Multi-hop Clustering Based on Neighborhood Benchmark in Mobile Ad-hoc Networks

Stephen S. Yau and Wei Gao
Department of Computer Science and Engineering
Arizona State University, Tempe, AZ 85287-8809, USA
{*yau, w.gao@asu.edu*}

***Abstract*— Large-scale mobile ad-hoc networks require flexible and stable clustered network structure for efficient data collection and dissemination. In this paper, a scheme is presented to construct multi-hop clusters with balanced sizes, based on the neighborhood benchmark (NB) which quantifies the connectivity and link stability of mobile nodes. By exploiting autonomous clusterhead selection and a specialized handshake process with the clusterheads, the nodes with highest NB scores are selected as clusterheads and all the clusters constructed are connected. The deviation of cluster sizes is kept small using a partial probability-based approach. Our scheme generates highly stable multi-hop clusters with low overhead, and provides the flexibility of controlling the cluster radius adaptively for various network applications.**

I. Introduction

The highly dynamic nature and severe resource constraints of mobile ad hoc networks (MANETs) make the flat network architecture difficult to achieve scalability and cost effectiveness in data collection and dissemination [1]. Clustered network structure can be used to facilitate such tasks because the clusterheads form a virtual backbone for restricting inter-cluster data transmission from flooding [2]. The effects of node mobility to the network structure are localized and the network appears smaller in view of individual mobile nodes because the virtual backbone reduces the lengths of paths between node pairs. Currently, most of the clustering schemes in MANETs construct clusters with a fixed one-hop cluster radius, and select clusterheads casually without considering practical network conditions. Hence, the clusters constructed have low stability and flexibility.

In this paper, we will present a scheme to construct multi-hop clusters with balanced sizes based on the neighborhood benchmark (NB) scores of mobile nodes in MANETs. Our approach conducts network initialization and cluster formation at run-time without the "frozen period" assumption [2], and hence is more realistic in practice. The NB can quantify the connectivity and link stability of mobile nodes using the nodes' neighbor degrees and the link failures encountered in unit time. The clusterheads selected autonomously based on their NB scores are hence efficient as the aggregation points of data flows, and stable to avoid frequent clusterhead changes. We will show that the clusters constructed are connected through a handshake process, and that the deviation of cluster sizes due to autonomous clusterhead selection is controlled without degradation of the quality of clusters.

II. Current State Of the Art

Network clustering has been well studied in various types of decentralized p2p and wireless networks. Some researchers designed combinatorial clustering algorithms based on abstracted network topology, regardingless of the realistic network constraints. In these algorithms, clusterheads are selected randomly, and a "frozen period" is assumed, such that all the mobile nodes keep static in the cluster formation [2]. With this assumption, mobile nodes are able to exchange accurate information with their neighbors, and thus the clusters can be formed with some specific characteristics. However, the assumption is unrealistic because mobile nodes are usually moving all the time.

The most common scheme of random clusterhead selection is the lowest-ID scheme [3], in which each node is assigned with a random ID, and the node with the lowest ID in a neighborhood is assigned as the clusterhead. Various clustering algorithms have been proposed based on the lowest-ID scheme. In [4], clusterheads are selected to construct a connected dominating set (CDS). Since the problem of finding a minimum CDS in a connected graph is NP-complete, approximation algorithms are developed to minimize the size of the dominating set and the computational complexity. In [5], a weakly CDS (WCDS) with smaller size is constructed by relaxing the requirement of direct connection between neighboring dominating nodes. Least Cluster Change (LCC) [6] increases the cluster stability by relinquishing the requirement that a clusterhead should always bear some specific attributes in its local area. In [7], the scope of CDS is expanded to d-hop, and a max-min heuristic is proposed for clusterhead selection based on the NP-completeness proof of the problem of finding a d-hop CDS.

All the above clustering algorithms have very limited usage in practice because they generally ignore the actual network conditions and constraints, such as the mobility pattern and communication capability of nodes. Alternative approaches have been developed to overcome these difficulties [8], [9], [10], [11], [12]. Highest Connectivity Clustering (HCC) [8] takes the node connectivity into account, but clusters constructed are unstable because a node is forced to change its clusterhead once it finds another clusterhead with higher connectivity. Some methods were developed to improve connectivity-based clustering [9], [10]. However, simply using the connectivity of mobile nodes as the selection criterion of clusterheads is not sufficient to incorporate the actual network conditions in to the clustered network structure, such as the dynamic network topology and volatile wireless channel characteristics. Mobility support is another issue which is ignored in these approaches because the existence of mobility impedes the proof of algorithm features. Mobility-aware clustering has been considered [11], [12], but most of them are based on certain assumptions of the network mobility patterns, which only fit the specialized network scenarios.

Comparing to the 1-hop clustering schemes based on connected dominating sets [4], [5], which are currently widely used in scalable routing, multi-hop clusters with controllable cluster radius provide more flexibility for sub-structures within clusters, and are more stable in dynamic network topology because a node has a higher chance to be multi-connected to a cluster, and a lower chance to lose its connection to its clusterhead. Kim, et al. [13] first defined a k-hop cluster for multi-hop clustering in MANETs. Some extensions have been made to ensure the connectivity among clusters and to reduce the number of gateway nodes connecting the clusterheads and the overhead for the k-hop cluster construction [14], [15]. In [11], multi-hop clusters are constructed by grouping the nodes with a similar mobility pattern together. However, possible inconsistency and conflict during the process of clusterhead selection is generally ignored. None of these schemes guarantee their constructed multi-hop clusters are connected with balanced sizes.

III. Our Approach

A mobile ad-hoc network normally consists of nodes with heterogeneous mobility and link characteristics. We assume that all the mobile nodes in the network have omni-directional antenna and all the network links are bi-directional. The NB score of a mobile node N_i used to indicate the qualification of this node to be a clusterhead is defined as

$$NBS_i = d_i / LF_i \tag{1}$$

where d_i is the neighbor degree of N_i indicating the connectivity of N_i's neighborhood, and LF_i is the number of link failures encountered by N_i in unit time indicating the link stability of N_i's neighborhood.

Our approach to constructing R-hop clusters consists of the following four steps:

1) *Network initialization*. Every mobile node periodically sends hello beacons to its neighbors, for its neighbors to calculate and exchange their initial NB scores. The interval of hello beaconing is set according to the mobility and communication range of mobile nodes.
2) *Autonomous clusterhead selection*. After the network initialization, every node starts to select its clusterhead autonomously based on the NB scores of its neighbors, in R consecutive selection iterations.
3) *Handshake with clusterheads*. After a node finishes its clusterhead selection, it starts a handshake process with its selected clusterhead, to build up its cluster membership and cluster structure.
4) *Cluster maintenance.* After a node completes the handshake with its clusterhead, it starts to conduct bilateral beaconing with its clusterhead to maintain the cluster structure. Such cluster maintenance last in the entire network lifetime.

These steps will be elaborated in the following sections.

IV. Network Initialization

The network initialization is done via hello beaconing, i.e., all the mobile nodes periodically send hello beacons to their neighbors. Such beaconing process is conducted independently on individual nodes, which does not rely on global time synchronization. The interval of such hello beaconing is $T_H = r_c/v_c$, where r_c is the average transmission range of the nodes, and v_c is the average moving speed of mobile nodes in the network. Hence, such an interval is positively proportional to the global network connectivity, and inversely proportional to the network mobility level.

Since in MANETs the link failures may be due to node mobility, channel interference and/or node power depletion. Such network initialization must be conducted at run-time, instead of a "frozen period".

The hello beacons sent by node N_i are encapsulated as $\{$*IP*$_i$, *NBS*$_i$, *Head_IP*$_i$, *HEAD_NBS*$_i$, *hop*$_i$, *size*$_i\}$, where *IP*$_i$ and *NBS*$_i$ are for N_i, and *Head_IP*$_i$ and *HEAD_NBS*$_i$ are for the current clusterhead of N_i. *hop*$_i$ indicates the hop count from N_i to its current clusterhead, with the range $[0, R]$. *size*$_i$ is the size of the cluster to which N_i belongs.

Every node maintains a neighborhood information table (NIT), which records and updates all the information included in the hello beacons from the node's neighbors. If the record of a neighbor in the NIT of any node has not been updated longer than $2T_H$, the neighbor is considered unreachable by the node, and one link failure of the node is counted. Nodes count the numbers of their neighbors and link failures before they send out the hello beacons, and update their NB scores using (2). *LF*$_i$ is calculated iteratively as follows:

$$LF_i^{(k)} = (LF_i^{(k-1)} \cdot (k-1) \cdot T_H + LF_{new})/(k \cdot T_H), \tag{2}$$

where k is the current hello beaconing period, and *LF*$_{new}$ is the number of link failures in this period.

Such network initialization lasts $\lceil SIZE/r_c \rceil \cdot T_H$ to initialize the NB scores of mobile nodes in the entire network, where *SIZE* is the size of network application area,. Hello beaconing will be conducted continuously in the network lifetime to keep updating the NB scores of mobile nodes.

V. Construction of Multi-hop Clusters

A. Autonomous clusterhead selection

The clusterheads are selected in an autonomous manner based on the NB scores of mobile nodes. As defined in Section III, the NB score of a mobile node quantifies the connectivity and link stability of the node. Our approach ensures that every clusterhead has its NB score higher than the NB score of any

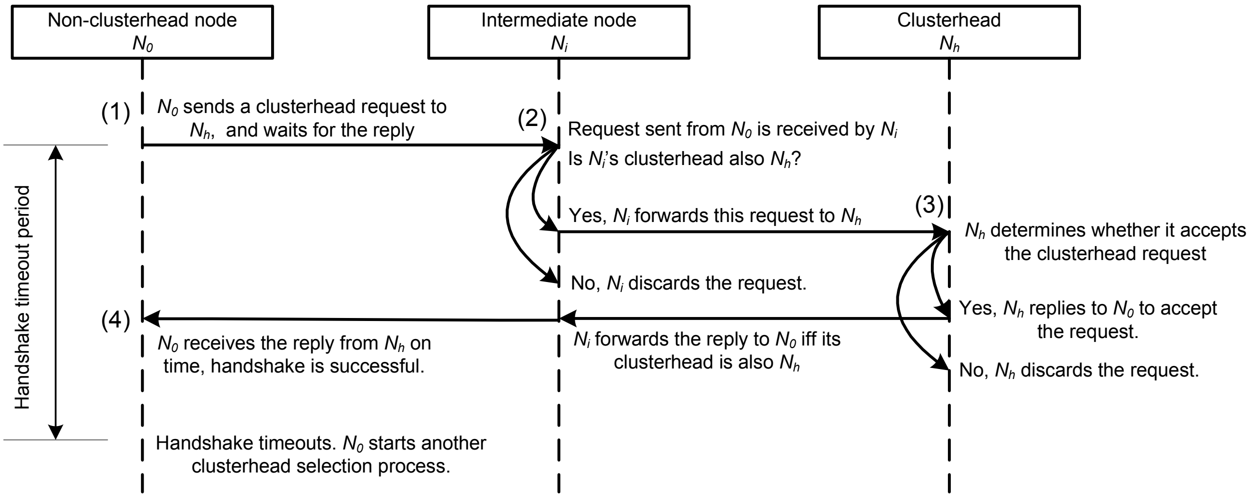

Fig. 1. Handshake with clusterheads

of its cluster members, and hence the selected clusterheads are as efficient as the aggregation points for forwarding data, and are stable to avoid frequent clusterhead changes.

Autonomous clusterhead selection is conducted on all the mobile nodes in parallel after the network initialization. The clusterhead selection process on each mobile node consists of R iterations, where R is the cluster radius in terms of the maximum number of hops from a node in the cluster to the clusterhead. In each iteration, a node N_i puts all the clusterheads of its 1-hop neighbors, and its own clusterhead into a selection pool. If N_i does not have its clusterhead yet, it uses itself as its clusterhead. Then, N_i selects the node with the highest NB score in the selection pool to be its clusterhead. N_i's clusterhead is updated in each iteration, and is finalized in the last iteration of the selection process.

The cluster radius R is pre-selected by the node owners, and different clusters can have different cluster radii. We will show in Section VII that by choosing different cluster radius according to the specific conditions of network environments, the constructed clusters can achieve different tradeoffs among the cluster stability, construction overhead, and cluster coverage.

The interval between two iterations is set to be the same as the interval of hello beaconing described in Section III. Hence, on a mobile node, each iteration in the clusterhead selection process is corresponding to a hello beaconing period. Any iteration will not start until the node receives all the corresponding hello beacons from its 1-hop neighbors. The correctness of such clusterhead selection process is shown as follows:

Theorem 1: The kth iteration on a node N_i selects the node with the highest NB score within the k-hop neighborhood of N_i to be N_i's clusterhead.

Proof: We prove this theorem by induction. For the first iteration, the theorem is self-evident. Assume that the theorem holds after the mth iteration. Because each iteration corresponds to a hello beaconing period, N_i will be notified of the clusterheads of its 1-hop neighbors before the $(m+1)$th iteration. These clusterheads are the nodes with the highest NB scores in the m-hop neighborhoods of N_i's 1-hop neighbors. Hence, in the $(m+1)$th iteration, all the possible new elements in N_i's selection pool are N_i's $(m+1)$-hop neighbors. ■

Based on Theorem 1, we can easily derive the following corollary, which ensures that a selected clusterhead has the larger NB score than any of its cluster members:

Corollary 1: Given a cluster C=$\{N_i|$ *i=0, 1, …..n*$\}$, if a node $N_k \in C$ is the clusterhead of C, then for $\forall i \in [0, n]$, $i \neq k$, $NBS_k \geq NBS_i$.

B. Handshake with clusterheads

In a multi-hop cluster, the clusterhead has the complete member list of the cluster, and the cluster members only record their clusterhead. After clusterhead selection, a mobile node handshakes with its selected clusterhead to construct the multi-hop cluster. Our approach can detect possible inconsistency during the handshake, and ensures that all the constructed clusters are connected. Such handshake process is described in Fig. 1 using a case including a node N_0, its selected clusterhead N_h, and an intermediate node N_i. It is noted that because of the autonomous and simultaneous clusterhead selection, all the mobile nodes also handshake with their selected clusterheads simultaneously. In such cases, if a node in its handshake process is requested to be a clusterhead by some other nodes, and the request is accepted, it should send another message to its selected clusterhead to cancel the ongoing handshake process, and start to handshake with the requesters as a new clusterhead.

An example of 2-hop clusters is shown in Fig. 2. Because every mobile node only has the knowledge of its neighborhood, there will be possible conflict and inconsistency during the autonomous cluster selection and handshake process. For example, in Fig. 2(a), in the first cluster selection round, N11 selects N3 as its clusterhead, and notifies N9 of this via hello beaconing. In the second round, because the NB score of N3

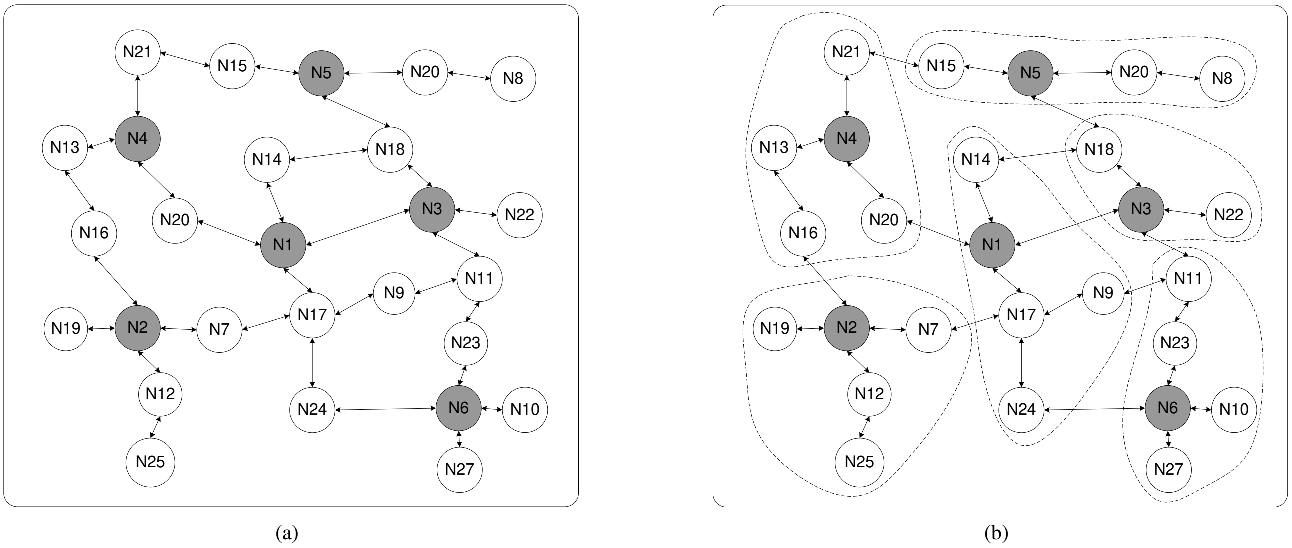

Fig. 2. An example of 2-hop clusters: (a) before handshake, and (b) after handshake

is higher than that of N1, N9 also selects N3 as its clusterhead. At the same time, N11 hears from N23 that N6 has a higher NB score. Since N11 has no idea about N9's choice by that time, N11 will change its clusterhead to N6. However, this change will break the link from N9 to N3, and thus constructs a disconnected cluster.

This problem is avoided by the handshake process such that a node will only forward the clusterhead requests if it is in the same cluster. Hence, in the above example, N9's clusterhead request cannot reach N3 because N11 will not forward it. Hence, N9's handshake process will timeout and another clusterhead selection process starts. The clusters constructed in this example are shown in Fig. 2(b). In this way, we can guarantee that all the constructed clusters are connected. This is described as follows:

Theorem 2: For any node N_i in an R-hop cluster C with the clusterhead N_h, there exists a path from N_i to N_h, such that the path only contains the nodes in C.

Proof: If N_c is 1-hop neighbor of N_h, the theorem is self-evident. Because only through the nodes which also select $N_h$as their clusterhead, $N_c$can complete the handshake process with N_h, N_c must have at least one of its 1-hop neighbors which is also in C. The theorem can be proved by induction. ■

C. Cluster maintenance

Cluster maintenance is conducted via bilateral beaconing. A clusterhead multicasts beaconing messages periodically at the interval T_{cb} to all the cluster members, where T_{cb} is set to be the same as the hello beaconing interval T_H. Every cluster member returns an acknowledgment to its clusterhead upon receiving the beacon message. Only the nodes within the same cluster will forward the beaconing messages and acknowledgments.

In order to keep stable cluster structures, a non-clusterhead node only searches for its new clusterhead if its current clusterhead is unavailable. If a non-clusterhead node has not received the beacon message from its current clusterhead longer than $2T_{cb}$, it considers its current clusterhead unreachable, and reselects its clusterhead. If the clusterhead has not received the acknowledgment from a cluster member longer than $2T_{cb}$, the clusterhead considers the member unreachable and deletes the member from its member list.

VI. Balancing Cluster Sizes

In a clustered network structure, clusterheads are the aggregation points of data flows and hence consume their resources faster. On the other hand, clusterhead changes are not desirable because a large number of nodes will be involved in clusterhead reselection, and cause large communication overhead. Clusters with balanced sizes are preferred to increase the network stability and lifetime.

In Section V, since every node selects its clusterhead deterministically based on the NB scores without considering size balancing, most of the clusterhead selections are focusing on a few nodes with higher NB scores, and hence form clusters with large size deviation. However, to construct clusters with balanced sizes will impair the quality of clusters represented by the NB scores of clusterheads. We exploit a predefined preference parameter P_t between 0 and 1 to balance between cluster quality and size balancing according to network situations.

Based on the notation of P_t, a partially probability-based approach cab be used for clusterhead selection. In a clusterhead selection round at node N_0, assume that the set $H = \{N_{h1}, N_{h2}\ldots, N_{hk}\}$ indicates the clusterhead selection pool, and *size*$_i$ indicates the current cluster size of the clusterhead N_{hi}. Each N_{hi} in H has a probability p_i to be selected as the clusterhead of N_0. Such a probability is made up of a singular part and a common part.

- The singular part p_{si} is only valued $1 - P_t$ when N_{hi} has the highest NB score in H. Otherwise, p_{si}=0.
- The common part p_{ci} is defined as

$$p_{ci} = ((1 - P_t) \cdot NBS_i + P_t \cdot E_i) \cdot P_t, \tag{3}$$

where E_i is the normalized entropy of N_i, i.e.,

$$E_i = \frac{\ln(S/size_i)}{\sum_{j=1}^{k} \ln(S/size_j)} \tag{4}$$

$$S = \sum_i size_i \tag{5}$$

in H, the entropy of N_i is inversely proportional to *size*$_i$, and the normalized entropy ensures that

$$\sum_i p_i = \sum_i (p_{si} + p_{ci}) = 1 \tag{6}$$

The common part of p_i makes N_{hi} with higher NB score or smaller size have higher probability to be selected as clusterhead, and the singular part of p_i gives N_{hi} with the highest NB score in H an extra preference. When $P_t = 0$, p_{ci}=0, clusterheads are selected deterministically without considering size balancing. When $P_t = 1$, p_{si}=0, and $p_{ci} = E_i$, clusterheads are selected probabilistically purely according to the entropy of N_i defined in (4), without considering nodes' NB scores, to minimize the deviation of cluster sizes. Any intermediate value of P_t between 0 and 1 produces a probabilistic distribution to tradeoff between cluster qualities and balanced cluster sizes.

VII. Performance Evaluation

We have simulated our scheme using ns-2 with the CMU wireless extensions. We have evaluated the performance of our scheme in terms of cluster stability, clustering overhead, and cluster coverage. We have also evaluated the performance of our size balancing mechanism on the deviation of cluster sizes and the stability of size-balanced clusters. In the simulations, we uniformly deployed 50 mobile nodes in a $1000 \times 1000 m^2$ square area. 802.11 WLAN is used as the underlying MAC protocol. We assume that all the nodes have omni-directional antennas and uniform communication range of 250m, and the two-way ground propagation model is used. The node mobility follows the random-walk mobility model [11] with the node moving speeds normally distributed in a range $[0, v_{max}]$. Each simulation lasts 5000 secs, and each point in the simulation figures is averaged over 10 random simulation scenarios.

A. Performance of multi-hop clustering based on neighborhood benchmark

In the first set of simulations, we have evaluated the performance of multi-hop clustering based on neighborhood benchmark (MCNB) with different cluster radius, comparing to two traditional clustering methods LCC [6] and HCC [8]. Fig. 3 shows the evaluation result of cluster stability represented by the average number of clusterhead changes per node during the simulation. Compared to HCC, because

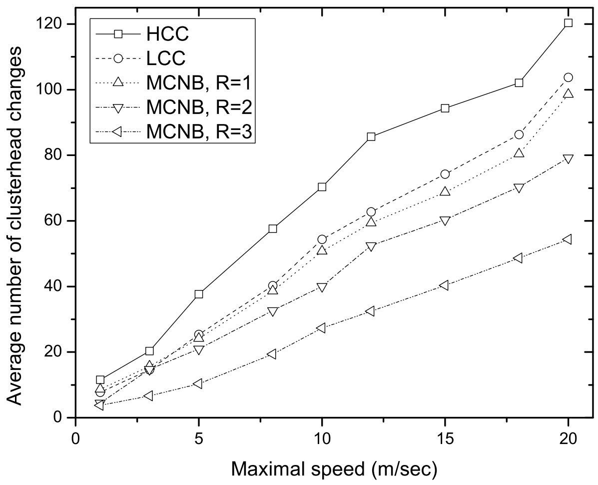

Fig. 3. Evaluation of cluster stability

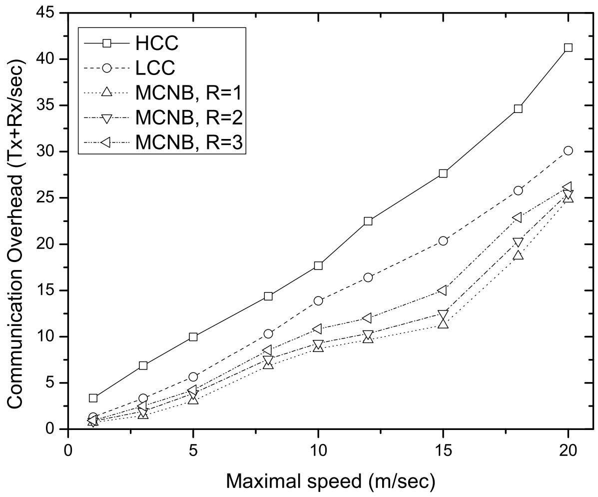

Fig. 4. Evaluation of communication overhead

MCNB does not force a node to change its clusterhead when it discovers another clusterhead with higher NB score, the cluster stability is greatly improved. The cluster stability inevitably degrades when the node mobility increases. It is shown that larger cluster radius can mitigate the effects of node mobility because a node has a smaller chance to lose its clusterhead connection in a larger and heavier connected cluster, which enables every node to reach its clusterhead via multiple paths. Even when R=1, the performance of MCNB is similar to LCC which is aiming at minimizing the cluster changes.

Fig. 4 shows the communication overhead for clustering evaluated by the average transceiving and receiving messages per sec per node through the entire simulation process. Such overhead includes both cluster construction and maintenance

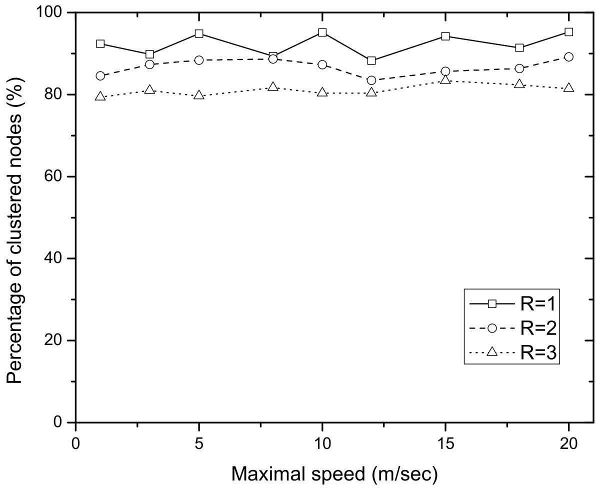

Fig. 5. Evaluation of cluster coverage

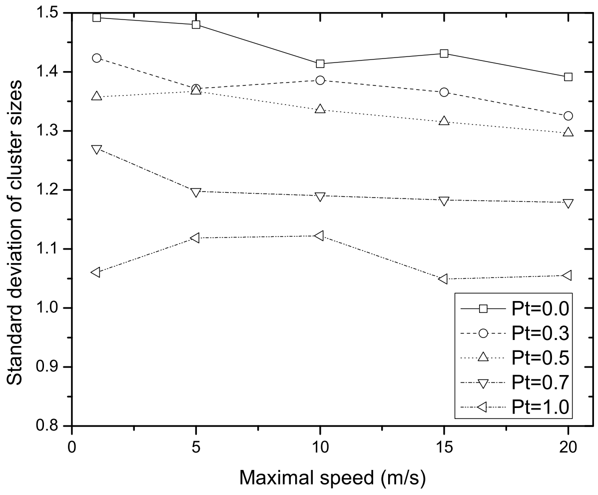

Fig. 6. Effects of size balancing

cost. Since MCNB restricts all the beaconing mechanisms to a localized scope, and does not enforce proactive clusterhead reselection, its clustering overhead is greatly reduced, compared to HCC and LCC. The overhead increase in higher mobility is due to more frequent beaconing.

Fig. 5 shows the cluster coverage, i.e., the percentage of nodes being clustered with various cluster radii. A number of nodes may have their clusterhead lost due to network topology changes. However, our approach can guarantee that a majority of nodes are clustered over a long time. On the other hand, using a larger cluster radius decreases the cluster coverage. This is because the construction of larger clusters needs more complicated clusterhead selection and handshake process. When the cluster radius is larger and it needs more clusterhead selection rounds to decide the clusterhead, the possibility of inconsistent cluster membership will increase.

The advantages of multi-hop clusters constructed based on neighborhood benchmark are their stability and flexibility. Different cluster radius can be used for our scheme to adapt to various network applications. A larger cluster is more stable with network topology changes, and provide more flexibility to construct sub-structures within the cluster, but the construction of the cluster causes more communication overhead among network nodes and the cluster can only cover up to 80% of the network nodes as shown in Fig. 5. On the other hand, a network structure with smaller clusters is more volatile and sensitive to network topology changes, but smaller clusters are also easier to be constructed, and can provide better node coverage. Hence, larger clusters are suitable for those data-intensive applications to achieve higher efficiency of data delivery, and small clusters can be used in severe physical environments to achieve higher node coverage and more flexible reconfigurability.

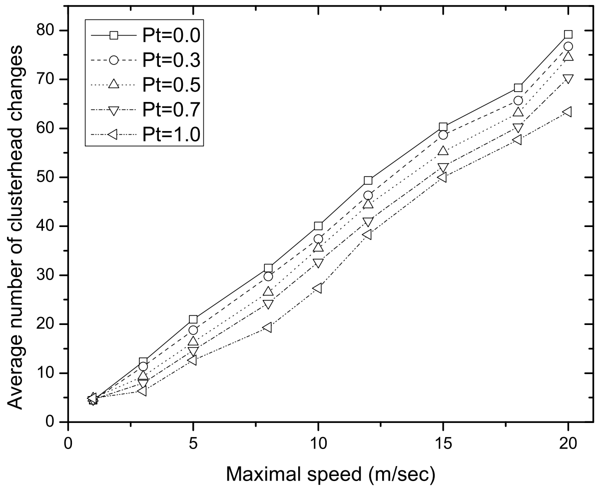

Fig. 7. Effects of size balancing to cluster stability

B. Performance of our scheme with balancing cluster sizes

We have evaluated the performance of our scheme with balancing cluster sizes in various values of the preference parameter with a cluster radius R=2. In Fig. 6, the effect of balancing cluster sizes is evaluated by the standard deviation of cluster sizes. Fig. 6 shows that the deviation of cluster sizes will be greatly reduced when larger P_t is used to put more emphasis on balanced cluster sizes. However, since the singular part of the clusterhead selection probability described in Section VI gives the node with the highest NB score an extra preference, the effect of size balancing can be apparent only when $P_t \geq 0.7$. On the other hand, the deviation of cluster sizes is not affected much by the node mobility. Such deviation is smaller when the node mobility is high because an arbitrary

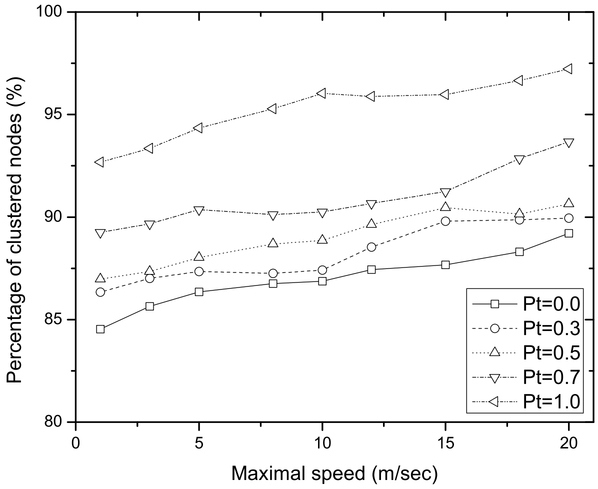

Fig. 8. Effects of size balancing to cluster coverage

node has more clusterhead choices in a high-mobility network scenario.

Fig. 7 shows that to construct clusters with size balancing will not impair the cluster stability. Instead, when clusters are constructed purely for balanced cluster sizes ($P_t = 1.0$) and the deviation of cluster sizes is minimized, the average number of clusterhead changes is reduced up to 20%.

It is also shown in Fig. 8 that our size balancing approach increases the cluster coverage. When P_t is increased from 0.0 to 1.0, the cluster coverage in different mobility scenarios gains an average increase of 10%. When the size balancing scheme is applied in the cluster construction process, clusters are expanded evenly in all directions, and hence has higher possibilities to cover more nodes in the network.

VIII. Conclusion

In this paper, we have presented a scheme to construct multi-hop clusters in MANETs, based on the neighborhood benchmark (NB) scores of mobile nodes, to provide more flexible and stable clustered network structure for efficient data collection or dissemination. We construct multi-hop clusters by letting every node autonomously select its clusterhead based on the NB scores and handshake with the clusterhead. Hence, we guarantee that all the clusters constructed are connected. We also present a partial probability-based approach to control the possible deviation of cluster sizes. The results of intensive simulation have indicated that our clustering scheme can provide stable clustered network structures with balanced cluster sizes in various network scenarios, and provide users with the flexibility of controlling the cluster radius adaptively for different applications.

Acknowledgement

This work was supported by the National Science Foundation under grant number ITR-CYBERTRUST 0430565.

References

[1] E. M. Belding-Royer, "Hierarchical routing in ad-hoc mobile networks", *Wireless Communications and Mobile Computing*, vol. 2(5), 2002, pp. 515-532.

[2] J. Yu and P. Chong, "A survey of clustering schemes for mobile ad hoc networks", *IEEE Communication Surveys & Tutorials*, vol. 7(1), 2005, pp. 32-48.

[3] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling", *Proc. IEEE*, vol. 75(1), 1987, pp. 56-73.

[4] J. Wu, and H. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks", *Proc. 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, 1999, pp. 7-14.

[5] Y.-Z. Chen, and A. Liestman, "Approximating Minimum Size Weakly-Connected Dominating Sets for Clustering Mobile Ad Hoc Networks", *Proc. MobiHoc*, 2002, pp. 165-172.

[6] C.-C. Chiang, et al., "Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel", *Proc. IEEE Singapore International Conference on Networks (SICON)*, 1997, pp. 197-211.

[7] A. D. Amis, et al., "Max-min D-cluster formation in wireless ad hoc networks", *Proc. IEEE INFOCOM*, 2000, pp. 32–41.

[8] J.T. Tsai, and M. Gerla, "Multicluster, Mobile, Multimedia Radio Network", *ACM/Kluwer Journal of Wireless Networks*, vol. 1(3), 1995, pp. 255-265.

[9] M. Chatterjee, S. K. Das and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks", Journal of Cluster Computing, vol. 5(2), 2002, pp. 193-204.

[10] L. Ramaswamy, B. Gedik, and L. Liu, "A Distributed Approach to Node Clustering in Decentralized Peer-to-Peer Networks", *IEEE Trans. on Parallel and Distributed Systems*, vol. 16(9), 2005, pp. 814-829.

[11] A. McDonald, and T. Znati, "A mobility-based framework for adaptive clustering in wireless ad hoc networks", *IEEE Journal on Selected Areas in Communications*, vol. 17(8), 1999, 1466-1487.

[12] B. An and S. Papavassiliou, "A Mobility-based Clustering Approach to Support Mobility Management and Multicast Routing in Mobile Ad-hoc Wireless Networks", *Int'l Journal of Network Management*, vol. 11(6), 2001, pp. 387-395.

[13] D. Kim, S. Ha, and Y. Choi, "k-hop cluster-based dynamic source routing in wireless ad-hoc packet radio networks", *Proc. IEEE Vehicular Technology Conference (VTC)*, 1998, pp. 224-228.

[14] F. G. Nocetti, J. S. Gonzalez, and I. Stojmenovic, "Connectivity based k-hop clustering in wireless networks", *Telecommunication Systems*, vol. 22(1-4), 2003, pp. 205-220.

[15] J. Wu, and W. Lou, "Forward node set based broadcast in clustered mobile ad hoc networks", *Wireless Communications and Mobile Computing*, vol. 3(2), 2003, pp. 155-173.