WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2011; **11**:1341–1357
Published online 1 March 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/wcm.931

RESEARCH ARTICLE

Local replacement for route recovery on a collaborative mobile *ad hoc* network

Jenn-Wei Lin* and Yi-Ting Chen

Department of Computer Science and Information Engineering, Fu Jen Catholic University, Taipei, Taiwan

ABSTRACT

This paper presents an efficient route recovery approach for a collaborative mobile *ad hoc* network (MANET). In the collaborative MANET, the movement of a node can be controlled by other nodes. Based on the property of the controllable movement, each active node (each node on a route) can be protected by its neighboring nodes. When the movement of an active node causes a link breakage on a route, its neighboring nodes compete with each other to move to the original location of the moving active node. Using the movements of neighboring nodes, the proposed approach can achieve route recovery without pre-establishing or dynamically finding backup routes. The proposed approach also has much less recovery overhead than previous approaches because of the local recovery. Finally, we perform extensive simulations to compare the proposed approach with previous approaches. The simulation results show that the proposed approach has better recovery performance under dense and sparse MANET architectures. Copyright © 2010 John Wiley & Sons, Ltd.

KEYWORDS
route recovery; mobile *ad hoc* network (MANET); controllable movement; link breakage; local recovery

***Correspondence**
Jenn-Wei Lin, Department of Computer Science and Information Engineering, Fu Jen Catholic University, Taipei, Taiwan.
E-mail: jwlin@csie.fju.edu.tw

1. INTRODUCTION

Mobile *ad hoc* network (MANET) has emerged as one important research area in the field of wireless networks. In the MANET, there is no infrastructure. Each node has a limited radio communication range, operates on a constrained battery power, and moves randomly. Due to these characteristics, it is a big challenge to design a routing protocol for MANETs. The Internet Engineering Task Force (IETF) has formed a MANET working group [1] to standardize routing protocols for MANETs.

In this paper, we propose an efficient route recovery approach for a collaborative MANET, e.g., *ad hoc* robot network [2–4]. The *ad hoc* robot network uses robots to replace human to perform dangerous tasks, such as cleanup of toxic waste, nuclear power processing, planetary exploration, military or civilian search and rescue, etc. Such dangerous tasks require dispersing a number of robots into a bounded region to form an *ad hoc* robot network. In the *ad hoc* robot network, each robot node is equipped with a GPS module to be aware of its location. The robot nodes also collaborate with each other to accomplish a common task. With the collaborative way, the movement of a robot node can be controlled by other robot nodes. Due to holding the controllable movement property, the proposed approach makes each active node (the node on a route) in a collaborative MANET be protected by its neighboring nodes. This can be also said that each active node uses the nodes within its communication range as its backup nodes. When an active node moves to break its route, its neighboring nodes move to compete with each other for replacing the moving active node. With the node replacement, the broken route can be locally repaired to continuously transmit packets.

The proposed approach modifies the well-known MANET routing protocol: *ad hoc* on-demand distance vector (AODV) [5] to perform route recovery based on node movements. There have been a lot of research efforts on improving the route recovery of the AODV protocol [6–9]. The literature can be classified into two categories: pre-establishment method and on-demand discovery method. The pre-establishment method establishes one or more backup routes for each route in advance. When a route is broken due to a node movement, this route is repaired using one of the pre-established backup routes. For the on-demand discovery method, the backup route is not pre-established. Instead, the backup route is dynamically found when detecting a link breakage. Compared to previous approaches [6–9], the proposed approach is not required to pre-establish or dynamically find any backup routes.

In the proposed approach, the backup nodes of an active node are its neighboring nodes. These backup nodes are pre-specified before the occurrence of a link breakage. However, due to node mobility, the protecting relationship between an active node and each of its backup nodes will be easily terminated. For correctly repairing a broken route, the backup node maintenance is also concerned in the proposed approach. In contrast, the backup route maintenance is not discussed in previous approaches [6–9]. To examine the effectiveness of the proposed approach in recovery cost reduction and recovery capability enhancement, we perform extensive simulations to compare the proposed approach with previous approaches under the dense and sparse MANET architectures.

The remainder of this paper is organized as follows. Section 2 introduces the background of this paper. Section 3 presents our recovery approach. Section 4 makes the comparison between the proposed approach and previous approaches. Finally, conclusions are made in Section 5.

2. BACKGROUND

This section describes the background materials of this paper. First, we briefly introduce existing routing protocols for MANETs. Then, we focus on reviewing previous work on the route recovery based on AODV protocol.

2.1. Routing protocols

In the IETF MANET working group [1], routing protocols can be classified into two categories: proactive (table-driven) and reactive (on-demand). Proactive protocols attempt to maintain routes from a node to all other nodes. Examples of proactive protocols include destination sequenced distance vector (DSDV) [10], cluster switch gateway routing (CGSR) [11], etc. Unlike the proactive protocols, the routes in the reactive protocols are established based on demand. When two nodes would like to transmit packets, the reactive protocols discover a route between these two nodes. The existing reactive protocols include dynamic source routing (DSR) [12], *ad hoc* on-demand distance vector routing (AODV) [5], etc.

With the on-demand property, the DSR and ADOV protocols are extensively used in MANETs. The feature of the DSR protocol is the use of source routing. When a source node needs to send a packet to a destination node, the source node puts a *source route* in the header of the packet. The source route keeps a sequence of hops for indicating packets how to be transmitted to the destination node. Normally, the source node can obtain a suitable source route from its route cache. If not, the source node will flood a *Route Request (*RREQ*)* message over MANET. The source node first broadcasts a RREQ message to its neighboring nodes. Each neighboring node further broadcasts the message to it neighbors, and so on. Upon receiving the RREQ message, if the node knows a route to the destination node in its route cache or itself is the destination node, it will send a *Route Reply (*RREP*)* message to the source node. The RREP message contains an accumulated route learned from the RREQ message. The source node then puts the route in its route cache.

The AODV protocol is also based on the above route discovery mechanism to find a route. However, the AODV protocol adopts a different way to maintain routing information. It uses the routing table, one entry per destination node. The RREP message is sent to the source node along the reverse path of the RREQ message. As the RREP message passes through intermediate nodes, these nodes update their routing tables for assisting the establishment of the route. In the AODV protocol, it also proposes a hello mechanism [13] to detect route failure. Each node in the MANET periodically broadcasts a *hello* message with TTL = 1. If the movement of an active node causes a link breakage on its route, the neighboring nodes can be aware of this link breakage since they do not receive a hello message from the active node for a period of time. Based on the periodical transmission of the hello message, the route failure can be detected.

2.2. Related work

The DSR and AODV protocols already have the route recovery capability using the *Route Error* (*RERR*) message to dynamically find backup routes [5]. Here, the backup route is found based on the above mentioned route discovery mechanism. This may introduce significant flooding overhead. To avoid incurring flooding overhead, the DSR protocol allows a node to keep multiple routes to a destination node in its route cache. These multiple routes are learned from the received RREQ messages. If a node detects a link breakage on its route, it can find an alternative route from the route cache. However, the cached routes are not updated as network topology changes. Some cached routes are stale. As indicated in Reference [14], due to using stale routes, the DSR protocol is worse than the AODV protocol in packet delivery. The AODV protocol has better packet delivery ratio. Therefore, several approaches have been proposed to improve the route recovery mechanism of AODV protocol.

In Reference [6], Lee and Gerla provide multiple alternative routes for each source–destination node pair. The proposed approach is called AODV-BR (AODV with multiple backup routes) which modifies the propagation process of the RREP message. When the RREP message is sent back from the destination node to the source node, this message can be overheard by the neighbors of each active node. Taking advantage of this property, the neighbors can also build respective routes to the destination node. After the RREP message is finally propagated to the source node, one primary route and multiple alternative routes are built. When a link breakage is detected, AODV-BR also sends a RERR message for discovering a new route. Before discovering the new route, AODV-BR asks the active node detecting

DOI: 10.1002/wcm

the link breakage to continuously send packets to its neighbors. Then, the neighbors use their respective alternative routes to forward packets to the destination node. Due to using multiple alternative routes, the same packet will be sent with several copies to the destination node. In addition, the alternative routes in Reference [6] are also not maintained as the active or neighboring nodes moves. The link breakage may easily occur in alternative routes. The AODV-BR cannot guarantee that packets can be correctly delivered to the destination node using alternative routes.

In Reference [7], authors propose a new multi-path routing protocol called AODV-CF (AODV with control flooding) to establish alternative routes from each active node to the destination node. The AODV-CF is similar to AODV-BR [6]. However, unlike AODV-BR, the alternative routes are not overlapped with the primary route. Therefore, the reliability of each alternative route in AODV-CF is better than AODV-BR.

In Reference [8], the route recovery is performed from a different viewpoint. The proposed approach is called AODV-RBA (AODV with route breakage avoidance). The main idea of this approach is to avoid route breakage. Each active node periodically measures the danger situation of a link breakage. If a danger situation is higher than a predefined threshold, the active node sends a RERR message to the source node to actively discover a new route to the destination node.

In Reference [9], a route update message is used to substitute the hello message of the AODV protocol. The proposed approach is called the AODV-LU (AODV with local update). Each active node and its neighboring nodes periodically broadcast a route update message between them for establishing multiple routes in advance. When detecting a link breakage on a route, an alternative route is verified whether it can really transmit packets to the destination node. If all existing alternative routes cannot be used, a RERR message is finally sent back to the source node to dynamically find a new route to the destination node.

3. PROPOSED APPROACH

In this section, we propose an efficient route recovery approach for the collaborative MANET. Unlike previous approaches, the backup routes are not used to perform route recovery. If no backup route exists in the MANET, the broken route is possible to be repaired using node movements.

3.1. Basic idea

The proposed approach is based on the following characteristics in a collaborative MANET to perform route recovery.

- In a MANET, each node periodically sends a hello message with TTL = 1 (see section 2.1) for detecting route recovery. When receiving a hello message, if the receiving node does not have a route to the sending node, it will establish a route in its routing table. The routing table also contains the neighbor information. Each node can know its neighboring nodes from its routing table.
- For each node in a MANET, it can know its location using global position system (GPS). The GPS has been extensively used in MANET literature to achieve the location awareness [15,16].
- In a general MANET, each node randomly moves by itself. However, in the collaborative MANET (e.g., *ad hoc* robot network), the movement of a node can be controlled by other nodes.

Based on the above characteristics, the proposed approach can work as follows. While establishing a route for a source–destination node pair, if a node receives the RREP message for being an active node on a route, its neighboring nodes will be simultaneously specified as backup nodes to protect the active node. For example, in Figure 1(a), if node C receives the RREP message for being an active node, nodes B, D, C_1 and C_2 will be designated as the backup nodes of node C. Later, if node C moves to break its located route, the backup nodes (neighboring nodes) of node C will move to compete with each other for replacing node C. As shown in Figure 1(b), the broken route can be recovered by moving node C_1 to the original location of node C.

To implement the proposed approach, the following three problems are required to be concerned.

- How to establish the protecting relationship between an active node and its neighboring nodes.
- How to maintain the protecting relationship.
- How to direct the movements of the neighboring nodes for route recovery.

In the first problem, it is to make each active node be protected by its neighboring nodes. Each neighboring node acts the role of a backup node. However, due to node mobility, the protecting relationship between an active node and each of its backup nodes is easily terminated. If a backup node moves out the communication range of its corresponding active node, it will be removed from the backup node set. Conversely, if a node moves in the communication range of an active node, the node should be added into the backup node set. Therefore, the second problem is to solve how to keep up-to-date backup nodes for an active node. As for the third problem, it concerns how to use node movements to perform local recovery.

3.2. Detailed operations

In this subsection, we describe the detailed operations of the proposed approach based on the above three problem.

3.2.1. Protection establishment and maintenance.

In the proposed approach, each active node is protected by its neighboring nodes. If an active node can know its

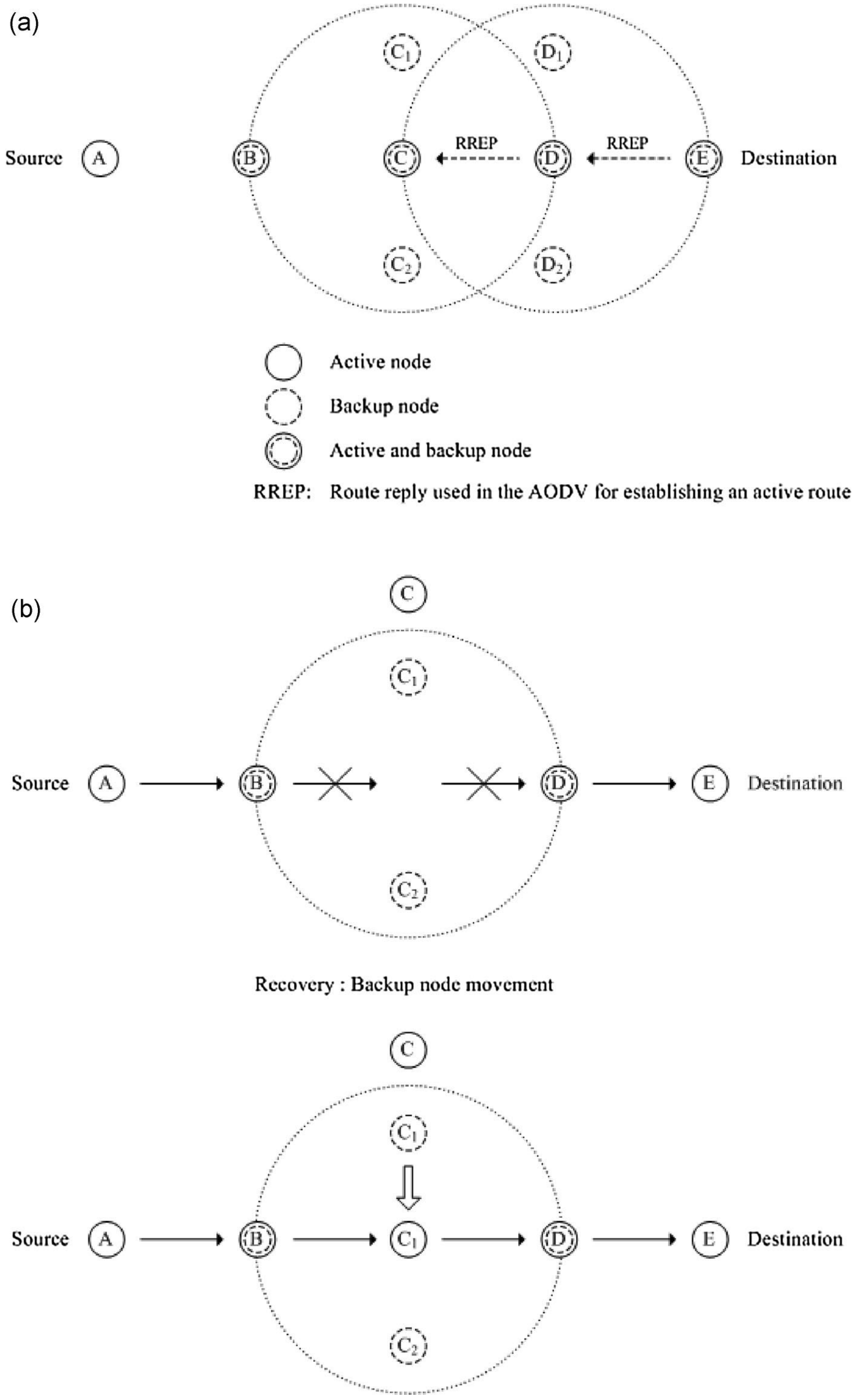


Figure 1. The illustration of basic idea: (a) backup node selection and (b) route recovery.

neighbor information, the establishment and maintenance of the protecting relationship can be easily done. As mentioned above, the hello mechanism can make each node know its up-to-date neighbor information in addition to the route failure detection. In the proposed approach, we extend the hello mechanism to establish and maintain the protecting relationship between an active node and each of its backup nodes.

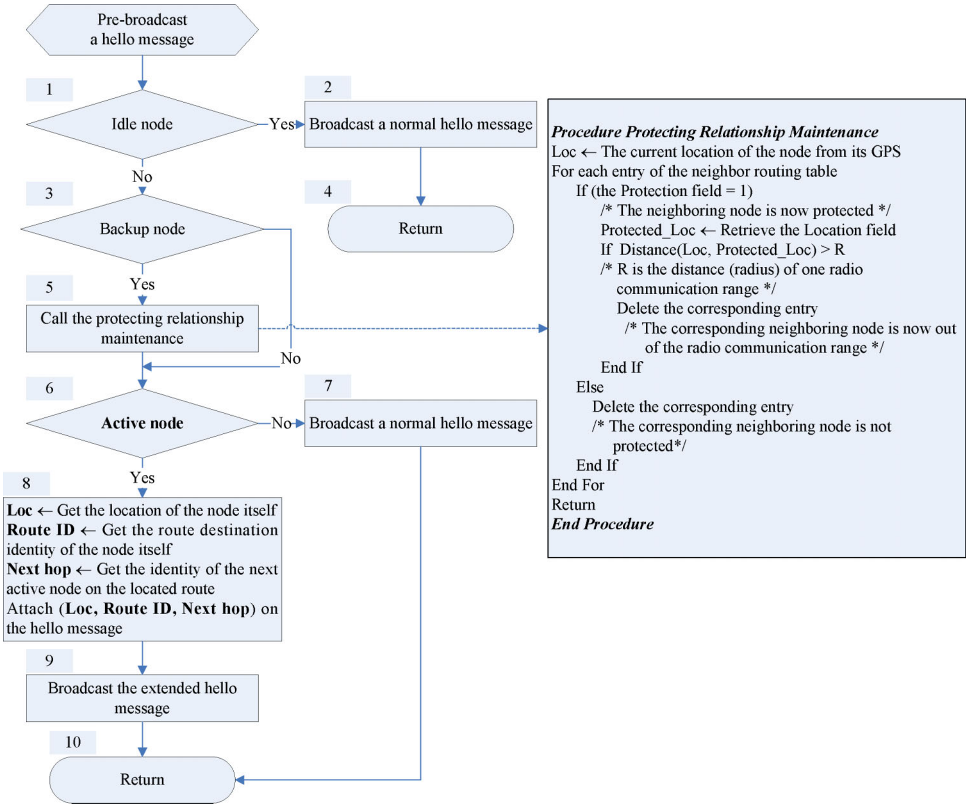
Neighboring routing table	
Neighbor ID:	The identity (address) of a corresponding neighboring node
Location:	The current location of a corresponding neighboring node
Protection:	0 (The corresponding neighboring node is not protected) or 1 (Protected)
Route ID:	The identity (address) of the route destination node if the corresponding neighboring node is an active node on a route
Next hop:	The identity (address) of the next active node if the corresponding neighboring node is an active node on a route

Figure 2. Data structure of a neighboring routing table.

- The routing table of a node is divided into two types: active routing table and neighboring routing table. Based on the AODV protocol [5], the routing table contains two kinds of route information: the routes to neighboring nodes and the routes to destination nodes. In the proposed approach, the routes to neighboring nodes are separated from the routing table to form another table called as neighboring routing table. The data structure of the table is given in Figure 2. The routes to destination nodes are still stored in the original routing table called as active routing table.
- The hello message is re-defined to have two formats: normal and extended. Compared to the normal hello message, the extended hello message adds three extra fields: *Location, Destination ID* and *Next hop*.
- A two-bit attribute vector is additionally stored in each node. If a node is on a route, the first bit of its attribute vector is set to '1'; otherwise '0'. The second bit is used as the indicator of a backup node. If a node is one backup node of an active node, the second bit of the attribute vector is set to '1'; otherwise '0'. It is possible to simultaneously set the first and second bits of the attribute vector to '1'. In such case, the corresponding node is an active node, and it is also a backup node of another active node. Conversely, if the two bits of an attribute vector are both set to '0', the corresponding node is an idle node that is neither an active node nor a backup node.

With the above given data structures, the establishment and maintenance of the protecting relationship can be integrated into the broadcasting (sending) and receiving procedures of a hello message, respectively. In the extended broadcasting procedure (see Figure 3), when a node broadcasts a hello message, it is according to its attributes (idle, backup, or active) to broadcast a normal or extended hello message. If the node has an idle attribute, it broadcasts a normal hello message (step 2 of Figure 3). If the node owns the backup attribute (step 3 of Figure 3), it executes the maintenance procedure of the protecting relationship to check whether it can continuously act as a backup node or not (step 5 of Figure 3).

In the maintenance procedure, the node computes the distance between it and each of its protected nodes. After computing the distance, if the distance is larger than the


Pre-broadcast a hello message
1
Idle node
Yes
2
Broadcast a normal hello message
No
3
Backup node
4
Return
Yes
5
Call the protecting relationship maintenance
No
6
Active node
No
7
Broadcast a normal hello message
Yes
8
Loc ← Get the location of the node itself
Route ID ← Get the route destination identity of the node itself
Next hop ← Get the identity of the next active node on the located route
Attach (Loc, Route ID, Next hop) on the hello message
9
Broadcast the extended hello message
10
Return

```
Procedure Protecting Relationship Maintenance
Loc ← The current location of the node from its GPS
For each entry of the neighbor routing table
    If (the Protection field = 1)
        /* The neighboring node is now protected */
        Protected_Loc ← Retrieve the Location field
        If Distance(Loc, Protected_Loc) > R
        /* R is the distance (radius) of one radio
           communication range */
           Delete the corresponding entry
             /* The corresponding neighboring node is now out
                of the radio communication range */
        End If
    Else
        Delete the corresponding entry
        /* The corresponding neighboring node is not
           protected*/
    End If
End For
Return
End Procedure
```

Figure 3. The broadcasting (sending) procedure of hello mechanism with the protecting relationship maintenance.

distance (radius) of one radio communication range, the node needs to abort acting as the backup node since it is out of the radio communication range of a protected node (it is not the neighboring node of a protected node again). In such case, the node needs to delete the entry corresponding to the protected node from its neighboring routing table. In addition to checking the backup attribute, the node also needs to check whether it also has the active attribute (step 6 of Figure 3) since an active node can also act as a backup node of another active node. If so, the node is on a route and it needs to broadcast an extended hello message; otherwise, it broadcasts a normal message. Before broadcasting an extended hello message, the node attaches the following information: location, identity of the destination node, and identity of the next active node on the hello message to form an extended hello message (step 8 of Figure 3). Note that the attached information is retrieved from the GPS module and active routing table of the node. The first attached information is for making the neighboring nodes of the active node act as backup nodes. The second and third attached information is used for performing route recovery.

In the extended receiving procedure (see Figure 4), when a node receives a hello message, it first checks the format of the hello message. If the hello message is with the extended format, the receiving node additionally executes the establishment procedure of the protecting relationship (steps 3–8 of Figure 4). The detailed operations are described as follows.

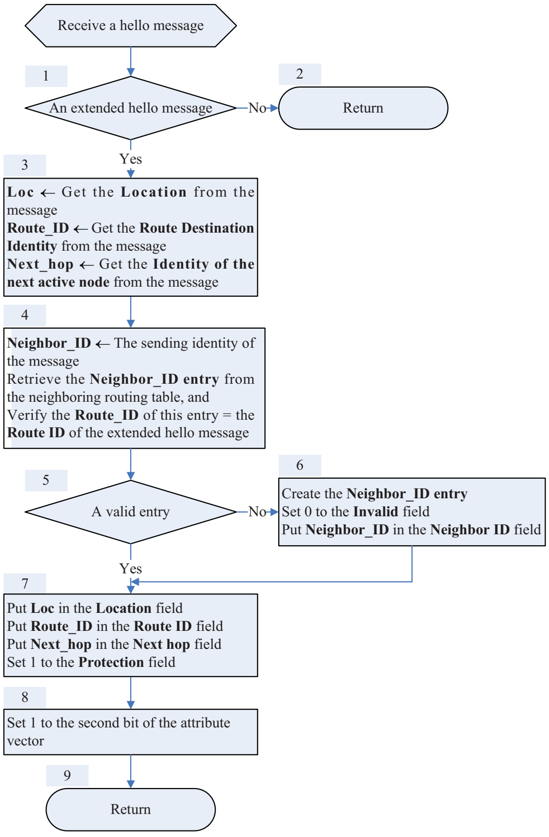

Receive a hello message
1
2
An extended hello message
No
Return
Yes
3
Loc ← Get the Location from the message
Route_ID ← Get the Route Destination Identity from the message
Next_hop ← Get the Identity of the next active node from the message
4
Neighbor_ID ← The sending identity of the message
Retrieve the Neighbor_ID entry from the neighboring routing table, and
Verify the Route_ID of this entry = the Route ID of the extended hello message
5
6
A valid entry
No
Create the Neighbor_ID entry
Set 0 to the Invalid field
Put Neighbor_ID in the Neighbor ID field
Yes
7
Put Loc in the Location field
Put Route_ID in the Route ID field
Put Next_hop in the Next hop field
Set 1 to the Protection field
8
Set 1 to the second bit of the attribute vector
9
Return

Figure 4. The receiving procedure of hello mechanism with the protecting relationship establishment.

- First, the receiving node finds the entry corresponding to the sending node from its neighboring routing table. If the neighboring routing entry cannot be found, this entry is created in the neighboring routing table (steps 4–6 of Figure 4). The sending node must be one neighboring node of the receiving node; otherwise, the receiving node cannot receive this hello message.
- Then, the receiving node puts the attached information (*Location*, *Destination ID*, and *Next hop*) of the extended hello message in the corresponding fields of the neighboring routing entry. A protection mark is also made in this entry by setting '1' in the protection field. The receiving node becomes a backup node of the sending node (step 7 of Figure 4).
- Next, the receiving node puts '1' in the second bit of its attribute vector (step 8 of Figure 4) to represent that it acts as a backup node.

3.2.2. Route recovery.

When an active node moves to break a route, the link breakage can be detected by the backup nodes of the active node, as shown in Figure 5. In Figure 5(a), the active node moves out the radio communication ranges of all its backup nodes. All the backup nodes can know this link breakage since they cannot continuously receive a hello message from the moving active node again. In Figure 5(b), the link breakage can be only detected by some backup nodes. However, in such case, the moving active node itself can also know the link breakage since it does not receive hello messages from some of its neighboring nodes. For making all the backup nodes can detect the link breakage, the moving active node also broadcasts a *recovery notification* message with TTL = 1 after detecting a link breakage, as shown in Figure 6(a). The recovery notification message can be received by the backup nodes that they are still within the radio communication range of the moving active node. Originally, these backup nodes do not know the link breakage. With the recovery notification message, these backup nodes can know a link breakage.

As shown in Figure 5, the link breakage is caused due to the movements of intermediate active nodes. When the movement of the source (destination) node causes a link breakage, the proposed approach can also tolerate such link breakage. The handling of this link breakage is similar to Figure 5(b), described as follows. When the source (destination) node detects a link breakage with its next (previous) active node, it also broadcasts a recovery notification message to notify its backup nodes. In such situation, these backup nodes compete with each other to move to the original location of the source (destination) node for connecting the moving source (destination) node with its next (previous) active node.

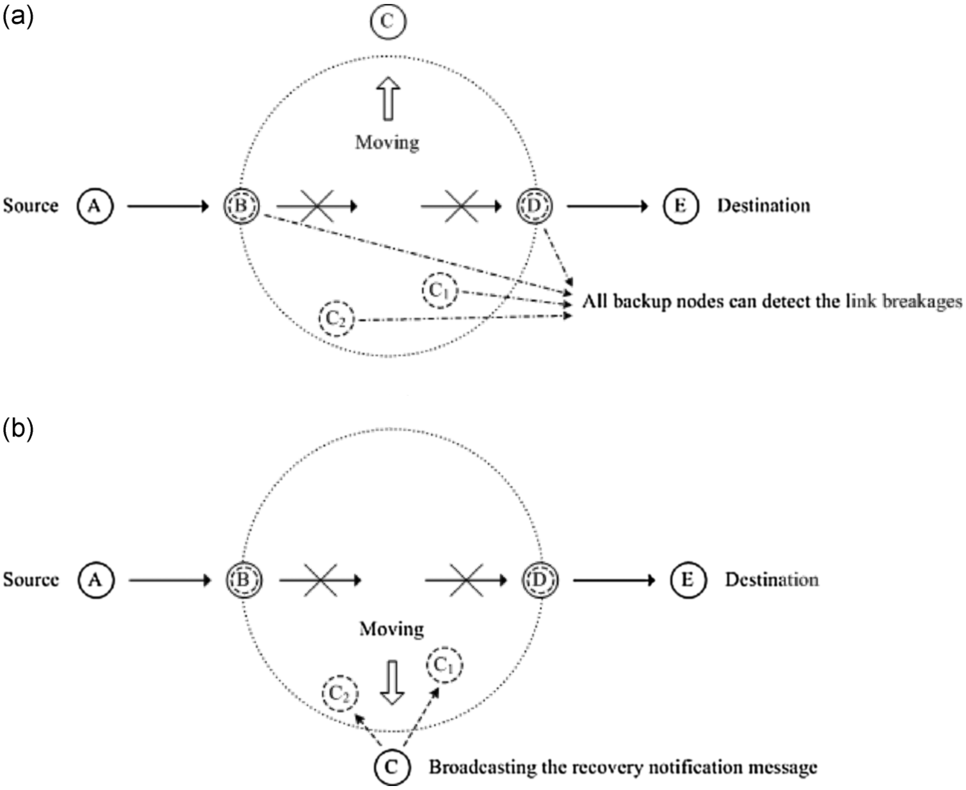
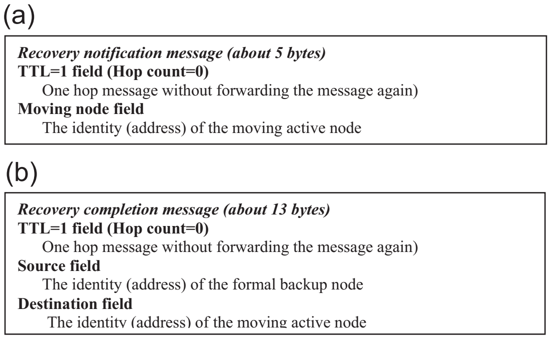

(a)
C
Moving
Source
A
B
D
E
Destination
C1
C2
All backup nodes can detect the link breakages
(b)
Source
A
B
D
E
Destination
Moving
C2
C1
C
Broadcasting the recovery notification message

Figure 5. Link breakage detection: (a) node moving away from all backup nodes and (b) node moving towards some backup nodes.



(a)

Recovery notification message (about 5 bytes)
TTL=1 field (Hop count=0)
One hop message without forwarding the message again)
Moving node field
The identity (address) of the moving active node

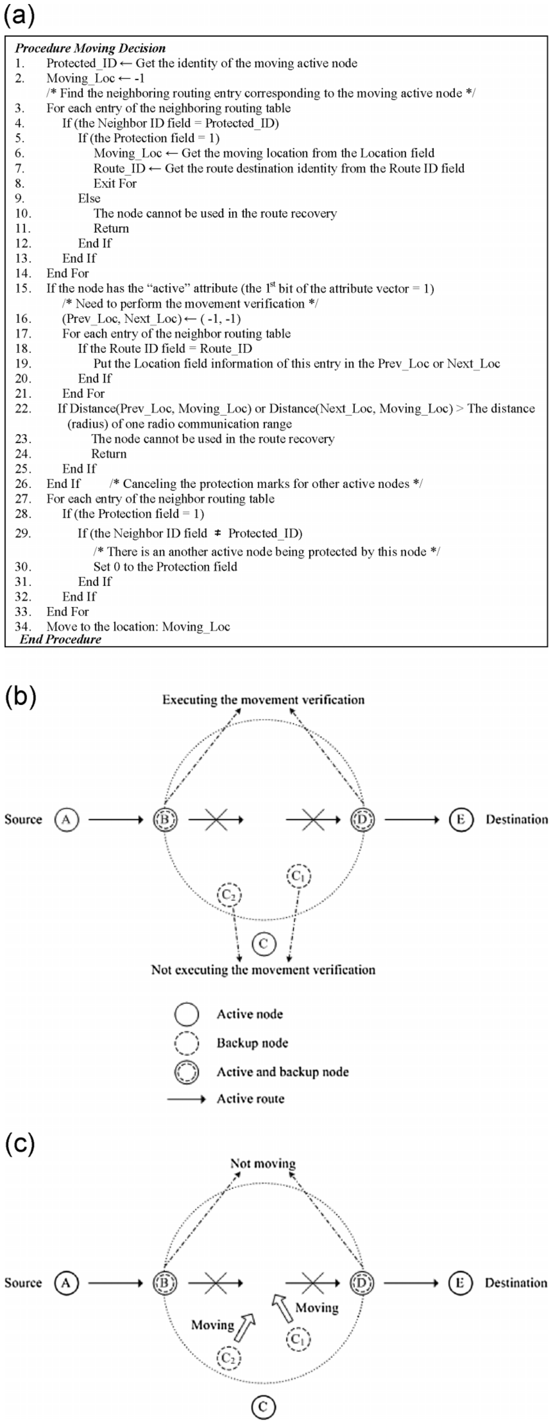
(b)

Recovery completion message (about 13 bytes)
TTL=1 field (Hop count=0)
One hop message without forwarding the message again)
Source field
The identity (address) of the formal backup node
Destination field
The identitv (address) of the moving active node

Figure 6. The formats of route recovery messages: (a) recovery notification message and (b) recovery completion message.

Before executing the node movement, each backup node also needs to execute the moving decision procedure to decide whether it should really move to the original location of the moving active node or not, as shown in Figure 7(a). The detailed operations are described as follows.

- First, the backup node searches its neighboring routing table to find the entry corresponding to the moving active node. If the entry is with a valid protection mark (protection field = '1'), it represents that the backup node can be used to replace the moving active node. Then, the backup node gets the original location of the moving active node from the location field of the entry. Conversely, if the entry is with an invalid protection mark (protection field = '0'), it represents that the backup node is now being used to perform recovery for another broken route. In such case, this backup node cannot be used to replace the moving active node again. Note that the number of backup nodes for an active node is not only one. Although the moving active node cannot be replaced by this backup node, it can be replaced by one of other backup nodes. The above operations are given in the lines 1–14 of Figure 7(a).
- Then, the backup node checks its attribute to determine whether it should really move to replace the moving active node or not. If it has an active attribute, it needs to execute the movement verification to guarantee that its movement cannot disrupt the connectivity of its original route. For example, in Figure 7(b), nodes *B* and *D* need to execute the movement verification. In the movement verification, if the verified distances are all less than the distance of one radio communication range, the backup node can move to the original location of the moving active node; otherwise not. The locations of the previous and next active nodes can be found from the neighboring routing table since they are also the neighboring nodes of the used backup node. The above operations are corresponding to lines



```
Procedure Moving Decision
1.  Protected_ID ← Get the identity of the moving active node
2.  Moving_Loc ← -1
    /* Find the neighboring routing entry corresponding to the moving active node */
3.  For each entry of the neighboring routing table
4.      If (the Neighbor ID field = Protected_ID)
5.          If (the Protection field = 1)
6.              Moving_Loc ← Get the moving location from the Location field
7.              Route_ID ← Get the route destination identity from the Route ID field
8.              Exit For
9.          Else
10.             The node cannot be used in the route recovery
11.             Return
12.         End If
13.     End If
14. End For
15. If the node has the "active" attribute (the 1st bit of the attribute vector = 1)
        /* Need to perform the movement verification */
16.     (Prev_Loc, Next_Loc) ← ( -1, -1)
17.     For each entry of the neighbor routing table
18.         If the Route ID field = Route_ID
19.             Put the Location field information of this entry in the Prev_Loc or Next_Loc
20.         End If
21.     End For
22.     If Distance(Prev_Loc, Moving_Loc) or Distance(Next_Loc, Moving_Loc) > The distance
          (radius) of one radio communication range
23.         The node cannot be used in the route recovery
24.         Return
25.     End If
26. End If        /* Canceling the protection marks for other active nodes */
27. For each entry of the neighbor routing table
28.     If (the Protection field = 1)
29.         If (the Neighbor ID field ≠ Protected_ID)
                /* There is an another active node being protected by this node */
30.             Set 0 to the Protection field
31.         End If
32.     End If
33. End For
34. Move to the location: Moving_Loc
End Procedure
```

Figure 7. Backup node movement for route recovery: (a) the procedure of the moving decision, (b) the execution of the movement verification, and (c) the movements of backup nodes.

15–26 of Figure 7(a). Conversely, if the backup node does not have the active attribute, it can directly move to the original location of the moving active node, e.g., nodes C_1 and C_2 in Figure 7(c).

- After replacing the moving active node, if the backup node also protects other active nodes, it is required to invalidate the protection marks for other active nodes. The invalidation is done by setting '0' in the protection field for one or more neighboring routing entries (see lines 27–33 of Figure 7a).

Based on the above description, it is possible that more than one backup node will move to replace the moving active node. When a backup node first moves to the original location of the moving active node, this backup node will actually replace the moving active node, and it is called the formal backup node. Next, the formal backup node will broadcast a *recovery completion* message with $\mathrm{TTL} = 1$ to notify other backup nodes to stop moving since the moving active node has been replaced by it. The format of the recovery completion message has been given in Figure 6(b). When receiving the recovery completion message, each of other backup nodes takes the following corresponding actions, as shown in Figure 8.

- If the backup node is moving for replacing the moving active node, it will immediately stop moving.
- If the backup node is the previous active node of the moving active node, it will update its active routing table to connect with the formal backup node.
- If the backup node is the next active node of the moving active node, it only responds with an acknowledgement message. The formal backup node will use this acknowledgement message to make the link connectivity with this next active node.

Next, the formal backup node can establish its new protecting relationship with other nodes when the formal backup node sends a new hello message. For the moving active node, the protecting relationship with some of its original backup nodes will be terminated if it moves out the radio communication ranges of these backup nodes. The termination of the protecting relationship is done when these backup nodes respectively send their new hello messages.

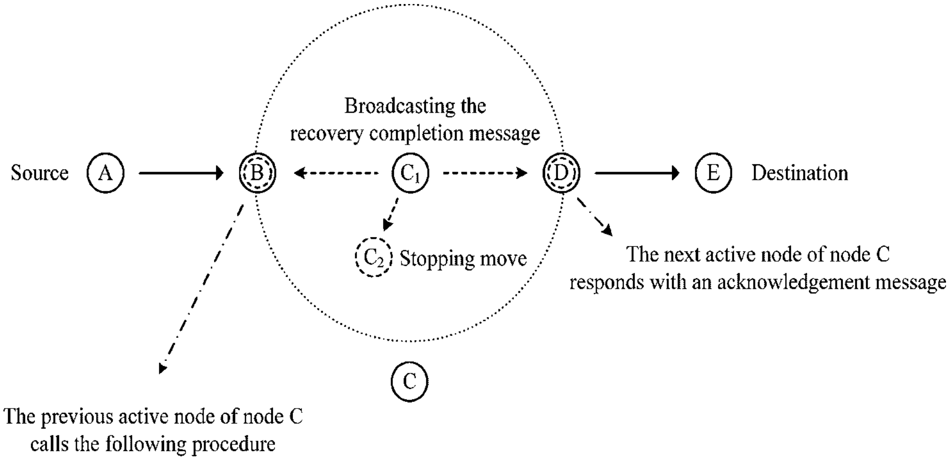
Finally, if a route is broken due to the status change of an active node (e.g., working → inactive), the backup nodes of the inactive node also compete with each other to replace the inactive node based on the abovementioned method. The protecting relationship between the inactive node and its backup nodes can be terminated when sending a new hello message.

4. COMPARISONS

This section makes the comparisons between the proposed approach and previous approaches mentioned in Section 2.2.

4.1. Qualitative comparisons

Table I gives the qualitative comparisons based on the recovery method, backup route (node) maintenance, recovery


Broadcasting the
recovery completion message
Source
A
B
C1
D
E
Destination
C2
Stopping move
The next active node of node C
responds with an acknowledgement message
C
The previous active node of node C
calls the following procedure

```
Procedure Connectivity of the Broken Active Route
    (Backup_ID, Invalid_ID) ← Retrieve the (Source, Destination) fields of
    the recovery complete message
    For each entry of the active routing table
        If (Next Hop field = Invalid_ID)
        /* This node is the previous active node of the invalid active node
           since it has a route entry corresponding to the invalid active node */
            Update the Next Hop field to Backup_ID
            Exit For
        End If
    End For
End Procedure
```

Figure 8. Local route recovery.

Table I. Comparisons.

Comparison metrics	AODV [5]	AODV-BR [6] AODV-CF [7]	AODV-RBA [8]	AODV-LU [9]	Proposed approach
Recovery method	On-demand route discovery	Backup route pre-establishment	Link break predication and on-demand route discovery	Backup route pre-establishment	Node movement
Backup route (node) maintenance	No	No	No	No	Yes
Recovery capability	Dependent on the network topology	Dependent on the pre-established backup route	Dependent on the network topology	Dependent on the pre-established backup route	Dependent on the pre-selected backup node
Failure-free overhead	No	Pre-establishing the backup routes	Predicting the link breakage	Pre-establishing the backup routes	Maintaining the backup nodes
Fault-tolerant overhead	Discovering the new route	Activating the pre-established backup route (worst case: discovering the active new route)	Discovering the new route	Activating the pre-established backup route (worst case: discovering the new active route)	Moving the pre-specified backup nodes

capability, failure-free overhead, and fault-tolerant overhead.

- *Recovery method*: In the original AODV protocol, it can dynamically find a new route instead of a broken route. The AODV-BR [6], AODV-CF [7], and AODV-LU [9] pre-establish one or more backup routes from each active node to the destination node. The backup routes will be used when a primary route is broken. The AODV-RBA [8] also dynamically finds a new route when it predicts a possible link breakage in a primary route. Unlike all the previous approaches, the proposed approach does not pre-establish or dynamically find any backup routes. Each active node on a route is protected using its neighboring nodes as backup nodes. When the movement of an active node causes a link breakage on a route, the moving active node will be replaced by one of its neighboring nodes based on node movements.
- *Backup route (node) maintenance*: The backup route maintenance is not considered in the previous approaches [5–9]. However, node mobility in a MANET is very frequent. The node movements can also occur in the nodes of a backup route. In contrast, the proposed approach considers the backup node maintenance. When an active node or one of its backup nodes moves, the proposed approach immediately updates the backup node set of the corresponding active node.
- *Recovery capability*: When an active route is broken, the AODV protocol and AODV-RBA [8] use the route discovery mechanism to dynamically find a new route as the backup route. Intuitively, the recovery capabilities of these two approaches are dependent on whether there is an alternative route between the source and destination nodes. However, due to node mobility, it is also possible that a link breakage later occurs in the found backup route. The probability of this situation has been evaluated by References [17,18]. First, the inter-packet arrival time (the waiting time between two consecutive packet transmissions) is assumed to have an exponential distribution. Similarly, the link breakage interval in each two neighboring active nodes is also assumed to follow an exponential distribution. The link breakage interval indicates the required time for causing a link breakage between two active nodes due to node movements. With the assumption of the exponential distribution, the probability P_{B} that a link breakage occurs in a backup route when the backup route is used to transmit packets is [17,18]:

$$P_B = \frac{\mu}{\lambda + \mu} \tag{1}$$

where λ is the mean packet arrival rate (the mean communication frequency), and μ is the mean link breakage rate (the mean link breakage frequency).Based on Equation (1), if the length (the number of links) of a found backup path is l, the probability P_T that the found backup route can be used to successfully transmit packets without incurring link breakages is

$$P_T = \left(1 - \frac{\mu}{\lambda + \mu}\right)^l \tag{2}$$

In Equation (2), it additionally makes the identical exponential distribution assumption for simplifying the derivation. Each node of a MANET has the same exponential distribution for the inter-packet arrival time. The link breakage interval in each two neighboring active nodes is also assumed to have the same exponential distribution. Note that the identical distribution is a common assumption used in many queuing models (e.g., *M/M/1*, *M/M/*, etc.). P_T can be also used to represent the recovery capability of a found backup route. If there are n backup routes between the source and the destination nodes, the recovery capabilities of the AODV protocol and the AODV-RBA [8] are

$$R_O = 1 - \prod_{i=1}^{n}(1 - \left(1 - \frac{\mu}{\lambda + \mu}\right)^{l_i}) \tag{3}$$

where l_i is the length of the ith backup route, and the term $\Pi_{i=1}^{n}(1 - (1 - \frac{\mu}{\lambda+\mu})^{l_i})$ is the probability that all n backup routes incur the link breakages.For AODV-BR [6], AODV-CF [7], and AODV-LU [9], these approaches do not maintain their pre-established backup routes. As mentioned above, if the stale backup routes are used in an approach, the approach will have a low recovery capability. In those approaches, if no pre-established backup routes can be used to successfully transmit packets, they will use the route discovery mechanism to dynamically find a backup route. In such case, the recovery capability is dependent on the network topology, which is same as the recovery capability of the AODV protocol. Therefore, AODV-BR [6], AODV-CF [7], and AODV-LU [9] do not enhance the recovery capability of the AODV protocol.In the proposed approach, it uses node movements to locally repair the broken route. The recovery capability R_P of the proposed approach is dependent on the average number of neighboring nodes for an active node, which can be evaluated as follows:

$$R_P = (1 - P_N)(1 - P_A) \tag{4}$$

where P_N is the probability that no nodes exists within the radio communication range of an active node. Therefore, $(1 - P_N)$ is the probability that an active node has one or more neighboring nodes. However, not all neighboring nodes can replace the moving active node. If a neighboring node is also an active node and its movement disrupts its original route, this neighboring node will not be used for replacing the moving active node. Therefore, $(1 - P_A)$ considers the probability how many neighboring nodes can be used for replacing the moving active node.The derivation of P_N has been given in References [17,18], as follows:

$$P_N = \binom{n-1}{0}(1-P_0)^{n-1} = (1-\frac{\pi r^2}{A})^{n-1} \quad (5)$$

where P_0 is the probability that a particular node Y is within the radio communication range of node X, n is the number of nodes in a MANET, r is the communication radius of node X, and A is the area size of a MANET.As for P_A, it can be evaluated as follows:

$$P_A = \sum_{k=1}^{n-1} \binom{n-1}{k} (P_0 P_1 P_2)^k (1 - P_0 P_1 P_2)^{n-1-k} \quad (6)$$

where P_0 can be also used to represent the probability the kth node in a MANET is the neighboring node of an active node (Note that the largest number of neighboring nodes for an active node is $n - 1$ since there are n nodes in the MANET), P_1 represents the probability that the neighboring node owns the active attribute, and P_2 is the probability that the movement of the neighboring node will disrupt its original route. Therefore, $P_0P_1P_2$ is the probability that a neighboring node is unsuitable to be a backup node. In Equation (6), it represents the probability how many neighboring nodes cannot be used for replacing a moving active node.From Equations (3) and (4), we can obviously observe that the AODV protocol and the proposed approach have different affected factors in the recovery capability. In the AODV protocol, the recovery capability is dependent on global connection topology of the whole network, such that there are how many backup routes for a source–destination node pair. In contrast, the recovery capability of the proposed approach relies on local connection topology of an active node, such that there are how many neighboring nodes for an active node. If no backup route exists between the source and destination nodes, the proposed approach still can repair the broken route by moving one of neighboring nodes to replace the moving active node.

- *Failure-free overhead*: During the normal time (the failure-free period), the AODV protocol does not take any actions against route recovery. The AODV-BR [6] needs to pre-establish backup routes, but the pre-established backup routes are highly overlapped with the corresponding primary route. When there is a link breakage on the primary route, it is very possible that the pre-established backup routes are also broken. To pre-establish better backup routes, the AODV-CF [7] uses two additional control messages: *Controlled* and *Controlled-Ack*. For the AODV-RBA [8], it needs to periodically measure the danger situation of each link on a route during the failure-free period. As for the proposed approach, it extends the hello mechanism to establish and maintain backup nodes for each active node. The failure-free overhead of the proposed approach is mainly determined by the execution cost of additional operations in the extended hello mechanism. From Figures 3 and 4, we can see that the additional operations are a few of low-cost instructions.
- *Fault-tolerant overhead*: To repair a broken route, the AODV protocol incurs the route discovery overhead. Due to flooding the RREQ message over the entire network, the route recovery may introduce a long delay. For AODV-RBA [8], if it measures a possible link breakage on a route, it also initiates the route discovery mechanism. Therefore, the AODV-RBA [8] has the same fault-tolerant overhead with the AODV protocol. To avoid flooding the RREQ message, the AODV-BR [6], AODV-CF [7], and AODV-LU [9] pre-establish backup routes in advance. If a route is broken, these three approaches activate the pre-established backup route instead of the broken route. The fault-tolerant overhead of each of these three approaches [6,7,9] seems to be small. However, if no pre-established backup routes can be used, these approaches [6,7,9] are also required to dynamically find a backup route. In such case, the fault-tolerant overhead of each of these three approaches [6,7,9] is same as the AODV protocol. As for the proposed approach, it locally repairs the broken route using neighboring node movements. The movement cost C_M is estimated as follows:

$$C_M = \frac{D_H}{M_S} \quad (7)$$

where C_M is mainly determined by two factors: the distance (radius) of the radio communication range of an active node D_H, and the moving speed of a neighboring node M_S. The values of these two factors are predetermined according to the hardware limitation (power supply) of a MANET node, which are not designated by the proposed approach.

4.2. Quantitative comparisons

For making the quantitative comparisons, we extend the *ad hoc* simulation module of the Network Simulator version 2.27 (ns-2) [19] to perform simulation experiments.

4.2.1. Simulation model.

In simulation experiments, we refer to Reference [14] to set up following simulation parameters. First, two MANET are modeled by distributing 50 nodes in a 2000 m × 600 m field and a 1500 m × 300 m field, respectively. In the two MANETs, the radius of the radio communication range for a node is set to 200 m. According to the node density indicated in Reference [20], if the node density of a MANET is less than or equal to 2π, the MANET is with sparse topology. If the node density is 3π or larger, the MANET is with dense topology. For a sparse MANET, it is difficult to find two or more routes for a source–destination node pair. In our simulation experiments, the MANET with 50 nodes in a $2000\,\mathrm{m} \times 600\,\mathrm{m}$ field represents a sparse MANET since the node density is 1.66π. With the large node density, the MANET with 50 nodes in $1500\,\mathrm{m} \times 300\,\mathrm{m}$ represents a dense MANET. To understand the effect of the radio communication range in the recovery capability, we also vary the value of the radio communication range from 50 to 200 m with an increment of 50 m in the heavy traffic scenario. The channel capacity of each node is set to 2 Mb/s.

For the node mobility, the random waypoint model [21] is used to make nodes with different moving speeds (1–20 m/s) and pause time (100–500 s). In the random waypoint model, the mobility of each node is independent of each other. To make nodes with a certain degree of dependent mobility, we additionally adopt another mobility model: community-based model [22]. In the community-based model, M hubs are set in the field of a MANET. Each node selects N_m of M hubs to form a hublist as its waypoints (moving positions). The movements of any two nodes are dependent if they have similar hublists. Whenever a node reaches a hub area, it stays in the hub area for a random time. After the timer expires, the node moves to the next waypoint of its hublist.

As for network traffic, 10 and 40 source–destination node pairs (communication routes) are respectively generated in the dense and sparse MANETs. The session time of a route is randomly set between 500 and 5000 s. Next, constant bit rate traffic is generated from each source node by periodically sending a 512-byte packet to the corresponding destination node.

Based on the above parameter settings, we performed 100 simulation runs. Each simulation run takes 5000 s to measure the following four concerned metrics: failure-free overhead, fault-tolerant overhead, average recovery capability, and average recovered route length. The fault-free overhead and fault-tolerant overhead are combined together as the normalized recovery routing overhead [14]. For an approach, if the approach has high-normalized recovery routing overhead, it represents that the approach incurs high failure-free and/or fault-tolerant overhead. Conversely, if the normalized recovery routing overhead is low, it represents that the approach has low failure-free and fault-tolerant overhead. In Reference [14], the normalized recovery routing overhead is defined as follows:

$$\frac{N_{\mathrm{ctl\_ff}} + N_{\mathrm{ctl\_}ft}}{N_{\mathrm{pkt}}} \tag{8}$$

Where $N_{\mathrm{ctl\_ff}}$ is the number of control messages issued during the failure-free period for route recovery, $N_{\mathrm{ctl\_ft}}$ is the number of control messages issued during the fault-tolerant period for route recovery, and N_{pkt} is the number of packets delivered successfully at the destination node.

For the metric of the average recovery capability, it measures the probability that the broken routes can be repaired successfully. As for the last metric, the average recovered route length is to measure the average length of a broken route after performing recovery.

4.2.2. Simulation results.

Figures 9–11 show the simulation results of the above concerned metrics, respectively.

As shown in Figure 9, the AODV-RBA [8] has the highest normalized recovery routing overhead in two mobility models and two MANETs since it needs to flood the RREQ message when detecting a danger link on a route. The normalized recovery routing overhead of the AODV protocol is little smaller than the AODV-RBA [8] since it floods the RREQ message only when actually detecting a link breakage on a route. For the proposed approach, it issues few control messages during the failure-free and fault-tolerant periods. The normalized recovery routing overhead is almost same with the AODV-LU [9]. However, the backup routes in AODV-LU [9] are not maintained as nodes move. If the pre-established backup routes in AODV-LU [9] cannot be used to repair a broken route, AODV-LU [9] is assisted by the AODV protocol to dynamically find a backup route. In such case, the AODV-LU [9] also incurs the RREQ flooding overhead, and its normalized recovery routing overhead is much larger than the proposed approach. Similarly, the AODV-BR [6] and AODV-CF [7] are also required to be supported by the AODV protocol if the pre-established backup routes are stale or broken.

Figure 10 shows the average recovery capabilities of various approaches in two mobility models and two MANETs, respectively. The AODV-BR [6], AODV-CF [7], and AODV-LU [9] have obviously smaller recovery capabilities than other approaches. These three approaches pre-establish backup routes and do not maintain backup routes as nodes move. When a route is broken, the corresponding pre-established backup route may not be used to transmit packets to the destination node. In this situation, the AODV protocol is required to support these three approaches to dynamically find a backup route. With the assistance of the AODV protocol, the average recovery capabilities of these three approaches can be enhanced. However, the largest recovery capability is same as the given recovery capability of the AODV protocol. From Figure 10, we also see that the AODV-RBA [8] has the same average recovery capability with the AODV protocol since it also uses the route discovery mechanism to dynamically find a backup route. Based on the above description, we can know that the AODV protocol has the largest average recovery capability among all previous approaches.

To compare the AODV protocol with the proposed approach, their average recovery capabilities are similar in the dense MANET. In this MANET environment, nodes are densely scattered. It can easily find one or more backup routes for a source–destination node pair. If a route is broken, the AODV protocol has a high probability to find another route. In contrast, the proposed approach is based on the neighboring node movements to repair a broken route.

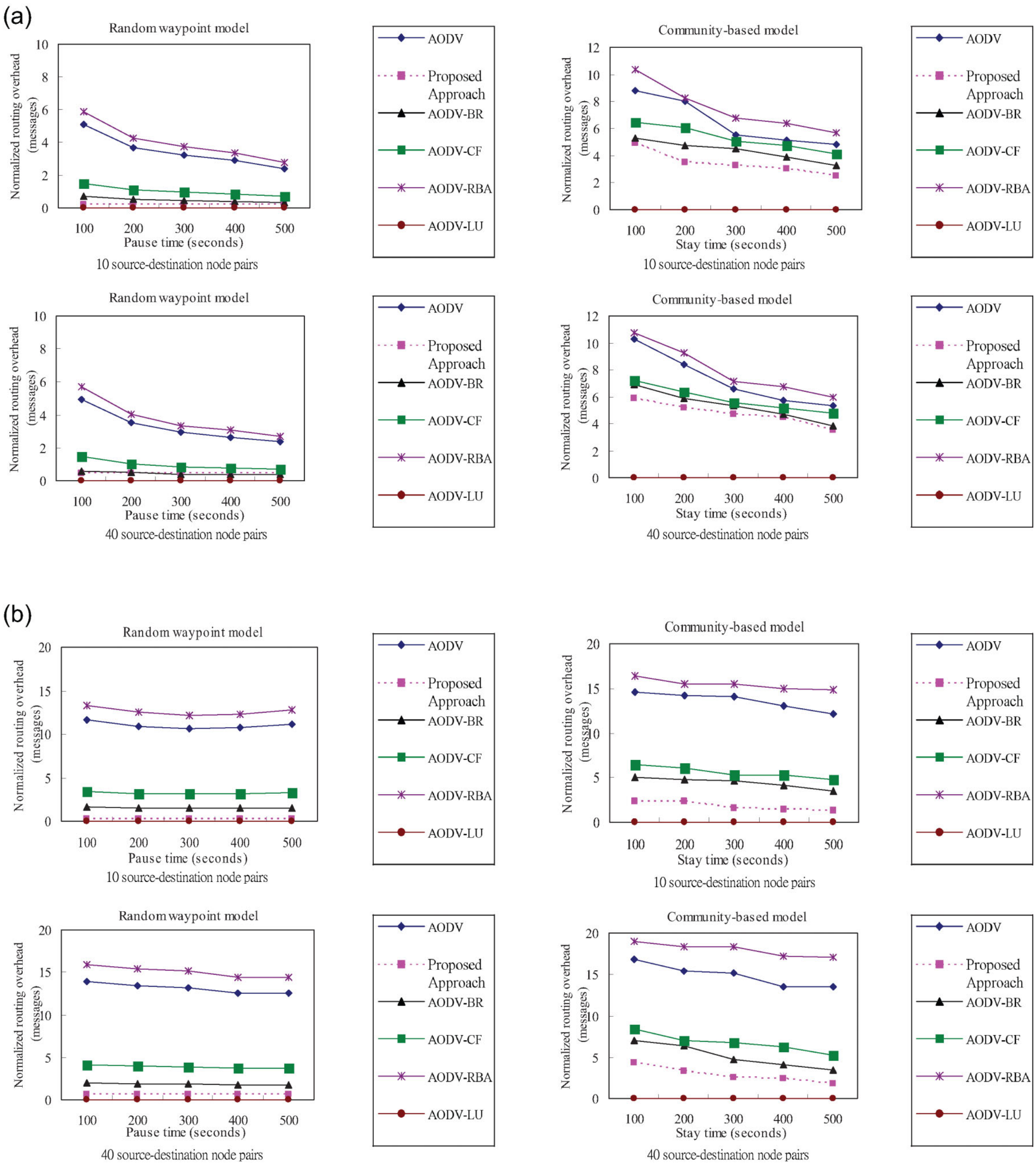
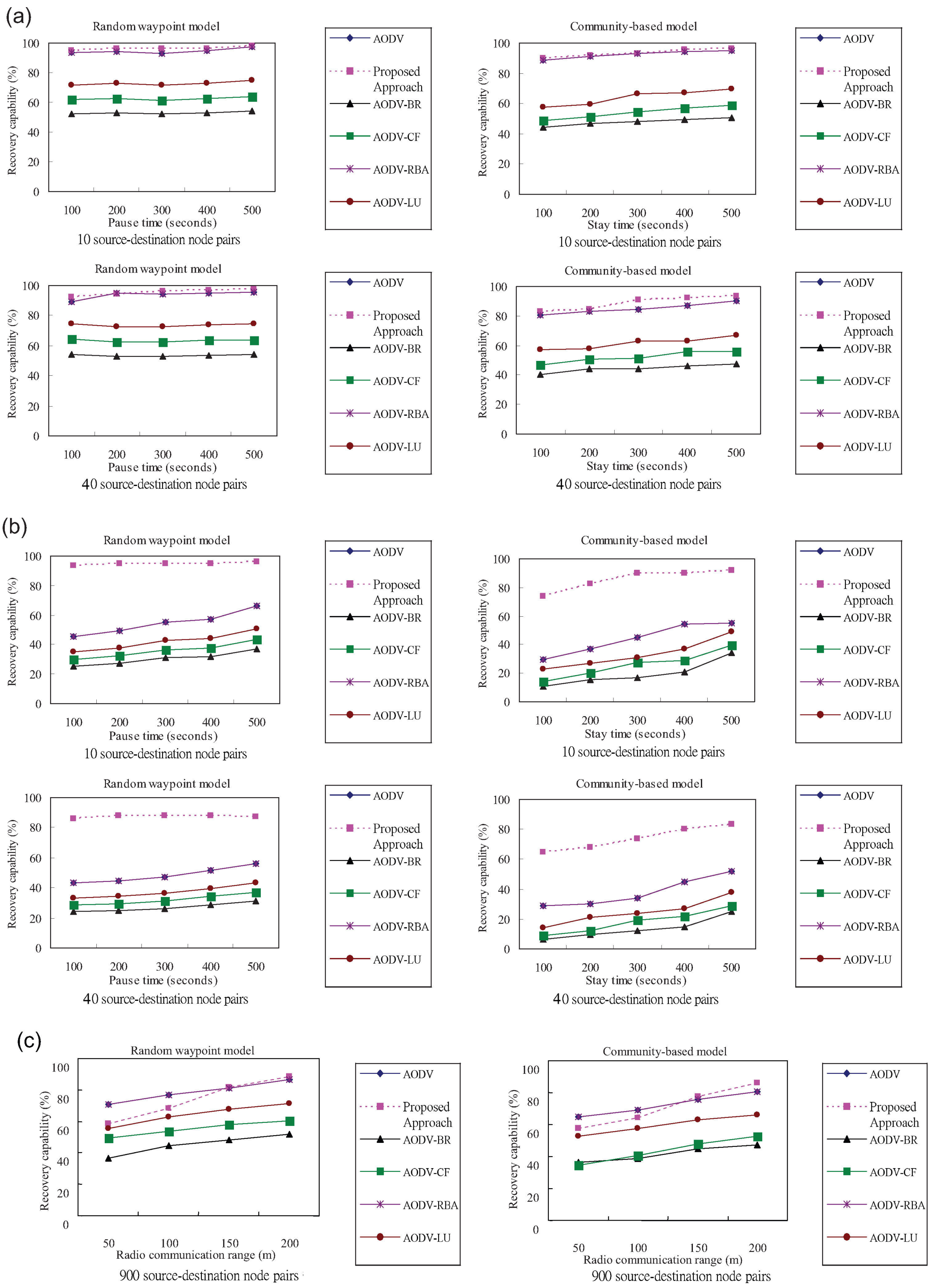


Figure 9. The simulation results of the normalized recovery routing overhead: (a) the dense MANET and (b) the sparse MANET.

In the dense MANET, each node can also easily find more than one neighboring node as backup nodes. Therefore, the proposed approach also has a large recovery capability. However, in the sparse MANET, nodes are sparsely scattered. In this MANET environment, the proposed approach has an obviously larger recovery capability than the AODV protocol. In the sparse MANET, it is difficult to find more than one route for a source–destination node pair. Therefore, the AODV protocol has a smaller average recovery capability. In contrast, the proposed approach concerns the distribution of the neighboring nodes In the simulation experiments with a sparse MANET, the average number of the neighboring nodes for an active node in the sparse MANET is 4. The proposed approach can make each active node with four backup nodes. Therefore, the proposed approach has a larger average recovery capability.

To understand the given recovery capability of the proposed approach in heavy traffic, we additionally generate



(a) Random waypoint model — 10 source-destination node pairs; Community-based model — 10 source-destination node pairs; Random waypoint model — 40 source-destination node pairs; Community-based model — 40 source-destination node pairs

(b) Random waypoint model — 10 source-destination node pairs; Community-based model — 10 source-destination node pairs; Random waypoint model — 40 source-destination node pairs; Community-based model — 40 source-destination node pairs

(c) Random waypoint model — 900 source-destination node pairs; Community-based model — 900 source-destination node pairs

Figure 10. The simulation results of the average recovery capability: (a) the dense MANET, (b) the sparse MANET, and (c) the heavy traffic in the dense MANET.

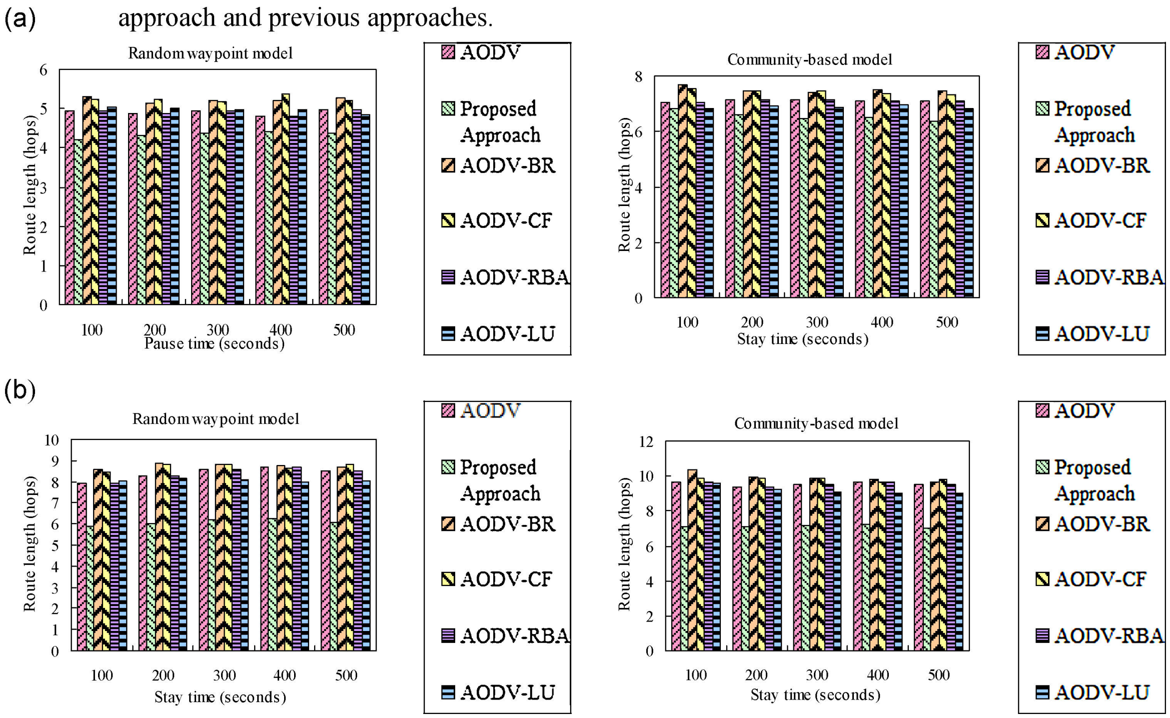


Figure 11. The simulation results of the recovered route length (a) the dense MANET and (b) the sparse MANET.

900 routes in the dense MANET to measure the recovery capability, as shown in Figure 10(c). Note that it is difficult to generate 900 routes in the sparse MANET due to the sparse node distribution. In Figure 10 (c), the recovery capability is also measured under various radio communication ranges. As expected, the recovery capability of the proposed approach decreases as the radio communication range shortens. This result also appears in each of the previous approach. However, the proposed approach has a more obvious effect on varying the radio communication range.

Note that the various radio communication ranges in Figure 10(c) can also represent various average node degrees. Given fixed simulation field and node density, if the radio communication range is designated with a large value, the average node degree is also large since each node has many neighboring nodes. Conversely, if a small value is assigned to the radio communication rage, the average node degree will be also small. In Figure 10(c), the average node degrees corresponding to various radio communication ranges are 2, 4, 8, and 12, respectively. When the radio communication range is 50 m, the corresponding average node degree is 2. In such case, the neighboring node movements cannot be easily done in the proposed approach because the movements will disrupt their original routes of the neighboring nodes. Therefore, the recovery capability of the proposed approach is smaller than the AODV and AODV-RBA, but it is still larger than other previous approaches. However, it is also known that AODV and AODV-RBA is based on large recovery overhead to achieve high recovery capability. If the radio communication rage is 150 m, the corresponding average node degree is 8. In this case, the recovery capability of the proposed approach is larger than all previous approaches.

As for the average recovered route length, Figure 11 shows that the proposed approach has the shortest length by comparing with all previous approaches in the two mobility models and two MANETS, respectively. In the proposed approach, when a route is broken, the moving active node is replaced by one of its neighboring nodes. After performing recovery, the length of the new route is same as the original route since the two routes have the same number of active nodes. In contrast, the previous approaches either pre-establish or dynamically find a backup route. The length of the backup route is often longer than the original route since the shortest route is usually as the primary route.

From the above descriptions, we can obviously see that the proposed approach has less recovery overhead and higher recovery capability in most of cases.

4.2.3. Complexity analysis.

In above simulation experiments, the computation cost of the route recovery is not shown since the value is very small. In this subsection, we use complexity analysis to make the computation cost and message comparisons between the proposed approach and previous approaches.

The previous approaches either pre-establish or dynamically find backup routes using RREQ message. For the previous approaches, the operation of forwarding one-hop RREQ message is the counting unit for analyzing the computation and message complexities. Due to flooding the RREQ message, the worst computation cost of each previous approach will involve all nodes to forward the RREQ message. The complexity of the computation cost is $O(n^2)$, where n is the number of nodes in a MANET. For the proposed approach, each active node is protected by its neighboring nodes. To set up the protection, the hello message in the proposed approach is extended to include the protection information. In the protection setup part of the proposed approach, the operation of broadcasting one-hop extended hello message is the counting unit for analyzing the computation and message complexities of this part. The complexity of the computation cost is $O(m \times n)$, where m is the maximum number of neighboring nodes for an active node. In the roué recovery part of the proposed approach, when detecting a link breakage on a route, the following three tasks are performed: moving decision (see Figure 7a), moving verification (see Figure 7b), and node movement (see Figure 7c). In the first task, the time complexity is $O(m)$. In the second task, the time complexity is also $O(m)$. In simulation experiments, the actual computation time taken by the above two tasks is nearly 0. For the computation time of the node movement, it has been analyzed as $C_M = \frac{D_H}{M_S}$ (see Equation (7)). The values of the two factors D_H and M_S are strongly dependent on the hardware limitation of a MANET node. If the backup node can move as quickly as possible and the distance (radius) of one radio communication range is not large, the movement cost is very small. Based on the above description, the proposed approach takes more computation cost in the protection setup part. The computation complexity of the proposed approach is $O(m \times n)$.

As for the message complexity, each of the previous approaches needs to take $O(n^2)$ in the worst case. If one of the previous approaches cannot use its pre-established or dynamically found backup routes to perform route recovery, it finally dynamically finds a new backup route. In such this worst case, the flooding overhead is incurred. For the proposed approach, the message complexity is $O(m \times n)$ since one-hop extended hello message is broadcast to make each active node be protected by its neighboring nodes. However, as shown in Figure 9, the number of messages taken for route recovery in the previous approaches and proposed approach are less than $O(n^2)$ and $O(m \times n)$, respectively.

5. CONCLUSIONS

This paper has presented a new route recovery approach for a collaborative MANET. The proposed approach makes each active node use its neighboring nodes as backup nodes. Due to node mobility, the neighboring nodes of an active node also change frequently. To maintain up-to-date backup nodes, the proposed approach extends the hello mechanism of the AODV protocol. When a link breakage is detected on a route, the broken route can be locally repaired based on the neighboring node movements.

Extensive simulation experiments were also performed to make the detailed comparisons between the proposed approach and previous approaches. The simulation results show that the proposed approach has better performance in the normalized recovery routing overhead, recovery capability, and recovered route length.

REFERENCES

1. Internet engineering task force (IETF) mobile ad hoc networks (MANET) working group charter, Chaired by Joseph Macker and M. Scott Corson. Available at: www.ietf.org/html.charters/manet-charter.html
2. Koutsonikolas D, Das SM, Hu YC, Lu YH, Lee CSG. CoCoA: coordinated cooperative localization for mobile multi-robot ad hoc networks. *Proceedings of IEEE International Conference on Distributed Computing Systems Workshops*, 2006; 9–9.
3. Basu P, Redi J. Movement control algorithms for realization of fault-tolerant ad hoc robot networks. *IEEE Network* 2004; **18**(4): 36–44.
4. Wang Z, Zhou MC, Ansari N. Ad-hoc robot wireless communication. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 2003; 4045–4050.
5. Perkins C, Belding-Royer E, Das S. Ad hoc on-demand distance vector (AODV) routing. *IETF RFC 3561*, July 2003.
6. Lee SJ, Gerla M. AODV-BR: backup routing in ad hoc networks. *IEEE Wireless Communications and Networking Conference*, September 2000; 1311–1316.
7. Costa LHMK, Amorim MDD, Fdida S. Reducing latency and overhead of route repair with controlled flooding. *ACM Wireless Networks* 2004; **10**(4): 347–358.
8. Tauchi M, Ideguchi T, Okuda T. Ad-hoc routing protocol avoiding route breaks based on AODV. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, January 2005; 322a–322a.
9. Tang S, Zhang B. A robust AODV protocol with local update. *The Joint Conference of the 10th Asia-Pacific Conference on Communications and the 5th International Symposium on Multi-dimensional Mobile Communications*, August 2004; 418–422.
10. Perkins CE, Bhagwat P. Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers. *ACM SIGCOMM Conference on Communications, Architectures, Protocols*, and *Applications*, October 1994; 234–244.
11. Chiang C-C, Wu H-K, Liu W, Gerla M. Routing in clustered multihop mobile wireless networks with fading channel. *Proceedings of IEEE Singapore International Conference on Networks*, April 1997; 197–211.

DOI: 10.1002/wcm

12. Johnson D, Maltz D. Dynamic source routing in ad hoc wireless networks. In Mobile Computing, Vol. 353. Kluwer Academic Publishers: Norwell, MA, 1996; 153–181.
13. Marandin D. Performance evaluation of failed link detection in mobile ad hoc networks. *The 3rd Annual Mediterranean Ad hoc Networking Workshop*, S10.3, June 2004; 398–404.
14. Perkins CE, Royer EM, Das SR, Marina MK. Performance comparison of two on-demand routing protocols for ad hoc networks. *IEEE Personal Communications* 2001; **8**(1): 16–28.
15. Blazevic L, Le Boudec J-Y, Giordano S. A location-based routing method for mobile ad hoc networks. *IEEE Transactions on Mobile Computing* 2005; **4**(2): 97–110.
16. Ko Y-B, Vaidya NH. Location-aided routing (LAR) in mobile ad hoc networks. *ACM/Baltzer Wireless Networks (WINET) Journal* 2000; **6**(4): 307–321.
17. Aron ID, Gupta SKS. Analytical comparison of local and end-to-end error recovery in reactive routing protocols for mobile ad hoc networks. *Proceedings of the 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, August 2000; 69–76.
18. Duggirala R, Gupta R, Zeng QA, Agrawal DP. Performance enhancements of ad hoc networks with localized route repair. *IEEE Transactions on Computers* 2003; **52**(7): 854–861.
19. UCB/LBNL/VINT Network Simulator Version 2, ns-2. Available at: www-mash.cs.berkeley.edu/ns
20. Zhao Y, Chen Y, Li B, Zhang Q. Hop ID: a virtual coordinate-based routing for sparse mobile ad hoc networks. *IEEE Transactions on Mobile Computing* 2007; **6**(9).
21. Broch J, Maltz DA, Johnson DB, Hu Y-C, Jetcheva J. A Performance comparison of multi-hop wireless ad hoc network routing protocols. *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, October 1998; 85–97.
22. Chen C, Chen Z. Exploiting contact spatial dependency for opportunistic message forwarding. *IEEE Transactions on Mobile Computing* 2009; **8**(10): 1397–1411.

AUTHORS' BIOGRAPHIES

Jenn-Wei Lin received the M.S. degree in computer and information science from National Chiao Tung University, Hsinchu, Taiwan, in 1993, and the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1999. He is currently an Associate Professor in the Department of Computer Science and Information Engineering, Fu Jen Catholic University, Taiwan. He was a researcher at Chunghwa Telecom Co., Ltd, Taoyuan, Taiwan from 1993 to 2001. His current research interests are fault-tolerant computing, mobile computing and networks, distributed systems, and broadband networks.

Yi-Ting Chen received the M.S. degree in computer and information science from Fu Jen Catholic University, Taiwan, in 2007. He is currently a Software Engineer in AboCom Systems Inc., Taiwan. His research interests include mobile networks and fault-tolerant computing.