PAPER *Special Section on Multi-dimensional Mobile Information Networks*

An Adaptive Multihop Clustering Scheme for Ad Hoc Networks with High Mobility*

Tomoyuki OHTA[†a)], **Shinji INOUE**[†], **Yoshiaki KAKUDA**[†], *and* **Kenji ISHIDA**[†], *Regular Members*

SUMMARY A clustering scheme for ad hoc networks is aimed at managing a number of mobile devices by utilizing hierarchical structure of the networks. In order to construct and maintain an effective hierarchical structure in ad hoc networks where mobile devices may move at high mobility, the following requirements must be satisfied. (1) The role of each mobile device for the hierarchical structure is adaptive to dynamic change of the topology of the ad hoc networks. The role of each mobile device should thus change autonomously based on local information in each mobile device. (2) The overhead for management of the hierarchical structure is small. The number of mobile devices in each cluster should thus be almost equivalent. This paper proposes an adaptive multihop clustering scheme for highly mobile ad hoc networks. The results obtained by extensive simulation experiments show that the proposed scheme does not depend on mobility and node degree of mobile devices in the network, which satisfy the above requirements.

key words: *ad hoc networks, clustering, autonomous decentralized systems, mobile computing*

1. Introduction

An ad hoc network is a collection of mobile devices with routing functions like routers in wired networks. When a mobile device communicates with another mobile device, data packets of the source mobile device are forwarded via other mobile devices. That is, mobile devices in the ad hoc network can communicate with each other without any aids of base stations like a cell phone system. Even if mobile devices move around in any directions and at any speeds in the network, they can communicate with each other if it is not away from the collection. This paper presents an adaptive clustering scheme for ad hoc networks, which divides a collection of mobile devices in the network into multiple clusters. A clustering scheme of ad hoc networks is aimed at managing a number of mobile devices by utilizing hierarchical structure of the network and at efficiently assigning network resources such as a frequency. Many researchers work in clustering schemes for ad hoc networks [4]–[6]. Based on them, many hierarchical routing protocols for ad hoc networks have been proposed so far [1]–[3], [7].

The representative hierarchical routing protocols for ad hoc networks rely on flooding. In case of flooding, the connectivity between each pair of a source mobile device and a destined mobile device can be kept regardless of the speed at which mobile devices are moving. However, the more the number of mobile devices is increasing, the more the number of control packets for flooding may be increasing exponentially. By introducing a clustering scheme, there is a possibility that the network can hold the connectivity with smaller control packets.

In case of conventional clustering schemes, each cluster has a clusterhead which manages the cluster, and the neighboring mobile devices which is reachable from the clusterhead by one hop number. The neighboring mobile device is referred to as a clustermember hereinafter. And if a clustermember has two or more clusterheads in the neighborhood, the clustermember works as gateway between two clusters.

The conventional clustering schemes have some problems as follows. Firstly, there are many possibilities that every time clusterheads change, clusters tend to be reconstructed frequently. This is because, when a mobile device moves out of the communication range of the clusterhead, the mobile device either becomes a new clusterhead or a clustermember of another clusterhead. As a result, if mobile devices move around at much higher speed, the clusters to which they belong would be reconfigured frequently.

Secondly, the overhead of each clusterhead depends on density of mobile devices in the network. Here, the density denotes how many clustermembers are included in a cluster. In case that the density of mobile devices in each cluster is much higher, the clusterhead of the cluster may have much overhead to manage all clustermembers of the cluster. Moreover, in case that the density of mobile devices is much lower, there are many small size clusters in the network. Therefore it is difficult to construct and maintain an effective hierarchical structure.

We have proposed the fundamental part of a clustering scheme for ad hoc networks [8]. In our scheme,

Manuscript received October 28, 2002.
Manuscript revised February 10, 2003.
Final manuscript received March 20, 2003.

[†]The authors are with the Faculty of Information Sciences, Hiroshima City University, Hiroshima-shi, 731-3194 Japan.
a) E-mail: ohta@ce.hiroshima-cu.ac.jp

*This work was supported in part by the Ministry of Education, Science, Sports and Culture of Japan under Grant-in-Aid for Scientific Research (C) (No.12650390) and Hiroshima City University under their research grants.

each cluster consists of a clusterhead and clustermembers which are reachable from the clusterhead by one or more hop (multihop) numbers. And also, we have proposed the hierarchical routing protocol called Hi-TORA based on the clustering scheme in ad hoc networks [11]. The routing within clusters and between clusters apply link state protocol and on-demand protocol called Temporally-Ordered Routing Algorithm (TORA [10]), respectively. In [11], we showed the experimental evaluation of Hi-TORA with respect to the number of control packets, accuracy of packet delivery and hop counts in comparison with TORA. Especially on the control packets, we obtained the result that the faster node moving speed is, the more effective Hi-TORA.

In order that effective hierarchical structure is constructed and maintained in highly mobile ad hoc networks, the following conditions should be satisfied.

1. Each mobile device changes its role such as clusterhead and gateway by receiving local information from its neighboring mobile devices. In other words, each mobile device autonomously acts as a member of a cluster.
2. The number of clustermembers which each clusterhead has to manage is bounded by constants. In other words, the overhead of each clusterhead is almost equivalent.

Due to these conditions and multihop property, the geographical region of each cluster is larger than that in the conventional scheme so that effective clustering can be realized even when mobile devices move at much faster speed.

In this paper, we will show a clustering scheme, especially on merger and division of clusters and show experimental evaluations of the proposed clustering scheme. Throughout the paper, mobile device is referred to node for convenience of description.

The rest of the paper is organized as follows. Section 2 explains key features of our proposed clustering scheme to satisfy the above conditions. Section 3 gives definitions to explain the proposed clustering scheme. Section 4 gives the proposed scheme in detail. The effectiveness of the proposed clustering scheme is shown by simulation experiments in Sect. 5. We discuss some related work in Sect. 6. Finally, Sect. 7 concludes this paper with future research.

2. Key Features

First of all, we describe the features of the proposed scheme.

Feature 1: Each node autonomously constitutes the cluster.

Feature 2: Each cluster autonomously maintains itself.

An ad hoc network is constituted with nodes moving around freely. Although a node discovers a route

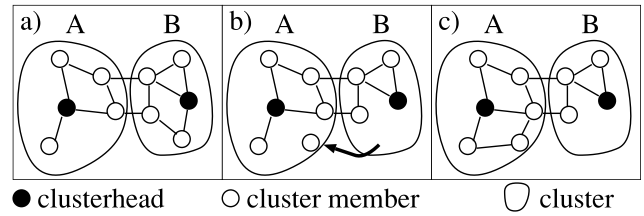

Fig. 1 Example of autonomous nodes.

to the destination node to forward the packets, it often occurs that the node cannot send the packets along the route because of nodes' movements. And also, in case of clustering, such a thing may occur frequently. The function of a cluster does not work well when some clustermembers leave from the cluster. However, to maintain the hierarchical structure in ad hoc networks, clusters need to be kept regardless of movement of nodes. The proposed scheme assures reconstructing of clusters.

Figure 1 provides an example which explains the Feature 1. Suppose that there are two clusters in the network as shown in Fig. 1(a). Let the left cluster be cluster A and the right cluster be cluster B, respectively. Suppose that one of clustermembers in cluster B moves in cluster A as shown in Fig. 1(b). As a result, the network topology changes as shown in Fig. 1(c). Although the clustermember left from cluster B, cluster B works regularly as if the clustermember did not exist in cluster B from the start. Similarly, the clustermember which moved from cluster B into cluster A works regularly as if it has joined in cluster A from the start.

Next, we explain the Feature 2. In the proposed scheme, the number of clustermembers must be kept a specified range. If the number of clustermembers in a cluster is less than the lower bound, the cluster needs to merge with one of the neighboring clusters. In order to merge two clusters into one cluster, a clusterhead always has to get the cluster size of all neighboring clusters. It prevents that the number of clustermembers in the merged cluster is over the upper bound. On the contrary, if the number of clustermembers in a cluster is greater than the upper bound, the cluster is divided into two clusters.

The actions and information of a clustermember are as follows intuitively.

Action 1: A clustermember periodically exchanges information with its neighboring clustermembers.

Action 2: A clustermember periodically broadcasts information to other clustermembers in the same cluster.

A clustermember can get information which the neighboring clustermembers have because of Action 1. Moreover, a clustermember can share information with other clustermembers in the same cluster because of Action 2.

Figure 2 provides an example which explains the cluster merger to describe the second feature. Suppose

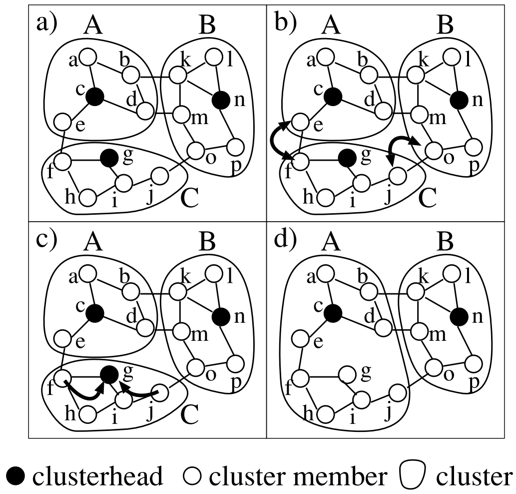

Fig. 2 Example of autonomous clusters.

that there are cluster A, cluster B and cluster C in the network as shown in Fig. 2(a). Cluster A has gateway b, d and e, cluster B has gateway k, m and o and cluster C has gateway f and j. Now, suppose that the number of clustermembers in cluster C is less than the lower bound. Then, cluster C tries to merge with cluster A or cluster B in the neighboring clusters. Clusterhead g in cluster C has to get the number of clustermembers of all neighboring clusters to merge with another cluster. We describe how clusterhead g gets them from cluster A and cluster B.

First of all, a clustermember can share the information with other clustermembers in the same cluster by Action 2. Therefore, gateway e in cluster A has the number of clustermembers in cluster A. As shown in Fig. 2(b), gateway f in cluster C exchanges the information with gateway e in cluster A so that gateway f can get the number of clustermembers in cluster A. After that, as shown in Fig. 2(c), gateway f broadcasts the information which it has in cluster A by Action 2 so that clusterhead g in cluster C can get the number of clustermembers in cluster A.

Through these procedures, clusterhead g can also get the number of clustermembers in cluster B from gateway j. Based on the number of clutermembers in cluster A and cluster B, clusterhead g in cluster C selects either cluster A or cluster B and merges with one of them. As a result, cluster C is merged with cluster A because the number of clustermembers in cluster A is small than that in Cluster B (see Fig. 2(c), and cluster C disappeared from the network as shown in Fig. 2(d)).

3. Preliminary

3.1 Ad Hoc Networks

Let the ad hoc network be denoted by an undirected graph $G = (V, E)$. A node $v_i \in V$ represents a mobile device with node ID i and it is simply called node v_i afterwards. An edge $(v_i, v_j) \in E$ represents a wireless link between nodes i, j and then node v_j is called the neighboring node of node v_i.

3.2 Structure of Cluster

A cluster C_i is a subclass of V. C_i consists of clusterhead v_i and some clustermembers. We define a node which has a cluster ID as a clustermember. A cluster satisfies the following two conditions. One is that all clustermembers must be connected within the cluster and its clustermembers have an identical cluster ID. The other is that the number of clustermembers ($|C_i|$) which the clusterhead has is bounded by the two constants, the upper bound (U) and lower bound (L).

3.3 Structure of Node

A node has an unique node ID, a cluster ID and a state. If a node is a clustermember, the node has all clustermember ID's. A node performs the following two actions as follows.

Action 1: A node exchanges the information with the neighboring nodes periodically by using *hello packet*.

Actoin 2: If a node is a clustermember, the node broadcasts its information to all clustermembers in the some cluster periodically by using control packets called *periodic notification packet*.

We define a node which manages and belongs to a cluster as a *clusterhead*. A clusterhead manages the number of clustermembers in the cluster. If the number of clustermembers which a clusterhead has is less than L, the clusterhead tries to merge with one of neighboring clusters. Here, the clusterhead collects the number of clustermembers of the neighboring clusters through gateways as mentioned in key feature 2 in Sect. 2. On the contrary, if the number of clustermembers which a clusterhead has is greater than U, the clusterhead tries to divide the cluster which the clusterhead manages into two clusters. We will explain the cluster division and merger in Sect. 4.4. The cluster division and merger of the proposed clustering scheme have been evaluated by theoretical considerations in [9].

In this paper, we define a node which has a neighboring node with different cluster ID's as a *gateway*. A clusterhead can communicate with the neighboring cluster through the gateways.

4. Adaptive Multihop Clustering Scheme

Each node holds an unique node ID, a cluster ID to which the node belongs, and all node ID's in the cluster which the node currently recognizes. In the hierarchical structure to be used in this paper, there are a clusterhead, gateways and clustermembers in each cluster. Each node also holds a state representing the role of the node, that is, clusterhead, gateway and clustermember. The cluster ID and the state which each node holds are changed according to node movement in the ad hoc network. In this section we will present an autonomous clustering scheme to maintain the hierarchical structure in the network by changing the cluster ID and the state which each node holds whenever nodes move.

4.1 States and Actions

First, we provide the definitions of states of each node and explain the actions which each node performs in a state.

Five types of states are defined as follows. Node v_i holds one of five states and performs the following actions in relation to the state.

NSN (Normal State Node)

Node v_i with state NSN is a clustermember which is neither any clusterhead nor any gateway. Node v_i periodically broadcasts control packets including node ID i and cluster ID j ($j \neq i$). v_i holds all nodes' IDs in the cluster which node i currently recognizes. When nodes with cluster ID j receive such control packets, they recognize that node v_i belongs to the cluster with cluster ID j and update information the nodes have.

CN (Control Node)

Node v_i with state CN plays the role of a clusterhead. Node v_i periodically broadcasts control packets including node ID i and cluster ID j ($j = i$). v_i holds all nodes' IDs in the cluster which node v_i currently recognizes. When nodes with cluster ID i receive such control packets, they recognize that node v_i is currently a clusterhead in the cluster with cluster ID i and update information they have.

BN (Border Node)

Node v_i with state BN plays the role of a gateway. Node v_i periodically broadcasts control packets in the same way as shown in NSN. Suppose that node v_i belongs to an cluster with cluster ID j ($j \neq i$). For a broadcast, information on a neighboring node of node v_i in the cluster with cluster ID k ($k \neq j$) is additionally included in control packets. This information is obtained by receiving a hello packet from the neighboring node of node v_i.

BCN (Border and Control Node)

Node v_i with state BCN plays the roles of both a clusterhead and a gateway. A broadcast is performed in the same way as shown in CN (Control Node) and BN (Border Node).

ON (Orphan Node)

Node v_i with state ON does not have any cluster ID. Note that, when a node is newly added, a state of the node becomes ON.

4.2 Assignment and Update of Cluster ID

Let NN_i denote a set of neighboring nodes of node v_i. Node v_i recognizes this set by receiving hello packets from them. Cluster IDs of the nodes in NN_i are obtained by receiving hello packets from them. Based on such cluster IDs, the cluster ID to be held by node v_i is assigned or updated in the following. Let v_i's cluster ID and v_i's state be denoted by $Cluster(v_i)$ and $State(v_i)$, respectively.

[Definition 1] Let NN_i^k denote a set of nodes such that they belong to NN_i and the cluster ID held by them is k, that is, let $NN_i^k = \{v_j | v_j \in NN_i \wedge k = Cluster(v_j)\}$. Let $NN_i^{\bar{k}}$ denote a set of nodes such that they belong to NN_i and the cluster ID held by them is not k, that is, let $NN_i^{\bar{k}} = \{v_j | v_j \in NN_i \wedge k \neq Cluster(v_j)\}$.

The cluster ID to be held by node v_i is assigned or updated as follows.

Case 1: $State(v_i) = ON$ and when v_i received a hello packet from an ON node, v_i recognizes that all neighboring nodes have been also orphan nodes (that is, $NN_i = \{v_j | v_j \in NN_i \wedge State(v_j) = ON\}$), v_i assigns the cluster ID to i (that is, lets $Cluster(v_i) = i$) with some probability.

Case 2: $State(v_i) = ON$ and when v_i received a hello packet from a neighboring node with its cluster ID = k, v_i recognizes a neighboring node holding cluster ID k, (that is, there is a node in NN_i^k), v_i assigns the cluster ID to k, (that is, lets $Cluster(v_i) = k$).

Case 3: There is a node v_i such that the cluster ID held by v_i is k and v_i has only a neighboring node holding cluster ID k, (that is, there was a node in NN_i^k). When v_i received a hello packet from the neighboring node v_t and the packet indicates $Cluster(v_t) = m$, v_i recognizes that all the neighboring nodes holding cluster ID k have disappeared, (that is, $NN_i = NN_i^{\bar{k}}$), v_i updates the cluster ID from k to m such that $m \neq k$ (that is, lets $Cluster(v_i) = m$).

Note that when nodes are newly added into the network, they become orphan nodes which do not have any cluster ID. In Case 1 or Case 2, the orphan node can have some cluster ID. In order to avoid that all orphan nodes have different cluster IDs and form a set of clusters in which there is only one node, in Case 2 some probability that cluster ID is assigned to orphan nodes is given.

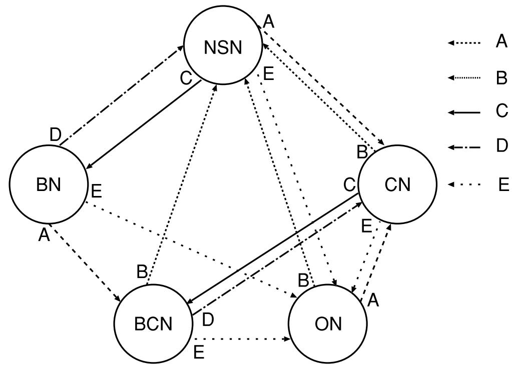

Fig. 3 State transition diagram in node v_i.

4.3 State Transitions in Nodes

Let NN_i denote a set of neighboring nodes of node v_i. Node v_i recognizes this set by receiving hello packets from them. Cluster IDs held by the nodes in NN_i are obtained by receiving hello packets from them. Based on such cluster IDs held by the nodes in NN_i, the states to be held by node v_i is changed in the following. This state change is called state transition.

Node v_i executes state transitions *A*–*E* as shown in Fig. 3 if the conditions for them described below are satisfied. Let $Cluster_{-T}(v_i)$ denote a cluster ID held by node v_i just before receiving hello packets from the neighboring nodes and updating the cluster ID. Let $Cluster(v_i)$ denote a current cluster ID held by node v_i just after receiving hello packets from the neighboring nodes and updating the cluster ID.

Condition for Transition A: $Cluster_{-T}(v_i) \neq Cluster(v_i) \wedge i = Cluster(v_i)$: This condition represents that the cluster ID is updated and the updated cluster ID equals to the node ID held by node v_i.

Condition for Transition B: $Cluster_{-T}(v_i) \neq Cluster(v_i) \wedge i \neq Cluster(v_i)$: This condition represents that the cluster ID is updated and the updated cluster ID does not equal to the node ID held by node v_i.

Condition for Transition C: $NN_i \supset NN_i^{\bar{k}}$: This condition represents that a neighboring node appeared, which holds a cluster ID being different with cluster ID held by node v_i.

Condition for Transition D: $NN_i = NN_i^k$: This condition represents that the cluster ID's held by all neighboring nodes became the same as the cluster ID held by node v_i.

Condition for Transition E: $NN_i = \phi$: This condition represents that all neighboring nodes of node v_i disappeared. And, exceptionally, in case that node v_i cannot receive the periodic notification packet from the clusterhead for some period, this transition occurs.

4.4 Division and Merger of Clusters

In order to maintain the hierarchical structure efficiently, load for each cluster should be equivalent. The number of nodes in each cluster, that is, the cluster size is a keypoint to make the load for each cluster equivalent. However, due to node movement, the cluster size increases or decreases as time proceeds. The proposed scheme adjusts each cluster size as much as possible autonomously so that it is bounded by the two constants, the upper bound and the lower bound. Suppose that these two constants are given in advance according to a network size, the number of nodes, issues on node movement such as speed and direction, and so on. In the proposed scheme, each clusterhead collects the sizes of neighboring clusters from gateways. When the cluster size is over the upper bound, the cluster is divided into two neighboring clusters, while when the cluster size is under the lower bound, two neighboring clusters are merged into one cluster.

This section describes how clusterheads manage these division and merger. Let U denote the upper bound and L denote the lower bound. First, the method to merge two neighboring clusters C_i and C_j is given below. Note that in the proposed scheme it is not possible to merge more than two clusters simultaneously.

[Definition 2] For any cluster C_i, let a set of clusters which is neighboring to cluster C_i be denoted by $NA_i = \{C_1, \cdots, C_m, \cdots, C_n\} (m \neq i)$.

If $|C_i| < L$, v_i determines the following cluster $C_j \in NA_i$ as the most preferable partner of merger.

1. If for some C_k $|C_i| + |C_k| \leq U$, let C_j be $C_k \in NA_i$ such that the number of nodes in C_j is the largest one among C_k.
2. If for any C_k $|C_i| + |C_k| > U$, let C_j be $C_k \in NA_i$ such that the number of nodes in C_j is the smallest one among C_k.

After determining C_j as the partner of merger, clusterhead $v_i \in C_i$ send a cluster merge request packet (shortly, $P_{bi_{req}}$) to clusterhead $v_j \in C_j$. If clusterhead $v_i \in C_i$ received the corresponding cluster merge reply packet (shortly, $P_{bi_{ack}}$) including 'answer yes' from clusterhead $v_j \in C_j$, the clusterhead in the merged cluster is determined as follows.

1. If $|C_i| > |C_j|$, $C_i \leftarrow C_i \cup C_j$, that is, clusterhead $v_i \in C_i$ becomes the clusterhead in the merged cluster.
2. If $|C_i| < |C_j|$, $C_j \leftarrow C_i \cup C_j$, that is, clusterhead $v_j \in C_j$ becomes the clusterhead in the merged cluster.
3. If $|C_i| = |C_j|$ and $i < j$, $C_i \leftarrow C_i \cup C_j$.

In the first case, since $v_j \in C_j$ is no more control node, it broadcasts cluster ID update packets (shortly, P_a) to all nodes in C_j. As a result, the cluster IDs held by all nodes in C_j are updated to i. The second case is the same as the first case except that i and j are interchanged with each other. The third case is the same as the first case.

Next, the method to divide cluster C_i into two neighboring clusters C_i and C_j is given below. As for C_i, if $|C_i| > U$, the cluster to be divided is determined to be C_i. Control node $v_i \in C_i$ determines a node $v_j \in C_i$ as a new control node in the divided cluster so that each size of the divided cluster is almost the same. After that, v_i sends a cluster divide packet (shortly, P_{de}) to v_j. Until time when a cluster is divided, which is included in P_{de}, v_i and v_j send cluster ID update packets (shortly, P_a) to the divided clusters, respectively. Nodes which first received P_a from v_i do not change their node ID, that is, node ID is i, while nodes which first received P_a from v_j change their node ID to j. As a result, the cluster is divided into two neighboring clusters C_i and C_j.

5. Simulation Results

5.1 Simulation Model and Parameters

Our simulation is based on an ad hoc network of 200 mobile nodes, whose initial locations are chosen from an uniform random distribution over a square of 750 m × 750 m. We used the random waypoint model. We explain the random waypoint model. It performs the following actions repeatedly. A node moves to a given position at a specified speed on average, and then, the node stops for a specified time at the position. We define the time as the pause time. The pause time is set 0 in the experiments. And as soon as the node arrives at the position, the node selects a new destined position and starts moving to the position again.

Each node has a radio transmission range of radius 100 m. At any time, each node can communicate with all the nodes within its transmission range. End-to-end delay including both data packet transmission time and processing time is 1 ms.

Each node periodically sends hello packets to nodes within its transmission range for a period of 200 ms to exchange its state and node ID. And each node broadcasts periodic notification packets to inform the information which the node has to all nodes within the cluster to which the node belongs. The duration of each simulation is 80 seconds. No data is collected for the first 5 seconds to avoid measurements during the transient period and to ensure that an initial hierarchical structure generated by the clustering scheme. And upon reaching the 65 seconds, the whole nodes are made to pause.

The average node speeds are 5 m/s, 10 m/s, and 20 m/s. The average node speed 20 m/s is a typical scenario of an ad hoc network of cars. The ranges of the cluster size (L, U) are $(10, 25)$, $(20, 50)$ and $(30, 75)$, respectively.

Table 1 Number of clusters.

Mobility (m/s)	Cluster Range		
	10-25	20-50	30-75
5	10.41 (18.31)	5.35 (36.74)	3.89 (51.26)
10	10.09 (17.81)	5.55 (34.65)	3.72 (54.12)
20	10.50 (18.38)	5.68 (35.13)	3.75 (53.60)

Table 2 Number of hop counts.

Mobility (m/s)	Cluster Range		
	10-25	20-50	30-75
5	1.97 (5.63)	2.74 (7.52)	3.12 (7.92)
10	2.00 (5.85)	2.76 (7.46)	3.29 (8.16)
20	1.95 (5.72)	2.62 (6.94)	3.13 (8.25)

5.2 Experiments on Overheads

In order to make the overhead for each cluster be almost equivalent, the number of clustermembers is bounded by the two constants, that is, upper bound and lower bound in the proposed clustering scheme. We call the range of the number of clustermembers bounded by these constants cluster range in this subsection.

First, we show simulation results on whether mobility (in other words, moving speed) of nodes influences the number of clustermembers in each cluster. We assume that the transmission range (distance within which radio is reachable) of each node is 100.

Table 1 denotes the average number of clusters and the average number of clustermembers in each cluster for mobilities of nodes, 5, 10, 20 and cluster ranges 10-25, 20-50, 30-75. The average number of clustermembers is surrounded by parentheses. Even when the mobility of nodes varies from 5 to 20, the number of clusters and the number of clustermembers for each cluster are almost equivalent. Therefore, the result of the experiments show that the proposed clustering scheme does not depend on mobility of nodes regardless of the cluster range.

Table 2 denotes the average of mean hop counts from the clusterhead to each clustermember for each cluster and the average of maximum hop counts from the clusterhead to the farthest clustermember for each cluster. The latter number is surrounded by parentheses. Each entry in this table is shown for mobilities of nodes 5, 10, 20 and cluster ranges 10-25, 20-50, 30-75. Even when the mobility of nodes varies from 5 to 20, the averages of the mean hop counts and the maximum hop counts are almost equivalent. The difference between the mean hop count and the maximum hop count for each entry are not large, we expect that clustermembers are uniformly distributed in each cluster. These results show that in each cluster the clustermembers are connected from the clusterhead by multihops

Table 3 Number of clusters and clustermembers versus node degree.

Mobility (m/s)	Transmission Range (m)		
	100	200	300
5	5.35 (36.74)	5.09 (38.86)	4.37 (42.14)
10	5.55 (34.65)	4.95 (40.13)	4.12 (41.41)
20	5.68 (35.13)	4.89 (39.02)	4.57 (41.17)

Table 4 Number of mean and maximum hop counts versus node degree.

Mobility (m/s)	Transmission Range (m)		
	100	200	300
5	2.74 (7.52)	1.64 (4.31)	1.25 (2.98)
10	2.76 (7.46)	1.90 (4.49)	1.35 (2.98)
20	2.62 (6.94)	1.97 (4.79)	1.54 (3.18)

and that the proposed clustering scheme is superior to the previous one in which all clustermembers in each cluster are reachable from the clusterhead by one hop count.

Next, we show simulation results on whether the average degree of nodes (in other words, the average number of neighboring nodes) influences the number of clustermembers in each cluster.

Table 3 denotes the average number of clusters and the average number of clustermembers in each cluster for mobilities of nodes, 5, 10, 20 and transmission ranges 100, 200, 300. The latter number is surrounded by parentheses. We assume that the cluster range is 20-50. Since the total number of nodes and the area within which nodes can move do not change, the longer the transmission range is, the larger the average degree of nodes is. Even when the mobility of nodes varies from 5 to 20, the number of clusters and the number of clustermembers for each cluster are almost equivalent. Therefore, we can say that the proposed clustering scheme does not depend on mobility of nodes regardless of transmission range.

Table 4 denotes the average of mean hop counts from the clusterhead to each clustermember for each cluster and the average of maximum hop counts from the clusterhead to the farthest clustermember for each cluster. The latter number is surrounded by parentheses. Each entry in this table is shown for mobilities of nodes 5, 10, 20 and transmission ranges 100, 200, 300. Along with increase of transmission range, the average degree of nodes increases and as a result both the average mean and the maximum hop counts decrease. These results show that the number of clustermembers in each cluster is kept almost constant in the proposed clustering scheme even when the transmission range changes.

From the above observations, we can say that if the cluster range is set appropriately, the proposed clustering scheme can constitute almost the same size clusters regardless of whether mobility and degree of nodes change. This means that overhead for management of hierarchical structure is uniformly distributed to clusters and the overhead of clusterheads in each cluster is made almost equivalent.

Table 5 Convergence time.

Mobility (m/s)	Cluster Range		
	10-25	20-50	30-75
5	101.00	80.40	0.40
10	360.70	301.50	100.70
20	583.30	481.00	161.00

unit: ms

Table 6 Number of changes of clusterheads.

Mobility (m/s)	Cluster Range		
	10-25	20-50	30-75
5	47.80	12.80	7.90
10	84.00	36.40	22.60
20	169.40	62.70	41.70

Table 7 Number of divisions of a cluster and mergers of clusters.

Mobility (m/s)	Cluster Range		
	10-25	20-50	30-75
5	43.70 (39.40)	11.60 (6.60)	7.40 (3.40)
10	73.50 (70.60)	33.40 (21.90)	20.80 (8.70)
20	151.40 (137.40)	58.50 (34.50)	38.50 (14.60)

5.3 Experiments on Adaptability

In order to construct and maintain the hierarchical structure in the ad hoc network so that the role of each node is adaptive to node movement, each node changes its role and its cluster ID autonomously based on the local information on the role and the cluster ID of the neighboring nodes and the size of neighboring clusters, which are obtained by receiving hello packets and periodic notification packets in the proposed clustering scheme.

The adaptability of node behavior is measured as a convergence time from an unstable state to a stable state of the ad hoc network. The stable state is recognized by observing that the role and cluster ID of all nodes does not change for five seconds, which have been determined by preliminary simulation experiments. In the simulation experiments, after 60 seconds node movement is stopped, however, at 60 seconds the network is unstable due to node movements performed before 60 seconds. The convergence time is thus a time interval from 60 seconds to a time at which the stable state is observed.

Table 5 denotes a convergence time for mobilities of nodes 5, 10, 20 and cluster ranges 10-25, 20-50, 30-75. The convergence time does not increase so much along with increase of node mobility and at most three times time intervals of hello packets or periodic notification packets even when node mobility is 20. The main reason for such short convergence time is that each node change its role and cluster ID autonomously.

Table 8 Number of changes of cluster ID and transitions.

Mobility (m/s)	Cluster Range		
	10-25	20-50	30-75
5	1132.00 (861.80)	658.10 (814.40)	572.60 (752.10)
10	2323.30 (1458.70)	1489.90 (1620.80)	1161.70 (1355.10)
20	4705.50 (3849.20)	2854.90 (3151.60)	2076.00 (2621.60)

Unstable states are caused by the following factors. (1) A current clusterhead moves from its cluster to another neighboring cluster and a new clusterhead is selected. (2) The division of a cluster and merger of clusters occur. (3) A clustermember except for a clusterhead moves.

Table 6 denotes the average number of changes of clusterheads for mobilities of nodes 5, 10, 20 and cluster ranges 10-25, 20-50, 30-75 and Table 7 denotes the average number of divisions of a cluster and mergers of clusters. These data are closely related because change of clusterheads is caused by factors (1) and (2), particularly factor (1) fractionally occurs due to the large cluster range. The number of divisions and mergers of clusters dominate the convergence time.

Table 8 denotes the average number of changes of cluster ID and the average number of changes of transitions for mobilities of nodes 5, 10, 20 and cluster ranges 10-25, 20-50, 30-75. The latter number is surrounded by parentheses. The results show that these numbers are almost proportional to the node mobility. Since these changes are autonomously performed by each node, they occur in parallel, which results in short convergence time.

From the above simulation results, we can say that the proposed clustering scheme has high adaptability to the node movement in ad hoc networks.

6. Related Work

In LCC scheme [2], a cluster consists of one clusterhead and its neighboring nodes called clustermembers. During node movement, each node recognizes the cluster to which the node belongs and the role of the node in the cluster as follows. When a node finds that its neighboring node is a clusterhead, the node belongs to the cluster which the clusterhead manages. When a node finds that more than one neighboring node of the node are clusterheads, it becomes a gateway shared with the clusters which these clusterheads manage. When a clusterhead finds that its neighboring node is another clusterhead, one of these clusterheads (that is, a clusterhead with the largest IP address) maintains the role of clusterhead and the others abandon their roles.

In the experimental evaluation, we compare our proposed scheme with the LCC scheme with respect to the number of changes of clusterhead and cluster ID. Tables 9 and 10 denote the average number of changes of clusterhead and cluster ID, respectively. The experimental evaluation is done at the same model and parameters in Sect. 5. The number of changes of clusterhead is an important factor to evaluate the clustering schemes in ad hoc networks. If the number is small, it means that the cluster is not frequently reconfigured. On the other hand, if the number is large, it means that the cluster is frequently reconfigured. As shown in the tables, the number of changes of clusterhead of our proposed scheme is much smaller than that of the LCC scheme. As a result, we can say that our proposed clustering scheme is much effective in ad hoc networks, especially those with high mobility.

Table 9 Number of changes of clusterheads of the proposed scheme versus LCC scheme.

Mobility (m/s)	Proposed scheme Cluster Range			LCC [2]
	10-25	20-50	30-75	
5	47.80	12.80	7.90	240.60
10	84.00	36.40	22.60	433.70
20	169.40	62.70	41.70	770.90

Table 10 Number of changes of cluster ID of the proposed scheme versus LCC scheme.

Mobility (m/s)	Proposed scheme Cluster Range			LCC [2]
	10-25	20-50	30-75	
5	1132.00	658.10	572.60	2159.10
10	2323.30	1489.90	1161.70	3861.30
20	4705.50	2854.90	2076.00	6620.60

The improved LCC clustering scheme [6] is based on [2]. When two clusterheads are neighboring with each other, they abandon the role of clusterhead, and a node which has the largest degree among nodes to which these two clusterheads are neighboring is selected as a new clusterhead. The improved LCC has a defect on changes of clusterheads, which is similar to that of the LCC.

The clustering scheme proposed in [4] configures the clusters based on the BFS tree. The authors hypothesize that topology changes in wireless networks will be slow and infrequent. Of course, if topology changes are frequent, reconfiguration of clusters based on the BFS tree needs a large time. The time increases along with increase of the network size.

7. Conclusion

We have presented an adaptive multihop clustering scheme for highly mobile nodes in ad hoc networks. The key features of the proposed scheme are as follows. (1) The role of each mobile device for the hierarchical structure is adaptive to dynamic change of the topology of the ad hoc networks. The role of each mobile device is thus changed autonomously based on only the local information. (2) The overhead for management of the hierarchical structure is small since the number of mobile devices in each cluster tends to be almost equivalent.

Simulation experiments demonstrate the advantages derived from the autonomy of the proposed scheme.

From the results on the overhead in the first experiment, we can say that if the cluster range is set appropriately, the proposed clustering scheme can constitute almost the same size clusters regardless of whether mobility and degree of nodes change. This means that the overhead for the management of the hierarchical structure is uniformly distributed to clusters and the overhead of clusterheads in each cluster is made almost equivalent.

From the results on adaptability in the second experiments, we can say that the proposed clustering scheme has high adaptability to node movements in ad hoc networks.

As a result, these experiments show that the proposed clustering scheme is adaptive and works well for highly mobile nodes in ad hoc networks.

In the future research, we are planning to design a generic function to evaluate the adaptability of clustering schemes to node movements in ad hoc networks.

References

[1] S. Basagni, "Distributed clustering for ad hoc networks," Proc. 99' Int'l Symp. on Parallel Architectures, Algorithms, and Networks (I-SPAN'99), pp.310–315, 1999.
[2] C. Chiang, H. Wu, W. Liu, and M. Gerla, "Routing in clustered multihop, mobile wireless networks with fading channel," Proc. IEEE Singapore Int'l Conf. on Networks (SICON), pp.197–211, 1997.
[3] C.R. Lin and M. Gerla, "Distributed clustering for ad hoc networks," IEEE J. Sel. Areas Commun., vol.15, no.7, pp.1265–1275, 1997.
[4] S. Banerjee and S. Khuller, "A clustering scheme for hierarchical control in multi-hop wireless networks," Proc. IEEE INFOCOM2001, pp.1028–1037, 2001.
[5] A.D. Amis, R. Prakash, T.H.P. Vuong, and D.T. Huynh, "Max-min d-cluster formation in wireless ad hoc networks," Proc. IEEE INFOCOM2000, pp.32–41, 2000.
[6] H. Taniguchi, M. Inoue, T. Masuzawa, and H. Fujiwara, "Clustering algorithms in ad hoc networks," IEICE Trans. Inf. & Syst. (Japanese Edition), vol.J84-D-I, no.2, pp.127–135, Feb. 2001.
[7] C.E. Perkins, Ad hoc networking, pp.221–253, Addison-Wesley, 2001.
[8] T. Ohta, K. Ishida, Y. Kakuda, S. Inoue, and K. Maeda, "Maintenance algorithm for hierarchical structure in large ad hoc networks," Proc. Int'l Conf. on Fundamentals of Electronics, Communications and Computer Sciences (ICFS2002), pp.S3-12–S3-17, 2002.
[9] T. Ohta, S. Inoue, Y. Kakuda, K. Ishida, and K. Maeda, "An adaptive maintenance of hierarchical structure in ad hoc networks and its evaluation," Proc. 22nd IEEE Int'l Conf. on Distributed Computing Systems Workshops (ADSN2002), pp.7–13, 2002.
[10] V.D. Park and M.S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," Proc. IEEE INFOCOM'97, pp.1405–1413, 1997.
[11] T. Ohta, M. Fujimoto, S. Inoue, and Y. Kakuda, "Hi-TORA: Hierarchical routing protocol in ad hoc networks," Proc. IEEE Int'l Symp. on High Assurance Systems Engineering (HASE2002), pp.143–148, 2002.

Tomoyuki Ohta received the B.E., and M.E. degrees from Hiroshima City University, Japan, in 1998 and 2000, respectively. He had been a Ph.D. candidate at the Graduate School of Hiroshima City University from 2000 to 2002. He is currently a research associate in the Department of Computer Engineering, Faculty of Information Sciences, Hiroshima City University, since 2002. His current research interests include mobile communication systems. He is a member of IEEE (U.S.A).

Shinji Inoue received the M.E. degree from Hiroshima University, Japan, in 1988. He has been a research associate in the Department of Computer Engineering, Faculty of Information Sciences, Hiroshima City University since 1994. His research interests include multi-agent systems and fault-tolerant systems. He is a member of IPSJ (Japan).

Yoshiaki Kakuda received the B.E., M.Sc., and Ph.D. degrees from Hiroshima University, Japan, in 1978, 1980 and 1983, respectively. From 1983 to 1991, he was with Research and Development Laboratories, Kokusai Denshin Denwa Co., Ltd. (KDD). He joined Osaka University from 1991 to 1998 as an Associate Professor. He is currently a Professor in the Department of Computer Engineering, Faculty of Information Sciences, Hiroshima City University, since 1998. His current research interests include network software engineering and assurance networks. He is a member of IEEE (U.S.A) and IPSJ (Japan). He received the Telecom. System Technology Award from Telecommunications Advanced Foundation in 1992.

Kenji Ishida received the B.E., M.Sc., and Ph.D. degrees from Hiroshima University, Japan, in 1984, 1986 and 1989, respectively. He was at Hiroshima Prefectural University from 1989 to 1997. He has been an Associate Professor in the Department of Computer Engineering, Faculty of Information Sciences, Hiroshima City University, since 1997. His interests include distributed computing systems and design of control procedures for computer networks. He is a member of IEEE (U.S.A), ACM (U.S.A), IPSJ (Japan).