



Coordinate-based addressing for MANET

Zhengxiong Dou¹ · Xiaonan Wang¹ · Yanli Li¹

Published online: 6 August 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

This paper proposes a Coordinate-based addressing solution for a mobile ad hoc network, and aims to configure each node with a unique address and reduce the addressing cost and latency. This solution configures all the nodes with unique addresses via one address initialization process and guarantees the address uniqueness in a merged or split network via address update. Finally, this solution is analyzed and evaluated, and the data results show that this solution effectively reduces the addressing cost and latency.

Keywords Mobile ad hoc network · Addressing · Uniqueness · Network merging · Network partitioning

1 Introduction and motivation

1.1 Issues in one MANET

A mobile ad hoc network (MANET) is a multi-hop network without any infrastructures [1]. In one MANET, neighbors can directly communicate with each other whereas nodes beyond each other's communication range must depend on intermediate nodes to perform indirect communications [2]. During a communication process, a node uses a unique address to identify itself, so a requisite of performing correct communications is that each node is configured with a unique address [3, 4]. In order to achieve address configuration for one MANET, some addressing approaches [5–22] are proposed. In these solutions, each node independently performs an addressing process, so the addressing cost and latency are relatively high. Moreover, in some cases the solutions cannot guarantee address uniqueness and address conflicts might happen. Therefore, one major issue in one MANET is how to achieve low-cost and low-latency address configuration and guarantee address uniqueness.

Node mobility results in frequent network merging/partitioning that might incur address conflicts [6], so it is important to effectively detect network merging/partitioning and guarantee address uniqueness during a network merging/partitioning process. Currently, network

merging/partitioning is usually detected by periodically updating and flooding a network ID, and address uniqueness is ensured by performing address reconfiguration. However, the periodical flooding and address reconfiguration inevitably degrade the network merging/partitioning performance. Therefore, another major issue in one MANET is how to achieve low-cost and low-latency network merging/partitioning and ensure address uniqueness during a network merging/partitioning process.

1.2 Motivation and contributions

Taking into account these issues, we are motivated to make use of coordinates to achieve the addressing and make use of address update instead of address reconfiguration to perform the network merging/partitioning, and aim to ensure the address uniqueness and reduce the addressing and network merging/partitioning costs and delays.

Based on the motivation, we propose a Coordinate-based addressing solution for one MANET that has the following contributions:

- (1) The Coordinate-based architecture and address structure are proposed. In the architecture one network is identified by one MANET ID and in the address structure one MANET ID is a part of the proposed address structure. Consequently, the network merging/partitioning can be effectively detected by checking the addresses of nodes.

✉ Xiaonan Wang
xnwang_2013@163.com

¹ Changshu Institute of Technology, Changshu 215500, Jiangsu, China

- (2) Based on the Coordinate-based architecture and address structure, the Coordinate-based addressing algorithm is proposed. In the algorithm, all the nodes can be configured with unique addresses via one addressing process by using their coordinates. Consequently, the addressing cost and latency are reduced.
- (3) Based on the Coordinate-based architecture and address structure, the network merging and partitioning algorithms are proposed. The algorithms make use of the address update rather than address reconfiguration to ensure the address uniqueness, so the network merging and partitioning costs and delays are lowered.

The rest of this paper is organized as follows. In Sect. 2, the related work on the addressing is discussed, and in Sect. 3 the Coordinate-based architecture is proposed. Based on the architecture, the Coordinate-based addressing algorithm and the network partitioning and merging algorithms are proposed in Sects. 4 and 5, respectively. The algorithms are analyzed and evaluated in Sect. 6 and 7, respectively, and this paper concludes with a summary in Sect. 8.

2 Related work

At present, some addressing solutions for one MANET are proposed and they are classified into two categories: the addressing with a special network topology and the addressing without a special network topology.

2.1 Addressing with a special network topology

In the addressing solutions with a special network topology, nodes are configured with addresses by joining the special topology, but maintaining a special network topology might bring the extra cost and delay to the addressing.

In [7], an addressing solution based on super nodes is proposed. One super node is made up of a few nodes and performs the addressing as a unit. Through one addressing process, all the nodes in one super node are configured with unique addresses, so the addressing performance is improved. In [8], a network is divided into clusters. One cluster head is responsible for configuring each cluster member with an address. In this way, the cluster members in different clusters might perform the addressing in parallel, so the addressing delay is reduced. In [9], an address method based on multi-hop clusters is proposed. In this method, a cluster head first calculates the weights of cluster members based on location information and then makes use of the weights to achieve the addressing for cluster members via one addressing operation. Consequently, the addressing cost and latency are reduced. In [10], a network is organized into a chain topology and the nodes in a chain acquire addresses from the

head of the chain via the stateful strategy, so the uniqueness of the address of each node can be guaranteed without duplicate address detection. In [11], a network is organized into grids and one grid is identified by location coordinates. One node in a grid is configured with a globally unique address based on the location coordinates of the grid without duplicate address detection. Moreover, the addressing process in one grid is independent of the one in another grid, so the addressing processes in different grids can be performed in parallel and the addressing latency is reduced. The addressing solutions [7–11] achieve the addressing, but they do not address the network partitioning and merging issues.

In [5], a network is organized into a tree and three kinds of nodes are defined: a normal node that acts as a relay, a leader node that is responsible for configuring each new node with an address, and a root that is in charge of network merging. A network merging process includes three stages: generating a new root, collecting addresses in use, and performing address reconfiguration. The second stage and the third stage are achieved in a flooding way, so the network merging cost and delay are relatively high. During a network partitioning process, each node in a split network performs an address configuration process to get a new address. In [12], a network is organized into a tree and a new node is configured with an address by joining a tree. During a network merging process, the ad hoc on-demand distance vector routing protocol (AODV) is used to detect address conflicts in a passive way, so this scheme does not guarantee address uniqueness in a merged network. In a network partitioning process, a new root is elected and the nodes in a split network acquire addresses by performing address reconfiguration. In [13], a network is also organized into a tree. In this scheme, a node acquires a unique address by joining a tree. A network merging process consists of two stages: electing a new root and address configuration. In a network partitioning process, the nodes in a split network perform address reconfiguration to acquire addresses. The addressing solutions [5, 12, 13] guarantee address uniqueness by performing address reconfiguration during a network partitioning/merging process, so the cost and latency are relatively high.

2.2 Addressing without a special network topology

Since the special network topology maintenance brings the extra cost and latency to the addressing, some addressing solutions without a special network topology are proposed.

In [14], an improved duplicate address detection (DAD) is proposed to achieve the addressing. In the algorithm, only a part of a target address is broadcasted in a network, so the addressing success rate is improved. However, this solution is performed in a flooding way, so the addressing cost is relatively high. In [15], a node acquires a unique address without DAD, so the addressing cost and delay are relatively

low. Moreover, the released addresses can be automatically reclaimed without any extra operations, so there is always an abundant address space available for allocation. Consequently, the addressing success rate is improved. In [16], the authors make a distinction between home addresses and care-of addresses and propose the addressing algorithms for home addresses configuration and care-of addresses configuration, respectively. The care-of address configuration is achieved based on the home address addressing, so the care-of address configuration latency is shortened. In [17], the authors propose a stateful addressing solution to achieve address configuration for nodes. This solution employs the idea of the dynamic host configuration protocol (DHCP) [18] and deploys a server to assign a unique address to each node, so the address configuration cost and latency caused by duplicate address detection are avoided. In [19], a network is organized into a tree and the root starts an addressing process. An address is 16-bit long and a new node acquires an address by joining a tree. In a tree, an intermediate node is configured with two addresses and a leaf node is configured with one address. The addressing solutions [14–17, 19] achieve the addressing, but they do not address the network partitioning and merging issues.

In [6], three types of nodes are defined: a type-1 node with reserved addresses, a type-2 node without reserved addresses and a new node without an address. If a new node requests an address from a neighbor type-1 node, the type-1 node directly allocates a reserved address. If a new node requests an address from a neighbor type-2 node, the type-2 node borrows an address and assigns the address. In [20], an agent node and a simple node are defined. One agent has an independent address space and a simple node acts as a relay. A new node acquires an address from the agent with the maximum address space. In a network merging process, the address uniqueness is checked and guaranteed in a flooding way. In [21], the authors propose a distributed address configuration scheme for a hybrid MANET. This scheme employs a stateful addressing strategy to achieve the addressing so that a new node can be configured with an address without DAD. In [22], a filter-based address auto-configuration protocol (FAACP) is proposed. In FAACP, a node randomly chooses an address and verifies the uniqueness of this address. If the address is in use, the node selects another address and verifies the address again. The node repeats the operations until it obtains a unique address. In the solutions [6, 20–22], during a network merging/partitioning process, nodes acquire unique addresses by performing address reconfiguration, so the cost and latency are relatively high.

2.3 Our solution

Based on the above discussion, the existing solutions are degraded by the following factors:

- 1) Each node independently performs an addressing process and in some cases address conflicts might happen.
- 2) The addressing solutions with a special network topology bring the extra cost and delay to the addressing.
- 3) The network merging and partitioning are usually detected by periodically updating and flooding a network ID and the address uniqueness is ensured by performing address reconfiguration.

Taking these factors into account, we propose the following strategies to improve the addressing and network merging/partitioning performance:

- 1) The nodes use their coordinates to acquire unique addresses via one addressing process.
- 2) The nodes achieve the addressing without special network topology support.
- 3) A network ID is included in an address so that the network merging/partitioning can be detected without updating and flooding a network ID. During a network merging/partitioning process, the address uniqueness can be guaranteed via address update rather than address reconfiguration.

3 Coordinate-based architecture

This scheme defines three types of nodes. A configured node is a node with an address and a new node is a node without an address. A coordinating node is a node on the border of one MANET when the MANET is initialized. Especially, the node located on the east border is called an east coordinating node (EC), the node on the south border a south coordinating node (SC), the node on the west border a west coordinating node (WC), and the node on the north border a north coordinating node (NC). There are only four coordinating nodes in one MANET, so a coordinating node actually refers to a role of EC/SC/WC/NC.

After a network starts, the EC/SC/WC/NC is first elected and then each node calculates and stores the 4-tuple coordinates (e, s, w, n) where $e/s/w/n$ is the distance (hop) from the node to the EC/SC/WC/NC. As shown in Fig. 1, in one MANET the EC/SC/WC/NC is elected and node X acquires the 4-tuple coordinates (3, 3, 1, 2).

4 Coordinate-based addressing

Based on the Coordinate-based architecture in Fig. 1, the Coordinate-based address structure is presented. Based on the architecture and address structure, the Coordinate-based addressing solution is proposed. The Coordinate-based addressing solution includes the address initialization and

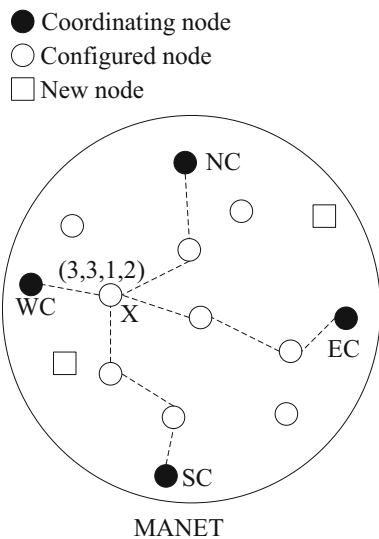


Fig. 1 Coordinate-based architecture

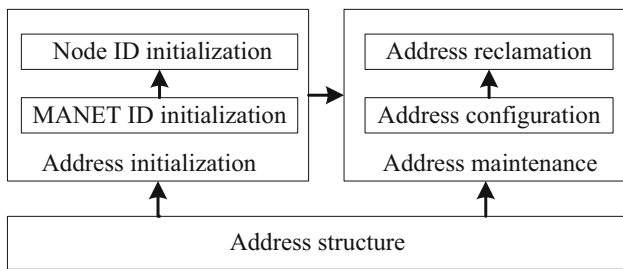


Fig. 2 Coordinate-based addressing framework

address maintenance algorithms, as shown in Fig. 2. The address initialization process consists of the MANET ID initialization and node ID initialization sub-processes. The address maintenance process is built on the address initialization process, and is made up of the address configuration and address reclamation sub-processes. The address initialization process is performed only once when one MANET is initialized. After the address initialization process is complete, the address maintenance process is performed to configure new nodes with addresses and reclaim released addresses.

4.1 Address structure

Based on the Coordinate-based architecture, the Coordinate-based address structure is proposed, as shown in Table 1. An address consists of four parts. The first part is the MANET ID which uniquely identifies one MANET, and all the nodes in a network have the same MANET ID. The second part is the 4-tuple coordinates (e, s, w, n) . The third part is the node ID which is used to distinguish among the nodes with the same 4-tuple coordinates. The fourth part is the new node ID which is used to perform the addressing for new nodes.

Table 1 Address structure

Number of bits: i	j	j	j	j	k	m
MANET ID	e	s	w	n	Node ID	New node ID
Coordinates						

In Table 1, the length of the MANET ID is i -bit, the length of each element in the 4-tuple coordinates is j -bit, the length of the node ID is k -bit, the length of the new node ID is m -bit, and i, j, k and m are positive integers. A bit has a single binary value, either 0 or 1, so the maximum unsigned number that $j/k/m$ bits in Table 1 can represent is $2^j - 1/2^k - 1/2^m - 1$. Since $e/s/w/n$ stores the distance from one node to the EC/SC/WC/NC, 2^j should be greater than a network diameter that is defined as the maximum hop between any two nodes in one MANET. The node ID is used to distinguish among the nodes with the same 4-tuple coordinates. Since the nodes with the same 4-tuple coordinates are neighbors, 2^k should be greater than the number of a node's neighbors. The new node ID is used to perform the addressing for new nodes, so 2^m should be greater than the total number of nodes in one MANET. For example, if the total number of nodes in one MANET is smaller than 1024 (namely, 2^{10}), m should be set to a number that is not smaller than 10.

Based on [23], the conflict probability $P_{Conflict}$ of x MANET IDs is shown in formula (1). If Condition 1 is satisfied, $P_{Conflict}$ tends to be zero, as shown in formula (2). In this scheme, the MANET IDs are used to determine if two networks merge, so x is equal to 2. If i is equal to 24 and x is 2, $P_{Conflict}$ is approximate to 0.000001226 based on formula (1).

Condition 1: 2^i tends to be infinite, and x is much smaller than 2^i .

$$P_{Conflict} = 1 - e^{-x} (1 - \frac{x}{2^i})^{x-2^i-\frac{1}{2}} \tag{1}$$

$$P_{Conflict} \approx 1 - e^{-x} (1 - \frac{x}{2^i})^{-2^i} = 1 - e^{-x} (1 - \frac{x}{2^i})^{-\frac{2^i}{x} \cdot x} \approx 1 - e^{-x} \cdot e^x = 0 \tag{2}$$

4.2 Address initialization

The goal of the address initialization process is to configure each node with one unique address and includes the MANET ID configuration and the node ID initialization sub-processes. The MANET ID initialization sub-process is to configure each node with one MANET ID and the node ID initialization sub-process is to configure each node with a node ID. After one node performs the MANET ID initialization and node

ID initialization sub-processes, it can be configured with a unique address.

4.2.1 MANET ID initialization

After a network starts, each node is a new node and uses a media access control (MAC) address as a temporary address (TA) where the MANET ID and 4-tuple coordinates are zero. Before a new node is configured with a unique address, it is identified by its TA. In this scheme, some nodes know their location coordinates (x, y) which might be pre-configured in advance or be acquired via GPS [24]. Each node periodically sends a *hello* to its neighbors [25], the source address in the *hello* is the TA of the node and the payload is the location coordinates of the node. If a node does not know its location coordinates, it sets the location coordinates to $(-1, -1)$. After a node receives a *hello* from N neighbors with location coordinates (x_c, y_c) ($x_c \neq -1, y_c \neq -1, 1 \leq c \leq N$), it employs the centroid localization algorithm to calculate an estimated coordinates (x_e, y_e) , as shown in Eq. (3–4) [26] and piggybacks the estimated coordinates in the next *hello*. In this way, each node can estimate its coordinates.

$$x_e = \sum_{c=1}^N x_c / N \tag{3}$$

$$y_e = \sum_{c=1}^N y_c / N \tag{4}$$

After node X (x, y) receives a *hello* from M neighbor nodes with estimated coordinates (x_b, y_b) ($1 \leq b \leq M$), it determines if it is a coordinating node based on the following election algorithm:

1. If x is smaller than $\text{Min}_{b=1}^{b=M}(x_b)$, X marks itself as one WC.
2. If x is greater than $\text{Max}_{b=1}^{b=M}(x_b)$, X marks itself as one EC.
3. If y is smaller than $\text{Min}_{b=1}^{b=M}(y_b)$, X marks itself as one SC.
4. If y is greater than $\text{Max}_{b=1}^{b=M}(y_b)$, X marks itself as one NC.

After a node changes into a coordinating node, it establishes the initial 4-tuple coordinates. The initial 4-tuple coordinates of one EC/SC/WC/NC is $(0,1,1,1)/(1,0,1,1)/(1,1,0,1)/(1,1,1,0)$. As shown in Fig. 3a, node X1 has two neighbors Y1 and Y2 and is elected as one WC based on the election algorithm.

Because the election algorithm is based on neighbors, it is possible that more than one EC/SC/WC/NC is elected, as shown in Fig. 3a where nodes X1 and X2 are elected as WCs.

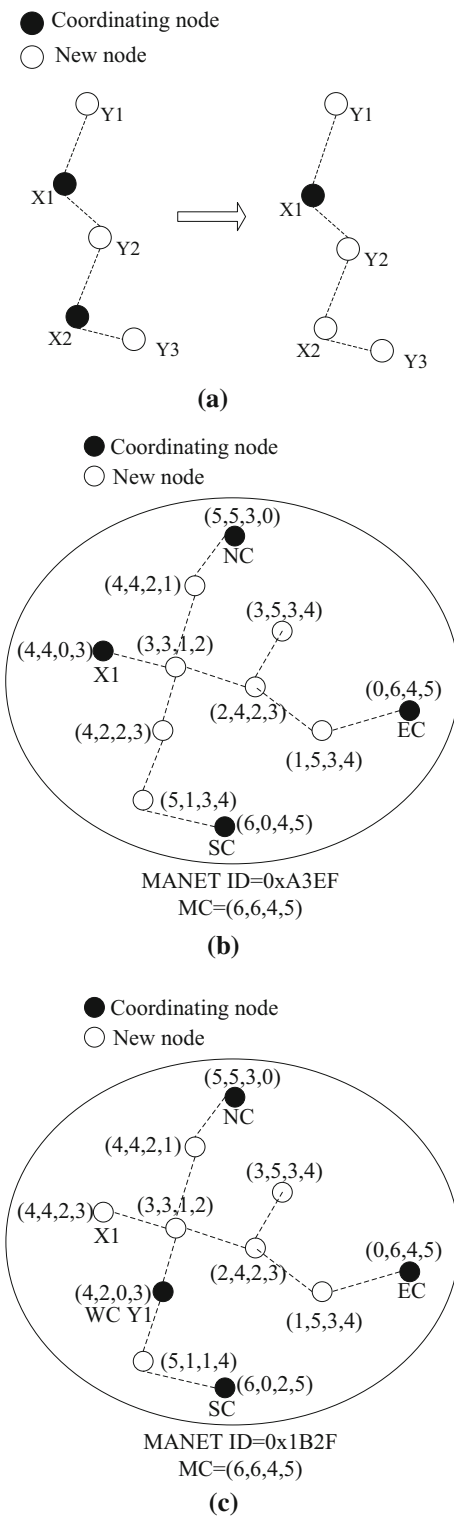


Fig. 3 MANET ID initialization. a WC election, b MANET ID generation, c estimation inaccuracy

In order to ensure that there is only one EC/SC/WC/NC in a network, each coordinating node does the following operations where a WC is taken as an example:

- (1) After X1 becomes a WC, it randomly generates one MANET ID and constructs a 2-tuple <TA, MANET ID> where the TA is the one of X1 and the MANET ID is the one generated by X1. Then, X1 sends a *hello* where the source address is the TA and the payload is the initial 4-tuple coordinates and the constructed 2-tuple.
- (2) Another coordinating node X2 receiving the *hello* compares its TA with the TA in the 2-tuple in the *hello*. If the former is greater, it stores the 2-tuple in the *hello* and changes into a new node.

Ultimately, only the WC with the smallest TA becomes the final WC, and the MANET ID generated by the coordinating node with the smallest TA becomes the final MANET ID. Similarly, the final EC/SC/NC is generated in the same way. If a node receives a *hello* from multiple EC/SC/WC/NC, it marks the EC/SC/WC/NC with the minimum TA as the final EC/SC/WC/NC, stores the hops from itself to the final EC/SC/WC/NC, establishes the 4-tuple coordinates and stores the final MANET ID. As shown in Fig. 3a, X1 has a smaller TA than X2, so it becomes one WC and X2 changes into a new node. As shown in Fig. 3b, the WC has a smaller TA than the EC/SC/NC, so the MANET ID 0xA3EF generated by the WC becomes the MANET ID.

The ultimate goal of electing the EC/SC/WC/NC is to configure a new node with the MANET ID and 4-tuple coordinates. In this scheme, the EC/SC/WC/NC is elected by using the estimated coordinates which are usually inaccurate. This scheme takes this estimation inaccuracy into account so that it can be still performed correctly with estimation errors. As shown in Fig. 3c, due to estimation errors, node Y1 is elected as a WC. Compared with the situation without estimation errors in Fig. 3b, each new node in the situation with estimation errors might acquire the different MANET ID and 4-tuple coordinates. Although the estimation inaccuracy leads to the different MANET ID and 4-tuple coordinates, the goal of electing the coordinating nodes is still achieved and a new node is configured with the MANET ID and 4-tuple coordinates in the same way as in the case without estimation errors. Therefore, the estimation inaccuracy has no impact on the performance of this scheme.

4.2.2 Node ID initialization

A new node acquires the MANET ID and 4-tuple coordinates via the MANET ID initialization algorithm. If the 4-tuple coordinates of a new node are unique, it uses the MANET ID and 4-tuple coordinates to construct an address where the node ID and new node ID are zero and directly changes into a configured node. As shown in Fig. 3b, new node X1 acquires the unique 4-tuple coordinates via the MANET ID initialization sub-process, so it constructs a unique address

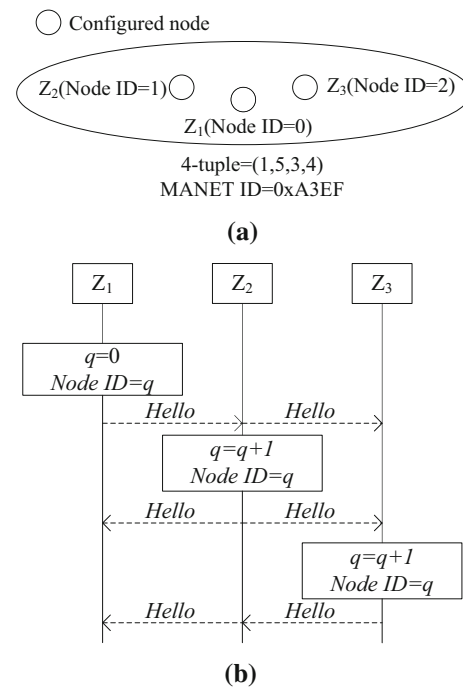


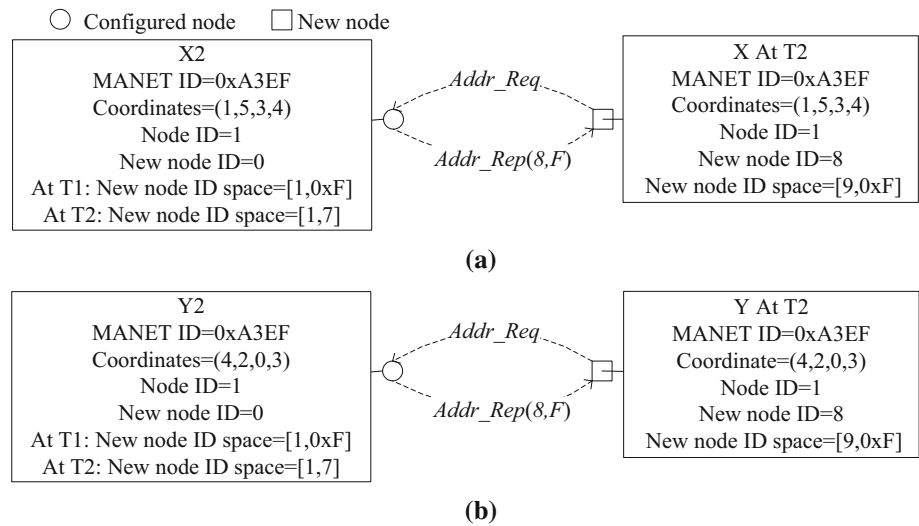
Fig. 4 Node ID initialization

where the MANET ID is 0xA3EF, the 4-tuple coordinates are (4, 4, 0, 3), and the node ID and new node ID are zero.

If neighbor nodes acquire the same 4-tuple coordinates via the MANET ID initialization sub-process, they perform the node ID initialization sub-process to acquire unique node IDs and then use the unique node IDs to construct the unique addresses. It is assumed that P ($P \geq 2$) neighbors denoted by Z_p ($P > p \geq 1$) have the same 4-tuple coordinates. After Z_p constructs an address where the node ID is q , it broadcasts a *hello* with the 4-tuple coordinates and q , and changes into a configured node. Z_{p+1} receiving the *hello* increases q by 1, sets its node ID to q , broadcasts a *hello* and changes into a configured node. The above process is repeated until P neighbors broadcast a *hello* and change into a configured node. Neighbor nodes share the wireless medium and only one node occupies the medium to send a message at a time [27], so these neighbors send a *hello* one by one. In this scheme, the first node that successfully occupies the medium to send a *hello* where q is 0 initiates the node ID initialization sub-process. Since the neighbors send a *hello* one by one, it is impossible that one node receives multiple *hello* messages from the neighbors at the same time.

As shown in Fig. 4a, b, new nodes Z_1 , Z_2 and Z_3 acquire the same 4-tuple coordinates via the MANET ID initialization algorithm, so they perform the node ID initialization sub-process to acquire a unique node ID. Z_1 is the first node that occupies the wireless medium and sends a *hello* where q is 0, so it sets its node ID to 0 and changes into a configured node. After Z_2/Z_3 receives a *hello* from Z_1 , it increases q

Fig. 5 Address configuration



in the *hello* by 1. Then, Z_2 occupies the medium and sends a *hello*, so it sets its node ID to q (namely 1), and changes into a configured node. After Z_3 receives a *hello* from Z_2 , it increases q in the *hello* by 1. Finally, Z_3 occupies the medium and sends a *hello*, so it sets its node ID to q (namely 2) and changes into a configured node. In this way, $Z_1/Z_2/Z_3$ acquires unique node ID 0/1/2 via the node ID initialization sub-process. Then, $Z_1/Z_2/Z_3$ constructs a unique address where the MANET ID is 0xA3EF, the 4-tuple coordinates are (1, 5, 3, 4), the node ID is 0/1/2 and the new node ID is zero.

4.3 Address maintenance

After the address initialization sub-process is complete, a configured node broadcasts a *hello* where the payload is the maximum 4-tuple coordinates (MC) ($E_{max}, S_{max}, W_{max}, N_{max}$) where $E_{max}/S_{max}/W_{max}/N_{max}$ is the maximum value among the first/second/third/fourth element of the 4-tuple coordinates of the configured nodes in a network. A configured node sets the initial value of one MC to its 4-tuple coordinates. After a configured node receives a *hello* from a neighbor, it compares its MC with the MC in the *hello*. If one element in the MC of the node is smaller, the node updates the element with the corresponding one in the *hello*. Finally, all the configured nodes have the same MC, as shown in Fig. 3b where the MC is (6, 6, 4, 5).

4.3.1 Address configuration

As shown in Fig. 5a, new node X starts after the address initialization sub-process. X first listens to a *hello* from a configured node and then requests an address from configured node X2 with the strongest signal strength based on the following algorithm:

- (1) X sends one *Addr_Req* message to X2 with new node ID space $[L, U]$ ($L \leq U$).
- (2) X2 receiving the *Addr_Req* assigns new node ID space $[L1, U]$ ($L \leq L1 \leq U$), returns one *Addr_Rep* message where the payload is the assigned new node ID space, and updates its new node ID space with $[L, L1-1]$.
- (3) X receiving the *Addr_Rep* sets its new node ID to $L1$, combines $L1$ with X2's MANET ID, coordinates and node ID to form an address, changes into a configured node, and sets its new node ID space to $[L1 + 1, U]$.

In Fig. 5a, at time T1, X requests an address from X2. At time T2, X acquires an address where the MANET ID is 0xA3EF, the coordinates are (1, 5, 3, 4), the node ID is 1 and the new node ID is 8.

After the address initialization sub-process is complete, each configured node has a unique address. That is to say, the combination of the 4-tuple coordinates and node ID of each configured node is unique. As shown in Fig. 5a, b, new nodes X and Y acquire an address from configured nodes X2 and Y2, respectively. The addresses of X and Y are unique even if X2 and Y2 assign the same new node ID space (namely, [8, 0xF]) and X and Y acquire the same new node ID (namely, 8). The main reason is that X2 has the different combination of the 4-tuple coordinates and node ID from Y2. After X acquires a new node ID from X2, it combines the new node ID with the MANET ID, 4-tuple coordinates and node ID of X2 to construct an address. Similarly, after Y acquires a new node ID from Y2, it also combines the new node ID with the MANET ID, 4-tuple coordinates and node ID of Y2 to form an address. Therefore, X has the different combination of the 4-tuple coordinates and node ID from Y because X2 has the different combination of the 4-tuple coordinates and node ID from Y2, X inherits the 4-tuple and node ID of X2 and Y inherits the ones of Y2. Consequently, X has a different address from Y even if they have the same new node ID.

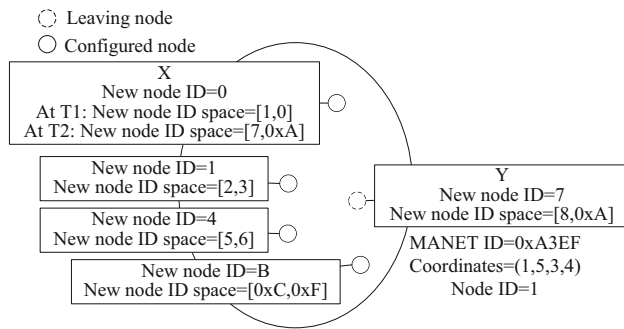


Fig. 6 Address reclamation

4.3.2 Address reclamation

If the size of the new node ID space of configured node X is zero, it launches the following address reclamation process:

- (1) X broadcasts one *Addr_Rec* message.
- (2) After a node with the same coordinates and node ID receives this *Addr_Rec*, it returns one *Addr_Ack* message where the payload is its new node ID space.
- (3) X checks all the received *Addr_Ack* and marks the unoccupied new node ID space as its own space.

As shown in Fig. 6, at time T1 the size of the new node ID space of configured node X is zero, so it launches the address reclamation operation. At time T2, X reclaims the new node ID space of node Y since Y leaves the network.

5 Network partitioning and merging

The network partitioning and merging are two main factors impacting on the address uniqueness, so these two issues are addressed in this section to ensure the address uniqueness.

5.1 Partitioning

A configured node determines if it leaves the current network via one of the following three cases:

- *Case 1*: A configured node does not receive a *hello*.

In *Case 1*, the configured node changes into a new node. As shown in Fig. 7a, at time T1, nodes X1 and X2 are located in MANET M1. At time T1', X1 and X2 leave the network, becomes the isolated new nodes and stores the original addresses.

- *Case 2*: The *hello* messages received by a configured node all come from new nodes.

In *Case 2*, the configured node changes into a new node. If the configured node has a smaller TA than the neighbors, it randomly generates a new MANET ID and broadcasts one *Addr_Update* message where the source address is its TA and the payload is the MANET ID pair <original MANET ID, new MANET ID>. A new node receiving this *Addr_Update* stores the MANET ID pair in the *hello*. If a new node receives more than one *Addr_Update*, it marks as the final MANET ID pair the one in the *hello* with the smallest source address. If the original MANET ID of a new node is the same as the original one in the final MANET ID pair, it updates the MANET ID of the original address with the new MANET ID in the final pair to form a new address and changes into a configured node. Otherwise, the new node acquires an address via the address configuration algorithm. As shown in Fig. 7b, at time T1, node X3 is located in MANET M1. At time T2, X3 leaves M1 and receives a *hello* from X1 and X2. Since X3 has a smaller TA than X1 and X2, it generates new MANET ID 0xEF36 for new MANET M2 and correspondingly X1 and X2 updates the MANET ID to acquire the new addresses.

- *Case 3*: The MANET ID of a configured node is different from the one of the source address in all the received *hello* messages.

In *Case 3*, the configured node changes into a new node and acquires an address via the address configuration algorithm. In Fig. 7c, at time T3 node X enters new MANET M2 and acquires a new address from X1.

5.2 Merging

If configured node X receives both a *hello* where the MANET ID of the source address is the same as its MANET ID and a *hello* where the MANET ID is different from its MANET ID, it can determine that the network merging happens. If X is located in MANET M1 and node Y is located in MANET M2. After X receives a *hello* from Y, it checks its MC ($E_{max1}, S_{max1}, W_{max1}, N_{max1}$) and Y's MC ($E_{max2}, S_{max2}, W_{max2}, N_{max2}$). If either *Condition 2* or *Condition 3* is satisfied, X initiates the following merging operations:

Condition 2: The maximum element is included in Y's MC.

Condition 3: The maximum element is included in X's MC and Y's MC, but the average value of four elements in X's MC is not larger than the one in Y's MC.

- (1) X broadcasts one *Merging_Addr* message where the payload is the MANET ID and MC in M2 and the MC in M1.

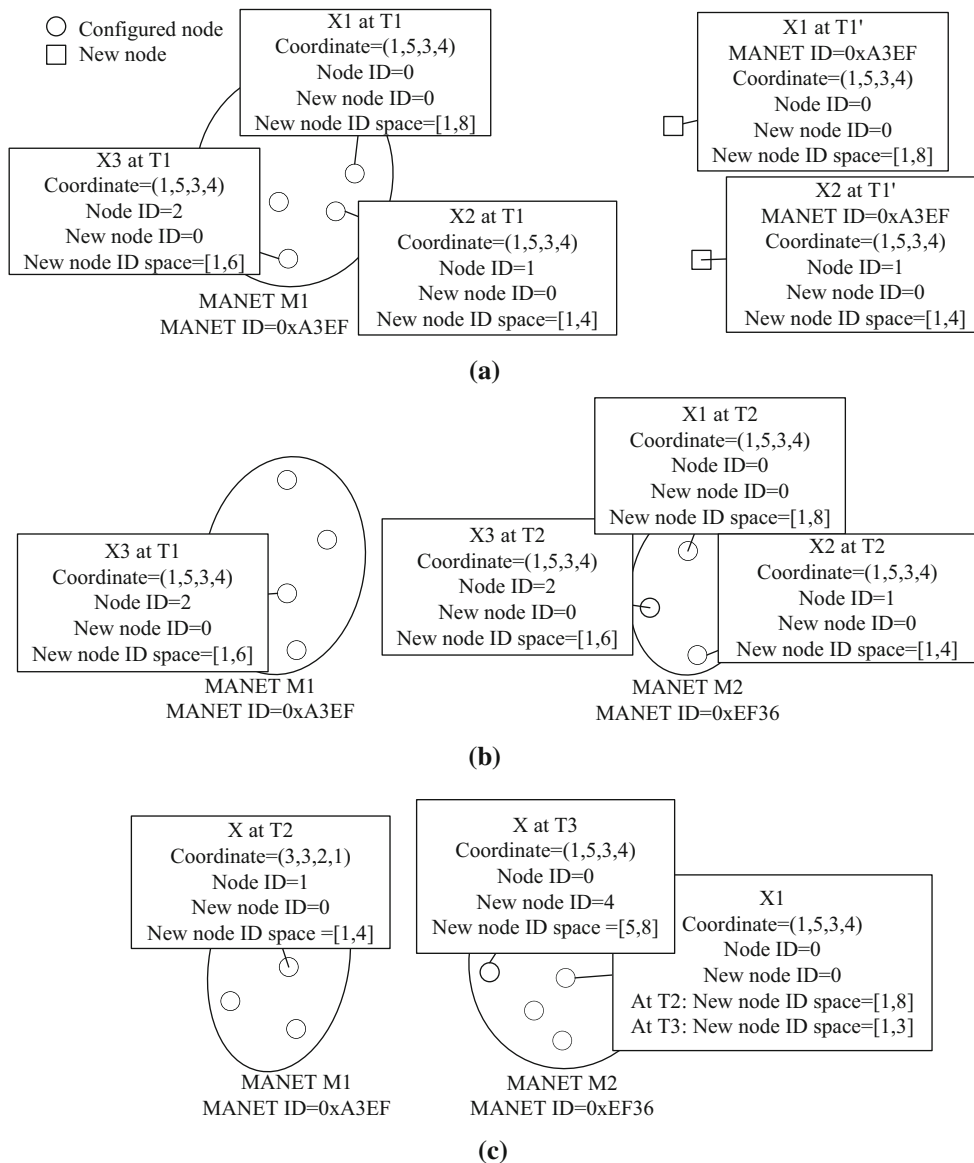


Fig. 7 Network partitioning. a Case 1, b Case 2, c Case 3

- (2) After a node in M1 receives the *Merging_Addr*, it does the following operations where the 4-tuple coordinates of the node is (e, s, w, n) :
 - Update the MANET ID with the one in M2.
 - Increase e by E_{max2} , s by S_{max2} , w by W_{max2} , and n by N_{max2} .
 - Update the 4-tuple coordinates of the address with the new 4-tuple coordinates.
 - Increase E_{max1} by E_{max2} , S_{max1} by S_{max2} , W_{max1} by W_{max2} , and N_{max1} by N_{max2} .
- (3) After a node in M2 receives the *Merging_Addr*, it increases E_{max2} by E_{max1} , S_{max2} by S_{max1} , W_{max2} by W_{max1} , and N_{max2} by N_{max1} .

- (4) The network merging process is complete.

As shown in Fig. 8a, b, before MANETs M1 and M2 merge, nodes X, X1 and X2 are located in M1 and nodes Y, Y1, Y2 and Y3 are located in M2. After M1 and M2 merge, MANET ID 0xA3EF in M2 becomes the MANET ID in the merged network, and correspondingly X, X1 and X2 updates the MANET ID and coordinates to acquire the new addresses.

6 Analysis

In this section, the addressing cost and delay, the network partitioning cost and delay and the network merging cost and

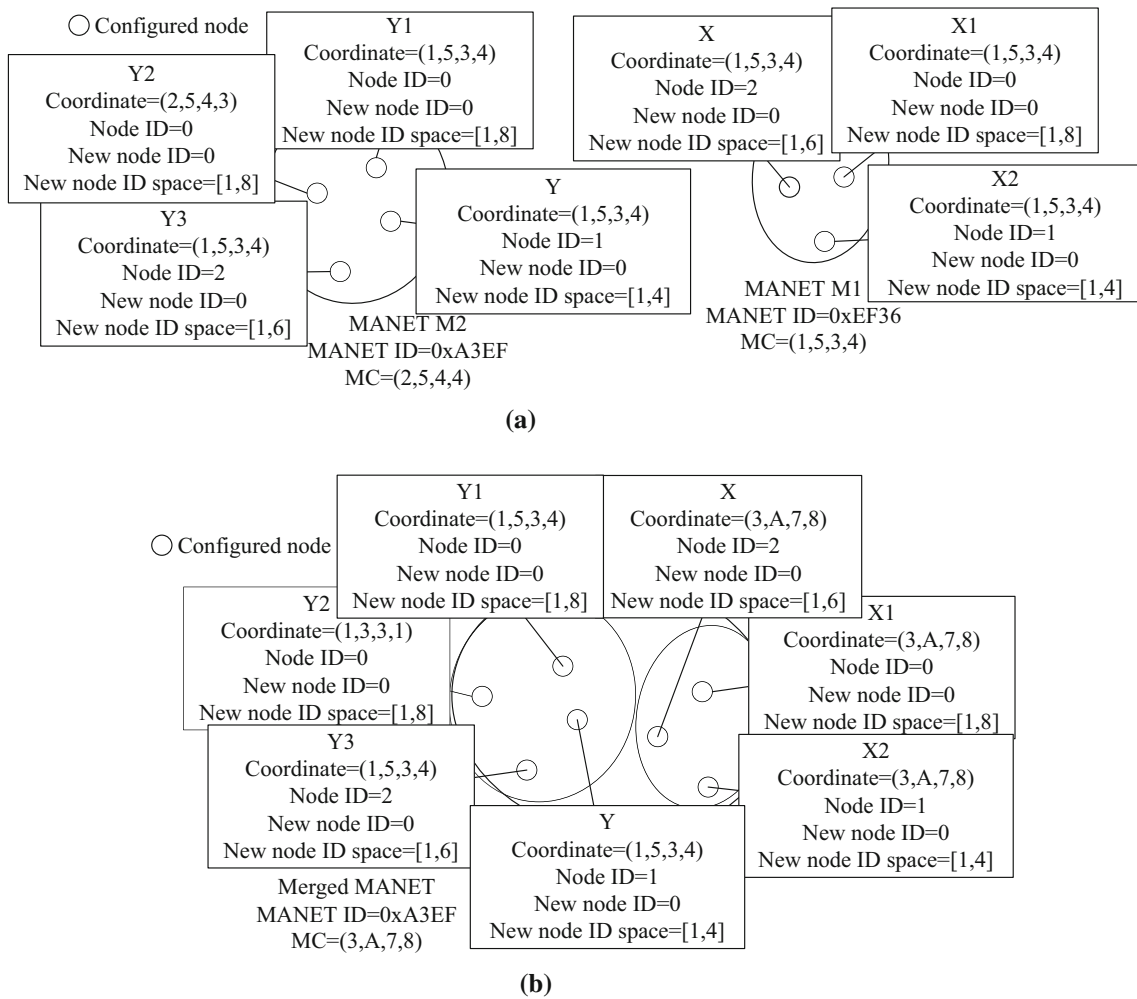


Fig. 8 Network merging. a Before merging, b After merging

delay of the proposed method are analyzed in the theoretical model, as shown in Fig. 9. In the theoretical model, in the initial state the nodes are evenly distributed around a network, the node density is d_{node} , the transmission radius of a node is r and the radius of a network is R .

6.1 Addressing analysis

In this scheme, a new node acquires a unique address via one of the three cases. In the first case (abbreviated as C1), a new node is configured with a unique address via the address initialization algorithm. Based on the address initialization algorithm, the addressing cost $C_{init-total}$ consists of the generation cost C_{gene} of coordinating nodes and the confirmation cost C_{conf} of address uniqueness, as shown in formula (5). Four coordinating nodes are included in a network, so C_{gene} is shown in formula (6) where n_{neig} is the number of a node's neighbors and c is the cost of transmitting a message between neighbors. Based on Fig. 4, C_{conf} includes the confirma-

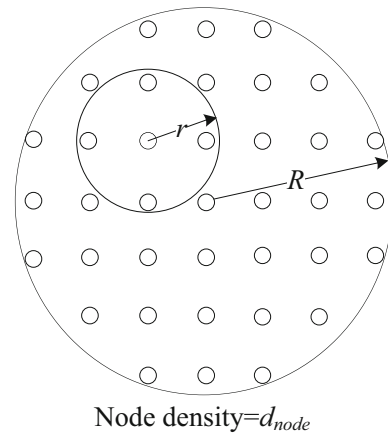


Fig. 9 Theoretical model

tion process of the final MANET ID and 4-tuple coordinates and the confirmation process of the node ID. The process of confirming the final MANET ID and 4-tuple coordinates

is achieved by flooding the MANET IDs generated by 4 coordinating nodes, so the cost is $4n \cdot c$ where n is the total number of nodes in a network. In the worst situation that none of new nodes has the unique 4-tuple coordinates, each new node broadcasts a *hello* to confirm the final node ID, so the cost is $n \cdot c$. Taken together, C_{conf} is equal to $5n \cdot c$, as shown in formula (7). Since $C_{init-total}$ is the total addressing cost for n nodes, the addressing cost $C_{addr-C1}$ for a new node is shown in formula (8). Similarly, the addressing delay $T_{init-total}$ includes the generation delay T_{gene} of coordinating nodes and the confirmation delay T_{conf} of address uniqueness, as shown in formula (9). The generation processes of four coordinating nodes tend to be performed in parallel, so T_{gene} is shown in formula (10) where t is the delay of transmitting a message between neighbors. Based on Fig. 4, T_{conf} is shown in formula (11). Since $T_{init-total}$ is the total addressing delay for n nodes, the addressing delay $T_{addr-C1}$ for a new node is shown in formula (12).

$$C_{init-total} = C_{gene} + C_{conf} \tag{5}$$

$$C_{gene} = 4n_{neig} \cdot c$$

where

$$n_{neig} = \pi r^2 d_{node} \tag{6}$$

$$C_{conf} = 5n \cdot c$$

where

$$n = \pi R^2 d_{node} \tag{7}$$

$$C_{addr-C1} = C_{init-total} / n \tag{8}$$

$$T_{init-total} = T_{gene} + T_{conf} \tag{9}$$

$$T_{gene} = n_{neig} \cdot t \tag{10}$$

$$T_{conf} = n_{neig} \cdot t + 2R \cdot t \tag{11}$$

$$T_{addr-C1} = T_{init-total} / n \tag{12}$$

In the second case (abbreviated as C2), a new node is configured with a unique address via the address configuration algorithm. Based on the addressing configuration algorithm, the addressing cost $C_{addr-C2}$ and delay $T_{addr-C2}$ for a new node are shown in formulae (13-14). In the third case (abbreviated as C3), a new node is configured with a unique address via the address configuration and address reclamation. In C3, the addressing cost $C_{addr-C3}$ includes the address reclamation cost $C_{addr-recl}$ and $C_{addr-C2}$, as shown in formula (15). Based on the address reclamation algorithm, $C_{addr-recl}$ is shown in formula (16) where n_{same} is the number of the nodes with the same MANET ID, coordinates and node ID. The addressing delay $T_{addr-C3}$ is made up of the address reclamation delay $T_{addr-recl}$ and $T_{addr-C2}$, as shown in formulae 16-17).

$$C_{addr-C2} = n_{neig} \cdot c + 2c \tag{13}$$

$$T_{addr-C2} = n_{neig} \cdot t + 2t \tag{14}$$

$$C_{addr-C3} = C_{addr-recl} + C_{addr-C2} \tag{15}$$

$$C_{addr-recl} = n \cdot c + n_{same} \cdot c \tag{16}$$

$$T_{addr-C3} = T_{addr-recl} + T_{addr-C2} \tag{17}$$

$$T_{addr-recl} = 4R \cdot t \tag{18}$$

6.2 Network partitioning analysis

If one of the following cases happens, the network partitioning algorithm is launched.

- (1) In the first case (abbreviated as C1), a node does not receive any one *hello*.
- (2) In the second case (abbreviated as C2), the *hello* messages received by a node all come from new nodes.
- (3) In the third case (abbreviated as C3), the MANET IDs in all the *hello* messages received by a node are different from the MANET ID of the node.

According to the network partitioning algorithm, in C1 a node only changes into a new node, so we only discuss C2 and C3. In C2, the nodes in a split network ensure the address uniqueness by updating the MANET ID, so the network partitioning cost $C_{splitting-C2}$ and delay $T_{splitting-C2}$ are shown in formulae (19-20) where $n_{splitting}$ is the number of the nodes and $R_{splitting}$ is the radius of the split network. In C3, the nodes in a split network ensure the address uniqueness by performing the address configuration algorithm, so the network partitioning cost $C_{splitting-C3}$ and delay $T_{splitting-C3}$ are shown in formulae (21-22).

$$C_{splitting-C2} = n_{splitting} \cdot c \tag{19}$$

$$T_{splitting-C2} = 2R_{splitting} \cdot t \tag{20}$$

$$C_{splitting-C3} = n_{splitting} \cdot C_{addr-C2} \tag{21}$$

$$T_{splitting-C3} = n_{splitting} \cdot T_{addr-C2} \tag{22}$$

6.3 Network merging analysis

The network merging algorithm is performed in the two cases. In the first case (abbreviated as C1), a node can receive both a *hello* with the same MANET ID and a *hello* with the different MANET ID. In C1, if two networks merge, the network merging cost $C_{merging-C1}$ and delay $T_{merging-C1}$ are shown in formulae (23-24). In the second case (abbreviated as C2), a node can receive a *hello* with the same address, so the network merging cost $C_{merging-C2}$ and delay $T_{merging-C2}$ are shown in formulae (25-26).

$$C_{merging-C1} = 2c + n \cdot c \tag{23}$$

$$T_{merging-C1} = 2t + 2R \cdot t \tag{24}$$

$$C_{merging-C2} = 2n \cdot c + C_{init-total} \tag{25}$$

Table 2 Parameters

Parameter	Value
Mobility model	Random waypoint
Pause time	10 s
Maximum speed	5 m/s
MAC protocol	IEEE 802.11
Transmission radius	100 m
Nodes	[50, 250]
Node Density	[100/km ² , 300/km ²]
Simulation time	1000 s

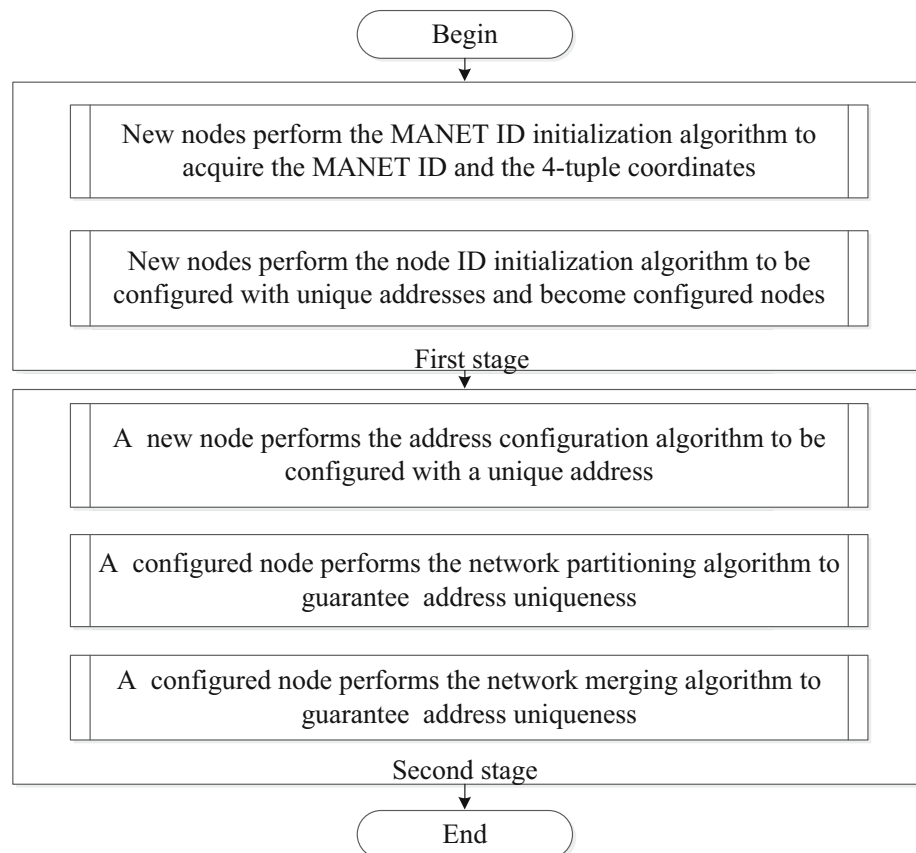
$$T_{merging-C2} = 4R \cdot t + T_{init-total} \quad (26)$$

7 Simulation

In this section, the effects of node population and node density on this proposal are evaluated by using *ns-2* and this proposal is also compared with the latest addressing solution. The simulation parameters are shown in Table 2. The random waypoint mobility model is the most common mobility model in one MANET [28, 29], so it is used to evaluate the proposal. In the random waypoint mobility model, a node

selects a random destination and speed, and then moves toward the selected destination at the chosen speed. Upon arrival, the node pauses for a specified period before selecting a new destination and speed, and the specified period is called the pause time. The pause time is usually set to 10 s and the maximum speed is 5 m/s [30]. The link protocol in one MANET is IEEE 802.11 where the transmission radius of a node is 100 m [31]. In the initial state, the number of nodes in one MANET ranges from 50 to 250 so that the effect of node population on the addressing performance can be evaluated. The node density ranges from 100/km² to 300/km² so that the effect of node density on the addressing performance can be assessed. In the evaluation process, the addressing cost is calculated by the total number of messages used for achieving the addressing divided by the total number of nodes. The network partitioning/merging cost is measured by the total number of messages used for guaranteeing the address uniqueness of each node in a split/merged network.

The implementation of the proposed method includes two stages, as shown in Fig. 10. The first stage is the address initialization. The second stage is the address maintenance including the network partitioning and merging, and works based on the first stage. In the first stage, the nodes perform the MANET ID initialization algorithm to elect one EC/SC/WC/NC and acquire the MANET ID and the 4-tuple

Fig. 10 Implementation Flow

coordinates. After one node obtains the MANET ID and the 4-tuple coordinates, it launches the node ID initialization algorithm to acquire one unique node ID. Via the first stage, a new node is configured with a unique address and become a configured node. Then, the second stage begins. In this stage, a new node is configured with a unique address by performing the address configuration algorithm. Also, the network partitioning and merging begin to happen due to node mobility. If a configured node detects that the network partitioning occurs, it starts the network partitioning algorithm to guarantee the address uniqueness. Similarly, if a configured node detects that the network merging happens, it launches the network merging algorithm to ensure the address uniqueness.

7.1 Effects of node population

The effects of node population on the proposal are evaluated, as shown in Fig. 11. In Fig. 11a, b, the addressing cost in C1 is mainly caused by the coordinating node election process. Since each node is involved in the coordinating node election, the election cost tends to be proportional to n . Consequently, the addressing cost is hardly affected by n and tends to be stable because n nodes achieve the addressing via one address initialization process. The addressing initialization delay is only related to the network diameter, so the addressing delay decreases with n . As n nodes achieve the addressing via one address initialization process, the addressing cost and delay in C1 are lower than the ones in C2 and C3. In C2, a new node acquires an address from a neighbor, so the cost and delay are hardly affected by n and tend to be stable. In C3, a node is configured with an address via performing both the address reclamation and address configuration. Since each node is involved in the address reclamation, the address reclamation cost and delay grow with n . Consequently, the addressing cost and delay in C3 also increase with n and are higher than the ones in C1 and C2. From Fig. 11a, b, we can find that the addressing algorithms in C1 and C2 are scalable. As shown in Fig. 11c, d, in C2 each node in a split network ensures the address uniqueness by updating its address, so the network partitioning cost and delay increase with $n_{splitting}$. In C3, after a node leaves the current network, it joins a new network and is configured with an address via the address configuration algorithm. Therefore, the network partitioning cost and delay grow with $n_{splitting}$ and are greater than the ones in C2 because the address reclamation is not involved in C2. The analysis assumes that in C3 the nodes in a split network perform the addressing one by one, but in the simulation some nodes execute the addressing in parallel. Therefore, the delay in the simulation is smaller than the one in the analysis. As shown in Fig. 11e, f, in C1 each node in a merged network guarantees the address uniqueness by updating the address, so the network merging cost and delay increase with n . In C2, the address initialization algorithm is performed to configure

each node with a unique address, so the network merging cost and delay also grow with n and are greater than the ones in C1. From Fig. 11c–f, we can find that the cost and delay for address update are smaller than the ones for address reconfiguration.

7.2 Effects of node density

The effects of node density on the proposal are evaluated, as shown in Fig. 12. In Fig. 12a, b, the addressing cost in C1 is mainly caused by the coordinating node election process. Since each node is involved in the election process, the cost and delay are hardly affected by d_{node} and tend to be stable. Because the nodes achieve the addressing via one address initialization process, the addressing cost and delay also tend to be stable and are lower than the ones in C2 and C3. In C2, a new node acquires an address from a neighbor, so the cost and delay grow with d_{node} . In C3, a node is configured with an address via performing both the address reclamation and address configuration. Since each node is involved in the address reclamation, the address reclamation cost and delay tend to be stable. Consequently, the addressing cost and delay in C3 tend to be stable and are higher than the ones in C1 and C2. From Fig. 12a, b, we can find that the addressing algorithms in C1 and C2 can work well in one MANET with relatively high node density. As shown in Fig. 12c, d, in C2 each node in a split network ensures the address uniqueness by updating its address, so the network partitioning cost is hardly affected by d_{node} and tends to be stable. The growth in d_{node} reduces the network diameter, so the network partitioning delay decreases. In C3, a node joins a new network and is configured with an address via the address configuration algorithm. Since the growth in d_{node} increases the address configuration cost and delay, the network partitioning cost and delay in C3 grow with d_{node} and are greater than the ones in C2. The analysis assumes that in C3 the nodes in a split network perform the addressing one by one, but in the simulation some nodes execute the addressing in parallel. Therefore, the delay in the simulation is smaller than the one in the analysis. As shown in Fig. 12e, f, in C1 each node in a merged network ensures the address uniqueness by updating the address, so the network merging cost tends to be stable. Since the growth in d_{node} reduces the network diameter, the network merging delay decreases. In C2, the address initialization algorithm is performed to configure each node with a unique address, so the network merging cost tends to be stable. Since the growth in d_{node} reduces the network diameter and increases the number of neighbors, the network merging delay first slightly decreases with d_{node} and then slightly increases with d_{node} . From Fig. 12c–f, we can find that the cost and delay for address update are smaller than the ones for address configuration, and the network partitioning in C2

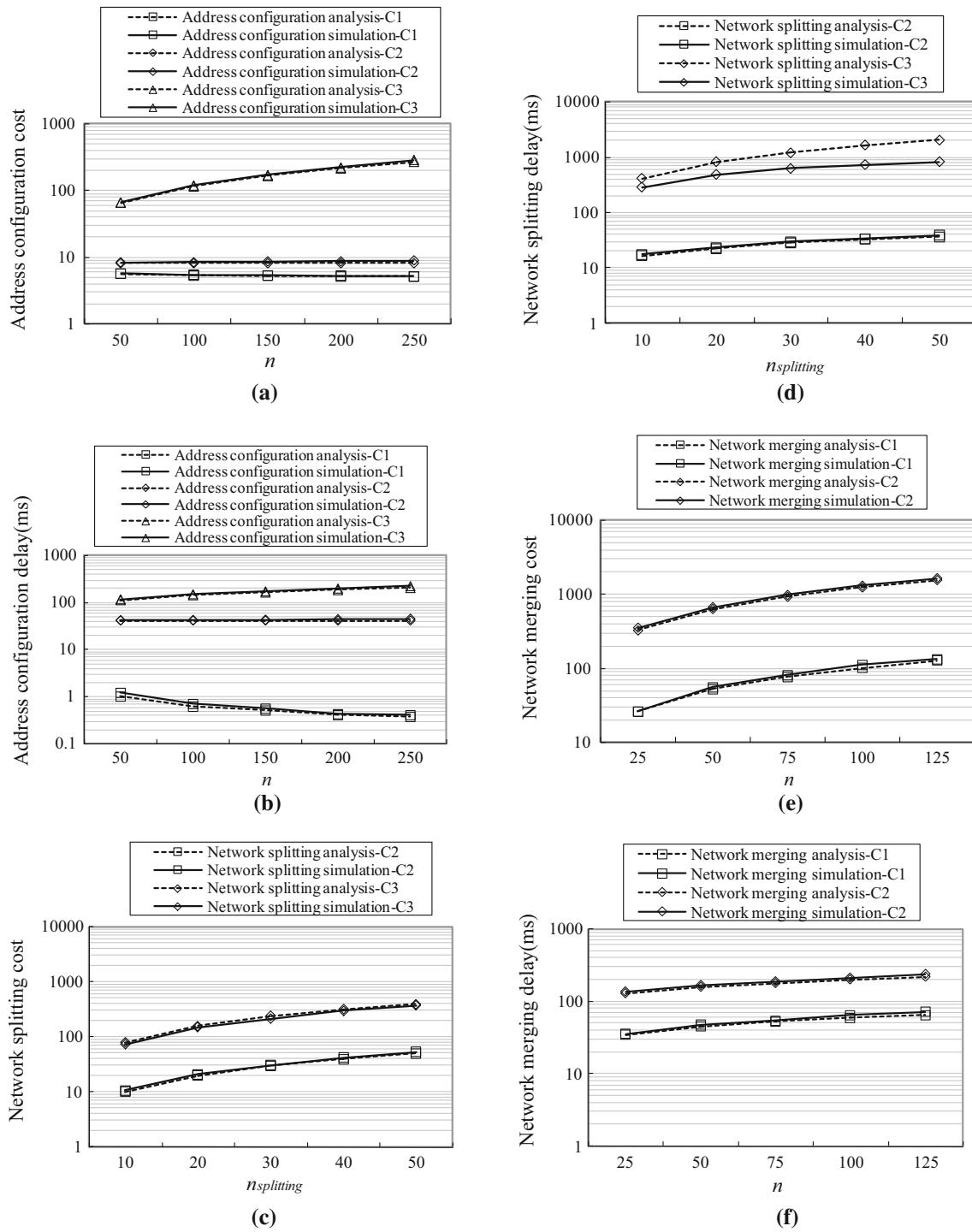


Fig. 11 Effects of node population. **a** Addressing cost based on n , **b** addressing delay based on n , **c** network partitioning cost based on $n_{splitting}$, **d** network partitioning delay based on $n_{splitting}$, **e** network merging cost based on n , **f** network merging delay based on n

and the network merging in C1 and C2 can work well in one MANET with relatively high node density.

7.3 Comparison

In order to evaluate the proposed addressing solution, FAACP [22] is selected to compare with our solution due to the following reasons:

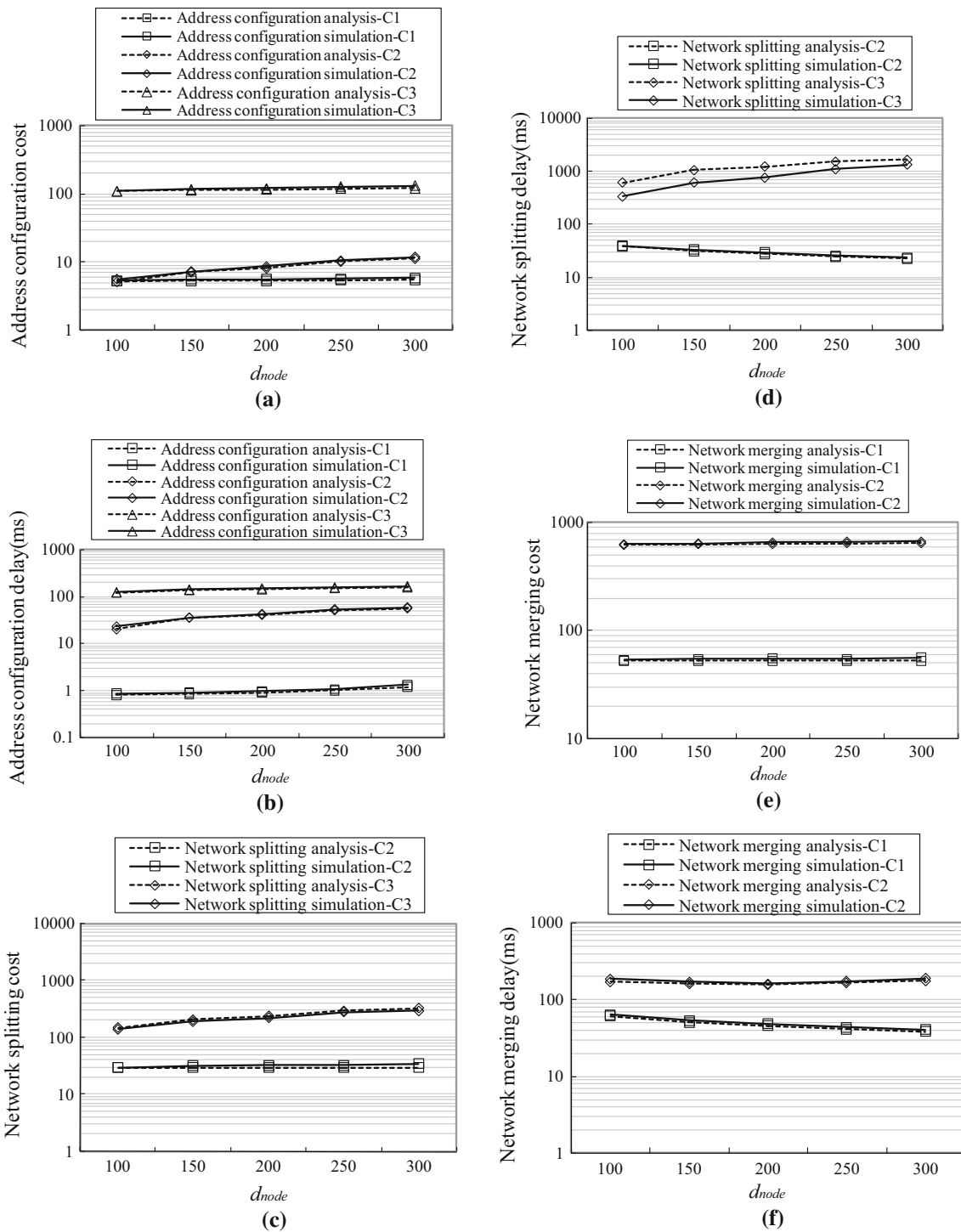


Fig. 12 Effects of node density. **a** Addressing cost based on d_{node} , **b** addressing delay based on d_{node} , **c** network partitioning cost based on d_{node} , **d** network partitioning delay based on d_{node} , **e** network merging cost based on d_{node} , **f** network merging delay based on d_{node}

- (1) FAACP is the latest addressing solution.
- (2) FAACP addresses the networking partitioning and merging issues.

The comparison results are shown in Figs. 13, 14. In the comparison, a new node might acquire a unique address via one of the three cases. The first case is the address initialization, the second case is the address configuration and the third case is the address reclamation and address configu-

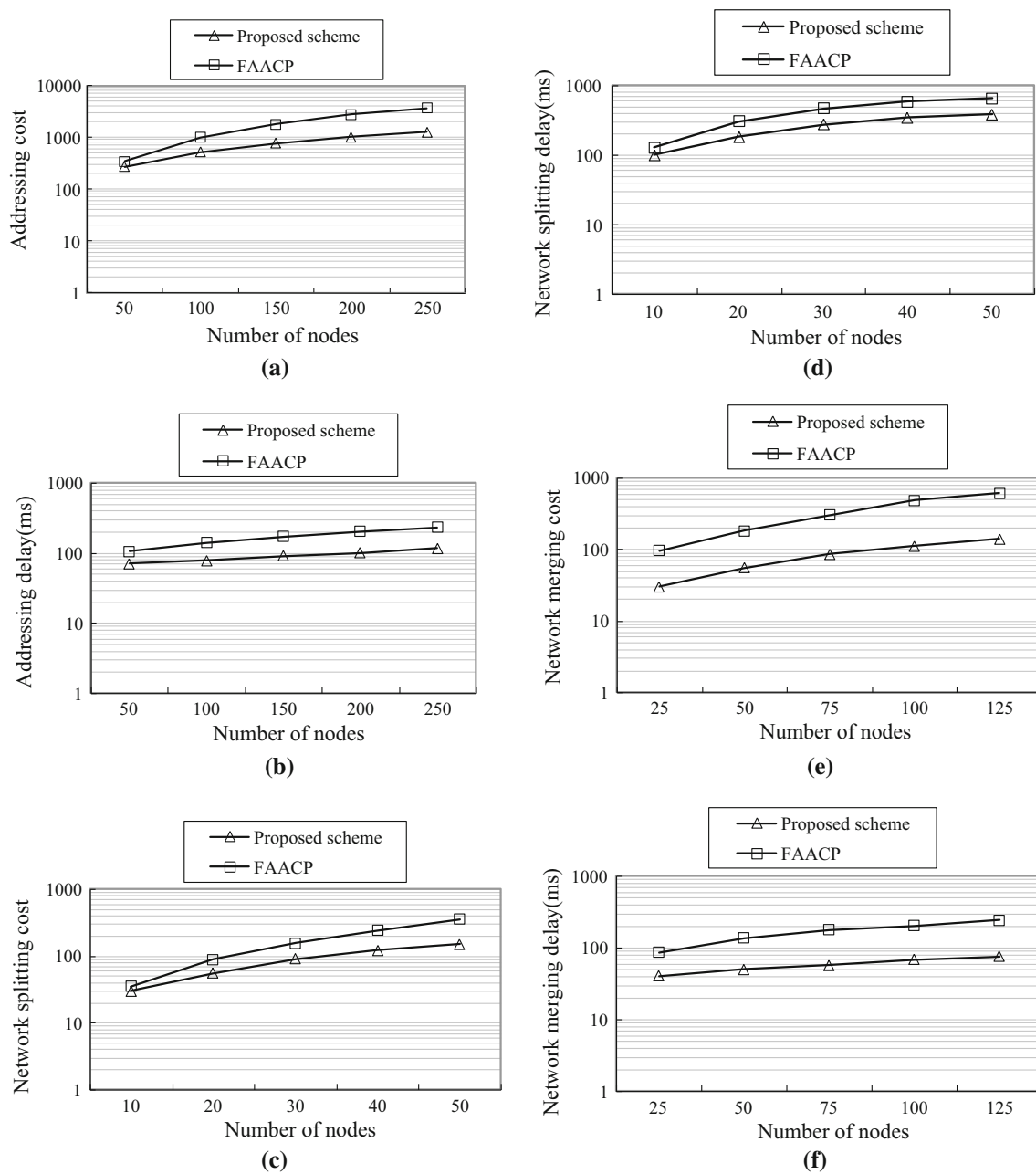


Fig. 13 Comparisons based on node population. **a** Addressing cost comparison, **b** addressing delay comparison, **c** network partitioning cost comparison, **d** network partitioning delay comparison, **e** network merging cost comparison, **f** network merging delay comparison

ration. The addressing cost and latency are evaluated by the average addressing cost and latency, namely the total addressing cost and latency for configuring all the nodes divided by the total number of the nodes. Similarly, the network partitioning might be performed via one of the three cases, as shown in Sect. 6.2. The network partitioning cost and latency are assessed by the average network partitioning cost and latency, namely the total network partitioning cost and latency divided by the number of performing the network partitioning. The network merging might be performed via one

of the two cases, as shown in Sect. 6.3. The network merging cost and latency are evaluated by the average network merging cost and latency, namely the total network merging cost and latency divided by the number of performing the network merging.

As shown in Fig. 13a, b, the addressing cost and delay in FAACP and this proposal grow with node population, and this scheme has lower addressing cost and delay than FAACP. The main reason is that in FAACP a node guarantees the address uniqueness from a remote node. In this scheme, most

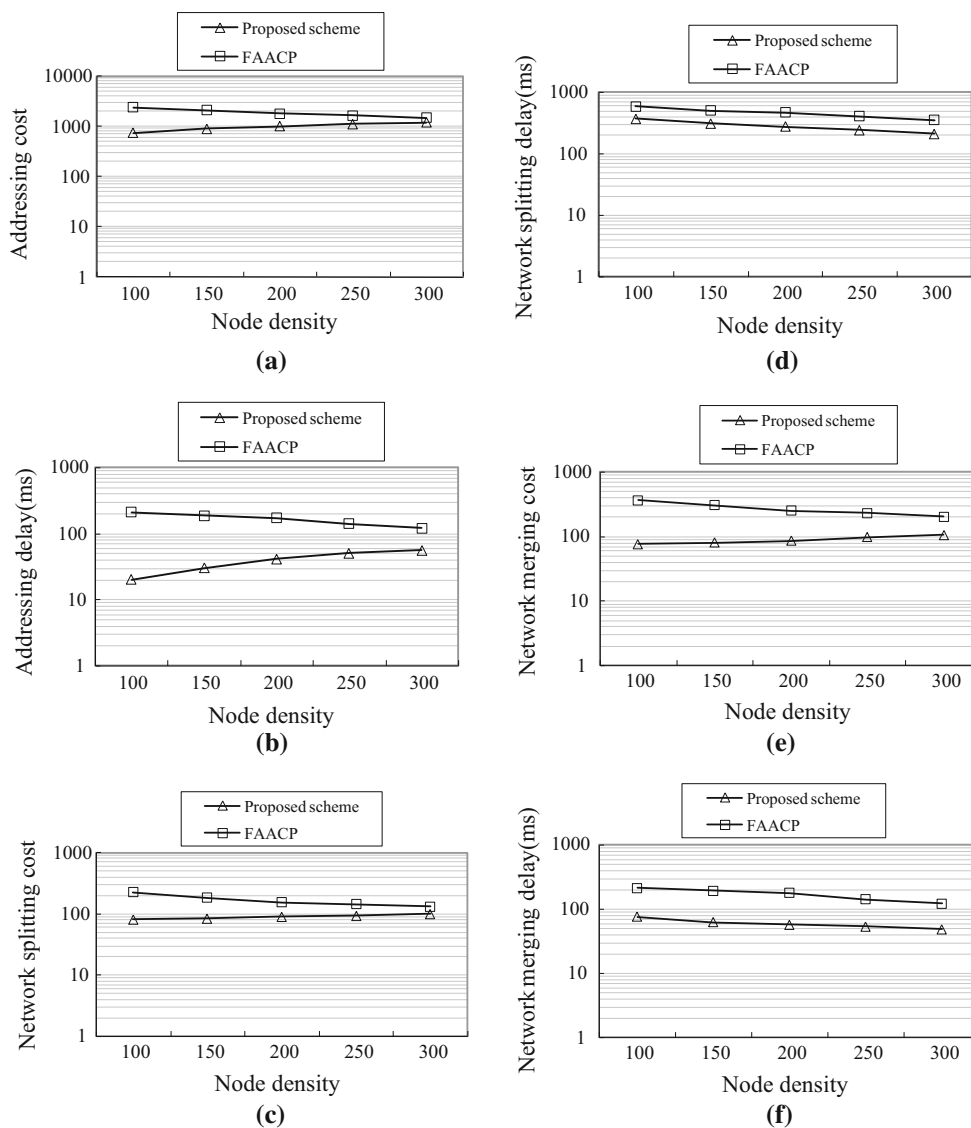


Fig. 14 Comparisons based on node density. **a** Addressing cost comparison, **b** addressing delay comparison, **c** network partitioning cost comparison, **d** network partitioning delay comparison, **e** network merging cost comparison, **f** network merging delay comparison

nodes achieve the addressing via one address initialization process or one address configuration process, so the addressing cost and latency are lower. As shown in Fig. 13c, d, the network partitioning cost and delay in FAACP and this proposal grow with node population, and this scheme has lower network partitioning cost and delay than FAACP. In FAACP, the nodes in a split network perform the address reconfiguration to acquire the new addresses. In this scheme, the nodes in a split network update their addresses to ensure the address uniqueness. As shown in Fig. 13e, f, this scheme has lower network merging cost and delay than FAACP. In FAACP, the nodes with duplicate addresses in a merged network perform the address reconfiguration to acquire the unique addresses. In this scheme, the nodes in a merged network ensure the

address uniqueness by updating their addresses, so the cost and latency are lower.

As shown in Fig. 14a, b, the addressing cost and delay in this proposal grow with node density whereas the addressing cost and delay in FAACP decrease with node density. However, this scheme has lower addressing cost and delay than FAACP. In FAACP, a node verifies the address uniqueness from a remote node. With the growth in node density, the network diameter reduces, so the distance from a source node to a remote node decreases. Consequently, the addressing cost and delay in FAACP decrease with node density. In this scheme, most nodes acquire the addresses via one address initialization process or one address configuration process, so the addressing cost and latency slightly increase with node density. As shown in Fig. 14c, d, the network partitioning cost

and delay in FAACP decrease with node density because the nodes in a split network perform the address reconfiguration to acquire the addresses. In this proposal, the network partitioning cost increases with node density whereas the network partitioning delay decreases with node density. The main reason is that the nodes in a split network update their addresses to ensure the address uniqueness. As shown in Fig. 14e, f, the network merging cost and delay in FAACP decrease with node density because the nodes in a merged network perform the address reconfiguration to acquire their addresses. In this proposal, the network merging cost increases with node density whereas the network merging delay decreases with node density. The main reason is that the nodes in a merged network update their addresses to ensure the address uniqueness. Since the growth in node density reduces the network diameter, the address update cost and latency decrease with node density. Consequently, the network partitioning and merging costs and delays are lower.

8 Conclusions

This paper presents a coordinate-based addressing solution for one MANET and aims to reduce the addressing cost and latency. The data results show that this work reduces the addressing cost and latency because the nodes can be configured with unique addresses via one addressing process. Moreover, this work also reduces the network partitioning and merging costs and delays because it makes use of the address update instead of the address reconfiguration to guarantee the address uniqueness during the network partitioning and merging.

The proposed solution can be potentially applied in real-life applications such as fieldwork and disaster rescue scenarios where no infrastructures are available. In these scenarios, nodes can be configured with unique addresses by performing the proposed solution and the address uniqueness can be ensured even if the network partitioning or merging happens.

In this work, if there is a lack of address space, a node has to acquire an address via address reclamation. In this case, the addressing cost and latency are considerable. In our future work, we plan to make use of nodes' positioning capabilities to exploit their potential for assisting address reclamation.

Acknowledgements This work is supported by CERNET Innovation Project (NGII20170106) and "333 Project" foundation (BRA2016438).

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

- Omar, M., Hedjaz, S., Rebouh, S., Aouchar, K., Abbache, B., & Tari, A. (2015). On-demand source routing with reduced packets protocol in mobile ad-hoc networks. *AEU-International Journal of Electronics and Communications*, 69(10), 1429–1436.
- Wang, X., Dou, Z., Wang, D., & Sun, Q. (2018). Mobility management for 6lowpan wsn. *Computer Networks*, 131, 110–128.
- Thoppian, M. R., & Prakash, R. (2006). A distributed protocol for dynamic address assignment in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(1), 4–19.
- Wang, X., Le, D., & Yao, Y. (2015). A cross-layer mobility handover scheme for IPv6-based vehicular networks. *International Journal of Electronics and Communications*, 69(10), 1514–1524.
- Al-Mistarihi, M. F., Al-Shurman, M., & Qudaimat, A. (2011). Tree based dynamic address autoconfiguration in mobile ad hoc networks. *Computer Networks*, 55(8), 1894–1908.
- Kim, N., Ahn, S., & Lee, Y. (2007). AROD: An address autoconfiguration with address reservation and optimistic duplicated address detection for mobile ad hoc networks. *Computer Communications*, 30(8), 1913–1925.
- Wang, X., Cheng, H., & Yao, Y. (2016). Addressing-based routing optimization for 6lowpan wsn in vehicular scenario. *IEEE Sensors Journal*, 16(10), 3939–3947.
- Wang, X., Le, D., & Cheng, H. (2018). Hierarchical addressing scheme for 6LoWPAN WSN. *Wireless Networks*, 24(4), 1119–1137.
- Wang, D., Wang, X., & Dou, Z. (2017). Addressing for 6lowpan based on multi-hop clusters. *International Journal of Internet Protocol Technology*, 10(4), 243–259.
- Wang, X., Cheng, H., & Le, D. (2018). A routing scheme for connecting delay-sensitive urban vehicular networks to the ipv6-based internet. *Telecommunication Systems*, 1–16.
- Wang, X., Le, D., Cheng, H., & Yao, Y. (2015). Location-based address configuration for 6lowpan wireless sensor networks. *Wireless Networks*, 21(6), 2019–2033.
- Ghosh, U., & Datta, R. (2011). A secure dynamic IP configuration scheme for mobile ad hoc networks. *Ad Hoc Networks*, 9(7), 1327–1342.
- Wang, X., & Qian, H. (2014). A distributed address configuration scheme for a MANET. *Journal of Network and Systems Management*, 22(4), 559–582.
- Wang, X., Cheng, H., & Yao, Y. (2016). Addressing with an improved DAD for 6LoWPAN. *IEEE Communications Letters*, 20(1), 73–76.
- Wang, X., Chen, H., & Le, D. (2016). A novel ipv6 address configuration for a 6lowpan-based wban. *Journal of Network & Computer Applications*, 61, 33–45.
- Wang, X., Sun, Q., & Wang, D. (2016). Addressing for 6lowpan wban. *Computer Methods and Programs in Biomedicine*, 137, 353–363.
- Wang, X., Gao, F., & Wu, H. (2017). Design and simulation of address management protocol based on DHCPv6 for large-scale wireless sensor networks. In *International Conference on Computer, Information and Telecommunication Systems* (pp. 68–72).
- Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., & Carney, M. (2003). Dynamic host configuration protocol for IPv6 (DHCPv6). RFC 3315.
- Ozturk, Y., & Nagarnaik, V. (2011). A scalable distributed dynamic address allocation protocol for ad-hoc networks. *Wireless Networks*, 17(2), 357–370.
- Gammar, S. M., Amine, E., & Kamoun, F. (2010). Distributed address auto configuration protocol for Manet networks. *Telecommunication Systems*, 44(1–2), 39–48.

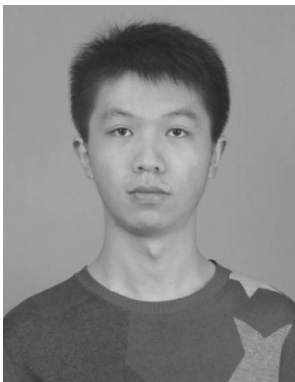
21. Wang, X., Le, D., & Cheng, H. (2016). Distributed ipv6 address configuration for hybrid manets. *Ad Hoc & Sensor Wireless Networks*, 33(1), 153–171.
22. Reshmi, T. R., & Murugan, K. (2015). Filter-based address auto-configuration protocol (FAACP) for duplicate address detection and recovery in MANETs. *Computing*, 97(3), 309–331.
23. Weniger, K. (2005). PACMAN: Passive autoconfiguration for mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(3), 507–519.
24. Patwari, N., Ash, J. N., Kyperountas, S., Hero, A. O., Moses, R. L., & Correal, N. S. (2005). Locating the nodes: cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE*, 22(4), 54–69.
25. Wang, X., & Zhu, X. (2018). Anycast-based content-centric MANET. *IEEE Systems Journal*, 12(2), 1679–1687.
26. Bulusu, N., Heidemann, J., & Estrin, D. (2000). GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5), 28–34.
27. Medepalli, K., & Tobagi, F. A. (2006). Towards performance modeling of IEEE 802.11 based wireless networks: A unified framework and its applications. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications*. Proceedings (pp. 1–12). IEEE.
28. Kim, S., Lee, J., & Yeom, I. (2008). Modeling and performance analysis of address allocation schemes for mobile ad hoc networks. *IEEE Transactions on Vehicular Technology*, 57(1), 490–501.
29. Camp, T., Boleng, J., & Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5), 483–502.
30. Suresh, H. N., Varaprasad, G., & Jayanthi, G. (2014). Notice of violation of IEEE publication principles designing energy routing protocol with power consumption optimization in MANET. *IEEE Transactions on Emerging topics in Computing*, 2(2), 192–197.
31. IEEE 802.11 Working Group. (2016). Part11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. ANSI/IEEE Std. 802.11



Xiaonan Wang is a full professor at Changshu Institute of Technology in China. Her research interests include Ad hoc networks, information-centric networks, etc.



Yanli Li is currently working at Changshu Institute of Technology. Her research interests include the sensor networks.



Zhengxiong Dou is currently working at Changshu Institute of Technology. His research interests include the sensor networks and ad hoc networks.