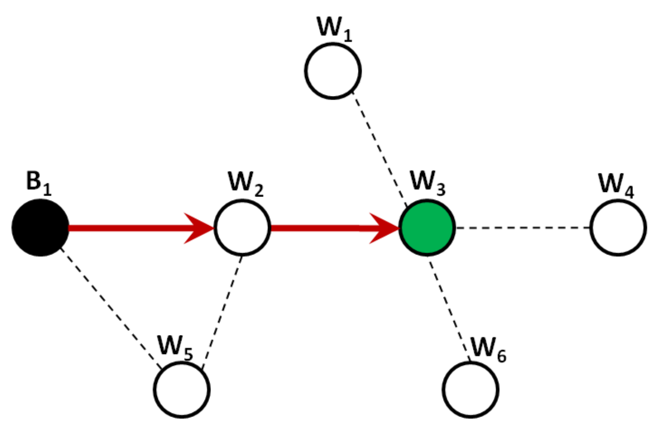

Figure 2.2: Unicast

In Figure 2.2, black node B_1 send a unicasts message to node W_3 which is separated in 2-hop distance to B_1. Hence, the unicast message is transmitted via W_2 as an intermediate node. In this study, we only use the unicast protocol to communication two nodes in 1-hop distance.

After sending a packet, sender waits for an acknowledgement (ACK) packet from recipient to sure that there is no collision and interference in sending and the packet is delivered correctly. After a waiting period, if sender does not receive an ACK packet from the recipient, it listens to the medium and retransmits the packet again after a random waiting time that is managed by a collision detection and prevention protocol (such as CSMA/CD).

Multicast

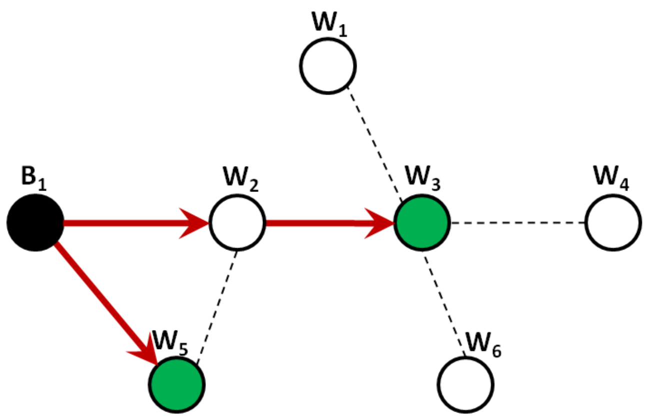

Figure 2.3: Multicast

Multicast is a class of techniques that is used to transmit packets from a single transmitter to a specific group of recipients simultaneously by only one transmission time from the transmitter. The packet will reach its destinations in one-hop (W_5) or multi-hops (W_3) like the example in Figure 2.3. There are two sub-classes of multicast protocols, which are unreliable multicast and reliable multicast. Reliable multicasts are developed to add packet loss detection and retransmission packet. The used techniques are the same as in unicast. In this work, the terminology multicast is used to indicate a 1-hop reliable multicast.

2.2 Definition of register

Given MANETs with all definitions and assumptions above, the register of MANETs is defined as follow:

Definition 2.4. Register is a concurrent object (or share data) that can be accessed by all nodes in MANETs. A register provides two basic operations to every nodes of MANETs:

- Write operation allows nodes to update the value of register.
- Read operation allows nodes to read and return the latest value written by write operations.

The read operation can be executed concurrently in every node. The write operation can be executed concurrently or non-concurrently. According to the permission to be executed concurrently or non-concurrently of write operation. A register can be classified into three types safe, regular and atomic register:

- A safe register can be written by one writer only. Moreover, the read operation is only executed if there is no concurrent write operation.
- Regular register can have any number of concurrent writers and readers. Concurrent write operations can be appeared differently in different nodes of MANETs.
- An atomic register is a regular register which does not allow the inversion of write operations in different nodes of MANETs. That means the results of write operations must be seen in the same order in all nodes.

Under the provided protocols of MANETs. The read and write operation of a register in MANETs is generally implemented as follow:

- Every read operation returns the valid replica of shared data stored in local node.
- Every write operation is terminated when replicas of shared data in all nodes are updated.

In this work, we propose a mechanism to propagate message in multi-hop MANETs. Based on underlying mechanism, we design a register and investigate its data consistency model.

2.2.1 Consistency models

Based on the proposed propagating mechanism, we design the system to manage the data consistency property of MANETs. As mentioned in the introduction part, data consistency is well studied for distributed systems but there are only a limited works for dynamic distributed system like MANETs. First of all, we introduce a overview of some different consistency models. There are many models of consistency. Some considerable models are Strict consistency, Linearizability, Sequential consistency, Causal consistency, FIFO consistency which are respectively introduced in the next part.

In a distributed system, all operations of all processes is recorded. A *trace* is a permutation of operations which are executed in the whole system. A consistency model defines all characteristics which must be satisfied by a *trace* to be considered as valid.

Strict consistency

Strict consistency reflect the true correspondent order of operations in real time. If a process executes a write operation $W(x := 2)$ at time t and this is a latest operation in the system, then every operation $R(x)$ executed at time $t' > t$ must return 2. Strict consistency requires an instantaneously update in all replicas in the whole system. It is too strong and difficult to implement especially for a mobile system like MANETs.

Linearizability

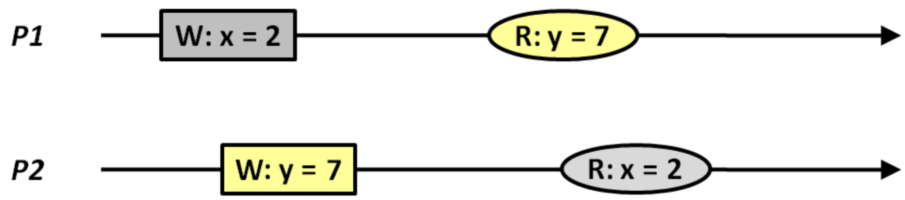

Figure 2.4: Linearizability

Strict consistency is too strong and difficult to implement because of the instantaneous update requirement. Thus, a slightly weaker model is introduced is Linearizability. A trace is linearizable when it is consistent and the consistent trace must respect the order of execution time of all operations. For example in Figure 2.4, the only trace is linearizable is $W(x := 2)W(y := 7)R(y := 7)R(x := 2)$.

Sequential consistency

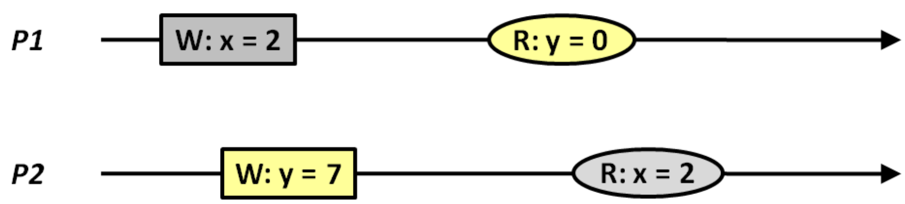

Figure 2.5: The behavior is sequentially consistent but not linearizable

A weaker model than Linearizability is Sequential consistency. It only concerns the first criterion of Linearzability is the global consistent but does not require the real-time order of all operations. The example given in Figure 2.5, which has all share variables are assigned to 0 initially, does not satisfy the Linearizability but satisfy the Sequential consistency. The reason is there not exist a consistent trace with real-time order. But, in the example, there exists trace $W(x := 2)R(y := 0)W(y := 7)R(x := 2)$ which satisfies the global order requirement of Sequential consistency.

Causal consistency

Causal consistency is a weaker model than Sequential consistency. It only requires satisfied traces to respect the causal order of write operations. The non-causal order operations can be seen by different processes in different orders, the global order is not ensured. For example, in the Figure 2.6, we have the results retrieved in process P_3 and P_4 are different.

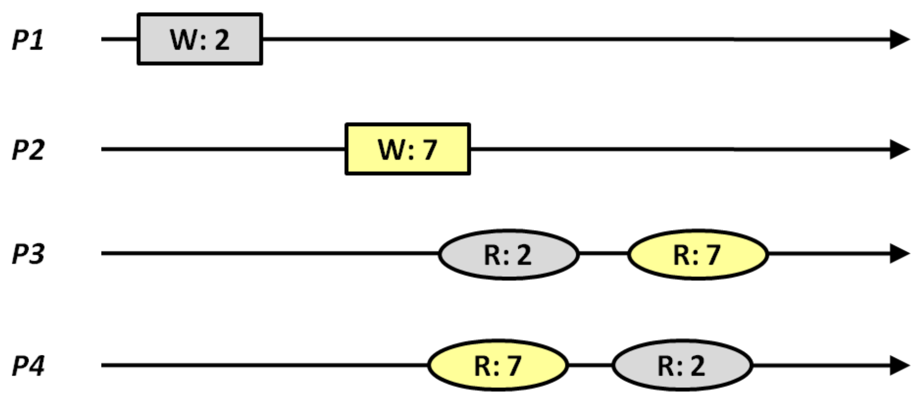

Figure 2.6: The behavior is causally consistent but not sequentially consistent

FIFO consistency

The causal consistency is violated if there are some write operations being in causal order but they are reflected in different order in particular process. The FIFO consistency allows the results of causally related write operations to be retrieved in different order. This consistency model ensures the FIFO order for write operations. Figure 2.7 is an example such that, the causal order of $W(x := 2)$ and $W(x := 7)$ are not reflected in P_3, but it is FIFO consistency.

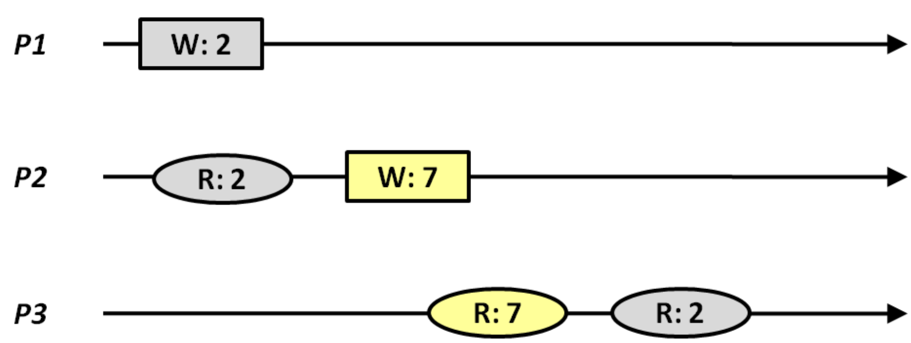

Figure 2.7: The behavior is FIFO consistent but not causally consistent

2.2.2 Current approaches

The data consistency management is a popular research area in the distributed database systems. It is also well studied with many different approaches especially for static distributed systems. However, there are only a few researches [2–8] focusing on consistency management in MANETs in recent years. These studies can be mainly classified into three general approaches. The first approach is doing broadcast in every intermediate node [2]. The second one is called cluster-based approach [7] and the third one is quorum system based [5, 6, 8]. Moreover, there are some other works using the combination between cluster-based and quorum system approaches [4].

Fully rebroadcast approach

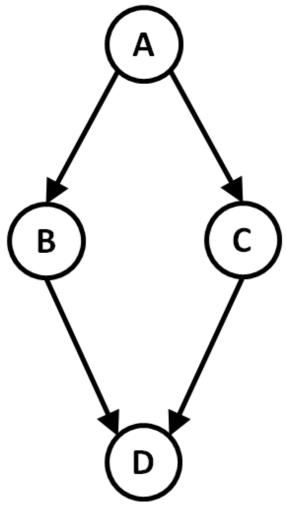

Figure 2.8: Broadcast storm

In the fully rebroadcast approach, when a node wants to write a new value to a variable in the shared data, it simply broadcasts a message to all other recipients in network. When a mobile node receives a broadcast message containing an updated value, it compares this value with the current value in its local memory and stores if the updated value is newer. In a 1-hop network, the receiving node does not retransmit broadcast message [2] but in a multi-hop network, with this approach, all mobile nodes need to retransmit received messages to the next hop. This part of protocol causes a broadcast storm problem in which the collision occurs when a node receives broadcast messages from two other nodes simultaneously.

For example, in Figure 2.8, when node A broadcasts a message, two neighbors B, C in 1-hop distance of A concurrently receive the broadcasting message and rebroadcast it. Consequently, their concurrent rebroadcasting cause the collision in D when it receives messages from B and C simultaneously. The collision causes the packet loss when D do not receive messages sent by B and C. This phenomenon called broadcast storm.

The approach based on fully rebroadcast mechanism contains a lot of problem of collisions, packet loss and performance reduction. Consequently, it is not appropriate to multi-hop network.

Cluster-based approach

In the cluster based approach, the network is separated into several clusters. In each cluster there are a particular degree of divergence of shared data and some nodes keep a part of the shared data. Besides, there are some nodes keeping a part of shared data, and a static node called cluster head or proxy [7]. For instance, in Figure 2.9, the network is separated into 6 clusters having black nodes as cluster heads. The cluster head is responsible for keeping the consistency of the shared data in all members of cluster. Each

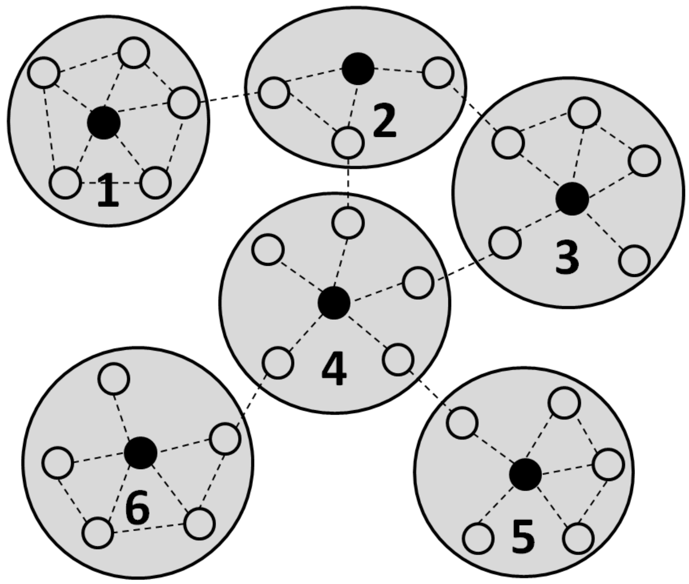

Figure 2.9: Clustering

member has to retrieve a right given by a mutual exclusion algorithm to write and read the shared memory.

Most of works in this approach require the existence of static nodes in the role of cluster heads. Because of the difficulty in maintaining the cluster structures, they do not support a dynamic distributed systems where all nodes can move.

Quorum system approach

In the Quorum system approach [5, 6, 8], a mobile node has to successfully set lock to QR from other nodes in MANETs whenever it wants to perform a read operation and successfully set lock to QW whenever it wants to perform a write operation. In this approach, the condition is $|QR| + |QW| > n$, where, n is the number of mobile nodes in the entire network.

In [4], the authors propose a combination of Cluster-based and Quorum system approaches. In this scheme, the global quorums QR and QW are defined by the number of cluster-heads or proxies (which also must be static nodes). Therefore, the quorum condition becomes $|QR| + |QW| > l$, where l is the number of regions in the entire network. In each region R_i, there are a local quorums QLR_i and QLW_i which are the number of nodes that need to be set lock to region R_i.

2.3 The Maximal Independent Set

Maximal Independent Set (MIS) is an important concept of Graph theory. In this work, MIS plays a central role of all proposed algorithms. Therefore, the definition of MIS is formally given below.

2.3.1 The definition of Maximal Independent Set

Definition 2.5 (Definition of Maximal Independent Set)**.** Given a graph $G(V, E)$, a subset of vertices M is Maximal Independent Set of G when:

- (1) $\forall u \in M, \forall v \in M, (u, v) \notin E$, where (u, v) is an edge between u and v.
- (2) $\forall u \notin M, \exists v \in M, (u, v) \in E$.

(1) is the condition of any 2 vertices belonging to M are not connected together. (2) means that any vertice which is not in M must be adjacent to some vertex in M.

In context of MANETs, an MIS is defined as follow.

Definition 2.6 (The Maximal Independent Set in MANETs)**.** Given a network, for any communication graph $G_t(V_t, E_t)$ at working time t, a subset of nodes M_t is an MIS when:

- (1) $\forall u \in M_t, \forall v \in M_t, (u, v) \notin E_t$, where (u, v) is a communication links between node u and node v.
- (2) $\forall u \notin M_t, \exists v \in M_t, (u, v) \in E_t$.

Such a set M_t is an MIS of $G_t(V_t, E_t)$. After a finite number of steps, every node knows whether it belongs to set M_t or not. In MANETs, the proposed scheme must be given in the distributed approach and use the message passing service. The definition of the problem does not require that members of MIS must be known by every nodes.

However, when mobile nodes move around network area, the communication graph of MANETs is changed very frequently. Consequently, the MIS structure may be broken. It gives the problem to maintain the MIS structure of a dynamic distributed system. In this work, we the problem above is considered as a self-stabilization problem.

2.3.2 Self-stabilization

Definition 2.7 (Definition of self-stabilization)**.** Self-stabilization is a concept of the fault tolerance problem in general distributed systems. A distributed system is self-stabilizing when it has the capability to automatically recover in a finite numbers of steps after the occurrence of faults.

The system can start with an arbitrary state and still converge to the legitimate state which has a desired behavior. The legitimate state is retrieved after a finite number of execution steps. In addition, the occurrence of faults can force the system to any arbitrary state but the system always recovers due to the self-stabilization characteristic.

In Figure 2.10, the inner points stand for legitimate states while the outer points stand for illegitimate states. When fault occurs, the system is changed from the desired state to the illegitimate state (black dotted line). But if it is self-stabilizing, after a define number of steps it is recovered to one of the legitimate states.

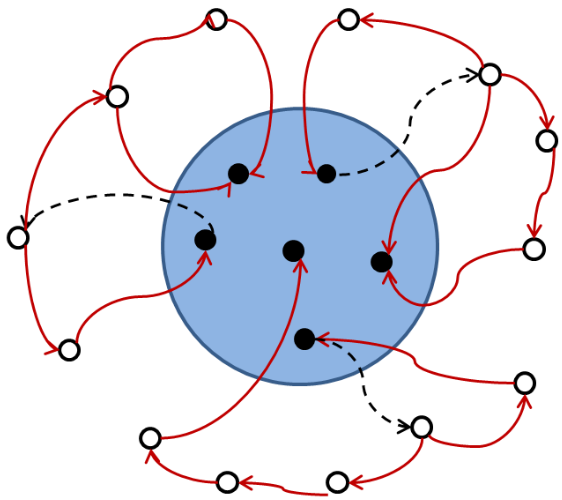

Figure 2.10: Self-stabilization

Self-stabilization of MIS in MANETs

MANETs has two important properties: the communication graph which saves the network topology of MANETs and the mobility. Let consider the mobility property be an external factor which affect to the communication graph $G_t(V_t, E_t)$ of MANETs. The problem of maintaining MIS structure becomes a self-stabilization problem of the communication graph $G_t(V_t, E_t)$.

In this self-stabilization problem, MANETs can start with an arbitrary state. However, the mobility works as an external factor that changes to communication graph and breaks the MIS structure of the network. Consequently, the MIS is changed to an illegitimate state. All legitimate states of the MIS structure of MANETs are defined by the definition of MIS in MANETs. A state is illegitimate when it violates one or both conditions in Definition 2.6. Therefore, there are two movements violating the legitimate states of MIS.

- Firstly, when two nodes belonging to MIS M_t at time t move close and connect together. Which violates condition (1).
- Secondly, when a mobile nodes which does not belong to MIS moves and disconnects to all neighbors belonging to MIS M_t. This movement violates condition (2).

In both cases, the definitions of MIS is violated and MANETs change to illegitimate state of MIS structure. The mechanism maintaining MIS structure is self-stabilizing when it automatically corrects faults generated by two movements above. In the next parts, we analyze the most famous construction of MIS in Wireless Ad-hoc Network and consider its self-stabilization property.

2.3.3 Wan's construction

The researches on Maximal Independent Set (MIS) concentrate on finding all MISs or only MIS which have the smallest/largest number of members. Besides that, there are a few algorithms to construct the MIS. Most of algorithms in finding the Maximal Independent Set mentioned above are centralized algorithms [10]. In [1], the authors propose a distributed algorithm to construct MIS. This algorithm is a part of scheme creating a Connected Dominating Set and it is based on the message passing service of Wireless Ad-hoc Networks.

In this algorithm, let $G(V, E)$ be the communication graph of the given ad-hoc network. The input of the proposed construction is a spanning tree T of G created by a distributed leader-election algorithm in [11] with $O(n)$ time complexity and $O(n \log n)$ message complexity. The spanning tree T has a root at v_0. The distance from every node to the root v_0, measured in number of hops, is the level of the node in the network. The ordered pair of the level of node and its ID called *rank*. Rank is a metric used to select a node into the MIS among possible candidates. Hence, the first stage of this algorithm is collecting a level for every member. This step is started by broadcasting a LEVEL message at the root of Spanning Tree and completed by continuously broadcasting LEVEL messages in every nodes of network receiving LEVEL messages from other nodes.

At the beginning of the second stage, all nodes are marked as WHITE. The target of the second stage marking all nodes of network as GREY or BLACK. In the end of algorithm, all BLACK nodes form a MIS. The second stage of the algorithm is described in the Algorithm 1 below.

Algorithm 1 Wan's distributed MIS construction

```
Every node colors itself WHITE;
Root node v_0 changes its color to BLACK and broadcasts a BLACK message to its
one-hop neighbors in G;
for all WHITE node u received a BLACK message do
    u colors itself GREY and broadcasts a message GREY to its one-hop neighbors in
    G;
end for
if a WHITE node w receives GREY from all its lower-ranked neighbors then
    w colors itself as black and sends BLACK message to all its one-hop neighbors in
    G;
end if
All BLACK nodes form a MIS.
```

The algorithm introduced in [1] has many steps and needs to retrieve further information to construct a MIS successfully. Firstly, the network needs to find a spanning tree T. Then it selects the root of T as the starting point of the of algorithm. Furthermore, whenever performing the MIS construction algorithm, every node has to retrieve its level via a broadcasting scheme as a metric to select members of a MIS in the next step. Retrieving further information requires extra cost and time.

In MANETs, as all nodes (including the root) move frequently, the distance from the root to each node is changed. Consequently, the level of a node in the network changes a lot and the MIS structure changes to the illegitimate state.

Starting from an illegitimate state, [1] does not introduce any way to recover the legitimate state. It is assumed that we must reset all information in every node and execute the algorithm from the first step which creates the spanning tree T. The lack of self-stabilizing makes the algorithm become appropriate to MANETs, it is necessary to design another scheme for maintaining the MIS structure when the mobile nodes move.

Chapter 3

Self-stabilizing Maximal Independent Set

3.1 Proposed MIS construction

In this research, we introduce a scheme for constructing and maintaining a MIS. This scheme is self-stabilizing and suitable to use in MANETs where all nodes can move freely. The construction of a MIS by this scheme is a distributed color marking procedure for all nodes of MANETs. In this scheme, according to the Assumption 2.2, every node of MANETs knows its 1-hop neighbors after the initialization phase. Here is the general idea of the proposed construction.

- Every node is initially marked as $TRANSPARENT$, a $TRANSPARENT$ node executes Algorithm 2 to change its color to $BLACK$ or $WHITE$
- After a finite number of steps, all nodes in MANETs are marked as $BLACK$ or $WHITE$.
- $BLACK$ and $WHITE$ node periodically executes the Algorithm 3, 4 respectively to maintain its own color or to change to appropriate color.

The proposed scheme is formally described by input definition and algorithms as follow.

Definition 3.1 (Definition of mobile node's properties)**.** Given a MANET having the communication graph $G(V, E)$. $\forall u \in V$, we have:

1. $color_u \in \{TRANSPARENT, BLACK, WHITE\}$

   It means that for every mobile node u, u can be marked as one of three different colors: TRANSPARENT, BLACK, and WHITE. The color of the node u is denoted by $color_u$.

2. Three sets $nbTransparents_u\{r_1..r_m\}$, $nbBlacks_u\{r_1..r_n\}$, $nbWhites_u\{r_1..r_p\}$, where r_i is $rank$ of i, denoted by $rank_i$. Node i can be a TRANSPARENT, BLACK and WHITE neighbor of u respectively.

3. The ordered pair $rank_u = \langle N_u, ID_u \rangle$ is $rank$ of u, in which ID_u is u's ID and:
    - $N_u = |nbWhites_u| + |nbTransparents_u|$ for TRANSPARENT nodes.
    - $N_u = |nbWhites_u|$ for BLACK nodes.
    - $N_u = |nbBlacks_u|$ for WHITE nodes.
4. u can broadcast one of color message I-AM-TRANSPARENT, I-AM-BLACK, or I-AM-WHITE denoted by msg according to its color is $TRANSPARENT$, $BLACK$, or $WHITE$ respectively. $rank_u$ is included in msg.

According to the input defined above, Algorithm 2 describes the execution of a TRANSPARENT node.

Algorithm 2 Execution of protocol in TRANSPARENT node u

Require: Executed in TRANSPARENT node u with its attributes
Ensure: u is marked as either BLACK or WHITE

```
procedure CheckingRank
    if (nbBlacks_u is empty) ∧ (∀v_i ∈ nbTransparents_u, rank_u > rank_{v_i}) then
        Change color_u to BLACK and broadcast an I-AM-BLACK message;
    end if
end procedure

procedure ReceivingMessage(msg, s)                ▷ s is sender of message msg
    if msg is I-AM-TRANSPARENT then
        Add rank_s into nbTransparents_u;
        Remove rank_s from nbWhites_u or nbBlacks_u;
    else if msg is I-AM-BLACK then
        Add the rank_s into nbBlacks_u;
        Remove rank_s from nbWhites_u or nbTransparents_u;
        Change color_u to WHITE and broadcast an I-AM-WHITE message;
    else if msg is I-AM-WHITE then
        Add rank_s into nbWhites_u;
        Remove rank_s from nbBlacks_u or nbTransparents_u;
        CheckingRank
    end if
end procedure
```

At the beginning, $\forall u \in V, |nbBlacks_u| = 0$. Following 2, every TRANSPARENT node checks its $rank$ by running CheckingRank procedure. There are two cases happen:

- (1) There always exists at least one node u such that:

$$\forall v_i \in nbTransparents_u,\ rank_u > rank_{v_i}$$

It means that u has $nbBlacks_u$ empty and the largest $rank_u$ when compared with its TRANSPARENT neighbors. Then, u marks itself to BLACK and broadcasts an I-AM-BLACK message.

- Other TRANSPARENT nodes v_i are still TRANSPARENT until:
  - (2) It receives an I-AM-BLACK message from a BLACK neighbor, such as the neighbors of u in case (1). Then, v_i marked itself WHITE and broadcast I-AM-WHITE (line 13 of Algorithm 2).
  - (3) It receives an I-AM-WHITE message which make v_i satisfies case (1).

The next section gives the proof that Algorithm 2 eventually marks all nodes in MANETs as BLACK or WHITE after a finite number of steps. The set of all BLACK nodes is a MIS of MANETs.

Algorithm 2 has a significant improvement when compared with the algorithm proposed by P. J. Wan [1] in the same condition of Wireless Ad-hoc Networks. It has the message complexity $O(n)$ while Algorithm 1 has the message complexity $O(n \log n)$. The complexity of algorithm will be proved in the next section.

Moreover, in MANETs, when the communication graph is changed the MIS structure created by Algorithm 2 can be maintained easily and recover the system to the legitimate state by periodically executing Algorithm 3 and 4. Which are consequently described in formal below:

Algorithm 3 Execution of protocol in BLACK node u

Require: Executed in BLACK node u with its attributes

procedure PeriodicBroadcasting
    Broadcast an I-AM-BLACK message
end procedure

procedure ReceivingMessage(msg, s) ▷ s is sender of message msg
    if msg is I-AM-TRANSPARENT **then**
        Add $rank_s$ into $nbTransparents_u$;
        Remove $rank_s$ from $nbWhites_u$;
    else if msg is I-AM-BLACK **then**
        if $(rank_s > rank_u)$ **then**
            Add $rank_s$ into $nbBlacks_u$;
            Changes $color_u$ to WHITE;
            Broadcasts REMOVE message;
        end if
    else if msg is I-AM-WHITE **then**
        Add $rank_s$ into $nbWhites_u$;
        Remove $rank_s$ from $nbWhites_u$;
    end if
end procedure

These algorithms executed in BLACK and WHITE nodes consequently. They are used to maintain the MIS structure when mobile nodes move.

Algorithm 4 Execution of protocol in WHITE node u

```
Require: Executed in WHITE node u with its attributes
procedure PeriodicChecking
    for all b_i ∈ nbBlacks_u do
        if u does not receive any I-AM-BLACK from b_i in iamWhiteCycle then
            Remove b from nbBlacks_u;
        end if
    end for
    if |nbBlacks_u| = 0 then
        Changes to TRANSPARENT;
    else
        Broadcast an I-AM-WHITE message;
    end if
end procedure

procedure ReceivingMessage(msg, s)                ▷ s is sender of message msg
    Upon receiving a color message msg;
    if msg is I-AM-TRANSPARENT then
        Add rank_s into nbTransparents_u;
    else if msg is I-AM-BLACK then
        Add rank_s into nbBlacks_u;
        Remove the sender's rank from nbTransparents_u;
    else if msg is I-AM-WHITE then
        Add rank_s into nbWhites_u;
        Remove rank_s from nbTransparents_u;
    else if msg is REMOVE then
        Remove rank_s from nbBlacks_u;
        Add rank_s into nbWhites_u;
        if |nbBlacks_u| = 0 then
            Change to TRANSPARENT
        end if
    end if
end procedure
```

Two execution in BLACK and WHITE are explained in more detail here:

- With BLACK nodes, the PeriodicBroadcasting procedure is executed to broadcast I-AM-BLACK every *iamBlackCycle*.
- With WHITE nodes w, the PeriodicChecking procedure is executed to check the connection between WHITE node to their BLACK neighbors.

- Let t_b be the last time when w receives I-AM-BLACK message from a BLACK neighbor b

  If $t_b < (currentTime - iamBlackCycle)$ **then** b will be removed from $nbBlacks_w$.
- After checking, if $nbBlacks_w$ is empty, w changes to TRANSPARENT and executes Algorithm 2 to have it new color.

When every mobile node moves in MANETs, based on the current color, mobile node execute the Algorithm 2, 3, or 4. By this execution every mobile node update its latest information and the change of its relative positions to its neighbors. Each algorithm has two procedures. The first one is executed periodically. Like the example in Figure 3.1, BLACK, WHITE and TRANSPARENT node periodically broadcast I-AM-BLACK, I-AM-WHITE and I-AM-TRANSPARENT respectively.

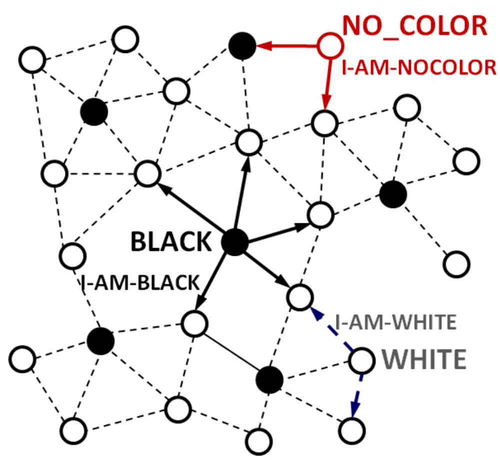

Figure 3.1: Broadcasting Color messages

The second procedure of each algorithm is executed when a mobile node receives a color message. Based on its current color, each mobile node has a different response when receiving a particular color message from its neighbors. The responses in these procedures help the mobile nodes to keep the information of their relative positions to the neighbors.

- When a BLACK node u receives an I-AM-BLACK message from BLACK s, the property of a MIS is violated because there are two BLACK nodes connecting together. In this case:
  - If $(rank_u > rank_s)$ then

    u still is BLACK. s is WHITE and broadcast REMOVE message.
  - Otherwise, if $(rank_u < rank_s)$ then

    s still is BLACK. u is WHITE and broadcast REMOVE message.

- Even if a TRANSPARENT mobile node u is moving, u will change to WHITE color as soon as it receives an I-AM-BLACK message. According to the definition of MIS, a BLACK node is a member of MIS, consequently neighbors cannot belong to MIS and they must be WHITE.
- When a WHITE node receives a color message, it simply updates its three sets $nbTransparents$, $nbBlacks$, and $nbWhites$.

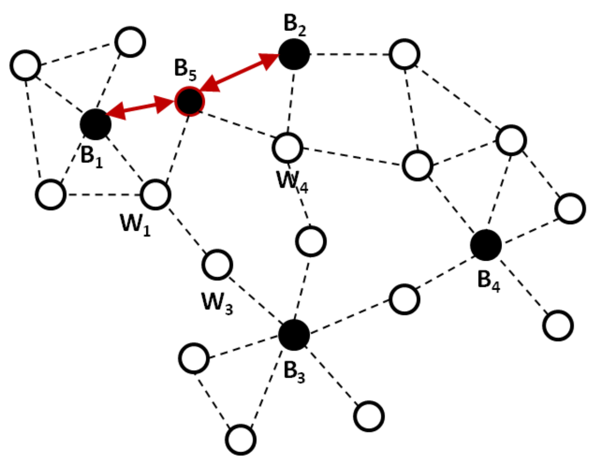

Figure 3.2: A BLACK node receives an I-AM-BLACK message

For example, in Figure 3.2, B_5 moves and receives an I-AM-BLACK messages from BLACK nodes B_1 or B_2:

- In both cases, B_5 has the smaller $rank$ than B_1 and B_2 then it has to change its color to WHITE and broadcasts REMOVE color. The BLACK nodes B_1 or B_2 stays as BLACK.
- WHITE nodes which receive REMOVE message from B_5 (W_1, W_4 in this example) remove B_5 from its variable $nbBlacks$.

The scenarios of color change in mobile nodes caused by the fault in MIS structure will be discussed in more detail in Self-stabilization section.

When all mobile nodes move and changes the communication links, the $rank$ of each node is changed, but it is also updated by the color message exchange service. Moreover, $rank$ value does not affect the correctness of MIS structure. It is only a reference metric used to select the BLACK node among its neighbors. So, even when a node moves, the outdated value of $rank$ still can be used in Algorithm 2, 3, 4.

From the proposed scheme (Algorithm 2, 3, 4), each type of color node has a different time cycle to broadcast color message. The time cycles can be configured differently based on the importance of each type. In this protocol, the BLACK node has the most important role because the WHITE and TRANSPARENT nodes decide their role base on the relation with the BLACK neighbors. Hence, $iamBlackCycle$ should have the smallest value compared to $iamWhiteCycle$ and $iamTransparentCycle$.

A protocol with the periodic sending scheme above provides the way to MANETs to correct the structure of MIS following the mobility of the mobile node. The communication graph $G_t(V_t, E_t)$ can start with an arbitrary state of MIS but it is always possible to converge to a legitimate state. In Section 3.4, we will prove that by applying Algorithm 2, the MIS structure of MANET will be constructed. After that, we will investigate the self-stabilization property of proposed protocol when Algorithm 3, 4 are run under context of mobility.

3.2 Correctness

First of all, we prove the correctness of the proposed scheme. According to Definition 2.6 of the Maximal Independent Set in MANETs, the expected result of a MIS construction algorithm is a subset $M \subset V$ in $G(V, E)$ such that:

1. $\forall u \in M, \forall v \in M, (u, v) \notin E$.
2. $\forall u \notin M, \exists v \in M, (u, v) \in E$.

Hence, the set of BLACK nodes created by the proposed scheme is proved to be MIS if:

Definition 3.2 (Conditions for set of BLACKs as MIS)**.** : Conditions such that set of BLACK nodes form an MIS:

(Condition 1) $\forall u$ is BLACK, $\nexists v$ is BLACK such that u and v are 1-hop neighbors.

(Condition 2) $\forall u$ is WHITE, $\exists v$ is BLACK such that u and v are 1-hop neighbors.

The correctness of Algorithm 2 is proved based on the assumption that:

Assumption 3.1. There is not any mobility that breaks the communication links between any two nodes (u, v) of network in a finite period time T. Where T has a upper bound that also given in the proof.

Lemma 3.1. *By Algorithm 2, all nodes of a MANET are marked as either BLACK or WHITE after a finite number of steps.*

Proof. The proof of this Lemma based on the Assumption 3.1.

According to the Algorithm 2, every node u is TRANSPARENT until one of three cases happens:

(Case 1) u knows it is the largest $rank$ among its TRANSPARENT neighbors. u changes to BLACK (Algorithm 2 line 3).

(Case 2) u receives an I-AM-BLACK message from a BLACK neighbor. u changes to WHITE (2 line 13).

(Case 3) u receives an I-AM-WHITE message which makes $nbTransparents$ of current node change such that it is the largest $rank$ among its TRANSPARENT neighbors. u changes to BLACK (Algorithm 2 line 17).

Considering MANETs as communication graph $G_t(V_t, E_t)$ where V_t is set of all vertices and E_t is set of all edges connecting the vertices, we have:

- $\forall u \in V_t, N(u)$ is the set of u's neighbors.
- Let $U_t \subset V_t$ such that: $\forall u \in U_t, u$ is TRANSPARENT.
- Let $M_{tk} \subset V_t$ such that: M_{tk} includes k vertices marked as BLACK and all their WHITE neighbors.
- Consequently, $U_t \cap M_{tk} = \varnothing$

Initially, with $k = 0$, we have:
$U_t \equiv V \wedge M_{t0} = \varnothing$;
According to the definition of $rank$ given in Definition 3.1 item 3.

$$\exists u_0 \in V_t, rank_{u_0} = \langle N_{u_0}, ID_{u_0} \rangle, \forall v \in N(u_0), rank_v < rank_{u_0}$$

It means that, u_0 has a highest $rank$ when compared with it all neighbors. Following Case 1, u_0 is changed to BLACK.

Then, resulting to Algorithm 2 line 3, mobile node represented by u_0 broadcast I-AM-BLACK message. Consequently, by Case 2, all neighbors of u_0 are changed to WHITE when receiving a I-AM-BLACK.

So, after the first step we have:

1. $M_{t1} = M_{t0} \cup (u_0 \cup N(u_0))$ are not connected by a communication link.
2. $U_{t_1} = U_t \ M_{t1}$.

Following the mechanism above, assume that at time t we have:

- Set M_{tk} includes k BLACK vertices and all their WHITE neighbors.
- Set U_t includes all TRANSPARENT vertices.

For example in Figure 3.3, we have set $M_{t4} (k = 4)$ of marked vertices, the other vertices with red circle in U_t are still marked as TRANSPARENT.

Suppose that:

$$(\exists u_i \in U_t \wedge \exists m_j \in M_{tk}, (m_j \text{ is BLACK}) \wedge (\exists e_{ij} \in E_t \text{ connects } u_i, m_j))$$

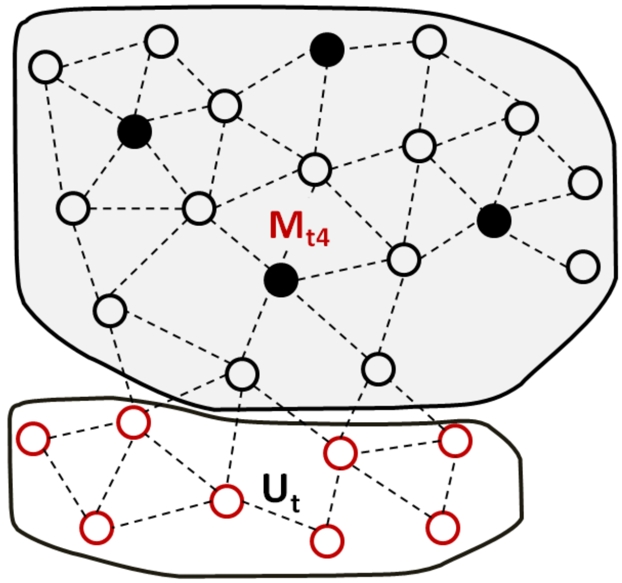

Figure 3.3: Expansion of MIS

then, u_i must be WHITE and $u_i \in M_{tk}$.
It contradicts to Definition of U_t and Definition of M_{tk}. Hence:

- $\forall u_i \in U_t, \nexists m_j \in M_{tk}, m_j$ is BLACK , $(u_i, m_j) \in V_t$
- $\forall u_i \in U_t$, u_i only receives I-AM-WHITE and I-AM-TRANSPARENT messages.

Moreover:

$$\exists u_m \in U_t, \forall (v_i \in N(u_m) \wedge v_i \in U_t), rank_{u_m} > rank_{v_i}$$

If u_m receives I-AM-WHITE from a $m_j \in M_{tk}$, u_m becomes BLACK following to Case 3.
If u_m does not receive I-AM-WHITE message and only receives I-AM-TRANSPARENT, u_m becomes BLACK according to Case 1.
Consequently:

- u_m marks itself BLACK.
- Neighbors of u_m mark themselves WHITE.
- $M_{tk} \cup u_m \cup N(u_m) = M_{t(k+1)}$, where $N(u_m)$ is all neighbors of u_m.
- $U_t' = U_t \setminus (u_m \cup N(u_m))$.
- $|U_t'| = |U_t| - (|N(u_m)| + 1)$

In other words, the set of nodes which are marked as either BLACK or WHITE is expanded like the illustration in Figure 3.4.

By induction on k, the procedure is repeated until $|U_t| = 0$.

Because $|U_t| \leq |V|$ where V is the set of all vertices, $|U_t|$ is finite.

Consequently, the number N of repeat steps is finite and $N \leq |V|$.

In conclusion, under Assumption 3.1, by executing the proposed mechanism, all the nodes of MANETs are eventually marked as BLACK or WHITE after a finite number of steps $N \leq |V|$, where V is the set of mobile nodes.

The upper bound time T in Assumption 3.1 is the time to execute Algorithm 2 N times. □

Theorem 3.2 (MIS construction)**.** *All BLACK vertices form a MIS of $G_t(V_t, E_t)$.*

Proof. According to Lemma 3.1, all vertices of G_t are marked as either BLACK or WHITE after a finite number of steps of proposed scheme. Moreover, we have:

Following Case 2 in Lemma 3.1, $\forall b$ is BLACK, $\forall w \in N(b)$ w receives I-AM-BLACK message from b and changes to WHITE. Hence:

$$\forall b \text{ is BLACK}, \nexists w \in N(b), w \text{ is BLACK. } \mathbf{(1)}$$

(1) insure **Condition 1** of **Definition 3.2**.

$\forall v \in V_t$ are WHITE. In all Cases [1-3] in Lemma 3.1, v only change to WHITE if it receives I-AM-BLACK from a BLACK vertex (Case 2). It means that:

$$\forall v \in V_t, \exists u \in N(v), u \text{ is BLACK } \mathbf{(2)}$$

(2) ensure **Condition 2** of **Definition 3.2**.

Therefore, the set of all BLACK vertices satisfies all conditions in Definition 3.2 of an MIS in MANETs, hence M_t is a MIS the network. □

Theorem 3.3 (Distance of MIS members)**.** *Any pair of complementary BLACK subsets are either connected by less than three hops or disconnected from each other.*

Proof. Let denote $B_t = \{b_i : 1 \leq i \leq k\}$ be a set of all BLACK vertices of G_t, where b_i is the ith vertex marked BLACK.

For any $1 \leq j \leq k$, let $W(B, E_w)$ be the weighted graph such that:

- $B = \{b_i : 1 \leq i \leq j \wedge b_i \in B_t\}$.
- E_w is set of multi-hop connection between any pair (b_m, b_n) where $b_m, b_n \in V$.

  The weight of $e_i \in E_w$ the number of hops in the connection between b_m and b_n.

We will prove that every connected partition of B always has a spanning tree T with all the edges having weight $w \leq 3$.

Assume that $\exists b_m, b_n \in B$, which cannot be connected together by any spanning tree $T(B, E_{Tw})$ such that $\forall e_i \in E_{Tw}, e_i \leq 3$.

Such as vertices B_1, B_3 in the Figure 3.4 and Figure 3.5, if there no B_2 exists.

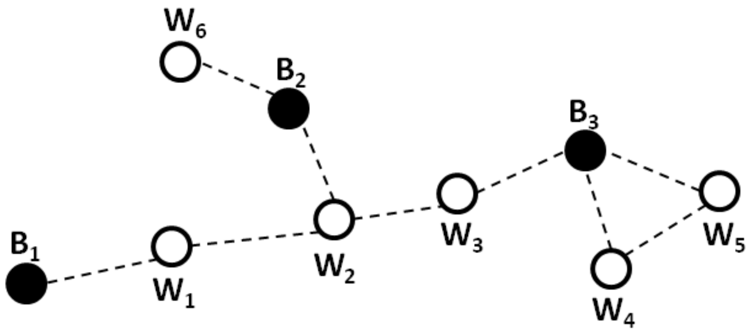

Figure 3.4: $G_t(V_t, E_t)$

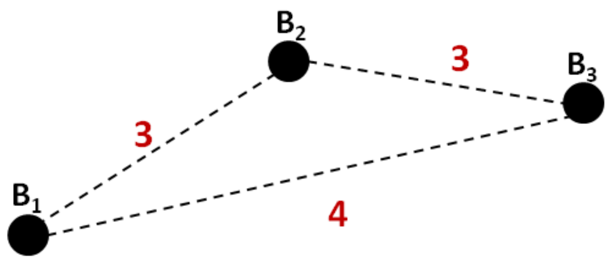

Figure 3.5: Corresponding $W(B, E_w)$.

In this case, as the example, the WHITE vertex W_2 is not connected to any BLACK vertex. It contradicts to **Condition 2** in **Definition 3.2**.

Therefore:

$$\text{Suppose that } (\exists b_i, b_j \in B, (b_i, b_j) \in E_{Tw}, (b_i, b_j) > 3)$$
$$\rightarrow (\exists b_k \in B, (b_i, b_k), (b_j, b_k) \in E_{T_w}, e_{ik} \leq 3 \wedge e_{jk} \leq 3).$$

In conclusion, with any subset B_i of BLACK nodes and its complementary subsets B_j of the other BLACK ones, they are either disconnected to each other or connected with by a connection edge having length less than 3 hops. □

3.3 Self-stabilization

Let consider the mobility of MANETs be an external factor which affect to the network graph. The self-stabilization problem is considered the self-stabilization mechanism for a MIS structure from the faults caused by the mobility as an external factor.

When we consider the self-stabilization properties of MANETs, we apply **Assumption 3.1** to make sure the MIS structure is not changed by external factors. According to **Definition 3.2**, the legitimate state of an MIS is established when following conditions are satisfied:

(Condition 1) $\forall u$ is BLACK, $\nexists v$ is BLACK such that u and v are 1-hop neighbors.

(Condition 2) $\forall u$ is WHITE, $\exists v$ is BLACK such that u and v are 1-hop neighbors.

The illegitimate states are states which violates one or both Condition 1 and 2 above. Hence, there are exactly two type of illegitimate states.

- $\exists u$ is BLACK $\wedge \exists v$ is BLACK such that u and v are 1-hop neighbors. We call it a fault **Type 1**.
- $\exists u$ is WHITE, $\forall v_i$ is BLACK, u and v_i are not 1-hop neighbors. We call it a fault **Type 2**.

The relationship between all illegitimate states and legitimate states of MIS is represented by Figures 3.6.

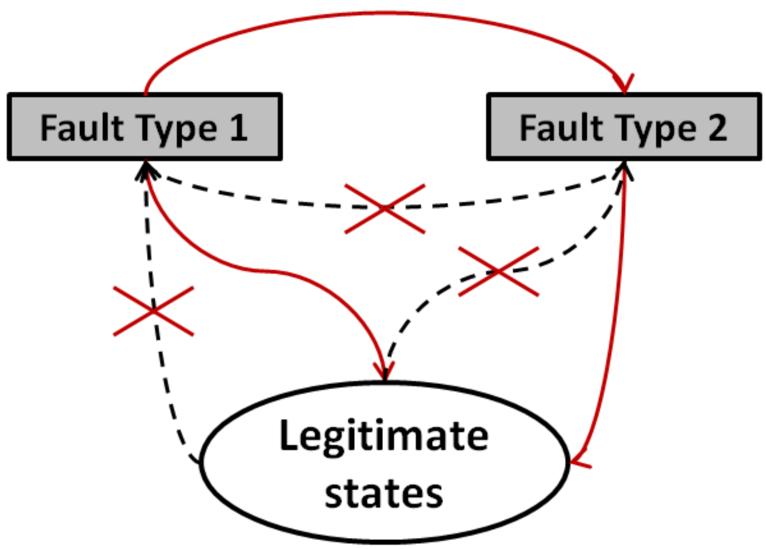

Figure 3.6: Corresponding $W(B, E_w)$.

Lemma 3.4. *Suppose that a MANET is in an illegitimate states containing only faults Type 2,*

1. *The network will not generate any fault Type 1*
2. *The network will be recovered in legitimate state in a finite number of steps.*

Proof. Suppose that at the time t, a MANET is in an arbitrary illegitimate state and contains a set $F2$ of $n2$ WHITE nodes in faults Type 2. It means:

$$\forall u_i \in F2, \text{ where } 1 \leq i \leq n2, \forall v_j \text{ is BLACK, } u_i \text{ and } v_j \text{ are not 1-hop neighbors.}$$

According to line 8 in Algorithm 4:

$$\forall u_i \in F2,\ u_i \text{ is changed to TRANSPARENT.}$$

Consequently, there are $n2$ TRANSPARENT nodes in MANETs.

Lemma 3.1 proves that $n2$ TRANSPARENT nodes of the network will be marked as either BLACK or WHITE after $n2$ steps. The new set of BLACK nodes created by process is proved that following 2 condition in Definition 3.2.

Therefore, the system does not generate any fault Type 1 and converges to a legitimate state in at most $n2$ finite steps. □

Lemma 3.5. *A MANET is in an illegitimate states containing some faults Type 1 and some fault Type 2 will be recovered in legitimate state in a finite number of steps.*

Proof. Suppose that at the time t, a MANET is in an illegitimate state and contains a $n1$ fault Type 1 and $n2$ fault Type 2.

Hence, there exists a set P of m pairs (u_i, v_i) such that:

$$u_i \text{ is BLACK } \wedge v_i \text{ is BLACK } \wedge (u_i \text{ and } v_i \text{ are 1-hop neighbors})$$

For instances, in Figure 3.7, there exists a pair of BLACK nodes B_1 and B_4 connecting together.

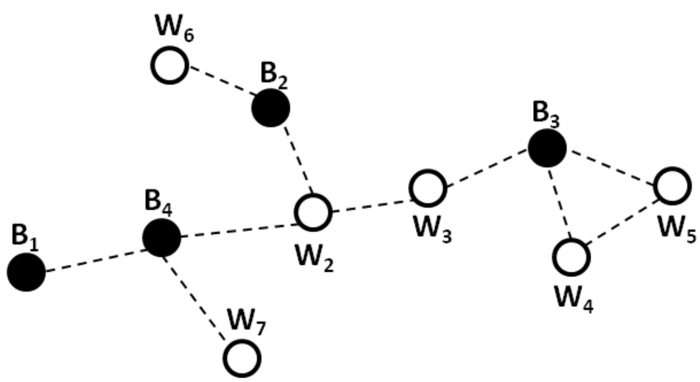

Figure 3.7: An illegitimate with fault Type 1

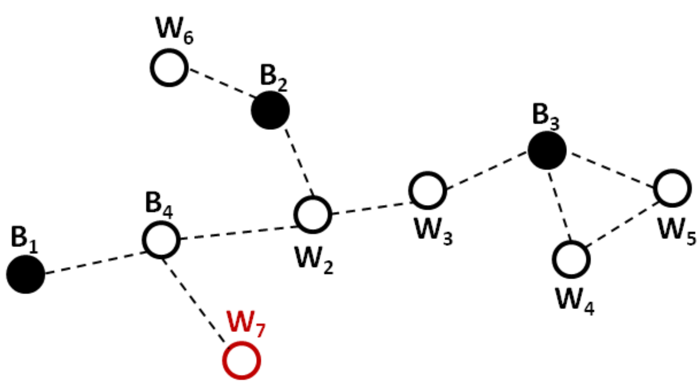

Figure 3.8: The correction of fault Type 1 causes fault Type 2

Assuming that the rank of B_1 is greater than the rank of B_4, according to Algorithm 3 line 8, B_4 is changed to WHITE and broadcast REMOVE messages. Upon receiving the REMOVE message, some neighbors of B_4 are still WHITE because they are adjacent to another BLACK neighbors (such as W_2), W_6 changes to TRANSPARENT because their $|nbBlacks_{W_6}| = 0$.

In general, for each pair (u_i, v_i), according to Algorithm 3 line 8, either u_i or v_i must change to WHITE and broadcast REMOVE message.

Assume that u_i change to WHITE, according to Algorithm 4 line 23−29, upon receiving REMOVE message from u_i, there are three cases for a WHITE neighbors $w \in N(u_i)$:

1. $|N(u_i)| = 0 \rightarrow \nexists w$.

   It means u_i does not have any neighbor.

2. w is still WHITE because $nbBlacks_w > 0$.

   It means w is adjacent to another BLACK neighbors.

3. w is changed to TRANSPARENT because $nbBlacks_w > 0$.

   It means there is no BLACK neighbors of w.

When one of BLACK in a pair (u_i, v_i) changes to WHITE (suppose that u_i), the number of fault Type 1 is decreased by 1 and the number of fault Type 2 is increased by m_i (Figure 3.8). Where $0 \leq m_i \leq |N(u_i)|$.

In the other words, fault Type 1 is changed to a number of fault Type 2 or change to legitimate state. (When $\forall i, m_i = 0$).

After all, all the faults Type 1 of a illegitimate system are changed to faults Type 2. And the number of faults is:

$$n_F \leq n2 + \sum m_i, \text{ where } 1 \leq i \leq n1 \text{ and } n2;$$

Because, $n2$, $n1$ and $\forall m_i$ are finite. Hence, n_F is finite.

However, according to Lemma 3.4, we have all faults Type 2 are corrected after a n_F steps.

Therefore, all faults Type 1 and faults Type 2 are corrected after n_F finite steps. □

Lemma 3.6. *A network in legitimate state does not generate any fault Type 1 and fault Type 2.*

Proof. Assuming that, there is no mobility as external factor, From the proposed algorithms, we have:

- According Algorithm 4, for each WHITE node u:
  - u always receives at least 1 I-AM-BLACK message from its BLACK neighbors. u also does not move and disconnect itself from its BLACK neighbors. So, there is no fault Type 2 generated by Algorithm 4.

- u does not change it to BLACK in any case. So, there is no fault Type 1 generated by Algorithm 4.

- According to Algorithm 3, for each BLACK node u:
  - u does not change to TRANSPARENT in any case. So there is no fault Type 2 generated by Algorithm 3.
  - There is no mobility, so u cannot close to another BLACK. Consequently, there is no fault Type 1 generated by Algorithm 3.

- Algorithm 2 run in mobile node u only:
  - Changes a u's color to BLACK when there is no BLACK neighbors adjacent to u.
  - Otherwise, u keeps it as TRANSPARENT or changes to WHITE when there is a BLACK neighbors.

  This algorithm also does not raise any fault.

Consequently, Algorithm 2, 3, and 4 do not internally make the system change from the legitimate states to any illegitimate state. □

From Lemma 3.4, 3.5 and 3.6 we have the Theorem 3.7:

Theorem 3.7 (Self-stabilization of MIS)**.** *With a MANET maintaining its MIS structure by the proposed mechanism, we have:*

- *The network in illegitimate state of MIS structure which contains faults Type 1 and faults Type 2 converges to legitimate state in a finite number of steps. (**Lemma 3.4, 3.5**)*
- *The network in legitimate state of MIS structure does not internal change to any illegitimate states. (**Lemma 3.6**)*

Hence, the proposed mechanism maintaining MIS is self-stabilizing.

3.4 Complexity

We consider the message complexity of proposed algorithm for creating the MIS in the same conditions with the scheme in [1]. When the network nodes do not move, each node only needs to send the color message once. There are two cases in which a color message is broadcasted by a node:

- When the condition at line 2 of Algorithm 2 is satisfied, a node decides itself BLACK and broadcasts an I-AM-BLACK message.

- When a node receive an I-AM-BLACK message (as line 10 of Algorithm 2), it changes to WHITE and broadcasts an I-AM-WHITE message.

After all nodes broadcast their color messages, they also have their colors. Hence, the message complexity of proposed algorithm is $O(n)$ in the same conditions with the scheme proposed in [1]. Each node needs only one time unit to compare and broadcast message, so the time complexity is also $O(n)$.

From the proof of complexity and self-stabilization property, the message complexity and time complexity to correct the fault of system are also $O(n)$.

Chapter 4

Data consistency management based on Maximal Independent Set

In MANETs, when a node wants to update the shared data in register, it has to propagate the updated value through the network. Therefore, the data consistency management is based on the message passing service. Motivated by the benefit of the self-stabilizing MIS, we design a message propagating mechanism. This mechanism has a solution for the broadcast storm problem of the fully broadcast service. In this section, firstly, we introduce the modification to retrieve the BLACK neighbors information of a WHITE node. Secondly, we introduce the propagation mechanism based on the information of MIS structure.

4.1 MIS-based propagation

As mentioned in chapter 2, the fully broadcast protocol of MANETs causes the broadcast storm with a lot of collisions. It increases the probability of the packet loss and decreases the throughput of network. So, the modification's objective is to eliminate the redundant transmissions in the intermediate nodes so that we can improve the network performance. Only a few nodes are selected to forward the propagating message. The main issue is defining the rules to select a set of forwarders. In this propagation mechanism, the variable $nbWhites$ is redefined.

Definition 4.1. In every node u:

$$nbWhites_u[\langle r_{w_1}, L_{w_1}\rangle..\langle r_{w_n}, L_{w_n}\rangle]$$

Where:

- r_{w_i} is the rank of WHITE neighbors w_i of u.
- $L_{w_i}[id_{b_1}..id_{b_m}]$, where $b_j (1 \leq j \leq m)$ is a 2-hop BLACK neighbor of u and 1-hop BLACK neighbor of w_i, id_{b_j} is ID of b_j

To maintain these information, we include IDs of BLACK and WHITE neighbors into the I-AM-WHITE message.

- IDs of BLACK neighbors are saved in $blackNeighborsID$.
- IDs of WHITE neighbors are saved in $whiteNeighborsID$.

Consequently, the size of message depends on the degree Δ, the maximum number of neighbors for a mobile node, of MANETs. We modify the receiving I-AM-WHITE procedure as follow:

Modification 4.2. As soon as receiving an I-AM-WHITE message msg_w from WHITE node w.

- A BLACK node creates a new pair $\langle r_w, L_w \rangle$, where $L_w = blackNeighborsID$ contained in msg_w.
- A WHITE node u:
  - Creates a new pair $\langle r_w, L_w \rangle$, where:
  $$L_w = \{b_1.ID, ..., b_n.ID\}$$
  such that:
  $$\forall 1 \leq i \leq n, (b_i.ID \in blackNeighborsID) \land (b_i \notin nbBlacks)$$
  where $nbBlacks$ is variable stored in u (Section 3.3).
  - $\forall v$ such that $(v.ID \in whiteNeighborsID) \land (v \notin nbWhites)$, update the pair $\langle r_v, L_v \rangle$ with:
  $$L_v = L_v \cup L_w;$$
- Finally, the current node adds the new pair $\langle r_w, L_w \rangle$ into variable $nbWhites$.

In the other enhancement, the mechanism for maintaining the MIS structure is scheduled. Every mobile node u sends its periodic color message in a particular time t_u to prevent the collision. The value of t_u is determined by a calculation such as the modulo of u's ID and the number of nodes N ($t_u = ID_u \bmod N$). Here, the time unit is the maximum time to transmit packet in 1 hop distance of MANETs ($< 1\mu s$).

The propagating mechanism is designed based on the background above. Behaviors of mechanism in WHITE node and BLACK node are described in Algorithm 5, 6 respectively.

In this mechanism, the message msg saves:

1. A set $coveredBlacks$ of IDs of BLACK nodes which were covered by the propagating message.
2. A set F as forwarders which are indicated that forward the propagating message.

4.1.1 Receiving behaviors of a BLACK node in MIS-based propagation

A BLACK node always forwards the receiving message msg. It knows what is the next covered BLACK nodes of the forwarding message msg so that it can choose the set of forwarders for msg.

The next covered BLACK nodes (members of $nextCoveredBlacks$) are formed by line 2 of Algorithm 5.

The forwarders F of the message msg is selected based on $nextCoveredBlacks$ such that $|F|$ is smallest and $|F|$ covers all BLACK neighbors in $nextCoveredBlacks$.

Algorithm 5 Receiving a message msg in BLACK node

```
1: procedure FindingListForwarderSet
2:     nextCoveredBlacks = nbBlacks'IDs \ (nbBlacks'IDs ∩ coveredBlacks)
3:     Select the set F of WHITE nodes which cover all BLACK nodes in
       nextCoveredBlacks as forwarders;
4:     coveredBlacks = coveredBlacks ∪ nextCoveredBlacks
5: end procedure
6: procedure ReceivingPropagatingMessage
7:     Execute FindingListForwarderSet;
8:     Forward the message including the list of forwarders F;
9:     Let S = F;
10:    Let t be the time of forwarding retry t = 0;
11:    repeat
12:        F = List of forwarders which receive and forward message successfully;
13:        S = S \ F;
14:        t++;
15:    until (|S| = ∅ ∨ t = MaxTry)
16: end procedure
```

4.1.2 Receiving behaviors of a WHITE node in MIS-based propagation

In Algorithm 6, when a WHITE node u receives message msg from its neighbor, based on list of forwarders F of msg, u decides to continue forward the message or not:

- If $u \in F$ of message msg, u continue forwards the propagating message.
- If $u \notin F$ of message msg, u checks whether there are BLACK neighbors in 2-hop distance from $nbWhites$.
    - If there exists BLACK neighbors in 2-hop distance of u and u belongs to the set of WHITE forwarders to these BLACK nodes, it will forwards the propagating message.

– Otherwise, it stays silent.

Algorithm 6 Receiving a message *msg* in WHITE node

```
if u's ID is in msg.blackNeighborsID then
    Forward message msg.
else
    Let C = ∅, C is set of forwarders candidates;
    Let B = ∅, B is set of 2-hop BLACK neighbors;
    for all ⟨r_w, L_w⟩ ∈ nbWhites do
        Let B_w = L_w \ nbBlacks;
        if (B_w ≠ ∅) then
            C = C ∪ w
            B = B ∪ B_w
        end if
    end for
    if B ≠ ∅ then
        C = C ∪ u
        Select F ⊂ C such that |F| is smallest and ∀b ∈ B, (∃f ∈ F) such that b ∈ L_f;
        if u belongs to F then
            Include IDs b_i ∈ B into coveredBlacks of msg.
            Forward message msg.
        end if
    end if
end if
```

Figure 4.1 gives the example for the propagating mechanism for two cases above. When B_1 sends a propagation message to all other nodes in MANETs.

- Since, B_1 has the only one 2-hop BLACK neighbor B_2, it broadcasts the propagation message *msg* with one forwarder W_2.
- When W_2 receives message *msg*, it updates value in *msg* to the local shared data and continue forwards *msg*. In this branch the next receivers of *msg* are B_2 and W_4.
- In the other branch, W_1 does not belong to *listCoveredBlack* when it receives message *msg*.
  - If does not forward *msg*, the propagating message cannot arrive to the BLACK node B_3 and its WHITE neighbors. Consequently, the propagating mechanism is not designed correctly.
  - However, by the Algorithm 4 and Modification 4.2, W_1 knows that:
    - W_3 is its WHITE neighbors.

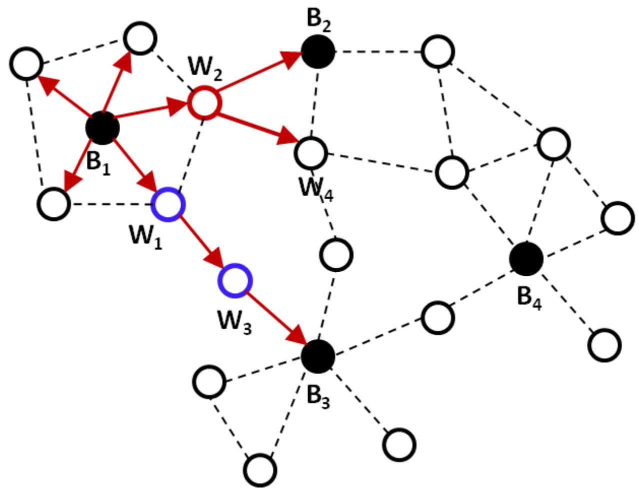

Figure 4.1: Propagating mechanism

    * B_3 is W_3's BLACK neighbor and also its 2-hop BLACK neighbor.

  – According to Algorithm 6:

    * W_1 forwards message *msg* to its WHITE neighbors.
    * W_3 is the next receiver of message *msg*.

According to Algorithm 5, 6, all nodes of network are eventually received the propagated messages by the proposed MIS-based mechanism.

- Every WHITE node receives the propagated message via its BLACK neighbors.
- A BLACK node u receives the propagated message from the closest BLACK v via two cases solved in Algorithm 5:
  – A common WHITE neighbors of both u and v. When u and v are separated by 2-hop distance.
  – Two WHITE connectors w_1, w_2 where w_1 is neighbors of v and w_2 is neighbors of u. When u and v are separated by 3-hop distance.
- When node u is disconnected to BLACK/WHITE neighbors or is marked as TRANSPARENT, it will receive the propagated message by a COLOR message as soon as it connects to any node in network.

Moreover, in the proposed mechanism, the forwarding operation of every mobile node is dynamically scheduled. Each mobile node is scheduled to forward message in a particular slot of time called *timeslot*. The unit of timeslot is a time for a packet to be transmitted over a transmission range r.

- In a BLACK node u, the *timeslot* is calculated by the order of u among its 2-hop BLACK neighbors in its *nbBlacks*.

- In a WHITE node, the *timeslot* is computed by the order of current WHITE node in the list of forwarders.

This simple scheduling reduces the collision of message instead of leaving it to be solved by CSMA/CD service of MAC layer.

4.2 Proposed data consistency management

According to the definition of the system model and the data consistency problem above, in this study, we introduce a register using the proposed message propagating mechanism called MIS-based register. MIS-based register is a regular register which supports multi-reader/multi-writer for a fully mobile network. In this register, we use the MIS structure to support the write operation. In general, a WHITE node has to request to BLACK node neighbor when it wants to write to the register. A BLACK node which receives request from WHITE node neighbors and propagates the write operation is called *committer*. The proposed register uses a logical clock called committer-based clock to manage the consistency of written data. In first part of this section, we introduce the committer-based clock.

4.2.1 Committer-based clock

Because of the absence of network time synchronization, we design a logical time system (or logical clock) for MANETs. The proposed logical clock is a *map clock* based on the BLACK *committer* in MANETs, so, it is called committer-based clock. A committer is a BLACK node which is responsible for propagate the written value:

Definition 4.3 (Committer-based clock)**.** Every mobile node u in MANETs maintains a map $m_u[\langle h_1, t_1\rangle..\langle h_n, t_n\rangle]$, where $\langle h_i, t_i\rangle$ is the local committer-based time t_i of the BLACK committer h_i in the last time h_i propagates a write operation.

- Local committer-based clock of a BLACK node is managed by itself.
- Local committer-based clock of a WHITE node is changed by its neighbors BLACK.
- The notation $m_u[h_i]$ is used to denote t_i.
- $\forall h, \langle h, \_\rangle \notin m_u \Leftrightarrow m_u[h] = 0;$

The entire map m_u reflects the u's view of the local logical time of all committers in MANETs and it is used to timestamp events issued by u. If u committed a write operation, local m_u contains the pair value $\langle u, t_u\rangle$ of u also. A BLACK node u uses the following two rules to update its local map clock:

Rule 1 When executing a write operation by itself, BLACK node u increase the second value t_u of the pair $\langle u, t_u\rangle$ of map m_u by 1.

$$m_u[u] = m_u[u] + 1;$$

Rule 2 Upon receiving a outcome write operation from another BLACK node including the timestamp m, BLACK node u executes following steps to maintain the local committer-based clock:

1. Update its committer-based time:

$$(\forall k : k \neq u) : m_u[k] = max(m_u[k], m[k])$$

2. Execute Rule 1;

The following definition and relations are defined to compare two committer-based timestamps, m_a and m_b:

Definition 4.4 (Intersection of two *committer-based timestamps*)**.** Let $I(m_a, m_b)$ be the intersect part of two map timestamps m_a, m_b, we have:

$$\forall m_a[\langle h_a, t_a \rangle] \in I(m_a, m_b) : \exists m_b[\langle h_b, t_b \rangle] \in I(m_a, m_b), h_a \equiv h_b.$$

According to Intersection of two map timestamps definition we have following relations:

Definition 4.5 (Relation of two *committer-based timestamps*)**.** The two committer-based timestamps have following relations:

- $m_a = m_b \Leftrightarrow \forall x \in I[m_a, m_b] : m_a[x] = m_b[x]$.
- $m_a \leq m_b \Leftrightarrow \forall x \in I[m_a, m_b] : m_a[x] \leq m_b[x]$.
- $m_a < m_b \Leftrightarrow m_a \leq m_b \wedge \exists x \in I[m_a, m_b] : m_a[x] < m_b[x]$.
- $m_a \| m_b \Leftrightarrow \neg(m_a < m_b) \wedge \neg(m_b < m_a)$.

Committer-based clock has a significant smaller size of timestamp than the normal vector clock which saves the local time of all nodes of MANETs. The number of nodes in MANETs can be thousands and it makes the size of vector timestamp exceeds the packet size. Consequently, network cannot send a packet in one transmission per hop and the network performance is slow down. Using the committer-based clock, we can increase the network performance and execution speed two compare two timestamps also.

Moreover, the number of records in a committer-based clock in a node can be reduced. A node can eliminate records corresponding to nodes which no longer propagate any write operation in a time period T. We can make the elimination because:

- Committer-based timestamp is only used to ensure the consistency of concurrent write operations, which are still being propagated in MANETs.
- In a particular node u, the write operation containing a timestamp $m \| m_u$ is accepted to write to MIS-based register.

In conclusion, the small size committer-based clock improves not only the network performance but also the execution speed in every mobile node when comparing two timestamps.

4.2.2 Concurrent MIS-based register

The concurrent MIS-based register with its read and write operation is defined as follow in two definitions:

Definition 4.6 (Read operation)**.** A read operation, executed by mobile node u, retrieves the value of the shared data from local variable in node u.

Definition 4.7 (Write operation)**.** A write operation is define differently in WHITE and BLACK nodes, whenever want to write to MIS-based register:

- A WHITE node u does:
  1. Unicasts a RTW (request-to-write) message to its highest rank BLACK neighbor.
  2. The RTW message includes updated value and its ID.
  3. The BLACK receiver is responsible for executing write operation of u.
- A BLACK node u executes following steps to issued a write operation requested by WHITE neighbors or by itself:
  1. Performs Rule 2.
  2. Forwards the UPDATE message included updated value, m_u, and u's ID (called $committerID$) by MIS-based propagation.

Algorithm 7 Receiving an UPDATE message msg in BLACK node u

procedure COMMITWRITING(msg)
    u write to the local shared data;
    Update committer-based clock m_u by m_{msg} according to Rule 2;
    $m_{msg} = m_u$;
    Propagate UPDATE message msg included m_{msg} by MIS-based propagation;
end procedure
procedure RECEIVINGUPDATEMESSAGE(msg)
    if $(m_{msg} > m_u)$ **then**
        Execute COMMITWRITING;
    else if $(m_{msg} \| m_u) \wedge (m_{msg}[committerID] > m_u[committerID])$ **then**
        Execute COMMITWRITING;
    else
        Deny write operation;
        Stopped propagating msg;
    end if
end procedure

When a WHITE node receives a UPDATE message from another node, it simply propagates the UPDATE message by MIS-based propagating mechanism.

On the other hand, when a mobile BLACK node u receives an UPDATE message msg including updated value and committer-based timestamp m, it executes Algorithm 7 to accept or deny a write operation.

In the next section we prove that the consistency model of MIS-based register is causal consistency and analyze the upper bound of respond time for the read/write operation in the proposed MIS-based register.

4.3 MIS-based register analysis

4.3.1 The consistency model of Concurrent MIS-based register

The proposed concurrent MIS-based register is proved that causal consistency if all cases that violate causal consistency are show that cannot appear according to definitions in Section 4.3.

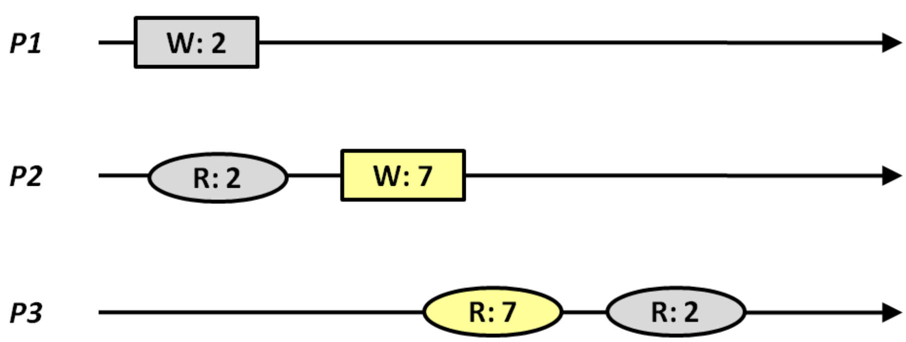

Figure 4.2: The behavior violates Causal consistency model

As mentioned in Section 4.2, in MANETs , the write operations may be executed by same node or different nodes. The causal consistency model ensures that all write operations that are causally related are seen by all nodes of MANETs in the same order. Consequently, values returned by read operations must respect this causal order. Write operations that are not causally related can be seen in any order in different nodes.

In Figure 4.2, the read operation $(R : x = 2)$ shows that $(W : x = 2)$ issued by $(P1)$ is completed written in $(P2)$. Therefore, write operation $(W : x = 7)$ is clearly executed after write operation $(W : x = 2)$. We call the relation of $(W : x = 2)$ and $(W : x = 7)$ are causal order or $(W : x = 2)$ and $(W : x = 7)$ are causally related. However, the read operations executed in $(P3)$ follow another order in which the result of $(W := 7)$ is reflected that issued before $(W : x = 2)$. Hence, the register with above behavior violates Causal consistency model.

In Figure 4.3, there is no evidence of the causal relation between write operations $(W : x = 2)$ and $(W : x = 7)$ issued by $P1$ and $P2$. It also means that, the operation $(W : x = 2)$ is not completely written in node $P2$ and the operation $(W : x = 7)$ is not completely delivered in node $P1$. So, the order of $(W : x = 2)$ and $(W : x = 7)$ can be seen by different nodes in any order.

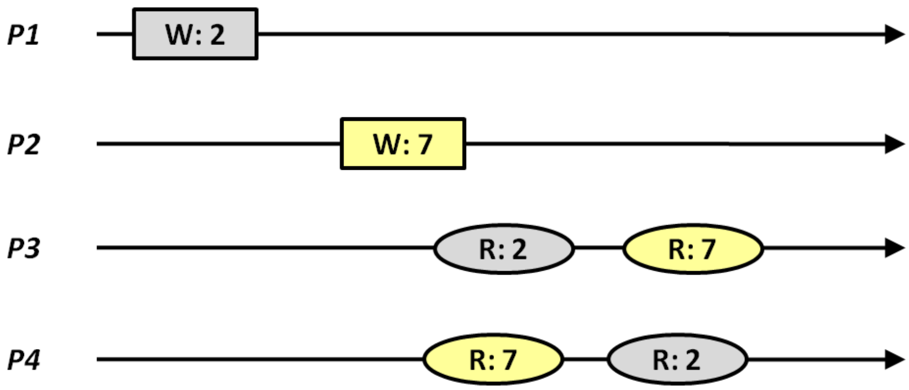

Figure 4.3: The behavior that is causally consistent

Next, we analyze the possible violated cases of causal consistency in MANETs using MIS-based register. In MANETs, whenever wants to write to the register, a mobile node send a message to the other nodes in network. Let consider a MIS-register of network consisting following events:

- Write operation $W : x = 2$ issued by mobile node 1 at time t_1
- Write operation $W : x = 7$ issued by mobile node 4 at time t_4
- $W : x = 2$ is carried by message $msg1$, $W : x = 7$ is carried in message $msg2$.

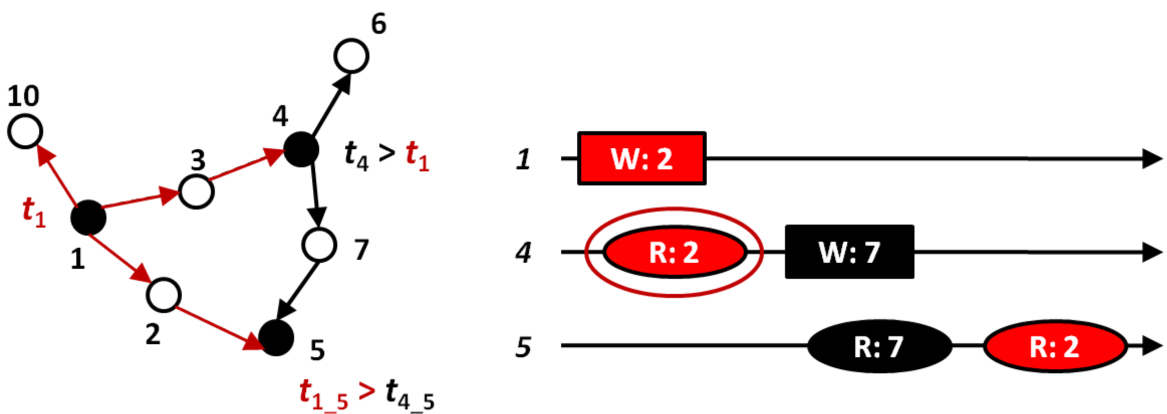

Figure 4.4: Two causally related write operations in MANETs

Suppose that, at $t_4 > t_1$, when mobile node 4 issues write operation $(W : x = 2)$, it completely received message $msg1$ and wrote updated value to local shared data. We call $(W : x = 2)$, $(W : x = 7)$ are causally related.

The causal consistency of register is violated at mobile node 5 when $msg2$ arrives at node 5 before $msg1$ $(t_{4\_5} < t_{1\_5})$.

In the Figure 4.4, red arrow lines represent to message $msg1$ and black arrow lines represent to message $msg2$. Suppose that, there are some reasons that make $msg1$ arrives to node 5 after $msg2$. Consequently, the causal consistency is violated if we follow the real time delivery of message.

But, in MIS-based register uses a committer-based clock. So that:

- When node 1 issues write operation $(W : x = 2)$ in UPDATE message $msg1$, $msg1$ has timestamp:

$$m_1 = [\langle 1, 1 \rangle]$$

- When m_1 arrive to 4 at time t_{1_4}, the committer-based clock in node 4 is updated according to Rule 2 $m_4 = [\langle 1, 1 \rangle]$

- When node 4 issues write operation $(W : x = 7)$, it update committer-based clock and attaches timestamps into $msg2$.

$$m_4 = [\langle 1, 1 \rangle, \langle 4, 1 \rangle]$$

- $msg2$ arrives to node 5 with timestamp m_2, and $msg1$ arrives to node 5 with timestamp m_1. According to the Definition 4.3 and 4.5, we have $m_1 < m_2$. Hence, the $(W : x = 2) \prec (W : x = 7)$ even if $msg1$ arrive after $msg2$.

Hence, the causal consistency is ensured in MIS-based register.■

4.3.2 The responses time

MIS-based register provide a immediate respond for a read operation because it reads register by accessing local shared data. Next, we consider the upper bound of response time for a write operation:

- A BLACK node will immediately commit the write operation of its.

- A WHITE node w has to send a RTW message to its highest rank BLACK node u. Firstly, it takes a period T_1 for unicasting a message in 1-hop and receive back the ACK message. Secondly, if there are concurrent write operations requested by other WHITE nodes to u, it costs at most Υ retry times for w to retransmit packet to u. Consequently, the time to transmit packet is at most $\Upsilon.T_1$. Moreover, it takes μ waiting time for CSMA/CD solve collisions.

  Therefore, the upper bound of response time for a write operation is $\Upsilon \cdot T_1 + \mu$.

In conclusion, based on the MIS-propagation mechanism, we propose a regular register which has:

- Causal consistency for concurrent write operations.

- The read operation is responded immediately because it does not wait for the concurrent write operation.

- The response time of a write operation issued by a WHITE node is $\Upsilon \cdot T_1 + \mu$. While a BLACK node immediately issues its write operation.

Chapter 5

Performance analysis

In this chapter, we analyze and compare the performance of our proposed algorithms and the reference algorithms. Firstly, we give the comparison of the mechanism maintaining MIS structure with the algorithm in [1]. Secondly, we evaluate the performance of MIS-based register in term of the percentage of forwarding nodes and the success rate of write operation. Finally, we give the comparison of the consistency levels between MIS-register and the other register suggested in [2–8].

5.1 Analysis of proposed MIS construction

The proposed algorithms to construct and maintain MIS is implemented in OMNET++. We compare the MIS created by our proposed algorithm with the algorithm given in [1] under properties of message and time complexity, self-stabilization.

The analysis in 5.1 shows that the MIS structure created by our algorithm has the internal distance between BLACK nodes is 3 hops while the MIS created by Wan's algorithm has the internal distance is 2 hops. Whereas our construction proposes a message complexity $O(n)$ when compared to $O(n \log n)$ of Wan's algorithm. Therefore, while Wan's algorithm is useful to create a strict MIS structure for stationary network, our proposed scheme is self-stabilizing and suitable to apply in MANETs where mobile nodes frequently move.

Table 5.1: Comparison of the two MIS constructions

Algorithm	Message complexity	Time complexity	Self-stabilizing	Internal distance
Wan's Algorithm	$O(n \log n)$	$O(n)$	No	2 hops
Our proposal	$O(n)$	$O(n)$	Yes	3 hops

5.2 Evaluation of MIS-based propagation

In this section, we evaluate the performance of MIS-based propagation in the comparison to the three results of [12], [13] (CDS-based flooding), [14] (Pure Flooding) and [15] (1-hop information flooding).

5.2.1 Simulation environment and metrics

In order to evaluate the performance of algorithms in this work, we use the standard MAC layer scheme following the IEEE 802.11 MAC specification. We use the 1-hop broadcast mode without RTS/CTS/ACK mechanism. According to the IEEE 802.11 the packet size is 2304 bytes plus 28 bytes of header fields. The bandwidth of wireless channel is set to 54Mb/s as default. Transmission range is set to $25m$. The network is deployed in the area of $300m \times 300m$. The mobility model used in our simulation is Random Waypoint Model (RWP) and Mass Mobility consequently. In RWP, each node moves to uniformly randomly chosen destination.

We concern the performance of a register in two aspects or metrics. The first one is the percentage of nodes forwarding the message when they receive. Whenever receiving a message, every node forwards the message only once. So the number of nodes forwarding the message is also the number of forwarding messages. This variable is given by the division of the number of forwarding messages on the total number of nodes in the network. The second metric is the success rate of a write operation. It is the average percentage of nodes in which a write operation is delivered successfully. We investigate how the two variables is affected by two parameters. The first parameter is the number of nodes that are deployed in the network. The second parameter is the speed of the mobility.

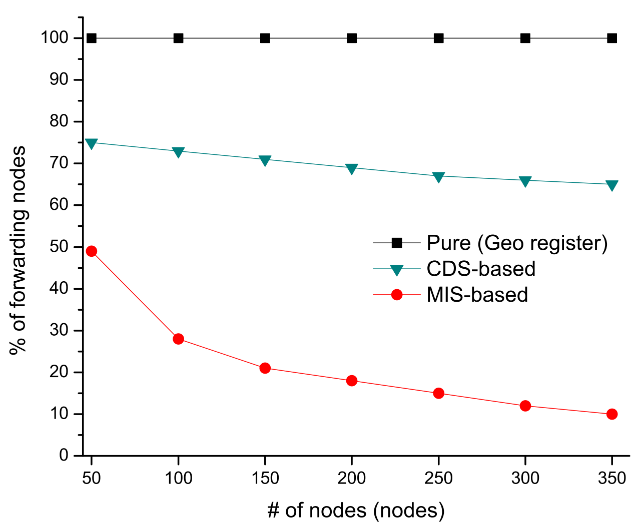

Figure 5.1: The ratio of retransmission on number of nodes

5.2.2 The percentage of forwarding nodes

The percentage of forwarding nodes shows the efficiency of the propagation algorithm used in MIS register. The lower percentage of forwarding nodes, the better efficiency of propagation algorithm. In order to evaluate the effect of the number of nodes on the percentage of forwarding nodes, we make a simulation with the transmission range is fixed at $25m$ and the moving velocity is fixed at $10m/s$. The parameter number of nodes is changed from 50 to 350 with the step is 50. The simulation results are plotted in Figure 5.1.

Because every nodes simply forwards the receiving message, the Pure flooding or the fully broadcast protocol always has the ratio is 1. The proposed MIS-based propagation has a significant improve when compared with two protocol CDS-based flooding [12] and 1-hop information flooding [15]. The number of retransmissions in our proposed protocol depends on ratio of the transmission range per the area of network. Because two independent BLACK nodes independent to each others. Hence, the percentage of forwarding nodes is reduced according to the increment of number of nodes in a specific area. Which is 50% better than the corresponding result of [15] and 70% − 80% better than the results of [12].

5.2.3 The success rate of write operation

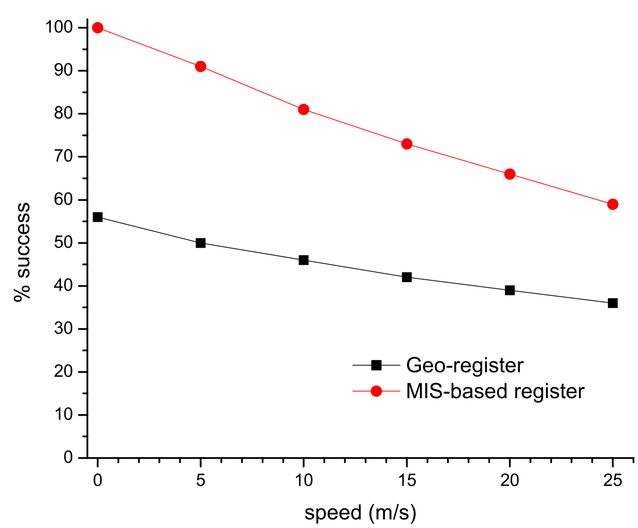

Figure 5.2: The success rate of write operation

The success rate of write operation shows the efficiency of proposed MIS-based register. The higher success rate of write operation, the higher confident of register and the higher consistency level of the shared data in the register. In MANETs, the success rate of write operation strongly depends on the speed of node mobility.

In order to evaluate the success rate of write operation we change the speed of nodes from $0m/s$ to $25m/s$. The simulation result in 3.4 shows that our MIS-based register has the success rate reducing from 100% down to 59%. This result is significant better than that of Geo-register using fully broadcasting protocol with success rate is $42-56\%$. Hence, our proposed MIS-based register is clearly better than Geo-register for multi-hop MANETs with medium speed of mobility.

5.3 Performance analysis of consistency level

In this analysis, we give the comparison between our proposed MIS-based register and the references works. The comparisons are given in the table 5.2 below.

Most of register based on cluster and quorum approach [4], [5] provide 2 mode of register. The first mode (1) is safe register, which is based on the quorum to set a lock/unlock for all partitions before starts writing to the shared value. Consequently, the safe register is automatically consistent. The second one (2) is regular register, which only ensure the local consistency for local cluster. Moreover, the works based on cluster/quorum approach require there are some stationary nodes, which play the role of cluster heads. Consequently, they do not support full mobility in which all nodes are allowed to move.

[6] and [8] support FIFO consistency for a regular register work with partial nodes move. But, these protocols requires some stationary nodes work as connectors which manages the consistency between two adjacent partitions.

[2] is the only work among the reference works that supports to the network having all nodes move. But this register is designed to 1-hop mobile network which have all nodes are connected in 1-hop connections. Moreover, in this register, when a node wants to write to shared data, it has to set a lock to all neighbors.

When compared with the related works, MIS-based register has a better supporting to MANEts in which all nodes are allow to move. Moreover, it ensures the Causal consistency model, which is various used in many applications. Hence, our MIS-based register can be used in the medium mobility MANETs in which the causal order of updated data is not a necessity.

Table 5.2: Comparison of the eight registers

Algorithm	Network	Type of register	Mobility	Consistency model
[2]	1-hop	safe	all nodes move	
[4]	multi hop	safe (1)	partial move	
	multi hop	regular (2)	partial move	local causal
[5]	multi hop	safe (1)	partial move	
	multi hop	regular (2)	partial move	local FIFO
[6], [8]	multi hop	regular	partial move	FIFO
MIS-base register	multi hop	regular	all nodes move	Causal

Chapter 6

Conclusion

6.1 Summary

In this work, motivated from Maximal Independent Set (MIS), we have proposed a regular register as the data management system Mobile Ad-hoc Networks (MANETs) called MIS-based register. MIS-based register supports multi-reader and multi-writer and ensures the causal data consistency model. This register is designed based on a set of algorithms including:

- A distributed algorithm for constructing and maintaining the MIS-structure for MANET. The proposed algorithm has the message complexity $O(n)$, where n is the number of nodes in network. Moreover, the proposed algorithm is proved that self-stabilizing after a $O(n)$ time.

- MIS-based propagating is a mechanism to select a set of forwarders and to transmit message to all mobile nodes in MANETs. The propagating mechanism significantly reduces about 50% the number of forwarders in transmitting when compared to the other flooding mechanisms.

Based on the proposed mechanisms of MIS-structure and message propagating, MIS-based register provides a mechanism supporting multi-writer with a short response time and ensuring the causal consistency property. The simulation result shows that MIS-based register has a high rate of success and costs a small number of forwarding message. Consequently, the response time to a write operation of MIS-based register is shorter than that of MIS-based register. Including such properties, MIS-based register is suitable to use in multi-hop MANETs having high rate updating and medium mobility.

6.2 Future work

The main issue is to design a data consistency management supporting various consistency models such as FIFO consistency, Sequential consistency or Linearizability. In the other words, according to the demand of application, there are different consistency models need to be provided. For instance, in a computation require the strict order of written value, a sequential consistency must be provided. To ensure the sequential consistency model there are two solutions. First, we can provide a total order broadcast scheme for multi-writer register. Second, we can design a lock/unlock mechanism to write operations.

Once the expansion is successfully studied, there will be a data consistency management system which provides many consistency models for different kinds of register.

Bibliography

[1] P. J. Wan, K. Alzoubi, and O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks," Mobile Networks and Applications, vol. 9, pp. 141-149, 2004.

[2] M. Roy, F. Bonnet, L. Querzoni, S. Bonomi, M. Killijian, D. Powell "Geo-registers: An Abstraction for Spatial-Based Distributed Computing," Proc. Int. Conf. On Principles Of DIStributed computing (OPODIS), pp. 534-537 2008.

[3] T. Hara, S. K. Madria, "Consistency Management among Replicas in Peer-to-Peer Mobile Ad Hoc Networks," Proc. IEEE Symp. Reliable Distributed Systems (SRDS), pp. 3-12, 2005.

[4] T. Hara, S. K. Madria, "Consistency Management Strategies for Data Replication in Mobile Ad Hoc Networks," IEEE Transactions on Mobile Computing, vol. 8, No. 7, pp. 950-967, 2009.

[5] G. Karumanchi, S. Muralidharan, and R. Prakash, "Information Dissemination in Partitionable Mobile Ad Hoc Networks," Proc. IEEE Symp. Reliable Distributed Systems (SRDS), pp. 4-13, 1999.

[6] J. Luo, J.P. Hubaux, and P. Eugster, "PAN: Providing Reliable Storage in Mobile Ad Hoc Networks with Probabilistic Quorum Systems," Proc. The ACM Int. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc), pp. 1-12, 2003.

[7] E. Pitoura and B. Bhargava, "Data Consistency in Intermittently Connected Distributed Systems," IEEE Trans. Knowledge and Data Eng., vol. 11, no. 6, pp. 896-915, Nov./Dec. 1999.

[8] K. Rothermel, C. Becker, and J. Hahner, "Consistent Update Diffusion in Mobile Ad Hoc Networks," Technical Report 2002/04, Computer Science Dept., Univ. of Stuttgart, 2002.

[9] L. Lamport, "On Interprocess Communication," Springer-Verlag, Dec. 1985.

[10] J. M. Byskov, "Algorithms for k-colouring and finding maximal independent sets," ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 456-457, 2003.

[11] I. Cidon and O. Mokryn, "Propagation and leader election in multihop broadcast environment, in: Proc. of 12th International Symposium on DIStributed Computing," (DISC), pp. 104-119, September 1998.

[12] F. Dai and J.Wu, "An Extended Localized Algorithm for Connected Dominating Set Formation in Ad hoc Wireless Networks," IEEE Trans. on Parallel and Distributed Systems, Vol. 15, No. 10, pp. 908-920, October 2004.

[13] J. Wu and H. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad hoc Wireless Networks", Proc. Third ACM Int'l Workshop DiaLM, pp. 7-14, August 1999.

[14] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The Broadcast Storm Problem in a Mobile Ad hoc Network", Proc. Fifth ACM/IEEE MOBICOM, pp. 151-162, August 1999.

[15] H. Liu, X. Jia, P. J. Wan, X. Liu, and F. F. Yao, "A Distributed and Efficient Flooding Scheme Using 1-Hop Information in Mobile Ad-hoc Networks", IEEE Trans. on Parallel and Distributed Systems, Vol. 18, No. 5, pp. 658671, May 2007.