

Hierarchical management protocol for constructing a QoS communication path in wireless Ad Hoc networks

Chih-Yung Chang ^{a,*}, Chao-Tsun Chang ^b, Tzung-Shi Chen ^c, Hsu-Ruey Chang ^a

^a *Department of Computer Science and Information Engineering, Tamkang University, Taiwan*

^b *Department of Information Management, Hsiuping Institute of Technology, Taiwan*

^c *Department of Information and Learning Technology, National University of Tainan, Taiwan*

Received 19 January 2005; received in revised form 12 September 2006; accepted 20 December 2006

Abstract

An Ad Hoc network consists of mobile hosts that can dynamically construct a wireless network without base stations. Due to the limited communication range, a source host usually needs other hosts to relay messages to the destination in a multi-hop manner. Consequently, establishing a routing path from the source to the destination is a basic requirement for providing communication service between any pair of mobile hosts. This study proposes a two-level management approach for efficiently constructing and maintaining a QoS routing path in Ad Hoc wireless networks, significantly reducing the quantity of control packets. In the first phase, the mobile hosts are partitioned into a number of complete graphs, each represented by a Supernode managed by an agent. The Ad Hoc network topology is thus transformed to an *Agent-based Graph (AG)*. In the second phase, some agents of a larger degree than neighboring agents are selected as core nodes. The core nodes then virtually construct a *Core Graph (CG)*. The proposed two-level hierarchical management and bandwidth-looking-ahead technologies can efficiently establish and maintain a QoS communication path at a low control packet cost. Simulation results indicate that the proposed management model significantly reduces the number of control packets in areas with very large numbers of mobile hosts.

© 2007 Elsevier Inc. All rights reserved.

Keywords: Clique; Agent-based Graph; Supernode; Core node; QoS; Ad Hoc network

1. Introduction

The progress of technology means that consumers increasingly demand high communication quality, enabling them to communicate with others at any time and place. A Personal Communication System provides one-hop communication in which the base station plays a prominent role in the communication of mobile hosts. However, base stations may not be set up in places due to high cost, low utilization or inadequate performance. Moreover, base stations are difficult to set up and easily destroyed in situations such as

* Corresponding author. Tel.: +886 2 26215656x2768; fax: +886 2 26209749.

E-mail addresses: cychang@cs.tku.edu.tw, cychang@mail.tku.edu.tw (C.-Y. Chang).

war or natural disaster. A MANET (mobile Ad Hoc wireless network) consists of low-cost mobile hosts, enabling mobile users to communicate with each other without the need for fixed base stations or access points.

In contrast to static networks, Ad Hoc networks have no infrastructure. Mobile hosts are selected to construct a multi-hop communication path from source to destination, acting as a router for relaying information from one neighbor to another. User mobility means that routing protocols adopted in wired networks cannot be adopted in MANET. Various routing protocols [1–4] have been developed recently to provide and enable communication among mobile users. These protocols can be categorized into various classes, based on the following criteria: whether the protocol is *on-demand* or *table-driven*; whether it adopts hierarchical management and whether the mobile hosts are equipped with GPS.

In on-demand routing protocols [5–13], each host maintains the next-hop routing information. The routing path from source to destination can be quickly established hop by hop as soon as the source host issues a communication request. Therefore, a mobile host can construct the shortest routing path from itself to any other host. However, the MANET topology changes frequently, forcing the mobile hosts to update their routing information frequently. Thus, routing tables are very expensive to maintain. Unlike table-driven protocols, most on-demand routing protocols dynamically construct a routing path from flooding operations when a host requests to communicate with other hosts. Constructing a route on demand takes a long time, since hosts do not maintain the whole route information, but it saves on the traffic and memory costs involved in maintaining the routing information. This study examines the advantages of the table-driven and on-demand techniques. A two-level management model is proposed to efficiently construct a QoS communication path with a low flooding overhead and small cost at maintaining table.

Many researchers [14–17] have previously presented hierarchical management models for reducing the number of control packets during path construction by partitioning MANET into several clusters. The host belonging to two clusters acts as a gateway to relay messages from one cluster to another. A manager is selected in each cluster to manage its hosts. When a source host intends to communicate with a destination host in another cluster, the source host asks its manager to construct a routing path. Only managers and gateways can participate in the broadcasting operation to relay the search packet during path construction. The advantage of hierarchical management is that it can significantly reduce the number of control packets, while maintaining the routing information in a mobile environment.

Since the host is mobile and the bandwidth resource is limited, constructing a QoS route for communication from one host to another in Ad Hoc networks is a significant challenge. Related studies [7,8,18–23] on the QoS problem have been proposed in recent years. Chen and Nahrstedt [8] built a QoS routing path with ‘tickets’. The DSDV mechanism should be adopted to collect the bandwidth information of the whole MANET such that each host can estimate the successful probability of the quality of each path from itself to any other destination. The tickets are then divided into several sets, each of which is assigned to an estimated QoS path. The host mobility increases the cost of using DSDV to accurately determine a possible QoS routing path. Refs. [7,18] propose core-based management systems to reduce the number of control packets during QoS path construction. In their management protocol, CEDAR, each host compares degree with its neighbors. The host with the largest degree among its neighbors is selected as the core node. The core node is responsible for recording all information for its members. When a source host requests to establish a QoS routing path to some destination host, its core node must construct a QoS routing path. The advantage of core-based management is that it reduces the number of control packets during path construction. However, single-level management performs poorly in densely populated areas, because too many cores participate in the flooding operations. The virtual grid architecture protocol (VGAP) [24] proposes a hierarchical management model that provides end-to-end statistical QoS guarantees by using location information and operating on a fixed virtual grid. The proposed hierarchical management model achieves good performance in terms of packet delivery ratio and end-to-end packet delay while expensive GPS equipment is required to obtain the location information.

This study proposes a two-level hierarchical management model without extra equipment. First, a protocol is developed for each host to efficiently determine a maximal k -complete graph, called a k -clique, containing itself. A host with a larger value of k than its neighboring hosts is then considered as an agent for managing other hosts located in its k -clique. By adopting agents to manage hosts in the same clique, the original

MANET topology is thus transferred to an *Agent-based Graph (AG)*. Next, in the second-level management, the agent with the largest degree of its neighboring agents becomes the core, that is, the manager of the neighboring agents. The *Core Graph (CG)* is then virtually constructed by only collecting core nodes. The *AG* reduces the number of control packets, looks ahead at the quality of link bandwidth and efficiently reconstructs a broken route. Hereafter, the *CG* is used to avoid flooding operations. Under the proposed two-level management protocols, a QoS communication path can be constructed and maintained efficiently in a high-population area. Compared to the single-level core-based management [7,18], experimental results indicate that the proposed two-level management has the following significant improvements. First, two-level management saves many control packets since the core node in the second-level management does not flood the search packets to hosts whose communication links do not satisfy the QoS bandwidth requirements. Second, when the mobility of some relaying hosts breaks the current path, the agent can efficiently select another host from its k -clique to reconstruct the QoS communication path efficiently.

The rest of this study is organized as follows: Section 2 defines k -cliques, and proposes a method of partitioning the MANET topology efficiently into an *AG*. Section 3 then presents rules for determining the core node from the *AG* and constructing the *CG*. Next, Section 4 discusses the QoS routing protocol developed from bandwidth-looking-ahead technology and two-level management. Section 5 describes the maintenance of hierarchical topologies and routes. Section 6 presents the performance study. Conclusions are finally drawn in Section 7.

2. Construction of an Agent-based Graph

This section discusses how to transform the MANET topology to an *Agent-based Graph (AG)*. The protocol initially partitions the mobile hosts greedily into a number of cliques. Each clique is then considered as a Supernode, and the Ad Hoc network topology is transformed to an *AG*. The k -clique structure is defined as follows. Each host estimates the k value of its k -clique structure.

Definition (Ad Hoc Communication Graph). An Ad Hoc Communication Graph $G = (V, E)$ represents MANET topology where V denotes the set of all mobile hosts and E denotes the set of links. There is a link between two hosts if they can communicate with each other directly.

Definition (k -clique; C_k). Given an Ad Hoc Communication Graph $G = (V, E)$, a k -complete graph C_k consists of k hosts, each of which can communicate with the other $k - 1$ hosts in a one-hop distance. The maximal k -complete graph is called a k -clique. The k -clique constructed by host h , denoted by hC_k , is a complete graph with the maximal value of k compared with all complete graphs constructed by h 's one-hop neighboring host.

Next, consider how to transform MANET topology into *AG*. Each host first performs the developed k -clique estimation process. The MANET then is partitioned into several subMANETs according to the estimated hC_k of each host h . Each subMANET is denoted as a Supernode S_h or hC_k for some value k . In a Supernode, a mobile host is selected as a *routing agent* for serving the route request issued by other hosts. Thus, in *AG*, a Supernode is labeled with the name of its routing agent. Two Supernodes S_x and S_y are linked if a member in S_x can communicate with a member in S_y within one-hop distance. In Fig. 1a, hosts c, d and e form a 3-clique

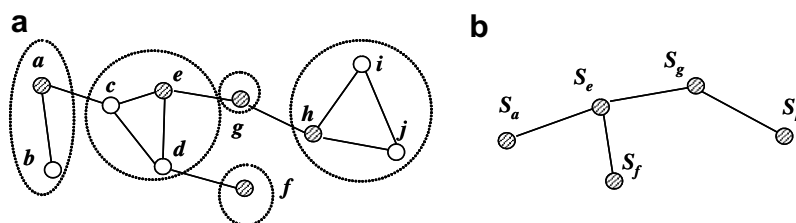


Fig. 1. The transformation from the MANET topology to *AG*. (a) An example of MANET topology. Hosts grouped in a circle construct a clique. (b) The corresponding *AG* of (a).

in MANET topology, and host e is selected as the Supernode agent. Thus, these three hosts are represented as a Supernode S_e , as shown in Fig. 1b. Since hosts a and c are linked in Fig. 1a, Supernodes S_a and S_e are linked in AG as shown in Fig. 1b.

Since the MANET topology changes frequently, the k -clique must be determined quickly. The distributed construction protocol for transforming MANET topology to AG is presented as follows. The AG construction protocol mainly comprises two phases. In the first phase, a k -clique estimation process is designed for each host h to estimate the k -clique hC_k efficiently. In the second phase, mobile hosts partition MANET into a number of separate subMANETs. A greedy algorithm is proposed to partition the MANET into a number of subMANETs according to the hC_k of each host h . The MANET topology can be reduced to an AG by considering each subMANET as a Supernode and determining a routing agent as the manager for each Supernode.

2.1. Process for determining a k -clique

This subsection describes the process used by each host h to determine the k value of its hC_k . Since the k -clique problem is an NP-complete problem, this paper proposes a heuristic mechanism for each node to determine the k value of its hC_k . Let r denote the communication radius of a mobile host. An obvious method for host h to construct a clique is to use $r/2$ as its communication radius by using power control to collect all the neighbors located within its transmission range. However, in a highly populated area, this protocol causes too many hosts to transmit messages simultaneously, resulting in packet collision and contention. The following protocol is proposed to reduce the collision and contention. Let $C_{a,r/2}$ represent the set of mobile hosts that are one-hop neighbors of host a with a signal strength of $r/2$. Let $|C|$ denote the number of elements in a set C . Since the maximum distance of any two hosts in $C_{a,r/4}$ is $r/4 + r/4 = r/2$, the mobile hosts in $C_{a,r/4}$ can constitute a $|C_{a,r/4}|$ -clique, while the mobile hosts in $C_{a,r/2}$ can constitute at most a $|C_{a,r/2}|$ -clique.

According to Fig. 2a, as host a derives the set of $C_{a,r/4}$, it sends a message containing the set of $C_{a,r/4}$ to its neighbors with a signal strength of $r/4$. After each host (denoted as x) contained in $C_{a,r/4}$ receives the message, it then replies to host a the set of $C_{x,r/4}$. Consequently, the value k of aC_k can be determined from $C_{a,r/4}$ and those sets of $C_{x,r/4}$, for all x in $C_{a,r/4}$. As shown in Fig. 2a, the set $C_{a,r/4}$ is $\{a, b, c, d, e, f\}$. Host a sends the set of $C_{a,r/4}$ by using signal strength $r/4$. Hosts b, c, d, e and f perform the same operations as host a at this point. Host a thus receives sets $C_{b,r/4}$, $C_{c,r/4}$, $C_{d,r/4}$, $C_{e,r/4}$ and $C_{f,r/4}$. As shown in Fig. 2b, all mobile hosts located in the gray area, denoted as A , approach the set of $C_{a,r/2}$ and the $C_{a,r/2}$ is evaluated by

$$C_{a,r/2} = \bigcup_{x \in C_{a,r/4}} C_{x,r/4}.$$

Notably, the area A approaches the circle area drawn with host a as the center point and $r/2$ as the radius, implying that if two hosts are both precisely on the circumference of $C_{a,r/4}$ in an opposite direction, then the maximum distance between them is $r/4 + r/4 = r/2$. If two hosts, say x and y , are both precisely on the circumference of $C_{a,r/4}$ in an opposite direction, then the maximal distance of two hosts respectively belonging to $C_{x,r/4}$ and $C_{y,r/4}$ is given by $(r/2 + r/4 + r/4) = r$, indicating that any pair of mobile hosts in $C_{a,r/2}$ can communicate with each other, and that mobile hosts in $C_{a,r/2}$ can constitute at most a $|C_{a,r/2}|$ -clique. Moreover, the

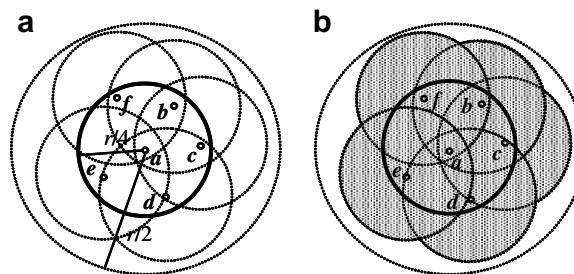


Fig. 2. Determining k -clique. (a) Host a measures the set $C_{a,r/4}$ by $r/4$ signal strength. (b) Mobile hosts in the gray circle area could constitute at least a $|C_{a,r/2}|$ -clique.

proposed protocol reduces the packet collision area and the contention of control packets, thus enhancing the network performance.

The protocol for transforming MANET topology to several disjoint subMANETs, and determining a routing agent in each, is introduced below. Some control packets used in the protocol are defined below to illustrate the protocol processes.

Message type	Description
$CSIZE(a,s)$	Host a informs the other hosts its clique size $s = C_{a,r/2} $.
$JMGR(a)$	Host a informs the other hosts that it will compete with the manager (or agent) of the clique.
$JMBR(a,i)$	Host a informs the host i that it wants to be a member of host i .

2.2. MANET partitioning and Agent-based Graph constructing

In the first phase, each host can estimate the k value of its k -clique. However, if MANET is partitioned according to the k -clique derived by each host, then a host may appear in several partitioned subMANETs. A protocol is proposed to partition MANET greedily into several disjoint subMANETs. The protocol partitions the MANET to minimize the number of partitioned cliques, and thus maximize the size of each partitioned clique. The partitioning protocol is described in detail as follows.

First, each host, denoted as x , sends a $CSIZE(a, |C_{x,r/2}|)$ message with the signal strength of $r/2$, and receives the same type of messages from neighboring hosts. Host x then sorts the received $|C_{i,r/2}|$, where i is in $C_{x,r/2}$, in decreasing order. If host x is the first position in the sorted list, then it can construct a k -clique as a Supernode, and act as the routing agent of this Supernode by broadcasting a $JMGR(x)$ to tell other hosts in $C_{x,r/2}$ that it is the agent of the $|C_{x,r/2}|$ -clique. If host x is not in the first position, then it should wait to receive other messages sent by the hosts ranked before it in the list. Assume that Pre_x denotes the set of hosts in $C_{x,r/2}$ that have the larger k value of k -clique than $|C_{x,r/2}|$. If host x receives a $JMGR(i)$ message sent from host $i \in Pre_x$, then host i intends to be the agent. Host x then sends a $JMBR(x,i)$ message to inform host i that it intends to be the member of host i . However, if host x receives a $JMBR(i,t)$ message, for some $t \neq a$, then i intends to be a member of host t , and host x removes host i from its sorted list. This is because host i is a member of another agent and cannot be the agent of $C_{x,r/2}$. After performing the remove operation, if host x appears at the first position of the sorted list, then it broadcasts a $JMGR(x)$ message to construct the $|C_{x,r/2}|$ -clique by taking host x as the center point. If the MANET has a size of $L \times L$, then the time complexity for executing the two tasks, partitioning MANET into cliques and determining the agent in each clique, is $O(L^2/r^2)$ in the worst case. The time complexity in the worst case is derived by

$$O\left(\frac{A_{L \times L}}{A_{C_{x,r/4}}}\right) = O\left(\frac{16L^2}{\pi r^2}\right) = O\left(\frac{L^2}{r^2}\right).$$

Consequently, the MANET topology can be fully transformed into AG and can be completed.

The proposed partition protocol is illustrated here by an example. As shown in Fig. 3, host a ranks the received values, $|C_{b,r/2}|$, $|C_{c,r/2}|$, $|C_{d,r/2}|$, $|C_{e,r/2}|$ and $|C_{f,r/2}|$ sent respectively by hosts b , c , d , e and f . These hosts also perform the same operations as host a . Fig. 3 shows the results of applying this protocol. The Pre_a set of host a is $\{b, c\}$, meaning that the best candidate agent of host a could be host b or c . Thus, host a waits for hosts b and c to send the $JMBR$ or $JMGR$ messages. Two possible cases are discussed below. As mentioned before, we discuss situations as the following:

Host a	b	c	a	d	e	f		
Host b	d	b	a					
Host c	j	c	a	h				
Host d	a	d	b					
Host e	a	e	f					
Host f	a	e	f	k				

Fig. 3. Sorted list maintained by every host.

Case 1. Hosts b and c respectively send messages $JMBR(b, a)$ and $JMBR(c, a)$ to host a , which then removes b and c from its sorted list. Host a is then in the first position, and therefore sends a $JMGR(a)$ message to ask hosts d, e and f to form a 4-clique, whose central point and agent is host a .

Case 2. Host j sends the $JMBR(j, c)$ message, and host c removes j from its sorted list. Host c then sends a $JMGR(c)$ message to ask hosts a and h to form a 3-clique. On receiving the $JMGR(c)$ message, host a sends a $JMBR(a, c)$ message to participate in the clique managed by host c .

The algorithm that transforms MANET to AG is illustrated as follows.

Algorithm (Transformation from MANET topology to Agent-based Graph (AG)).

- Step 1: Each host x uses $1/4$ power as its signal strength to estimate the set $C_{x,r/2}$.
- Step 2: Each host x sends $CSIZE(x, |C_{x,r/2}|)$ message to inform its neighbouring hosts in $C_{x,r/2}$ by the signal range of $r/2$.
- Step 3: Each host x sorts the received values $|C_{i,r/2}|$, where host i is in $C_{x,r/2}$, in decreasing order. Assume that Pre_x denotes the set of hosts with larger k values of k -clique than $|C_{x,r/2}|$. Host x performs Steps 4–6 repeatedly until it determines its agent.
- Step 4: If host x is at the first position of its sorted list (i.e. $Pre_x = \phi$), then host x sends a $JMGR(x)$ message and constructs a clique managed by itself.
- Step 5: If host x receives the $JMGR(i)$ message, $i \in Pre_x$, then host x sends a $JMBR(x, i)$ message to participate in the clique managed by host i .
- Step 6: If host x receives a $JMBR(i, *)$ message, then for any $i \in Pre_x$, host h removes i from its sorted list.
- Step 7: If host x is an agent, then it labels its clique as a Supernode and can compete to be a core node.

Theorem 1. The proposed protocol partitions a given MANET topology into a number of separate subMANETs and determines a routing agent for each subMANET. The maximum distance between two agents is 3 hops.

Proof. After executing the proposed algorithm, each host either selects an agent from its neighbors or is selected as an agent. Hence, the maximum distance between two agents is 3 hops, as shown in Fig. 4a. Consequently, the case shown in Fig. 4b is impossible. $\square$

After transforming the MANET topology to AG , the set of mobile hosts in i hops distance from a host is defined, and the QoS routing information stored in agents are then considered.

Definition. $N_i(x)$ represents the set of mobile hosts whose distance from Supernode S_x is not greater than i hops.

Each agent x stores the bandwidth of all links between hosts in $N_1(x)$. Since both control packets and data messages (such as data or beacon packet) can be sent over MANET, the agent and core can be elected passively [25] without creating additional control overhead. ‘Passive’ transmission exploits the neighborhood information carried by data packets. For instance, the data packets exchanged among neighbors hosts can piggyback the information required for agent or core selection, significantly reducing the control overhead

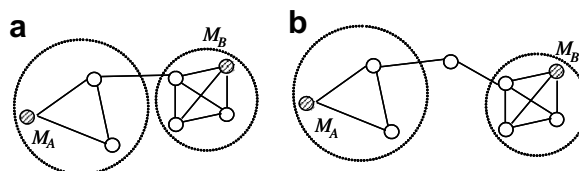


Fig. 4. The hop-relation between agents M_A and M_B . (a) There are at most 3 hops between any two agents. (b) Impossible case.

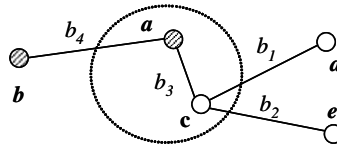


Fig. 5. Each agent records the bandwidth of its member and hosts with one-hop distance far from its member.

Table 1
Agent-to-Member table of agent *a*

Member <i>x</i>	$N_1(x)$	Bandwidth
<i>a</i>	<i>c</i>	b_3
<i>a</i>	<i>b</i>	b_4
<i>c</i>	<i>d</i>	b_1
<i>c</i>	<i>e</i>	b_2

for selection or maintenance of the agent or core. By this method, regardless of the routing protocol, the proposed hierarchical topology can be constructed as a by-product of the user data packet exchange.

In the proposed protocol, the agent maintains QoS routing information for a Supernode and looks ahead at the quality of the link bandwidth with a one-hop distance from its members. The evaluation of link bandwidth depends on the MAC protocol applied by the system. The bandwidth in a TDMA system is measured by the number of free slots in two hosts. The bandwidth of a link connecting hosts *x* and *y* is measured according to the following formula:

$$\text{Bandwidth}(\overline{xy}) = \frac{\text{The number of common free slots}}{\text{The total number of slots in a superframe}} \times \text{Transmission rate}.$$

For instance, assume that the transmission rate of 802.11 is 11 Mbps and a superframe consists of 100 slots. Assume that hosts *x* and *y* have 25 free common slots in the next superframe. Then the bandwidth of hosts *x* and *y* is

$$\text{Bandwidth}(\overline{xy}) = \frac{25 \times 11 \text{ Mbps}}{100} = 2.75 \text{ Mbps}.$$

Similarly, in a CDMA system, the bandwidth of each node depends on the number of independent codes reserved for the host pairs. The protocol developed herein belongs to a network layer. We simply consider the model that a link of each pair of hosts has a bandwidth, regardless of which MAC system is adopted. As shown in Fig. 5, hosts *a* and *c* belong to the same Supernode S_a , and host *a* is the agent of host *c*. Therefore, host *a* has to record the bandwidth of links *ac*, *ab*, *cd* and *ce* into its Agent-to-Member table as shown in Table 1. The bandwidths of links *ac*, *ab*, *cd* and *ce* are assumed to be b_3 , b_4 , b_1 , and b_2 , respectively.

This section describes the protocol transforming the MANET to AG. The next section discusses how to transform AG to CG for second-level management.

3. Construction of a Core-based Graph

Sivakumar [7,16] adopted a core to manage mobile hosts for reducing the number of control packets during QoS route construction. This idea is extended herein to hierarchical management constructed with agent-level and core-level structures. This section describes the construction of the core-level structure, CG.

In the constructed AG, each Supernode, labeled with the host ID of agent, represents a clique. The agent *v* chooses a node in $N_2(v)$ as its core node, which is denoted by $\text{core}(v)$. By applying the following algorithm, the core node can be selected from its member agents, and the bandwidth of paths between the core node and its member agent and between core nodes can be constructed. Some control messages are defined below to illustrate the CG construction process.

Message type	Description
$CSTATUS(a,b,c,d)$	Status of agent a for competing the core, where b denotes the number of a 's neighbors, c denotes the number of a 's neighbors who have chosen a as their core, and d denotes the core node of a
$SETC(a,b)$	Agent a sets agent b to its core
$SETP(a,b,c,path_traversed)$	Agent a sets a path, $path_traversed$, with a constrained length c in order to make core node b know the path to himself
$SETP(a, '*', c, path_traversed)$	Core node a sets a path, $path_traversed$, with a constrained length c in order to make neighboring core nodes know the path to himself
$CNFP(a,b,path)$	Host a confirms the $path$ from b to a

Algorithm (Transformation from Agent-based Graph (AG) to Core Graph (CG)).

- Step 1:* The degree $d(v)$ denotes the number of v 's neighbors, and the core degree $d^*(v)$ represents the number of v 's neighbors that have chosen v as their core nodes.
- Step 2:* Each agent v periodically broadcasts a $CSTATUS(v, d(v), d^*(v), core(v))$ message, where the value of $core(v)$ is NULL if none of v 's neighbors has chosen v as the core node.
- Step 3:* Agent v does not initially have a core. If agent v finds $d^*(v) > 0$, then it becomes the core itself. Otherwise, the following steps are used to identify the core node and the paths between agent and core, and between two cores.
- (1) Agent v elects an agent u in $N_2(v)$ where the maximum value of u is $(d(u), d^*(u))$. Consequently, the node with a larger degree has a greater opportunity to be chosen as a core node. If two agents have the same degree, then the agent with the most agent members is selected as a core node.
 - (2) Agent v then sends a $SETC(v, u)$ message to agent u to accept its management of agent u . This increases the value $d^*(u)$ stored in agent u .
 - (3) After agent v joins its core node u , it broadcasts a $SETP(v, u, 3, '')$ message to core node u . The $path_traversed$ denotes a path from agent v to core node u .
 - (4) Core node u broadcasts a $SETP(u, '*', 6, '')$ message to its neighboring core nodes in a six-hop distance. The $path_traversed$ denotes a path into core node u .
 - (5) When host w or agent w receives the $SETP(x, '*', i, path_traversed)$ message, it broadcasts a $SETP(x, '*', i - 1, path_traversed + 'w')$ message to host t in $N_{i-1}(u)$ if $i - 1 > 0$. If $i = 0$, then the $SETP$ message will be discarded.
 - (6) When core node w receives the $SETP(x, w, i, path_traversed)$ message, it sends a $CNFP(w, x, reverse(path_traversed + 'w'))$ message back to agent x to confirm the path. The reversed order of a path P is denoted as $reverse(P)$.
 - (7) When core node w receives the $SETP(x, '*', i, path_traversed)$ message, it sends a $CNFP(w, x, reverse(path_traversed + 'w'))$ message back to core node x to confirm the path, where the reversed order of a path P is represented as $reverse(P)$.

Each core node knows the paths for communicating with its neighboring core nodes or member agents. Fig. 6 shows an example of transformation from AG to CG by applying the algorithm. The data structures, namely Core-to-Core table stored in each core and Core-to-Member table, adopted in the proposed protocol and stored in each agent and member, are explained below. The Core-to-Core table has three fields: *To*, *Passed Agent* and *Bandwidth*. Table 2 describes the Core-to-Core table of core node Cr_a , shown in Fig. 6a which has two neighboring core nodes, Cr_g and Cr_i , virtually connected to core Cr_a . Thus, the Core-to-Core table of core node Cr_a has two rows. The first row of Table 2 records that core node Cr_a can route to core node Cr_g passing through the agent S_j . The value B_1 in the first row of Table 2 denotes the bandwidth resource for the virtual

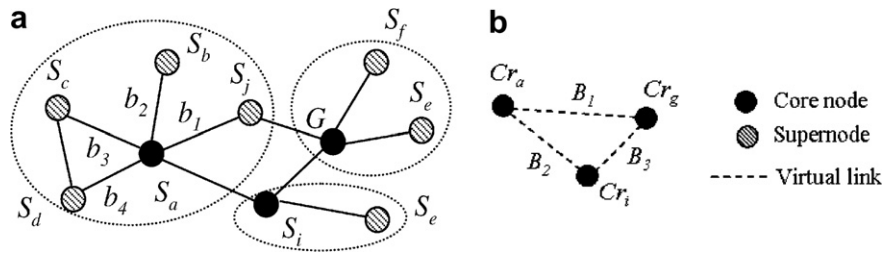


Fig. 6. An example of transforming *AG* to *CG*. (a) An example of *AG*. (b) An *CG* corresponding to *AG* in (a).

Table 2

Core-to-Core table of core node Cr_a

To	Passed agent	Bandwidth
Cr_g	j	B_1
Cr_i	–	B_2

link (Cr_a, Cr_g) . Similarly, the second row of Table 2 records that core node Cr_a can directly route to core node i without passing through another agent. The *Bandwidth* field of the Core-to-Core table is used to reduce the flooding operations performed for a QoS route discovery. As the core receives a route request packet indicating the required bandwidth, the core checks the bandwidth field of the Core-to-Core table to determine whether to flood the packet to the neighboring agents, according to whether the recorded bandwidth meets the requirement.

The Core-to-Member table records the agents managed by a core and the bandwidth of the paths between a core node and its member agents. Table 3 shows core a 's Core-to-Member table corresponding to the *AG* shown in Fig. 6a. As a core receives a route request packet, it forwards the packet to those managed agents whose bandwidth recorded in the Core-to-Member table meets the required bandwidth indicated in the packet. If the destination host is managed by one of its agents, then the core creates a route reply packet and sends it to the source host, thus constructing the QoS route. The next section describes the QoS routing protocol for the proposed hierarchical management system.

In the proposed hierarchical management system, the MANET topology is transformed to *AG* by the efficient clique construction protocol. Then, the *AG* is transformed into *CG* so that the agents with a larger degree than their neighboring agents are selected as core nodes. The link between two agents is a virtual link in *AG*. The bandwidth of a path is defined by the minimum bandwidth of the links on the path. The bandwidth of a virtual link connecting two agents is defined by the maximal bandwidth of paths between them. The bandwidth of the virtual link between two cores in *CG* is determined in the same way.

An example of how to determine the bandwidth of a virtual link is given below. The dotted circle in Fig. 7a denotes a Supernode. Assume that a and e denote the agents of the two Supernodes. Fig. 7b shows the transformed *AG*, which has two paths from a to e , path $a-b-e$ and path $a-c-b-e$. The bandwidth values of links $a-b$ and $b-e$ in path $a-b-e$ are 4 and 6, respectively. The bandwidth values of path $a-b-e$ and path $a-c-b-e$ are 4 and 5, calculated by $\min(4, 6)$ and $\min(5, 6)$, respectively. The maximum bandwidth value of the two paths is

Table 3

Core-to-Member table of core node Cr_a

Agent	Bandwidth
b	b_2
c	b_3
d	b_4
j	b_1

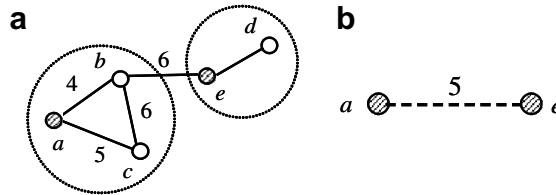


Fig. 7. Bandwidth calculation in *AG*. (a) An example used to illustrate the bandwidth calculation. (b) Bandwidth of the virtual link connecting agents *A* and *E* is 5.

then selected as the bandwidth value of the virtual link connecting those two agents in *AG*. Thus, the bandwidth of virtual link between connecting agents *a* and *e* in *AG* is computed as

$$\max(\text{bandwidth}(\text{path } a-b-e), \text{bandwidth}(\text{path } a-c-b-e)) = \max(\min(4, 6), \min(5, 6)) = 5$$

The bandwidth values in the Agent-to-Member, Core-to-Member, and Core-to-Core tables are calculated similarly.

4. QoS routing

This section presents a routing protocol for establishing a QoS route that satisfies the bandwidth requirements for each link on the path. A source host *s* is assumed to issue a route request to establish a multi-hop path, with required bandwidth *b*, to a destination host *d*. The agent adopts a basic function *shortcut(R)* to shorten path *R*. If two different members of the agent appear in path *R*, and the link bandwidth value between them is larger than *b*, then the agent connects the two different members to shorten the path *R*.

To illustrate the process of the QoS routing protocol, some control messages are defined as follows:

Message type	Description
$RREQ(s, d, b, \text{path_traversed})$	A route request message is sent for discovering a QoS path from <i>s</i> to <i>d</i> , with a required bandwidth value larger than <i>b</i> . The <i>path_traversed</i> denotes the path which the message traversed
$RREP(d, s, \text{path})$	Route reply for a <i>path</i> from <i>d</i> to <i>s</i>
$RERR(s, d)$	Routing error for discovering a QoS path from <i>s</i> to <i>d</i>
$NFND(s, d)$	Agent member <i>s</i> has not found host <i>d</i> recorded in the table or the bandwidth of a path from <i>s</i> to <i>d</i> is not larger than the required value. Agent <i>s</i> sends a <i>NFND(s, d)</i> message to inform its core node

The routing protocols for establishing a QoS routing path in *AG* and *CG* are described as follows:

Algorithm (*The QoS routing protocol*).

- Step 1:** Host *s* checks whether the bandwidth value of the path from *s* to its agent *m* is larger than *b*. If yes, then host *s* sends a $RREQ(s, d, b, 's')$ message to its agent *m*. Otherwise, host *s* stops the route discovery process.
- Step 2:** When agent *m* receives the $RREQ(s, d, b, p)$ message, it checks whether the host *d* exists in its Agent-to-Member table.
- (1) If agent *m* finds host *d* in its table, and the bandwidth of a path from *m* to *d* is larger than *b*, then it creates a route according to the table, and sending the $RREP(s, d, \text{reverse}(\text{shortcut}(p + 'md')))$ message, back to host *s*, and completes the route discovery.

- (2) If agent m does not find host d in the table but finds that the bandwidth of the path from m to its core node u is larger than b , then it relays the $RREQ(s, d, b, \text{shortcut}(p + 'm'))$ message to u and executes Step (3).
- (3) If agent m does not find host d in the table, and the bandwidth of the path from m to its core node u is smaller than b , it sends the $RERR(s, d)$ message back to host s and stops the route discovery.

Step 3: When core node u receives the $RREQ(s, d, b, p)$ message, it checks its Core-to-Core and Core-to-Member tables. The Core-to-Core table records u 's neighboring cores, given by w , the passed agent from core u to core w and the path bandwidth. The Core-to-Member table records the agents managed by core node u .

- (1) Core node u checks its Core-to-Member table and then sends a $RREQ(s, d, b, \text{shortcut}(p + 'u'))$ message to its agent members if the bandwidth values of the path from u to them are larger than b . On receiving the $RREQ(s, d, b, r)$ message, the agent members check their Agent-to-Member tables. If the agent member does not find host d in the table, or the bandwidth of the path from it to d is smaller than b , then it sends a $NFND(s, d)$ message to inform Core node u . If the agent member finds that host d is recorded in the table, and the bandwidth of a path from it to d is larger than b , then it establishes a route q from it to d according to the table, and replies the $RREP(s, d, \text{reverse}(\text{shortcut}(r + q)))$ message back to host s , and thus completes the route discovery.
- (2) If core node u finds that host d exists in its Core-to-Member table and the bandwidth of path from u to d is larger than b , then it establishes a route according to the table, and sends the $RREP(s, d, \text{reverse}(\text{shortcut}(p + 'ud')))$ message back to host s , thus completing the route discovery.
- (3) If core node u does not find host d in its Core-to-Member table, but finds that the bandwidth of the path from u to core node w is larger than b , then it relays the $RREQ(s, d, b, \text{shortcut}(p + 'u'))$ message to w and executes *Step 3* again.
- (4) If core node u does not find host d in its Core-to-Member table, and the bandwidths of path from u to other core nodes are smaller than b , then it sends the $RERR(s, d)$ message back to host s and stops the route discovery.

In the proposed QoS routing protocol, each agent maintains an Agent-to-Member table, and each core node maintains Core-to-Core and Core-to-Member tables. The two tables stored in each core node are used to establish a QoS route by a bandwidth-looking-ahead technique, significantly reducing the number of control packets. *AG* and *CG* are adopted to construct the QoS route. Thus, if the route is broken due to the mobility of a particular host, then the k -clique property enables the agent to immediately assign another host in the same clique to replace the host that has left. Moreover, the proposed two-level management, core and agent, can significantly reduce the number of control packets.

Fig. 8 shows an example of how to establish a QoS routing path by applying the proposed two-level management system. Fig. 8a shows the topology of an example MANET, and Fig. 8b shows the corresponding Supernodes. Fig. 8c illustrates the *AG* derived from Fig. 8b. Fig. 8d depicts the cores and their managed agents, and Fig. 8e displays the resultant *CG* transformed from *AG*. Assume that source host c creates a route request to destination host m for discovering a QoS communication path with a required bandwidth value 5. The operations are as stated below.

- (1) Initially, source host c finds that the bandwidth value of the path from c to its agent a is larger than 5, as shown in Fig. 8a. Therefore, host c sends an $RREQ(c, m, 5, 'c')$ message to its agent a .
- (2) On receiving the $RREQ(c, m, 5, 'c')$ message, agent a checks its Agent-to-Member table and does not find the host m recorded in it, as shown in Fig. 8b. Since agent a is its own core, it does not send the $RREQ(c, m, 5, 'ca')$ message to itself.
- (3) Core node a checks its Core-to-Member table, then sends a $RREQ(c, m, 5, 'ca')$ message to its agent member r since the bandwidth value of the path from a to r is larger than 5, as shown in Table 4. After receiving the $RREQ$ message, the agent r checks its Agent-to-Member table, but does not find host m

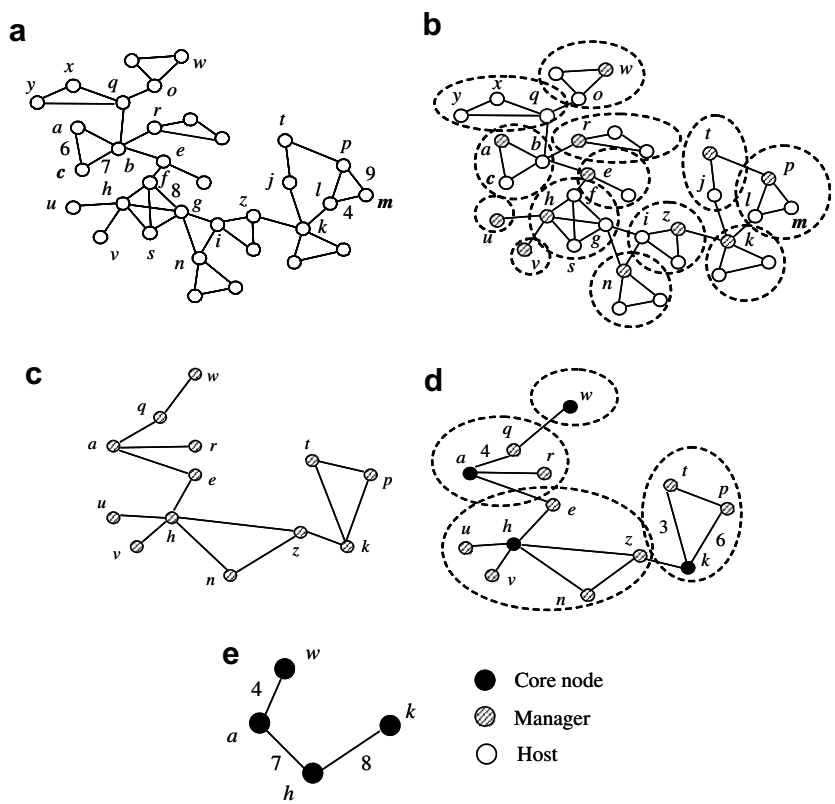


Fig. 8. An example MANET and its AG and CG. (a) Hosts and some bandwidth values of links in the topology of example MANET. (b) Hosts in the same Supernode are circled. (c) The AG of (a). (d) The agents in a circle are managed by the same core, and some bandwidth values of virtual links are represented in AG of (a). (e) CG corresponding to AG and some bandwidth values of virtual links are represented in CG of (a).

Table 4
Core-to-Member table of core node Cr_a

Agent	Bandwidth
q	4
r	8

Table 5
Core-to-Core table of core node Cr_a

To	Passed agent	Bandwidth
Cr_w	q	4
Cr_h	e	7

recorded in the table, as shown in Fig. 8b. Agent r immediately sends a *NOTFOUND* message $NFND(c,m)$ to agent a .

- (4) When core node a receives the $NFOUND(c,m)$ message, it checks the Core-to-Core table in Table 5, and finds that the bandwidth value of the path from a to h is larger than 5. Core a then relays the $RREQ(c,m,5, 'ca')$ message to the bridge agent e .

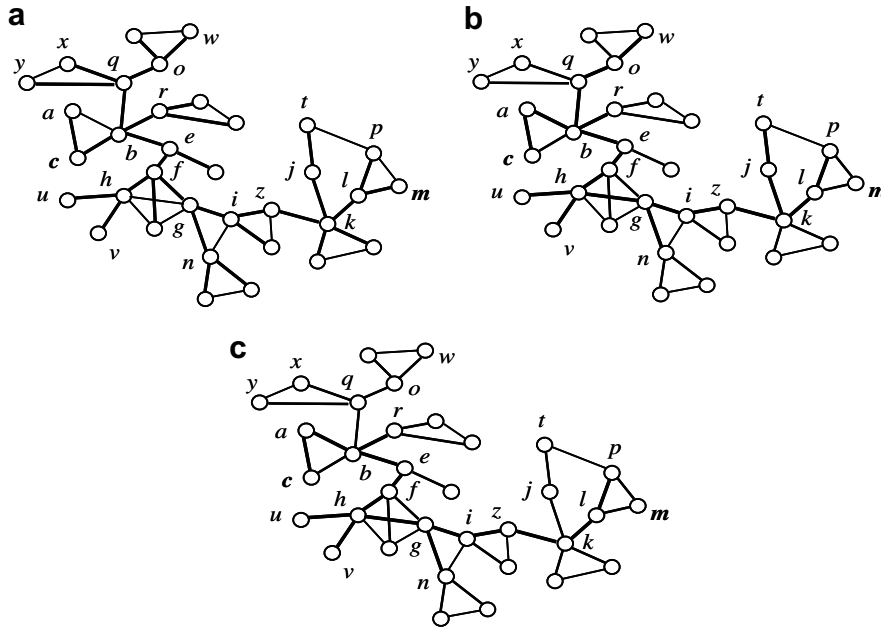


Fig. 9. Comparison of the blind flooding mechanism and the proposed mechanism in terms of control overhead for discovering a QoS route, from c to m in the example topology of Fig. 8a. (a) The routing protocol by applying the blind flooding results the large number of route request messages for discovering a QoS route, from c to m in the example topology of Fig. 8a. (b) The routing protocol by applying the two-level management reduces a number of control packets compared with (a). (c) The routing protocol by applying two-level management and bandwidth-looking-ahead mechanisms further reduces the number of control packets compared with (b).

- (5) By executing the operations described in Steps (3) and (4), as shown in Fig. 8e, core node e receives a $RREQ(c, m, 5, \text{shortcut}('cab'))$ message and then relays the $RREQ(c, m, 5, 'cbe')$ message to the core node h . Core node h receives a $RREQ(c, m, 5, 'cbef')$ message and then relays the $RREQ(c, m, 5, \text{shortcut}('cbefhg'))$ message to the bridge agent z . Core node z then receives an $RREQ(c, m, 5, 'cbefgi')$ message and then relays the $RREQ(c, m, 5, 'cbefgiz')$ message to the core node k . Core node k then checks its Core-to-Member table and (sends OR transmits) an $RREQ(c, m, 5, 'cbefgizk')$ message to its agent member p , since the bandwidth values of the paths from k to p are greater than 5, as shown in Fig. 8d. After receiving the $RREQ(c, m, 5, 'cbefgizkl')$ message, the agent p checks its Agent-to-Member table, and finds that host m is recorded in the table, and that the bandwidth of the path from it to m is larger than 5, as indicated in Fig. 8a. Agent p therefore establishes a route shortcut ('cbefgizklpm') from c to m , and then replies with a $RREP(c, d, 'mplkzigfebc')$ message back to host c , thus completing the route discovery.

The proposed routing protocol can significantly reduce the number of control packets raised from flooding operations. Consider the MANET topology shown in Fig. 8a. Fig. 9a and b show the control packets required to construct a QoS route by applying blind flooding and the proposed two-level management protocol, respectively. The bold lines in Fig. 9 denote the link passed through by control packets during the route discovery. Applying both bandwidth-looking-ahead technologies and two-level management, the proposed routing protocol significantly reduces the number of control messages, as shown in Fig. 9c. Specifically, applying the two-level management and both the two-level management and bandwidth-looking-ahead mechanisms reduce the number of control packets by 41% and 56%, respectively, in comparison with the blind flooding mechanism.

5. Maintenance protocols

This section presents the maintenance protocol including the route maintenance, structure maintenance and route information update. Due to the host mobility in MANET, the route maintenance and the structure

maintenance are critical for any QoS routing protocol when the existing path becomes broken. This section, first introduces the strategies for handling the route and structure maintenance, and then discusses the update of bandwidth information for the Agent-to-Member, Core-to-Member and Core-to-Core tables.

5.1. Route and structure maintenance

MANET hosts in very densely populated areas are assumed to have slow mobility. If r denotes the radio communication range, then a link is broken if its distance is larger than r . A link is said to be *weak* if its distance is larger than $4r/5$. The clique structure is redefined when a member host moves out $r/2$ from its agent. Since the MANET is partitioned into several cliques, the agents of cliques should be responsible for maintaining the backup path for any weak link and the clique structure. This section first introduces the route maintenance then the structure maintenance. Some control messages are defined below to illustrate the route and structure maintenance.

Message type	Description
$LEAVE(x, y, flag)$	Host x informs its agent y that it will leave clique $C_{y, r/2}$. The values ‘busy’ and ‘free’ of the $flag$ respectively denote x is serving a route or not, respectively
$JOIN(x, y)$	Host x informs its agent y that it will join clique $C_{y, r/2}$ again
$BACKUP(x, y)$	Hosts x and y require their agent to replace the weak link xy by a backup path
$ESCAPE(h, a)$	Agent h informs host a that host a is not its member

As a link xy served in a route becomes weak in a clique, the end host of the weak link that is closer to source sends a $BACKUP(x, y)$ message to request its agent to replace the link by a backup path. A k -clique has $k - 1$ links between pairs of members. Therefore, $k - 2$ candidate links can be used to back up a weak link in a k -clique. If $k < 2$, or the agent cannot prepare a backup path to replace the weak link, then the route fault occurs and the route discovery is reinitiated. Fig. 8 shows an example of this procedure. The proposed routing protocol constructs a QoS route path: $c-b-e-f-g-i-z-k-l-p-m$, as shown in Fig. 8a. Assume that host g is moving away from its peer communicating host f , and that link fg becomes weak as shown in Fig. 10a. Host f sends a $BACKUP(f, g)$ message to require its agent h to replace the weak link by a backup path. On receiving the message, agent h selects a QoS path from paths: $f-h-g$ or $f-s-g$ to backup the weak link fg rapidly, as shown in Fig. 10b.

The structural maintenance of the proposed protocols is considered next. As a member host x moves out $r/2$ from its agent y , the member host sends a $LEAVE$ message to inform the agent that it will leave. The message has a $flag$ to indicate whether it serves any route. On receiving the message, if the member host is serving any existing routes and is in $5r/6$ distance from its agent, then the agent neither updates the managed structure nor assigns any work until the member host finishes the data relaying task. Otherwise, the agent initiates the structural maintenance process. If the structural maintenance is initiated but the member host

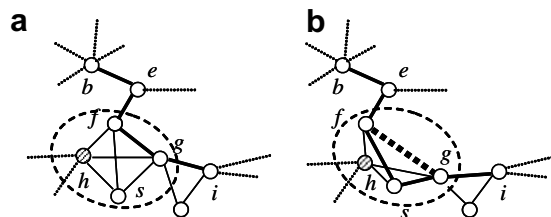


Fig. 10. An example of route maintenance, taken the QoS routing illustrated in Fig. 8 as an example. (a) The bold line denotes the constructed QoS route from c to m in the example topology of Fig. 8a. The clique agent h manages hosts f , g and s . (b) The agent h selects the QoS path, path: $f-s-g$ to rapidly backup the weak link fg .

is serving for existing routes, then route fault occurs and another route discovery is initiated. Conversely, when host x moves in $r/2$ from agent y , it sends a $JOIN(x,y)$ message to inform its agent to join clique $C_{y,r/2}$.

When a host leaves a clique, it completes the data relaying service for the QoS route, and sends a $LEAVE$ message with a *flag* to indicate that it is free. The agent initiates the structure maintenance process on receiving the message. The agent h removes host a from its Agent-to-Member table and sends an $ESCAPE(h,a)$ message back to host a . After receiving the message, host a broadcasts a $JMBR(a,*)$ message at a signal strength of $r/2$ power to invite its neighboring agent to collect it. The neighboring agents that receive the message then send a $CSIZE$ message back to host a . If agent h has the largest $|C_{h,r/2}|$, then host a sends a $JMBR(a,h)$ message to be a member of agent h . On receiving the $JMBR(a,h)$ message, agent h sends a $JMGR(h)$ message to confirm that host a is a member of $C_{h,r/2}$. Host a then sets host h as its agent, and finishes the structural maintenance process.

5.2. Bandwidth information update

In the proposed QoS routing protocol, the agent records the bandwidth of links in $N_1(v)$ in Agent-to-Member table and the bandwidth of virtual link from itself to its core node. In addition, the core node records the bandwidth of virtual links between the core node and each of its members in the Core-to-Member table and the bandwidth between neighboring core nodes in Core-to-Core table. In the previous section, we have illustrated how to determine the bandwidth of a virtual link using the examples of Fig. 7. The following introduces how to update the bandwidth information of the three tables.

In the proposed protocols, every host periodically sends to its agent the bandwidth of links that connects to neighboring agents. Every agent records the bandwidth of all members in its clique in Agent-to-Member table. In addition, every agent and core node periodically sends a bandwidth update message in each of virtual links between the core node and its each member agent. The bandwidth update message has a field to record the minimal bandwidth of the virtual link. When the agent or the core node receives the bandwidth update message, it records the minimal bandwidth of the virtual link in its Core-to-Member table. In addition, every core node periodically sends a bandwidth update message in each of virtual links between the core nodes. When the core node receives the bandwidth update message, it records the minimal bandwidth of the virtual link in the Core-to-Core table. In developing a QoS route maintenance protocol, the bandwidth information update usually raises a large number of control packets to collect the bandwidth information of each link. To reduce the maintenance overheads, the passive maintenance mechanism proposed in [25] can be adopted herein.

6. Performance study

The system performance, in terms of the number of agents and cores, the system traffic overhead, the route maintenance overhead, route failure ratio, the size of routing cache and the average time of route discovery, was evaluated using simulation. The simulation environment was as follows. The MANET was $1500 \times 1500 \text{ m}^2$ and had with 250–2000 hosts. The radio range was set to 100 and 200 m. Mobile hosts were randomly placed initially, and moved in random directions at random speeds of 0–4 m/s. The source and destination of each route were randomly selected from the hosts in MANET. The bandwidth requirement for each route was randomly determined from 1 to 10, while the full bandwidth of each host was assumed as 30. Each performance result was obtained from the average results of 100 experiments. Herein, the system traffic overheads were evaluated from the number of route request packets for QoS route construction, and the number of control packets for route-recovering was adopted to measure the route maintenance overheads.

The number of agents (called cores in CEDAR) generally rises with the number of hosts in both the proposed protocols and CEDAR, as shown in Figs. 11 and 12. Fig. 11 compares the number of agents and cores in the proposed protocols by varying the radio range between 100 and 200 m. The number of agents or cores generally fell with the radio range. A small radio range produces many small cliques. As shown in Fig. 11, the number of cores constructed in the proposed protocols is far smaller than the number of agents. Since core

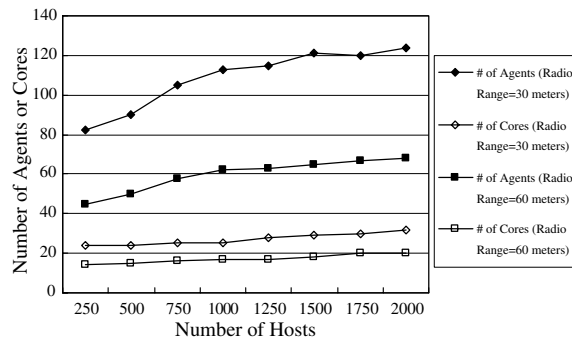


Fig. 11. Comparison of number of agents and cores with different number of hosts in the proposed protocols.

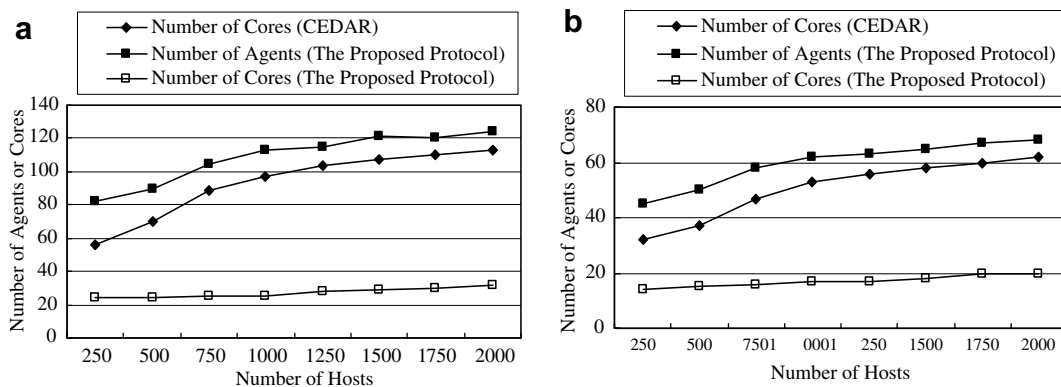


Fig. 12. Comparisons of the proposed protocol and CEDAR in the number of agents and cores with different radio ranges. (a) Radio range = 100 m. (b) Radio range = 200 m.

nodes perform bandwidth-look-ahead to reduce the number of control packets during QoS route construction, the core nodes can be considered as the main source of control overhead, which also indicates that the proposed mechanism has significantly better control of overheads than CEDAR.

Fig. 12 compares the number of agents or cores in MANET under the proposed two-level management and CEDAR. Fig. 12a and b evaluate the number of agent and cores at radio ranges of 100 and 200 m, respectively. The number of agents or cores falls as the radio range of the host rises under both of the proposed two-level management and CEDAR. Fig. 12 compares the number of agents or cores in MANET under the proposed two-level management and CEDAR, because a host with a large radio range creates a big cluster, thus reducing the number of clusters. Hence, a large radio range helps to reduce the number of route request packets flooded in MANET. In comparison, a host is much easier than an agent to invite as a member in the core of CEDAR in the proposed protocol, so the core contains more members than the constructed clique. Fig. 12a and b indicate this effect. The number of cliques in the proposed protocol is larger than the number of cores in CEDAR. The number of cores is proportional to the number of hosts in both protocols, but the number of cores constructed in the proposed protocols is far smaller than the number of cores in CEDAR, and the core of the proposed protocols are considered as the main source of control overheads. As shown in Fig. 12a and b, CEDAR creates a large number of cores to perform route discovery, and thus floods more route request packets over MANET than does the proposed protocol.

Fig. 13 compares the traffic overhead for QoS route construction in the proposed method with CEDAR. The simulation randomly generates a pair of source and destination hosts and the level of bandwidth requirement ranging from 1 to 10, and then measures the number of route request packets required OR needed in

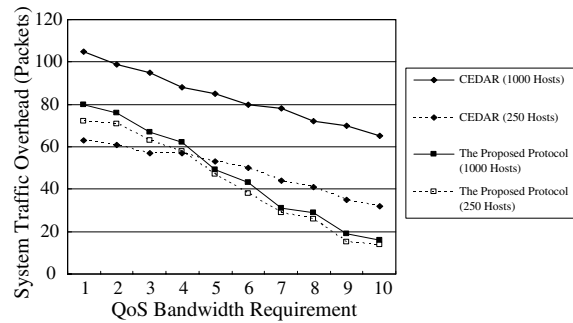


Fig. 13. Comparison of system traffic overhead with different QoS bandwidth requirements.

MANET. As shown in Fig. 13, the proposed two-level management generally has lower traffic overhead than CEDAR, mainly because the proposed protocol applies bandwidth-looking-ahead techniques to protect the route request packets from ineffective flooding. Additionally, the proposed two-level management has fewer agents and cores participating in the flooding operation. The number of route request packets rises with the number of cores participating in the flooding operations in both compared protocols. As shown in Fig. 12a and b, the increasing number of hosts in CEDAR causes the system traffic overheads to increase with the number of cores, but the proposed protocols have stable traffic overheads while increasing the number of hosts and cores are increased. When the number of hosts is 250 or 1000, and the bandwidth requirement is smaller than 4, the two-level management has higher traffic overheads than CEDAR because the control packets sent between cores, and between cores and agents, result in ineffective bandwidth-looking-ahead. Hence, the proposed mechanism shows significant improvement against CEDAR as regards the system traffic overheads in a densely populated area.

The constructed route may fail due to node mobility. A good maintenance protocol may replace the weak link in time to prevent the route from failing. Fig. 14 compares the route failure ratio of the proposed protocols and CEDAR with a radio range of 100 m. The route failure ratio is the ratio of the number of failure route to the number of routes with weak links. When a weak link leads to route recovery, a smaller route failure ratio means more opportunity to find the backup path and successfully replace the weak link. As shown in Fig. 14, increasing the number of hosts in MANET simultaneously increases the number of member hosts managed by the core in CEDAR and the k value of each k -clique, thus increasing the opportunity to find the backup path and decreasing the route failure ratio. Since the number of cores in CEDAR is smaller than the number of agents as shown in Fig. 12a and b, the number of member hosts managed by each core in

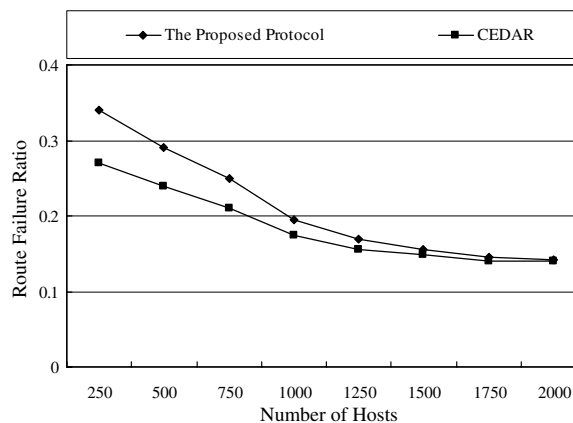


Fig. 14. Comparison of route failure ratio with radio range 100 m.

CEDAR is larger than the k value of each k -clique. However, in a very densely populated area, the existence of adequate opportunities to find the backup path does not imply that the weak links are fully recovered, because a weak link between two cliques or between two member hosts belonging to different cores in CEDAR cannot be recovered by the local route, meaning that the route discovery must be initiated at the source host. As shown in Fig. 14a and b, the proposed two-level management and CEDAR have similar route failure ratios while the radio range of the host rises, which indicates that the proposed mechanism and CEDAR have the same route failure ratio in a highly densely populated area.

Fig. 15 compares the cost of route maintenance of two-level management and CEDAR at a radio range of 100 m. When a constructed route fails due to node mobility, the route maintenance protocol generally replaces the weak link locally in time to repair the failed route. If no local route can be used as the backup path, then the source is requested to initiate the route discovery again. The compared route maintenance protocols create a small number of control packets for backup route reconstruction. However, the number of cores constructed in the proposed protocols is far smaller than that in CEDAR, so CEDAR has greater route maintenance overhead than the proposed protocols, as shown in Fig. 15. This finding indicates that the proposed two-level management has lower traffic overheads for route discovery and also less overheads for route maintenance costs than CEDAR.

To discover the QoS routing path, the proposed protocols and CEDAR record the bandwidth of links and routing information into the caches of agents and cores. Fig. 16a and b show the size of cache required by the

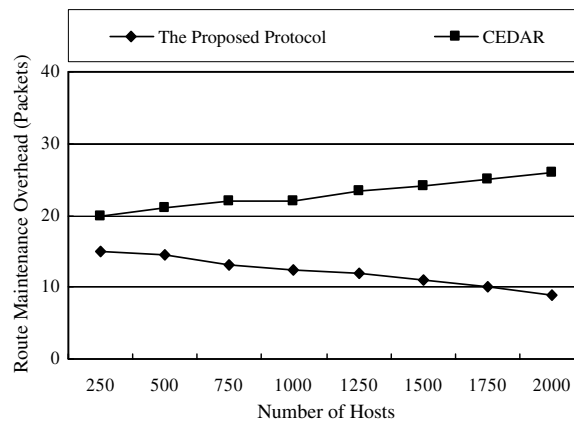


Fig. 15. Comparison of route maintenance overhead with radio range 100 m.

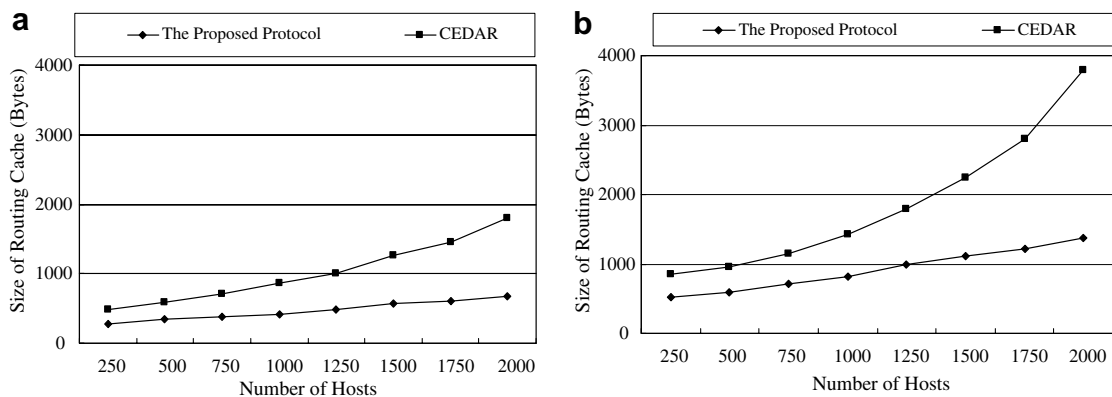


Fig. 16. Comparison of the size of cache with different radio ranges. (a) Radio range = 100 m. (b) Radio range = 200 m.

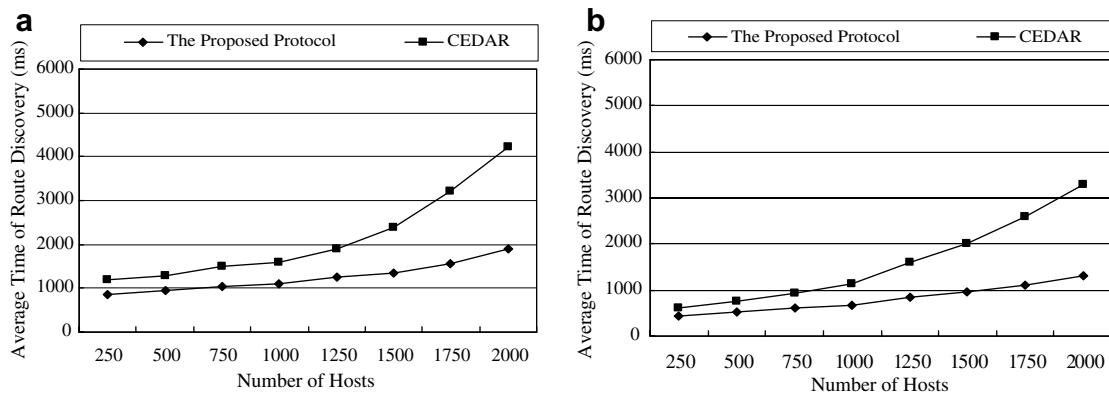


Fig. 17. Comparison of the discovery time with different radio ranges. (a) Radio range = 100 m. (b) Radio range = 200 m.

considered protocols at different radio ranges. Increasing the number of hosts in MANET increases the number of member hosts managed by the core in CEDAR above the k value of each k -clique. A core in CEDAR caches its local topology. Specifically, a core in CEDAR caches information of a set of its member hosts, given by V . The cached information includes the node ID, state, bandwidth information of links between nodes in V , and the paths (or virtual links) between neighboring cores. Most agents in the proposed protocols simply record the Agent-to-Member table and the bandwidth of virtual link from itself to its core. As compared with CEDAR, the proposed protocol requires a smaller cache size for each agent. The average cache requirement of agents and cores is small since the number of cores in the proposed management is far smaller than the number of agents. The average cache requirement of agents and cores in the proposed mechanism is smaller than that of CEDAR, as shown in Fig. 16a. Since the number of agents or cores falls as the radio range increases, the routing information cached in the agents or cores increases as the number of hosts increases. Fig. 16a and b also indicate that the proposed protocol has a smaller cache requirement than CEDAR, because the CEDAR cores cache much more routing information than in the agents in the proposed protocol.

Fig. 17a and b compare the route discovery time of two-level management and CEDAR with different radio ranges. As shown in Fig. 12a and b, the number of cores constructed in the proposed protocols is far smaller than that of CEDAR. Since CEDAR has too many cores participating in the flooding operation, it spends more time on route discovery than the proposed protocols. Moreover, the core of CEDAR caches much routing information and determines the QoS route by Dijkstra's single source shortest path algorithm, which means that the route discovery of CEDAR is more time-consuming than the proposed protocols, as shown in Fig. 17a. Increasing the radio range of the host reduces the number of cores participating in the flooding operations in both protocols. However, the core's cache information significantly increases in CEDAR as the radio range is enlarged, significantly increasing the route discovery time due to the execution of Dijkstra's algorithm. Fig. 17a and b indicate that the proposed mechanism has a smaller route discovery time than CEDAR, revealing that the proposed mechanism has significantly better QoS route discovery performance than CEDAR.

7. Conclusions

This study proposes a two-level management approach for efficiently constructing and maintaining a QoS routing path in Ad Hoc wireless networks. In the first phase, the proposed technique partitions the MANET into several k -cliques, and constructs an AG as first-level management. In the second phase, a Core-based mechanism is presented to transform AG to CG for second-level management. By applying the Agent-based and Core-based scheme to look ahead the required bandwidth, the proposed two-level management significantly reduces the number of route request packets flooding links with insufficient bandwidth for the QoS

requirement. The *AG* reduces the number of packets flooding by looking ahead to find the link bandwidth quality, and efficiently reconstructs a route when a link of the route is becoming weak. Hereafter, the *CG* further reduces the flooding overhead performed by improper agents. A QoS communication path can be constructed and maintained efficiently in a highly populated area in the proposed two-level management protocols. The performance analysis indicates that the proposed management protocols have significantly fewer control packets, lower route maintenance than CEDAR. In high density areas, CEDAR requires a large cache size to store the local topology information, and has a longer route discovery time than the proposed protocols, while the route failure ratio of the proposed protocols is close to that of CEDAR.

References

- [1] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, J. Jetcheva, A performance comparison of multi-hop wireless Ad Hoc network protocols, in: *Proc. of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Dallas, TX, USA, 1998, pp. 85–97.
- [2] S. Corson, A. Ephremides, A distributed routing algorithm for mobile wireless networks, *ACM/Baltzer Journal of Wireless Networks* 1 (1995) 61–81.
- [3] D.B. Johnson, D.A. Maltz, Protocols for adaptive wireless and mobile networking, *IEEE Personal Communications* 3 (1996) 34–42.
- [4] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers, *ACM SIGCOMM Computer Communication Review* 24 (1994) 234–244.
- [5] R. Dube, C.D. Rais, K.Y. Wang, S.K. Tripathi, Signal stability based adaptive routing (SSA) for Ad Hoc mobile networks, *IEEE Personal Communications* February (1997) 36–45.
- [6] D.B. Johnson, D.A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, Kluwer Academic, Dordrecht, The Netherlands, 1996.
- [7] R. Sivakumar, P. Sinha, V. Bharghavan, CEDAR: a core-extraction distributed Ad Hoc routing algorithm, *IEEE Journal on Selected Areas in Communications* 17 (1999) 1454–1465.
- [8] S. Chen, K. Nahrstedt, Distributed quality-of-service routing in Ad Hoc networks, *IEEE Journal on Selected Areas in Communications* 17 (1999) 1488–1505.
- [9] W.H. Liao, Y.C. Tseng, J.P. Sheu, GRID: a fully location-aware routing protocol for mobile Ad Hoc networks, *Telecommunication Systems* 18 (2001) 37–60.
- [10] C.Y. Chang, C.T. Chang, Hierarchical cellular-based management for mobile hosts in wireless Ad Hoc networks, *Computer Communications Journal* 24 (2001) 1554–1567.
- [11] S. Doshi, S. Bhandare, T.X. Brown, An on-demand minimum energy routing protocol for a wireless Ad Hoc network, *ACM SIGMOBILE Mobile Computing and Communications Review* 6 (2002) 50–66.
- [12] J.H. Song, W.S. Wong, C.M. Leung, Efficient on-demand routing for mobile Ad Hoc wireless access networks, *IEEE Journal on Selected Areas in Communications* 22 (2004) 1374–1383.
- [13] C.Y. Chang, S.C. Tu, Active route-maintenance protocol for signal-based communication path in Ad Hoc networks, *Journal of Network and Computer Applications* 25 (2002) 161–177.
- [14] M. Gerla, J.T. Tsai, Multicluseter, mobile, multimedia radio network, *ACM/Baltzer Journal of Wireless Networks* 1 (1995) 255–265.
- [15] W. Chen, N. Jain, S. Singh, ANMP: ad hoc network management protocol, *IEEE Journal on Selected Areas in Communications* 17 (1999) 1506–1531.
- [16] P. Sinha, R. Sivakumar, V. Bharghavan, Enhancing Ad Hoc routing with dynamic virtual infrastructures, in: *Proc. of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*, Anchorage, AK, USA, 2001, pp. 1763–1772.
- [17] J. Sucec, I. Marsic, Location management for hierarchically organized mobile Ad Hoc networks, in: *Proc. of the Wireless Communications and Networking Conference (WCNC)*, New York, USA, 2002, pp. 603–607.
- [18] C. Zhu, S. Corson, QoS routing for mobile Ad Hoc networks, in: *Proc. of the 21th Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*, New York, USA, 2002, pp. 958–967.
- [19] W.I. Kim, Y.J. Suh, S. An, A reliable route selection algorithm in mobile Ad-Hoc networks, *Journal of KISS: Information Networking* 29 (2002) 314–323.
- [20] S. Sha, K. Chen, K. Nahrstedt, Dynamic bandwidth management for single-hop Ad Hoc wireless networks, *ACM/Kluwer Mobile Networks and Applications (MONET) Journal* 10 (2005) 199–217.
- [21] C. Chaudet, I.G. Lassous, BRuIT: bandwidth reservation under interferences influence, in: *Proc. of the European Wireless (EW02)*, Saint-Emilion, France, 2002, pp. 466–472.
- [22] L. Georgiadis, P. Jacquet, B. Mans, Bandwidth reservation in multihop wireless networks: complexity and mechanisms, in: *Proc. of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, Hachioji, Tokyo, Japan, 2004, pp. 762–767.
- [23] Y.H. Wang, C.F. Chao, Dynamic backup routes routing protocol for mobile ad hoc networks, *Information Sciences* 176 (2006) 161–185.

- [24] J.N. Al-Karaki, A.E. Kamal, End-to-end support for statistical quality of service in heterogeneous mobile ad hoc networks, *Computer Communications Journal* 28 (2005) 2119–2132.
- [25] M. Gerla, T.J. Kwon, G. Pei, On demand routing in large Ad Hoc wireless networks with passive clustering, in: *Proc. of the Wireless Communications and Networking Conference (WCNC)*, Chicago, IL, USA, 2000, pp. 23–28.