FIRST QUARTER 2005, VOLUME 7, NO.1

IEEE COMMUNICATIONS SURVEYS
The Electronic Magazine of Original Peer-Reviewed Survey Articles

www.comsoc.org/pubs/surveys

# A SURVEY OF CLUSTERING SCHEMES FOR MOBILE AD HOC NETWORKS

JANE Y. YU AND PETER H. J. CHONG, NANYANG TECHNOLOGICAL UNIVERSITY

## ABSTRACT

Clustering is an important research topic for mobile ad hoc networks (MANETs) because clustering makes it possible to guarantee basic levels of system performance, such as throughput and delay, in the presence of both mobility and a large number of mobile terminals. A large variety of approaches for ad hoc clustering have been presented, whereby different approaches typically focus on different performance metrics. This article presents a comprehensive survey of recently proposed clustering algorithms, which we classify based on their objectives. This survey provides descriptions of the mechanisms, evaluations of their performance and cost, and discussions of advantages and disadvantages of each clustering scheme. With this article, readers can have a more thorough and delicate understanding of ad hoc clustering and the research trends in this area.

With the proliferation in personal computing devices and the development in wireless communication technologies, ad hoc wireless networks have gained worldwide attention in recent years. The great popularity of Internet services makes more people enjoy and depend on the networking applications. However, the Internet is not always available and reliable, and hence it cannot satisfy people's demand for networking communication at anytime and anywhere. MANETs, without any fixed infrastructures, allow mobile terminals to set up a temporary network for instant communication. Hence, MANETs bear great application potential in these scenarios, including disaster and emergency relief, mobile conferencing, sensor dust, battle field communication, and so on [1, 2].

Dynamic routing is almost the most important issue in MANETs. However, it has been proved that a flat structure exclusively based on proactive or reactive routing schemes cannot perform well in a large dynamic MANET [3–5]. In other words, a flat structure encounters scalability problems with increased network size, especially in the face of node mobility at the same time. This is due to their intrinsic characteristics. The communication overhead of link-based proactive routing protocols is $O(n^2)$, where $n$ is the total number of mobile terminals in a network [6]. This means that the routing overhead of such an algorithm increases with the square of the number of mobile nodes in a MAENT. For a reactive routing scheme, the disturbing RREQ (route request) flooding over the whole network and the considerable route setup delay become intolerable in the presence of both a large number of nodes and mobility. Consequently, a hierarchical architecture is essential for achieving a basic performance guarantee in a large-scale MANET [1].

Since a cluster structure is a typical hierarchy, many papers focus on presenting an effective and efficient clustering scheme for MANETs. However, until now no overviews of ad hoc clustering issues have been presented. In this article we present a comprehensive survey of several proposed clustering schemes for MANET research, and classify and analyze these schemes based on their main objectives.

## CLUSTERING SCHEME OVERVIEW

### WHAT IS CLUSTERING?

In a clustering scheme the mobile nodes in a MANET are divided into different virtual groups, and they are allocated geographically adjacent into the same cluster according to some rules with different behaviors for nodes included in a cluster from those excluded from the cluster. A typical cluster structure is shown in Fig. 1. It can be seen that the nodes are divided into a number of virtual groups (with the dotted lines) based on certain rules. Under a cluster structure, mobile nodes may be assigned a different status or function, such as clusterhead, clustergateway, or clustermember. A clusterhead normally serves as a local coordinator for its cluster, performing intra-cluster transmission arrangement, data forwarding, and so on. A clustergateway is a non-clusterhead node with inter-cluster links, so it can access neighboring clusters and

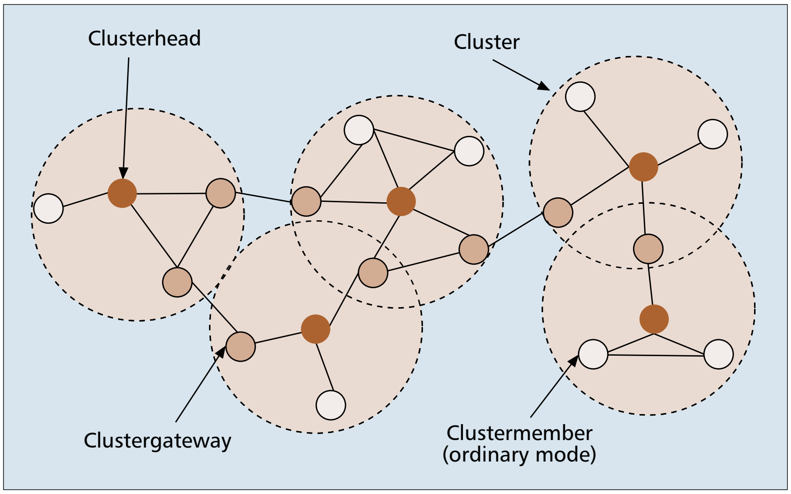



**Figure 1.** *Cluster structure illustration.*

forward information between clusters. A clustermember is usually called an ordinary node, which is a non-clusterhead node without any inter-cluster links.

## WHY DO AD HOC NETWORKS REQUIRE CLUSTERING?

It has been shown that cluster architecture guarantees basic performance achievement in a MANET with a large number of mobile terminals [1, 6]. A cluster structure, as an effective topology control means [7], provides at least three benefits [8–10]. First, a cluster structure facilitates the spatial reuse of resources to increase the system capacity [9, 11]. With the non-overlapping multicluster structure, two clusters may deploy the same frequency or code set if they are not neighboring clusters [10]. Also, a cluster can better coordinate its transmission events with the help of a special mobile node, such as a clusterhead, residing in it. This can save much resources used for retransmission resulting from reduced transmission collision. The second benefit is in routing, because the set of clusterheads and clustergateways can normally form a virtual backbone for inter-cluster routing, and thus the generation and spreading of routing information can be restricted in this set of nodes [12, 13]. Last, a cluster structure makes an ad hoc network appear smaller and more stable in the view of each mobile terminal [8]. When a mobile node changes its attaching cluster, only mobile nodes residing in the corresponding clusters need to update the information [14, 15]. Thus, local changes need not be seen and updated by the entire network, and information processed and stored by each mobile node is greatly reduced.

## WHAT IS THE COST OF CLUSTERING?

As we know, clustering is important for a network to achieve scalability in the presence of a large number of mobile nodes and high mobility. However, a cluster-based MANET has its side effects and drawbacks because constructing and maintaining a cluster structure usually requires additional cost compared with a flat-based MANET. The cost of clustering is a key issue to validate the effectiveness and scalability enhancement of a cluster structure. By analyzing the cost of a clustering scheme in different aspects qualitatively or quantitatively, its usefulness and drawbacks can be clearly specified. The clustering cost terms are described as follows:

• To maintain a cluster structure in a dynamically changing scenario often requires explicit message exchange between mobile node pairs [9, 16–28]. When the underlying network topology changes quickly and involves many mobile nodes, the clustering-related information exchange increases drastically. Frequent information exchange may consume considerable bandwidth and drain mobile nodes' energy quickly so that upper-layer applications cannot be implemented due to the inadequacy of available resources or the lack of support from related mobile nodes.

• Some clustering schemes may cause the cluster structure to be completely rebuilt over the whole network when some local events take place, e.g. the movement or "die" of a mobile node, resulting in some clusterhead re-election (re-clustering) [16, 18–20, 22, 24, 25, 28, 29]. This is called the ripple effect of re-clustering. In other words, the ripple effect of re-clustering indicates that the re-election of one clusterhead may affect the structure of many clusters and arouse the clusterhead re-election over the network [18]. Thus, the ripple effect of re-clustering may greatly affect the performance of upper-layer protocols.

• In addition, most schemes separate the clustering into two phases, cluster formation and cluster maintenance, and assume that mobile nodes keep static when cluster formation is in progress [9, 5, 6, 8–21, 24, 25]. This is because for the initial cluster formation of these schemes, a mobile node can decide to become a clusterhead *only* after it exchanges some specific information with its neighbors and assures that it holds some specific attribute in its neighborhood. With a frozen period of motion, each mobile node can obtain accurate information from neighboring nodes, and the initial cluster structure can be formed with some specific characteristics. However, this assumption may not be applicable in an actual scenario [29, 30], where mobile nodes may move randomly all the time.

• Another metric is the computation round, which indicates the number of rounds in which a cluster formation procedure can be completed. For clustering schemes relying on a frozen period of motion assumption, the computation round is an important metric since the more rounds that a clustering scheme requires for its cluster formation, the longer the frozen period that is required for mobile nodes. But in fact, the topology of a MANET changes frequently with the movement of mobile nodes. For most of the clustering schemes, their cluster formation procedure can be performed in parallel in the whole network, which should result in fast time convergence in cluster formation. But in these schemes, not all mobile nodes can decide their status at the same time (within one round), and they may require a non-constant number of rounds to finish the initial cluster construction. Thus, the time required for these algorithms cannot be bounded and may vary noticeably for different network topologies.

Hence, the required explicit control message exchange, the ripple effect of re-clustering, and the stationary assumption for cluster formation are the main costs of a cluster-based MANET compared with a MANET with a flat structure. In the study of clustering schemes in this article, we will discuss their cost in the categories shown in Table 1 so that readers can gain a better understanding and make a thorough evaluation of the clustering schemes.

## HOW TO CLASSIFY CLUSTERING SCHEMES?

The clustering schemes of MANETs can be classified according to different criteria. For example, depending on whether a special mobile node with extra functions, named a *clusterhead*, is required for a cluster, clustering protocols can be classified as clusterhead-based clustering [16–28] and non-clusterhead-

| Cost of clustering | Definition and description |
|---|---|
| Explicit control message for clustering | Clustering requires explicit clustering-related information exchanged between node pairs. Clusters cannot be formed or maintained by non-clustering-related messages, such as routing information or data packets. |
| Ripple effect of re-clustering | Ripple effect indicates that the re-election of a single clusterhead may affect the cluster structure of many other clusters and completely alter the cluster topology over the whole network. |
| Stationary assumption for cluster formation | Mobile nodes must be assumed static in the cluster formation phase so that mobile nodes are able to obtain accurate neighbor information and cluster structure can be promised with specific attributes. |
| Constant Computation round | Computation round is the number of rounds that a cluster formation procedure can be completed. The non-constant computation round of a clustering scheme indicates its unbounded time complexity. Only clustering schemes that require the *frozen period of motion* assumption need to consider this metrics. |
| Communication (message) complexity | Communication complexity represents the total amount of clustering-related message exchanged for the cluster formation. For clustering schemes with ripple effect, the communication complexity for the re-clustering in the cluster maintenance phase may be the same as that in the cluster formation phase. But for those with no ripple effect, the communication complexity of re-clustering should be much lower. |

Table 1. *Description of cost terms for clustered-based MANET.*

| | Objectives |
|---|---|
| DS-based clustering | Finding a (weakly) connected dominating set to reduce the number of nodes participating in route search or routing table maintenance. |
| Low-maintenance clustering | Providing a cluster infrastructure for upper layer applications with minimized clustering-related maintenance cost. |
| Mobility-aware clustering | Utilizing mobile nodes' mobility behavior for cluster construction and maintenance and assigning mobile nodes with low relative speed to the same cluster to tighten the connection in such a cluster. |
| Energy-efficient clustering | Avoiding unnecessary energy consumption or balancing energy consumption for mobile nodes in order to prolong the lifetime of mobile terminals and a network. |
| Load-balancing clustering | Distributing the workload of a network more evenly into clusters by limiting the number of mobile nodes in each cluster in a defined range. |
| Combined-metrics-based clustering | Considering multiple metrics in cluster configuration, including node degree, mobility, battery energy, cluster size, etc., and adjusting their weighting factors for different application scenarios. |

Table 2. *Summary of six clustering schemes*.

based clustering [8, 9]. Or, based on the hop distance between node pairs in a cluster, clustering schemes can be divided into 1-hop clustering [16–22, 25, 26, 28] and multi-hop clustering [23, 27].

In this article we classify the clustering protocols based on their objectives. According to this criterion, the proposed clustering schemes for MANETs can be grouped into six categories, as shown in Table 2. Dominating-Set-based (DS-based) clustering [16, 17, 31, 32] tries to find a DS for a MANET so that the number of mobile nodes that participate in route search or routing table maintenance can be reduced. This is because only mobile nodes in the DS are required to do so. Low-maintenance clustering schemes [9, 18, 21, 29] aim at providing stable cluster architecture for upper-layer protocols with little cluster maintenance cost. By limiting re-clustering situations or minimizing explicit control messages for clustering, the cluster structure can be maintained well without excessive consumption of network resources for cluster maintenance. Mobility-aware clustering [8, 22, 23] takes the mobility behavior of mobile nodes into consideration. This is because the mobile nodes' movement is the main cause of changes to the network topology. By grouping mobile nodes with similar speed into the same cluster, the intra-cluster links can be greatly tightened and the cluster structure can be correspondingly stabilized in the face of moving mobile nodes. Energy-efficient clustering [24–26] manages to use the battery energy of mobile nodes more wisely in a MANET. By eliminating unnecessary energy consumption of mobile nodes or by balancing energy consumption among different mobile nodes, the network lifetime can be remarkably prolonged. Load-balancing clustering schemes [24, 27] attempt to limit the number of mobile nodes in each cluster to a specified range so that clusters are of similar size. Thus, the network loads can be more evenly distributed in each cluster. Combined-metrics-based clustering [28] usually consider multiple metrics, such as node degree, cluster size, mobility speed, and battery energy, in cluster configuration, especially in clusterhead decisions. With the consideration of more parameters, clusterheads can be more properly chosen without giving bias to mobile nodes with specific attributes. Also, the weighting factor for each parameter can be adaptively adjusted in response to different application scenarios. Based on this classification, we can more easily study the common criteria underlying each category, and the similarities and differences between schemes in the same category. At the same

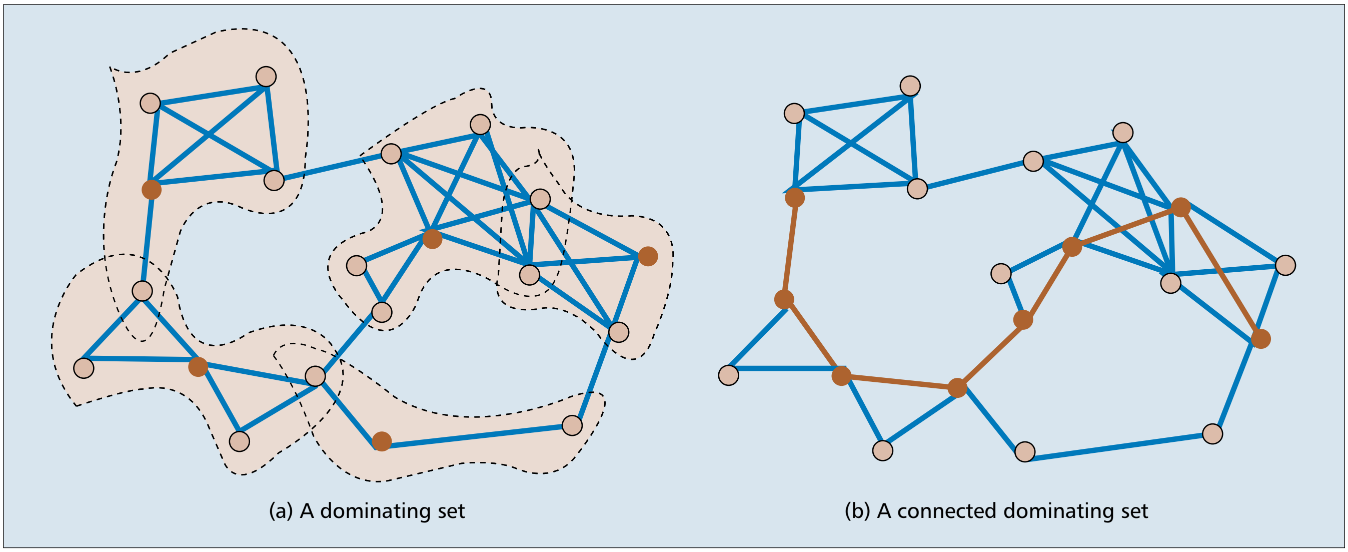



**Figure 2.** *Dominating set illustration.*

time, we can find out the best application scenario for each clustering category. The brief summary of these six categories is presented in Table 2.

## Classifying Clustering Schemes

In this section, typical clustering schemes of MANETs are classified into the six categories listed in Table 2 and are studied based on their objectives.

### DS-Based Clustering

Routing based on a set of dominating nodes [16, 31, 32], which function as the clusterheads to relay routing information and data packets, is a typical technique in MANETs. Such a set of nodes is called a DS. Taking a MANET as an un-weighted graph $G$, a vertex (node) subset $S$ of $G$ is a DS if each vertex in $G$ either belongs to $S$ or is adjacent to at least one vertex in $S$. For example, in Fig. 2a the black vertices form a DS. Each area surrounded by the dash line is a dominating area of some dominating node. Then we can utilize the vertices of a DS as clusterheads and assign each vertex to a cluster corresponding to a vertex that dominates it [16, 31]. A DS is called a connected DS (CDS) if all the dominating nodes are directly connected with each other. In Fig. 2b the black vertices form a CDS and the black lines indicate the corresponding induced subgraph of the CDS.

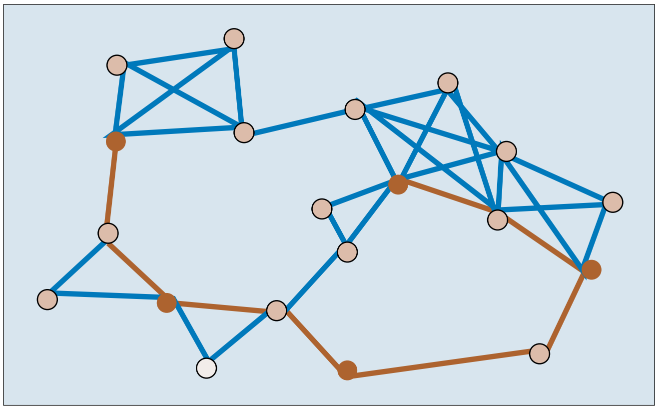
The idea of finding a CDS for a MANET comes from the fact that the routing process is only aggregated on mobile nodes in the DS [16]. Hence, when table-driven routing is applied, only nodes in the CDS are required to construct and maintain the routing tables. When on-demand routing is adopted, the route search space is limited to the CDS. However, to keep a DS connected and with approximately minimum size is not a trivial task in a dynamic environment.

***Wu's CDS (Connected Dominating Set) Algorithm*** — Wu [16] proposed a distributed algorithm to find a CDS in order to design efficient routing schemes for a MANET. Initially, every node $v$ exchanges its neighbor list with all its neighbors. A mobile node sets itself as a dominating node if it has at least two unconnected neighbors. This is called a *marking* process. Then some extension rules are implemented to reduce the size of a CDS generated from the marking process. A node deletes itself from the CDS when its close (open) neighbor set is completely included in the neighbor set(s) of a (two connected) neighboring dominating node(s) and it has smaller ID than the neighboring dominating node(s). The close neighbor set of a mobile node includes all its direct neighbors as well as itself, whereas the open neighbor set only includes a mobile node's direct neighbors. The connection relationship of mobile nodes may change and the CDS needs to be updated if any one or more of these events occur, including switch-on, switch-off, and movement of mobile nodes.

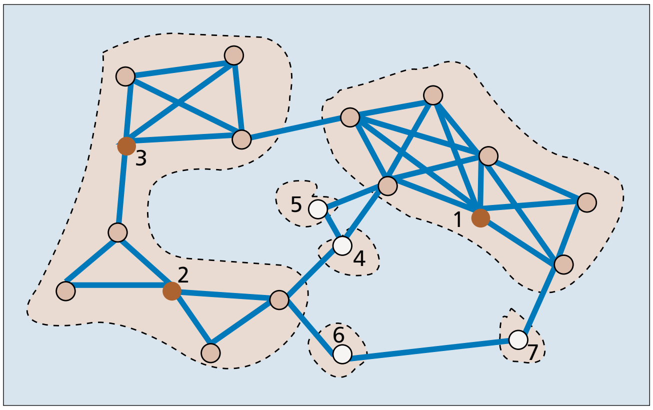
A CDS with small size reduces the number of nodes involved in routing-related tasks. The extension rules proposed in Wu's algorithm are effective for reducing the size of the DS [16]. Also, Wu's algorithm promises that the cluster construction can be completed in just two rounds, one round for the marking procedure and the other round for extension rules, so that the time complexity of this algorithm can be highly bounded.

For CDS maintenance, any moving node needs to continue to send out a beacon message every $\tau$ second during its movement. Other related mobile nodes keep monitoring the messages from that moving node. Therefore, a single mobile node's movement may suppress many mobile nodes from transmitting or receiving their own packets. Also, if many mobile nodes in the network are in movement, the network topology may be greatly affected and thus, the complete recalculation of a CDS with a large amount of message exchange is required. Thus we can conclude that the cluster maintenance events in Wu's CDS algorithm are expensive in terms of transmission efficiency and control overhead. Since no simulation performances for moving scenarios are provided, the effectiveness of the cluster maintenance mechanisms of Wu's scheme under a dynamic environment is with doubt.

***Chen's WCDS (Weakly Connected Dominating Set) Algorithm*** — Chen [17] pointed out that although a CDS provides explicit information for inter-cluster routing, the number of clusters produced by the CDS clustering is rather large and the formed cluster structure is likely highly overlapping. Chen proposed schemes to build a weakly CDS (WCDS) by relaxing the requirement of direct connection between neighboring dominating nodes. Normally, a WCDS is smaller in size than a CDS, and hence it is better for further simplifying the network structure. The induced subgraph of a WCDS, which can be adopted for routing, includes all the dominating nodes, some non-dominating nodes, each of which can link two unconnected dominating nodes together just by itself, and the



■ **Figure 3.** *Network structure from a WCDS.*




■ **Figure 4.** *A snapshot of pieces.*

links between all these mobile nodes. In Fig. 3, the black vertices form a WCDS, and the black lines indicate paths that can be used for inter-cluster data relay. Compared with Fig. 2b, the size of the DS is decreased from 8 to 5.

Five algorithms have been proposed in [17]. Because of space limitation, here we only discuss Algorithm I and V. In both algorithms, each vertex (mobile node) in a graph *G* is associated with a color (white, gray, or black) and a piece. Initially all vertices are white-colored. When a vertex is colored black, all its direct white neighbors are changed to gray. *Piece* is a term to describe a substructure of a graph. A *white piece* is just a single white-colored vertex. A *black piece* includes a black vertex plus all its direct gray-colored neighbors. If two black vertices can be connected through a single gray vertex, their corresponding black pieces can be merged into a larger one. Hence, a black piece refers to a maximal set of black vertices, which can form a WCDS, plus all their gray-colored neighbors. The idea of both algorithms is to elect the optimal white/gray to dye black, and gradually grow the vertices of a graph into a single black piece. Finally, the black vertices form a WCDS.

In Algorithm I, a gray/white vertex with the maximum improvement over the whole network is chosen each time to change to black. The improvement of a gray/white vertex indicates the number of pieces that would be merged into a single black piece if that vertex is colored black. The process continues until all vertices are included into a single black piece. For example, Fig. 4 shows the piece structure of a graph after the third iteration of coloring. Then we can see that dyeing vertex 4 to black would merge four pieces together. However, dyeing any other vertices, 5, 6, or 7, would merge only three pieces. Thus, vertex 4 would be selected to dye black in the fourth iteration.

Later, Chen proposed a fully distributed approach, namely, Algorithm V for the WCDS construction. A mobile node can be considered as a *candidate* of its piece for dyeing black if it has the maximum improvement in its closed neighborhood. Therefore, a black piece may have more than one candidate at the same time. In each run, by choosing the best candidate in each piece to color black, the dyeing procedure can be performed in parallel in the network. A newly merged black piece always chooses the ID of its newly colored black node as the piece ID. The algorithm ends when a new black piece finds that its best candidate has an improvement value of 1.

The size of the DS in Algorithms I and V is reduced compared with that of a typical CDS [17]. However, how to maintain the WCDS structure as the network topology changes is not addressed in [17]. Algorithm I only allows one mobile node to be colored black in each run. This may cause slow convergence on WCDS construction. Also, as a centralized algorithm, Algorithm I probably requires a powerful centralized device to grasp the information of all mobile nodes. But this is not feasible for a typical MANET, where all terminals are homogeneously equipped and with limited computing capability. Compared with Algorithm I, Algorithm V brings faster WCDS construction because it allows the dyeing procedure to perform in parallel. The size of the constructed WCDS of Algorithm V is shown to be very close to that of Algorithm I [17] as well. However, in Algorithm V, a piece can decide its best candidate for dyeing only after it collects the complete "improvement" information from all vertices inside it. As the piece size grows, this procedure needs a large amount of message exchange, and the information collection delay may become significant. In addition, it is difficult to calculate the computation and communication complexity for Algorithm V because it requires non-constant rounds to complete the dyeing procedure and a vertex may change its color and piece ID in each round. We feel the communication complexity for this algorithm should be rather high since each vertex needs to inform its neighbors whenever it changes its color or piece ID, and the way to collect the "improvement" information is not a trivial task.

***Summary of DS-based Clustering*** — Features and objectives of the two DS-based clustering schemes are summarized in Table 3. Only clusterhead-based clustering bears the attribute of "1-hop" or "multi-hop" in this article. The term 1-hop clustering means that the hop distance between any clusterhead and its member nodes is limited to one hop. Multi-hop clustering indicates that there is no hop limit between a clusterhead and its member nodes, and normally the distance between a clusterhead and its farthest cluster-member is larger than one hop.

The objective of the two DS-based clustering schemes is to attempt to select a small number of mobile nodes as dominating nodes to form the DS, which can form an inter-cluster routing backbone in a network. Both schemes form 1-hop clusters with dominating nodes serving as clusterheads. Compared with Wu's algorithm, Chen's algorithm can form fewer clusters, resulting in less overlapping cluster architecture by relaxing the direct connection requirement between dominating nodes. Hence, Chen's algorithm should be more effective in simplifying a network structure.

The cost comparison of the two algorithms is shown in Table 4. The symbol $|V|$ represents the number of mobile nodes in a network. When calculating the communication complexity of a clustering algorithm, the assumption is that a mobile node transmits in all directions so that each message is a local broadcast [31] and can be received by all direct neighbors of the sending node. The communication complexity of Wu's algorithm only considers the marking procedure.

Wu's and Chen's schemes both require explicit message

| | With clusterhead? | 1-hop or multi-hop | Objectives |
|---|---|---|---|
| Wu [16] | Yes | 1-hop | Find a minimum number of nodes as dominating nodes to construct a CDS. Decrease the number of nodes participating in routing and reduce the DS size to eliminate unnecessary dominating nodes without breaking the direct connection between neighboring dominating nodes. |
| Chen [17] | Yes | 1-hop | Find a minimum number of nodes as dominating nodes to construct a WCDS. Decrease the number of nodes participating in routing and further reduce the DS size by relaxing the direct connection requirement between dominating nodes. |

■ Table 3. *Summary of DS-based clustering schemes.*

| | Explicit control message for clustering | Ripple effect of re-clustering | Stationary assumption for cluster formation | Constant computation round | Communication complexity |
|---|---|---|---|---|---|
| Wu [16] | Yes | Yes | Yes | Two rounds | |
| Chen [17] | Yes | N/A | Yes | No | $> O(2\|V\|)$ |

■ Table 4. *Cost comparison of DS-based clustering schemes.*

exchange for clustering, and have a high restriction on the DS structure (connected/weakly connected). Hence, local network topology updates may require global adjustment of the structure of CDS/WCDS, although Chen does not address cluster maintenance mechanisms in his article. In other words, local clusterhead re-election may cause a ripple effect of re-clustering globally, and cause large communication overhead for the maintenance. Thus, DS-based clustering is more feasible for static networks or networks with low mobility. The marking process of Wu's algorithm can be completed with $O(2|V|)$ messages, because in this procedure each mobile node needs to send out only two messages, one for its neighbor list broadcast and the other for its cluster-related status claim. Wu's algorithm is less complex than Chen's in terms of computation and communication, especially with the number of rounds needed for the initial cluster construction. This is because Wu's algorithm could be completed in just two rounds while Chen's (Algorithm V) probably needs a larger and non-constant number of rounds.

## Low-Maintenance Clustering

The main criticism of a clustered network comes from the need for extra explicit message exchange among mobile nodes for maintaining the cluster structure. When network topology changes frequently, resulting in frequent cluster topology updates, the control overheads for cluster maintenance increases drastically. Thus, clustering behavior may consume a large portion of network bandwidth, drain mobile nodes' energy quickly, and override its improvement on network scalability and performance [29]. Hence, it is important to reduce the communication overhead caused by cluster maintenance. Most low-maintenance clustering protocols aim at providing stable cluster architecture by reducing the re-affiliation rate and especially minimizing re-clustering situations. This is because re-clustering is more disturbing than re-affiliation since it causes more communication overhead, route invalidation, and even ripple effect. Here, re-affiliation refers to a non-clusterhead changing the attached cluster without affecting the corresponding clusterhead(s). By limiting re-affiliation and re-clustering events, the clustering-related control overhead can be reduced accordingly. A newly proposed scheme tries to eliminate the control overhead for clustering completely by constructing and maintaining cluster architecture based on data traffic forwarding [29], which means the cluster-related status for a mobile node can be naturally updated when data packets pass.

***LCC (Least Cluster Change)*** — **LCC** [18] is considered to be a significant enhancement of Lowest ID Clustering (LIC) [19] or Highest Connectivity Clustering (HCC) [20]. Before LCC was proposed, most protocols executed the clustering procedure periodically, and re-clustered from time to time in order to satisfy some specific attribute of clusterheads. In HCC, the clustering scheme is performed periodically to check the "local highest node degree" attribute of a clusterhead. When a clusterhead finds a member node with a higher degree, it is forced to relinquish its clusterhead role. This mechanism, of course, involves frequent re-clustering.

In LCC the clustering algorithm is divided into two steps: cluster formation and cluster maintenance. The cluster formation simply follows LIC, i.e. initially mobile nodes with the lowest ID in their neighborhoods are chosen as clusterheads. Re-clustering is event-driven and invoked in only two cases:
- When two clusterheads move into the reach range of each other, one gives up the clusterhead role.
- When a mobile node cannot access any clusterhead, it rebuilds the cluster structure for the network according to LIC.

Hence, LCC significantly improves cluster stability by relinquishing the requirement that a clusterhead should always bear some specific attributes in its local area. But the second case of re-clustering in LCC indicates that a single node's movement may still invoke the complete cluster structure re-computation, and once this happens, the large communication overhead for clustering may not be avoided.

***3hBAC (3-hop Between Adjacent Clusterheads)*** — **3hBAC** [21] forms a 1-hop non-overlapping cluster structure with three hops between neighboring clusterheads by the introduction of a new node status, named clusterguest. Clusterguest is a mobile node that cannot directly connect to any clusterhead, but can access some cluster with the help of a clustermember.

The cluster construction procedure always begins from the neighborhood of the mobile node with the lowest ID (assuming it is mobile node $m_0$) in a MANET. The mobile node with the highest node degree in $m_0$'s closed neighbor set is chosen to be the first clusterhead. All the direct neighbors of this

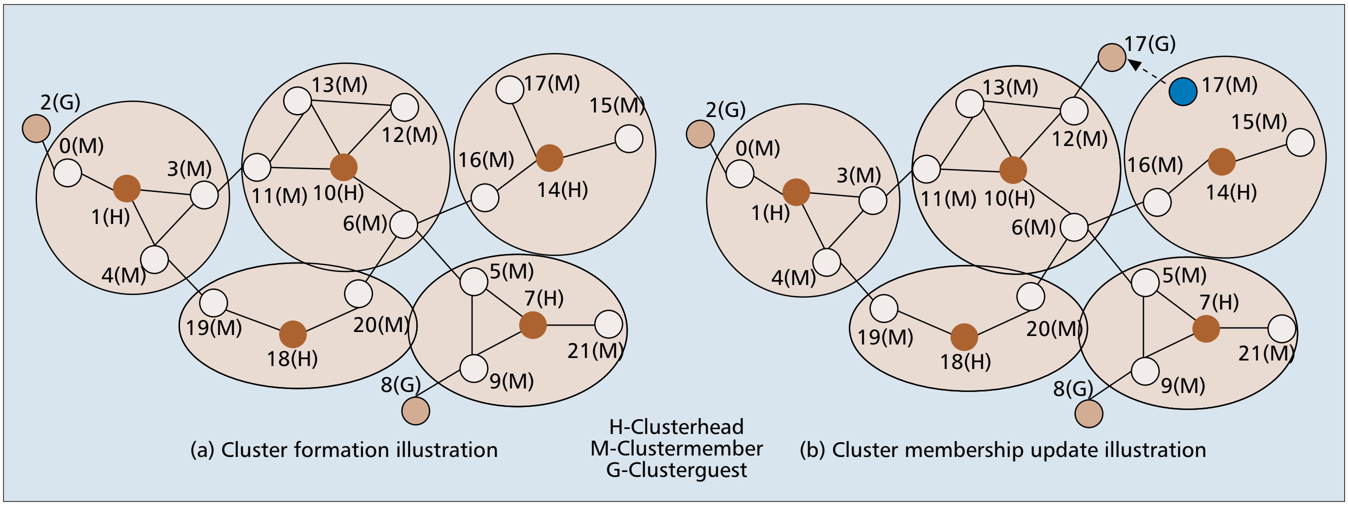



**Figure 5.** *Clustering scheme of 3hBAC.*

clusterhead change status to "clustermember." After the completion of the first cluster, the cluster formation procedure can be performed in parallel in the network. A clustermember or a direct neighbor of any clustermember with status "unspecified" (indicating it is not included in any cluster yet) are declined the capability of serving as a clusterhead. A mobile node, without being declined clusterhead capability, declares as a new clusterhead when it is with the highest node degree in its neighborhood. When a mobile node finds out that it cannot serve as a clusterhead or join a cluster as a clustermember, but some neighbor is a clustermember of some cluster, it joins the corresponding cluster as a clusterguest. Figure 5a shows an initially established cluster structure by 3hBAC. In Fig. 5a we can see that mobile node $m_2$ is a clusterguest of cluster $C_1$ and is two-hops away from its clusterhead $m_1$. Also, clustermember $m_0$ of cluster $C_1$ can help $m_2$ communicate with clusterhead $m_1$.

For cluster maintenance, 3hBAC keeps the adjacent clusterheads at least two-hops away. So when two clusterheads move into the reach range of each other, one is required to give up its clusterhead role. With the help of clusterguest, 3hBAC does not raise ripple effect when re-clustering, which means the clusterhead re-election will have no affect on the status of mobile nodes outside these two clusters. For another case, when a mobile node moves out of the ranges of all clusters, it can join a cluster as a clusterguest if it can reach some clustermember(s) of that cluster. Hence, there is no need to form new clusters in order to cover such a single node as in LCC and the cluster topology stays unchanged. For example, in Fig. 5b, when mobile node $m_{17}$ moves to the new location, it cannot be covered by its former clusterhead or any other clusterhead, but it can join cluster $C_{10}$ as a clusterguest since it can directly communicate with clustermember $m_{12}$ of $C_{10}$.

Some papers have proved that most 1-hop clustering schemes form a highly overlapping cluster structure [18, 29] with a large number of small clusters. A highly overlapping structure may cause difficulty in channel spatial reuse [10], lead to inefficient flooding [29], and likely form longer hierarchical routes. In 3hBAC, a non-overlapping cluster structure can be achieved with the introduction of clusterguest. This can reduce the number of clusters and eliminate small unnecessary clusters. In the cluster maintenance phase, 3hBAC invokes the clusterhead update in very limited cases, and brings no ripple effect [21]. Thus, 3hBAC successfully satisfies its cluster-stability-aware objective. However, the stationary assumption is required for 3hBAC's cluster formation, since it is mandatory to exchange information between direct neighbors to decide the first cluster and to update mobile nodes' clusterhead serving capability accordingly for following procedures.

***Lin's Algorithm*** — **Lin** [9] proposed an adaptive clustering scheme to form a non-overlapping cluster architecture without clusterheads because clusterheads bear extra work compared with ordinary member nodes and likely will become the bottlenecks of a network.

In Lin's scheme, every mobile node $i$ keeps its own ID and the ID of its direct neighbors in a set $\Gamma_i$. Each mobile node that declares to be a clusterhead will set its own ID as its cluster ID (CID). Initially, mobile nodes with the lowest IDs in their local area become clusterheads. A mobile node $i$ can broadcast its own CID information only when it is with the lowest ID in its set $\Gamma_i$. The CID information includes a mobile node's ID and CID. If the mobile node's ID is the same as its CID, then the CID information is a clusterhead claim; otherwise, it is just an ordinary node claim. When a mobile node $i$ receives CID information from a neighbor $j$, it deletes $j$ from its set $\Gamma_i$. If the CID information from $j$ is a clusterhead claim, the mobile node checks its own CID attribute. If its CID is unspecified (it is not involved in any cluster yet) or larger than the ID (CID) of $j$, it sets $j$ as its clusterhead. The process continues till all mobile nodes access some cluster. This mechanism promises small communication overhead for building clusters because each mobile node broadcasts only one CID message for the cluster construction.

After cluster formation is completed, clusterheads are no longer used in any further cluster maintenance phase. Lin's algorithm requires that mobile nodes in the same cluster are mutually reachable with at most two hops. In the maintenance phase, when a mobile node $i$ finds out that the distance between itself and some node $j$ in the same cluster becomes 3-hop, it invokes a cluster maintenance mechanism. If node $i$ is a direct neighbor of the node with the highest intra-cluster connectivity in its cluster, it remains in the cluster and removes node $j$; otherwise, it joins a neighboring cluster. As soon as there is no proper cluster to join, it forms a new cluster to cover itself. Since this mechanism likely forms new clusters but without any cluster elimination or merge mechanisms, the cluster size decreases and the number of clusters increases as time advances. Eventually, almost every mobile node forms a single-node cluster, and the cluster structure disappears [10].

***PC (Passive Clustering)*** — Conventional clustering schemes require all the mobile nodes to advertise cluster-dependent information repeatedly to build and maintain the cluster

| | With clusterhead? | 1-hop or multi-hop | Objectives |
|---|---|---|---|
| LCC | Yes | 1-hop | Limiting re-clustering situations and reducing clustering control overhead. |
| 3hBAC | Yes | 1-hop | Limiting re-clustering situations, reducing clustering control overhead, and eliminating ripple effect of re-clustering; producing non-overlapping cluster structure and eliminating unnecessary small clusters. |
| Lin [9] | Clusterhead only for cluster formation | N/A | Limiting re-clustering situations and reducing clustering control overhead; eliminating clusterheads in cluster maintenance to avoid overloaded nodes. |
| PC | Yes | 1-hop | Eliminating explicit control message for clustering and forming and maintaining a cluster structure only when some mobile nodes have packets to send; suppressing the number of gateways to achieve flooding efficiency. |

■ Table 5. *Summary of low-maintenance clustering.*

| | Explicit control message for clustering | Ripple effect of re-clustering | Stationary assumption for cluster formation | Constant computation round | Communication complexity |
|---|---|---|---|---|---|
| LCC | Yes | Yes | Yes | No | $O(m\|V\|)$ |
| 3hBAC | Yes | No | Yes | No | $O(3\|V\|)$ |
| Lin [9] | Yes | No | Yes | No | $O(\|V\|)$ |
| PC | No | Yes | No | Unnecessary to consider | Zero |

■ Table 6. *Cost comparison of low-maintenance clustering.*

structure, and thus clustering is one of the main sources of control overhead. PC is a clustering protocol that does not use dedicated clustering-protocol-specific control packets or signals [29], so it is called Passive Clustering. In PC, a mobile node can be in one of the following four states: *initial*, *clusterhead*, *gateway*, and *ordinary node*. All the mobile nodes are with 'initial' state at the beginning. Only mobile nodes with "initial" state have the potential to be clusterheads. When a *potential clusterhead* with "initial" state has something to send, such as a flood search, it declares itself as a clusterhead by piggybacking its state in the packet. Neighbors can learn the clusterhead claim by monitoring the "cluster state" in the packet, and then record the CID and the packet receiving time. A mobile node that receives a claim from just one clusterhead becomes an ordinary node, and a mobile node that hears more claims becomes a gateway. Since PC does not send any explicit clustering-related message to maintain the cluster structure, each node is responsible for updating its own cluster status by keeping a timer. For example, when an ordinary node does not receive any packet from its clusterhead for a given period, its status reverts to "initial."

Also, PC proposes a heuristic to control the number of gateways in a local area without breaking its passive nature. The main idea is that a mobile node serves as a clustergateway only when the number difference between clustergateways and clusterheads, $|D_{H-G}|$, in its coverage area is beyond some range; otherwise, it stays in "ordinary node" status even if it can hear from more than one clusterhead.

PC can form and maintain its cluster structure without explicitly exchanging the clustering control packets. Thus, it can completely eliminate the control overhead caused by active clustering. Since a potential clusterhead can send out clusterhead claims without bearing neighbors' information, PC does not require a "frozen period" for initial neighboring learning and cluster construction, as is true in many active clustering schemes such as LCC and 3hBAC. In PC, ordinary nodes are not permitted to relay flooding packets, and thus the replicated flooding traffic can be remarkably reduced. By adopting the clustergateway control heuristic, PC keeps $|D_{H-G}|$ of a local area at a constant level. This mechanism can further reduce the generated duplicated flooding traffic and keep good connectivity between clusters. However, how to decide the optimal $|D_{H-G}|$ according to operation situations, such as traffic pattern and channel quality, is not addressed in PC.

***Summary of Low-Maintenance Clustering*** — Table 5 summarizes the features and objectives of the four low-maintenance clustering schemes. The four protocols attempt to build 1-hop clusters with the involvement of clusterheads in the cluster formation phase. In the cluster maintenance phase, Lin's algorithm removes the clusterhead role to distribute traffic load more evenly among all mobile nodes, whereas the other three schemes still keep the clusterhead-based cluster structure.

Table 6 shows the cost comparison of these four schemes. In Table 6, $m$ indicates the average cluster degree of a mobile node, i.e. the average number of clusters that a mobile node is involved in. The value of $m$ is related to the cluster overlapping. If a cluster structure is highly overlapping, $m$ is larger; otherwise, $m$ is smaller.

LCC, 3hBAC, and Lin's algorithm are considered to be active clustering schemes, since they require explicit clustering-related control message exchange among mobile nodes. Also, they need frozen period of motion of mobile nodes for the initial cluster construction because their clusterheads are required to bear a specific attribute, i.e. lowest ID or highest connectivity, in their local area. These three schemes need a non-constant number of rounds to complete the cluster formation since not all the clusterhead claims can be declared at the same time, because some mobile nodes can serve as clusterheads *only* after they can confirm some neighbors' status.

For example, in LCC and Lin's algorithm, a mobile node cannot declare itself as a clusterhead until it finds that all direct neighbors with a lower ID are involved in some clusters as clustermembers. So the time complexity for those algorithms is difficult to determine. In the worst case, the number of rounds for completing the cluster formation procedure is equal to the number of clusters, which indicates that only one clusterhead is decided in each round. However, normally their cluster construction can be performed in parallel in the whole network and the number of rounds needed for cluster formation should be less. The three schemes re-cluster only when the current cluster structure violates some basic maintenance rules, for example, when two clusterheads move into the reach range of each other. Hence, the cluster structure can be stabilized and the clustering-related control overhead can be reduced by limiting the clusterhead change situations. However, for Lin's algorithm, it likely forms small and new clusters when the topology changes and the cluster size decreases with time. The cluster structure of Lin's algorithm may finally disappear. For LCC and 3hBAC, they can maintain their cluster structure well throughout the operation, and effectively prolong the average lifetime of a cluster [18, 21] even under a highly mobile environment. 3hBAC can outperform LCC in terms of cluster stability and clustering overhead since it guarantees no ripple effect of re-clustering by the introduction of "clusterguest" status. 3hBAC can provide a non-overlapping cluster structure in the cluster formation and cluster maintenance phases, but sometimes the cluster overlap is somewhat desired in order to diversify inter-cluster routes [1, 18]. Hence, 3hBAC may be more feasible for a somewhat *dense scenario*, where mobile nodes are highly connected. Lin's algorithm guarantees that the cluster formation procedure can be completed with each mobile node sending out only one message [9] so that the communication complexity of Lin's cluster formation mechanism can be kept at a low level. 3hBAC requires that each mobile node send out at most three messages to help the initial cluster construction. The first message is its node degree claim; the second message is its clusterhead serving capability claim if changed to "declined;" the last message is the cluster-related status claim. Hence, the communication complexity of 3hBAC is $O(3|V|)$. LCC forms an overlapping cluster structure. Since a mobile node broadcasts its cluster-related status whenever it is involved in a cluster, a clustergateway node may broadcast more than one message in the cluster formation phase. Thus, the communication complexity of LCC can be represented by $O(m|V|)$.

PC is different from the other three schemes because cluster formation and cluster maintenance are performed when the mobile nodes have data to send. Thus, PC completely eliminates the control overhead for clustering; in other words, it keeps the cluster structure for upper-layer applications without any additional cost. Thus, PC can be deployed in networks with limited *channel capacity* where reducing the control overhead is mandatory. PC should be suitable for a network with high mobility, where mobile nodes' continuous movement greatly affects the cluster topology. This is because the cluster maintenance of PC is traffic-dependent and immune from increased control overhead caused by frequent change of cluster structure. However, for a network with bursty traffic, the cluster structure is difficult to maintain and cannot be promised to be ready for serving upper-layer routing or data forwarding.

## Mobility-Aware Clustering

Mobility is a prominent characteristic of MANETs, and is the main factor affecting topology change and route invalidation [22]. Some believe that it is important to take the mobility metric into account in cluster construction in order to form a stable cluster structure and decrease its influence on cluster topology. Mobility-aware clustering indicates that the cluster architecture is determined by the mobility behavior of mobile nodes. The idea is that by grouping mobile terminals with similar speed into the same cluster, the intra-cluster links can become more tightly connected. Then the re-affiliation and re-clustering rate can be naturally decreased.

***MOBIC* — MOBIC** [22] suggests that cluster formation, especially clusterhead election, should take mobility into consideration. It points out that clusterhead election is a local activity so that a clusterhead should be determined only by its neighbors and itself. MOBIC proposes an aggregate local mobility metric for the cluster formation process such that mobile nodes with low speed relative to their neighbors have the chance to become clusterheads.

In MOBIC, by calculating the variance of a mobile node's speed relative to each of its neighbors, the aggregate local speed of a mobile node can be estimated [22]. The main rationale behind calculating the variance of the relative mobility values of a mobile node with respect to each neighbor is that a low variance value indicates that this mobile node is relatively less mobile to its neighbors [22]. Consequently, mobile nodes with low variance values in their neighborhoods take the clusterhead responsibility.

For cluster maintenance, MOBIC follows that of LCC except that a timer, named *CCI*, is used to avoid unnecessary clusterhead relinquishing when two clusterheads incidentally pass by each other in a short period. That means if two clusterheads are in the reach range of each other longer than the *CCI* time period, one gives up its clusterhead role. Otherwise, they both keep the clusterhead status. This mechanism reduces the clusterhead change rate by avoiding re-clustering for incidental contacts of two passing clusterheads. However, the mobility behavior of mobile nodes is not always considered in cluster maintenance, so a clusterhead is not guaranteed to bear a low mobility characteristic relative to its members during maintenance phase. As time advances, the mobility criterion is somewhat ignored.

It is easy to see that MOBIC is feasible and effective for MANETs with group mobility behavior, in which a group of mobile nodes moves with similar speed and direction, as in highway traffic. Thus, a selected clusterhead can normally promise the low mobility with respect to its member nodes. However, if mobile nodes move randomly and change their speeds from time to time, the performance of MOBIC may be greatly degraded.

***DDCA (Distributed Dynamic Clustering Algorithm)* — DDCA** [23] attempts to partition a number of mobile nodes into multi-hop clusters based on $(\alpha, t)$ criteria. The $(\alpha, t)$ criteria indicate that every mobile node in a cluster has a path to every other node that will be available over some time period $t$ with a probability $\geq \alpha$ regardless of the hop distance between them. The purpose is to support robust and efficient routing, and adaptively adjust its dominant routing scheme depending on the network mobility manner. How to relate the probability of path availability with time and how to adaptively choose the values for $\alpha$ and $t$ can be referred to in [8].

As a distributed dynamic clustering scheme, DDCA requires no periodic re-clustering. When a mobile node is powered-on, it becomes unclustered and seeks a cluster to join by sending out *JoinRequest* message. A mobile node can join a cluster if it has a mutual path to satisfy $(\alpha, t)$ criteria between itself and the clusterhead of that cluster. If a mobile node receives multiple replies from different clusters indicat-

| | With clusterhead? | 1-hop or multi-hop | Objectives |
|---|---|---|---|
| MOBIC | Yes | 1-hop | Minimizing the influence of mobile nodes' movement on cluster topology updates in terms of re-affiliation and re-clustering based on mobile nodes' explicit relative speed; tightening the connection between mobile nodes residing in the same clusters. |
| DDCA | Yes | Multi-hop | Minimizing the influence of mobile nodes' movement on cluster topology updates in terms of re-affiliation and re-clustering to meet (a, t) criteria; adjusting the dominant routing mechanisms based on a network's mobility behavior. |

■ Table 7. *Summary of mobility-aware clustering schemes.*

| | Explicit control message for clustering | Ripple effect of re-clustering | Stationary assumption for cluster formation | Constant computation rounds | Communication complexity |
|---|---|---|---|---|---|
| MOBIC | Yes | Yes | No | Unnecessary to consider | |
| DDCA | Yes | No | No | Unnecessary to consider | N/A |

■ Table 8. *Cost comparison of mobility-aware clustering schemes.*

ing the availability of these paths, it chooses the cluster with the highest path availability probability to join. If a mobile node does not receive any *JoinResponse* message after a certain period of time, it creates a new cluster just to cover itself. As a clusterhead of such a single-node cluster, the mobile node periodically monitors possible *JoinRequest* messages from unclustered nodes or other single-node clusters or sends out its *JoinRequest* message until it successfully expands its own cluster or joins some other cluster as a clustermember.

DDCA can adaptively adjust its cluster size, considering the same level of stability, α. In a network with low mobility, it forms large-sized clusters, but in a highly mobile network, it diminishes the cluster size. DDCA suggests using table-driven schemes for intra-cluster routing, and on-demand schemes for inter-cluster routing. Thus, a network can adaptively adjust its dominant routing mechanism according to its mobility features. As a multi-hop clustering scheme, DDCA can better tolerate mobile nodes' movement with less re-affiliation, because a clusterhead and its clustermembers require no direct connection. As long as there is a path to meet $(\alpha, t)$ criteria between the clusterhead and its clustermembers, they can always be kept in the same cluster. In DDCA each mobile node runs the clustering scheme independently and a clusterhead does not need to have specific attributes in its neighborhood. Hence, mobile nodes do not need to be stationary during cluster formation in order to get complete and accurate information of a local area.

***Summary of Mobility-aware Clustering*** — Table 7 provides a summary of the two mobility-aware clustering schemes, respectively. Both MOBIC and DDCA are clusterhead-based clustering schemes based on some mobility metrics. MOBIC keeps 1-hop clusters, whereas DDCA maintains multi-hop clusters. Both schemes try to group mobile nodes with low relative mobility with respect to each other into the same cluster. Thus, the impact of mobile nodes' movement on cluster topology may be minimized. MOBIC uses explicit relative speed to form clusters, but DDCA adopts path availability to determine mobility manners between mobile node pairs.

Table 8 shows the cost comparison between the two schemes. In Table 8 Δ indicates the maximum number of direct neighbors that a mobile node has. As a multi-hop clustering, DDCA normally forms clusters with larger size compared with 1-hop clusters. Thus, a clusterhead in DDCA is required to keep more clustermembers' information, and the status change of a mobile node may cause table updates of more mobile nodes, resulting in increased control overhead. Re-clustering of DDCA does not invoke ripple effect since the cluster size in DDCA is adaptive and there is no hop limit between two neighboring clusterheads. This is beneficial for keeping the stability of the cluster topology and reducing the control overhead for cluster maintenance. However, MOBIC likely causes ripple effect of re-clustering because of its 1-hop cluster structure, and thus its cluster maintenance may be more cost-expensive. Both schemes do not require the stationary assumption for the initial cluster formation because they utilize the mobility behavior of mobile nodes to decide the relative speed or path availability. However, in practice a mobile node in MOBIC needs to collect "speed" information from neighbors to decide whether it is with the lowest relative speed in its local area. Thus, the continuous movement of mobile nodes may cause inaccurate information collection. In the cluster formation phase of MOBIC, first each mobile node sends two consecutive messages to each of its direct neighbors to help that neighbor decide the relative speed of them. Thus, each node sends out up to 2Δ messages to determine the relative speed between itself and each neighbor. Then each mobile node calculates its own aggregate local mobility and broadcasts this information to its neighbors. Also, since MOBIC is with an overlapping cluster structure, a mobile node may broadcast more than one cluster-related message (cluster-related status) during the cluster formation procedure. Thus, each mobile node sends out $m$ such messages on average. Hence, a mobile node is required to send out a total of $(2\Delta + 1 + m)$ messages for the initial cluster construction. MOBIC forms clusters with clusterheads bearing low relative speed initially and neglects mobility manners in most situations in later cluster maintenance. Therefore, it is more suitable for scenarios of mobile nodes with *group mobility* feature for a long period of time. Because DDCA does not require a frozen period of motion for initial cluster formation, it should work well in a dynamic environment where mobile nodes are continuously moving from the very beginning.

## ENERGY-EFFICIENT CLUSTERING

Mobile nodes in a MANET normally depend on battery-power supply during operation, hence the energy limitation poses a severe challenge for network performance [33, 34]. A MANET should strive to reduce its energy consumption greedily in order to prolong the network lifespan. Also, a clusterhead bears extra work compared with ordinary members, and it more likely "dies" early because of excessive energy consumption. The lack of mobile nodes due to energy depletion may cause network partition and communication interruption [35]. Hence, it is also important to balance the energy consumption among mobile nodes to avoid node failure, especially when some mobile nodes bear special tasks or the network density is comparatively sparse.

***IDLBC (ID Load Balancing Clustering)*** — In **IDLBC** [24] each mobile node has a variable, virtual ID (VID), and the value of VID is set as its ID number at first. Initially, mobile nodes with the highest IDs in their local area win the clusterhead role. IDLBC limits the maximum time units that a node can serve as a clusterhead continuously, so when a clusterhead exhausts its duration budget (*Max_Count*), it resets its VID to 0 and becomes a non-clusterhead node. When two clusterheads move into the reach range, the one with higher VID wins the clusterhead role. Each non-clusterhead node keeps a circular queue for its VID and shifts the VID value by one every time unit in one direction. Thus, when a clusterhead resigns, a non-clusterhead with the largest VID value in the neighborhood can resume the clusterhead function.

IDLBC tries to avoid possible node failure due to energy depletion caused by excessively shouldering the clusterhead role. When a mobile node resigns its clusterhead status because of the expiration of its duration budget, another mobile node with the highest VID in the local area is chosen to resume the clusterhead function. The newly chosen mobile node is the one whose previous total clusterhead serving time is the shortest in its neighborhood, and this should guarantee good energy level for being a new clusterhead. However, this kind of new clusterhead selection may introduce ripple effect of re-clustering over the whole network without considering the network topology. In addition, the clusterhead re-election may require time synchronization of the VID value shift among different mobile nodes. Otherwise, the VID information may not be accurate enough to select the most suitable node to serve as new clusterhead. In addition, the clusterhead serving time alone may not be able to promise a good indication of energy consumption of a mobile node.

***Wu's Algorithm*** — Wu [25] proposed an energy-efficient clustering scheme based on the DS marking algorithm [20]. Mobile nodes inside a DS consume remarkably more battery energy than those outside a DS because mobile nodes inside the DS bear extra tasks, including routing information update and data packet relay. Hence, it is necessary to minimize the energy consumption of a DS.

One method is to decrease the size of a DS without impairing its function, and thus unnecessary mobile nodes can be excluded from the DS and save their energy consumed for serving as clusterheads. In [16] Wu proposed some extension rules to eliminate unnecessary dominating nodes. In [25], energy level (el) instead of ID or node degree is used in the extension rules to determine whether a node $u$ should remain in the DS. The mobile node $u$ can be deleted from the DS when its close (open) neighbor set is covered by one (two connected) dominating neighbor(s), and at the same time it has less residual energy than the dominating neighbor(s).

The proposed scheme is more energy-aware compared with other DS-based clustering algorithms because it tries to delete mobile nodes with less residual energy from the CDS when possible. However, it still cannot balance the great difference of energy consumption between dominating nodes and non-dominating nodes because the main objective in [25] is to minimize the DS updates rather than to balance the energy consumption among all mobile nodes. Hence, mobile nodes in the DS still likely deplete their energy at a much faster rate.

***Ryu's Algorithm*** — Ryu [26] proposed a clustering scheme for energy-conservation in a MANET. In [26] there are two types of nodes, master (clusterhead) and slave (member). A slave node can connect to only one master node, and a direct link between slaves is not allowed. Master nodes are somehow selected in advance, and each can only serve a limited number of slave nodes. The specific objectives of Ryu's scheme are to minimize the transmission energy consumption summed by all master-slave pairs and to serve as many slaves as possible in order to operate the network with longer lifetime and better performance.

Two schemes, single-phase clustering and double-phase clustering, are proposed in [26]. In single-phase clustering, initially every master node will page slave nodes with the allowed maximum energy. For each slave that receives one or multiple paging signals, it always sends a ACK (Acknowledgment) mesage back to the master from which it receives the strongest paging signal. Since a master node can serve only a limited number of slaves, it first allocates channels for slaves that only receive a single paging signal from itself. If any free channels remain, other slave nodes, which receive more than one paging signal, are allocated channels in the order of the power level of the paging signal received from the master node. For those slave nodes, which do not receive a channel from a master in the channel allocation phase, are dropped in the further communication phase. This mechanism can reduce the call drop rate by giving priority to those slave nodes that only receive single paging signals in channel allocation. Slave nodes, which receive multiple paging signals, always try to communicate with the nearest master. Then, by adopting power control to meet the required minimum receiving power, each connected master-slave pair communicates with the minimum transmission power in order to save energy. To further lower the call drop rate, double-phase clustering adds a channel re-allocation phase on the basis of single-phase clustering. Each master, with free channels after its first-round allocation, re-pages for slaves, which do not receive a channel in the first round, in its range. The channel allocation procedure also follows the received signal strength.

The proposed schemes can nearly achieve optimum performance [26]. However, master node election is not adaptive, and the method of selecting the master node is not specified. Peer-to-peer communication between slaves is forbidden. In addition, the method of maintaining the cluster structure when master or slave nodes move is not addressed. Because of these restrictions, Ryu's scheme may not be feasible for a typical MANET. In addition, the paging process before each round of communication consumes a large amount of energy for master and slave nodes.

***Summary of Energy-efficient Clustering*** — Table 9 provides some features and objectives of the three energy-efficient clustering schemes. The three schemes are all clusterhead-based clustering schemes. The difference is that Wu's and Ryu's algorithms form 1-hop clusters, whereas IDLBC does not specify its cluster size explicitly but uses

| | With clusterhead? | 1-hop or multi-hop | Specific objectives |
|---|---|---|---|
| IDLBC | Yes | N/A simulation with 2-hop | Avoiding possible node failure because of excessive serving as clusterheads by limiting the time that a mobile node can serve as a clusterhead continuously. |
| Wu [29] | Yes | 1-hop | Guaranteeing the CDS connected and at the same time decreasing the total energy consumption of a network by deleting unnecessary mobile nodes with poor energy level from the CDS. |
| Ryu [30] | Yes | 1-hop | Serving slave nodes as many as possible and minimizing the total energy consumption between all communicating master-slave pairs. |

■ Table 9. *Summary of energy-efficient clustering schemes.*

| | Explicit control message for clustering | Ripple effect of re-clustering | Stationary assumption for cluster formation | Constant computation round | Communication complexity |
|---|---|---|---|---|---|
| IDLBC | Yes | Yes | Yes | No | $O([1 + (1 + \Delta)m]\|V\|)$ |
| Wu [29] | Yes | Yes | Yes | Two rounds | $O(2\|V\|)$ |
| Ryu [30] | Yes | N/A | Yes | Two/four rounds | $\geq O(\|V\| + \|C\| \times h)$ |

■ Table 10. *Cost comparison of energy-efficient clustering schemes.*

2-hop clusters for simulation [24]. All these schemes attempt to consume the energy of a MANET more wisely, but they have a very different focus regarding energy conservation. IDLBC thinks balancing energy consumption among all mobile nodes is compelling because mobile nodes are equally important, and also believes that the clusterheads consume more energy. Thus, it proposes a scheme to balance the clusterhead serving time among all mobile nodes. This scheme is useful if some mobile nodes are with unique duties or the network density is somewhat sparse because the death of some node may cause sever problems. Wu's algorithm tries to reduce the size of a DS to avoid unnecessary energy consumption of mobile nodes serving as dominating nodes without affecting its performance. However, the energy saving may not be so remarkable for this scheme since the main objective is not to reduce the energy consumption but to reduce DS update and recalculation. Since Wu's scheme is based on the CDS algorithm, it is not suitable for a network with high mobility. As a result, Wu's algorithm is more effective for a network with a large population, high density, and low mobility, such as a static wireless sensor network. Ryu's algorithm adopts power control in order to use minimal transmission power between a communication pair. However, in practice, only a few wireless cards can be configured to use multiple energy levels, and the number of available energy levels is also quite limited [36]. Thus, this may make it difficult to deploy Ryu's algorithm accurately. Also, Ryu's algorithm sets restrictions on clusterhead election and communication mode. Thus, it may be more feasible for a Bluetooth scatternet.

Table 10 shows the cost comparison for the three schemes. In Table 10, $|C|$ is the number of clusterheads, and $h$ refers to the average number of channels that a clusterhead holds in Ryu's scheme.

All three schemes are active clustering schemes because they demand explicit clustering-related control messages, and they depend on the stationary assumption for mobile nodes in the cluster formation phase. This is because in IDLBC and Wu's algorithm, a mobile node can become a clusterhead only after it collects some attribute information from all its neighbors and confirms that it is the node with some specific attribute in its local area. In Ryu's algorithm, a master node also needs the paging round to get the information about slaves within its coverage. Referring to the simulation parameters, IDLBC can be considered a 2-hop clustering scheme, and is more complicated than Wu's energy-efficient algorithm because clusterheads need to grasp more member nodes' information and cluster-related information is required to propagate up to two hops. Also, IDLBC requires a non-constant number of rounds to finish its cluster formation phase. Wu's algorithm is efficient in time since it is based on Wu's CDS scheme and can be completed in two rounds. The time complexity of Ryu's algorithm can also be bounded since it requires only two rounds for single-phase clustering (one for paging and the other for channel allocation) and four rounds for double-phase clustering (plus one round for re-paging and one round for channel allocation). In other words, Ryu's algorithm can always be completed with a constant number of rounds. The communication complexity of Wu's energy-efficient clustering is the same as that of Wu's CDS algorithm. For the single-phase clustering algorithm in Ryu's paper, all $|C|$ master nodes may page first, resulting in $|C|$ messages in the network. If the total number of mobile nodes is $|V|$, the corresponding number of slave nodes should be $(|V| - |C|)$. In the optimum case, all the slave nodes can receive at least one page signal and send an ACK message back, resulting in $(|V| - |C|)$ messages in the network. If each master node can have as many as $h$ channels, then the maximum number of slave nodes that all master nodes can serve together should be $|C| \times h$. Thus, the maximum channel allocation message is $|C| \times h$. Hence, for the single-phase clustering of Ryu's algorithm, the communication overhead is about $O(|V| + |C| \times h)$. If double-phase clustering is considered, the communication overhead should be even higher, because further communication will occur between master nodes, which have free channels after the first-round channel allocation, and slave nodes, which do not receive channels in the previous round.

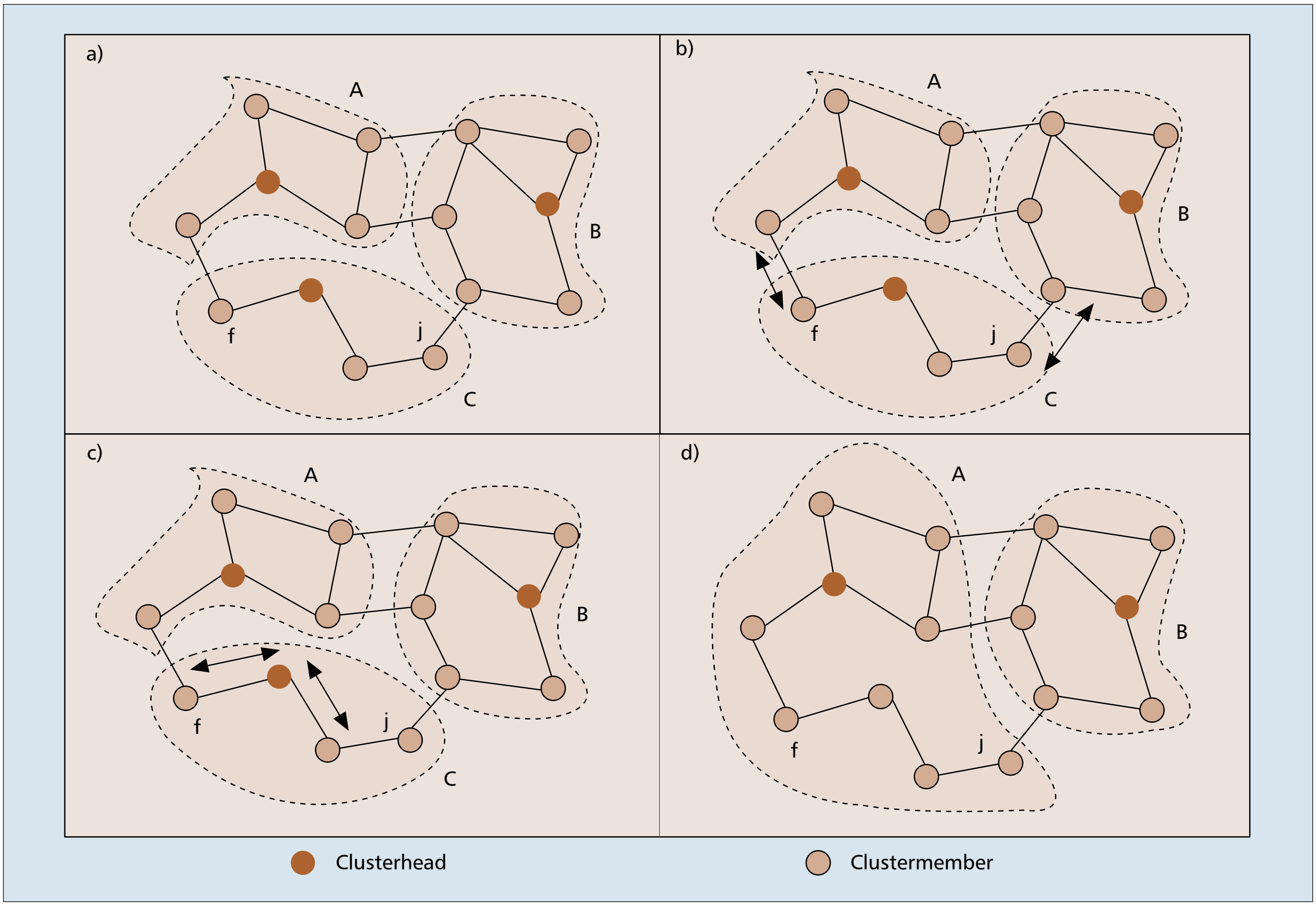



**Figure 6.** *Illustration of cluster merging.*

### Load-Balancing Clustering

Load-balancing clustering algorithms believe that there is an optimum number of mobile nodes that a cluster can handle, especially in a clusterhead-based MANET. A too-large cluster may put too heavy of a load on the clusterheads, causing clusterheads to become the bottleneck of a MANET and reduce system throughput. A too-small cluster, however, may produce a large number of clusters and thus increase the length of hierarchical routes, resulting in longer end-to-end delay. Load-balancing clustering schemes set upper and lower limits on the number of mobile nodes that a cluster can deal with. When a cluster size exceeds its predefined limit, re-clustering procedures are invoked to adjust the number of mobile nodes in that cluster.

***AMC (Adaptive Multi-hop Clustering)*** — **AMC** [27] maintains a multihop cluster structure based on load-balancing clustering. AMC does not describe how the clusters are initially constructed. However, for cluster maintenance each mobile node periodically broadcasts its information, including its ID, CID, and status (clusterhead/member/gateway) to others within the same cluster. By such message exchange, each mobile node obtains the topology information of its cluster. Each gateway also periodically exchanges information with neighboring gateways in different clusters and reports to its clusterhead. Thus, a clusterhead can recognize the number of mobile nodes of each neighboring cluster. AMC sets upper and lower bounds ($U$ and $L$) on the number of clustermembers that a clusterhead can handle.

In AMC, when the number of members, $|C_i|$, in a cluster $C_i$ is less than $L$, the merge mechanism is invoked. Cluster $C_i$ tries to find a neighboring cluster $C_j$ to satisfy $|C_i| + |C_j| \leq U$ and maximize the sum value. If $|C_i| + |C_j| > U$ for all neighboring clusters, it tries to find a $C_j$ to minimize the sum value. When two clusters merge, the clusterhead with more member nodes wins to continue the clusterhead role. Figure 6a shows the initial structure of cluster A, B, and C. Figure 6b shows that the gateways $f$ and $j$ of cluster $C$ exchange the information about the number of members with neighboring clusters. Figure 6c indicates that the information is forwarded to the clusterhead. Figure 6d shows that cluster $C$ invokes the merge process and merges with cluster $A$. The clusterhead of cluster $A$ becomes the clusterhead of the newly merged cluster because it has more member nodes than cluster $C$.

A cluster $C_i$ with $|C_i| > U$ performs the division mechanism. By choosing a suitable node $v_k \in C_i$ as a new clusterhead for the detached cluster, $C_i$ can be separated into two clusters with almost the same size. Nodes $v_i$, the original clusterhead of $C_i$, and $v_k$ send out CID update messages to member nodes. If mobile nodes receive the update message from $v_i$ first, then they keep their CID. Otherwise, mobile nodes change their CID to $v_k$. However, AMC does not address how to select a proper node $v_k$ to serve as the clusterhead for the newly detached cluster.

Although AMC mentions that the upper and lower bounds should be decided by network size, mobility, and so on, the values for $U$ and $L$ in [27] are given in advance in simulations without any justification.

***DLBC (Degree-Load-Balancing Clustering)*** — **DLBC** [24] periodically runs the clustering scheme in order to keep the number of mobile nodes in each cluster around a system parameter, $ED$, which indicates the optimum number of

| | With clusterhead? | 1-hop or multi-hop | Specific objectives |
|---|---|---|---|
| AMC | Yes | Multi-hop | Balancing the traffic load in each cluster by limiting the number of mobile nodes that a cluster can handle within a predefined range. |
| DLBC | Yes | N/A simulation with 2-hop | Balancing the traffic load in each cluster by limiting the number of mobile nodes that a cluster can handle around a predefined value. |

■ Table 11. *Summary of load-balancing clustering schemes.*

| | Explicit control message for clustering | Ripple effect of re-clustering | Stationary assumption for cluster formation | Constant computation round | Communication complexity |
|---|---|---|---|---|---|
| AMC | Yes | No | N/A | N/A | N/A |
| DLBC | Yes | Yes | Yes | No | $O([1 + (1 + \Delta)m]\|V\|)$ |

■ Table 12. *Cost comparison of load-balancing clustering schemes.*

mobile nodes that a clusterhead can handle. A clusterhead degrades to an ordinary member node if the difference between *ED* and the number of mobile nodes that it currently serves exceeds some value, *Max_Delta*. This mechanism tries to make all clusterheads almost serve the same and optimal number of member nodes.

Compared with HCC [20], DLBC can reduce the rate of clusterhead change because a clusterhead does not need to relinquish its clusterhead status whenever it has a member node with a higher node degree. However, since the clusterhead change is still based on node degree, DLBC likely will cause frequent re-clustering because the movement of mobile nodes and consequent link setup/break results in dynamic variation of mobile node degree. In addition, how to select a clusterhead is not addressed in DLBC if in a local area no mobile nodes can satisfy the degree difference requirement between *ED* and its current node degree (2-hop). Similar to AMC, how to decide the important system parameters, *ED* and *Max_Delta*, is not discussed in DLBC.

***Summary of Load-balancing Clustering*** — Table 11 presents a summary of the two load-balancing clustering schemes addressed before. Both AMC and DLBC are clusterhead-based clustering schemes. AMC keeps a multi-hop cluster structure for both the formation and maintenance phases. DLBC does not address its cluster size explicitly, but uses a 2-hop cluster for simulations. Both load-balancing clustering schemes manage to balance the workload of each cluster by limiting the number of mobile nodes around an optimal value in a cluster. DLBC always tries to replace the current clusterhead with a new clusterhead when the current clusterhead cannot satisfy the node degree requirement, but AMC merges neighboring clusters together or splits a cluster into two when the size of a cluster is too small or too large.

Table 12 provides the cost comparison between these two schemes. AMC does not address how to form the multi-hop cluster structure initially, so we cannot evaluate its cost in terms of stationary assumption, computation round, and communication complexity. Since AMC forms a cluster structure with multi-hop clusters and has no limits on the hop distance between neighboring clusterheads, the merge or division procedure only affects the manipulated clusters but no others. Thus, AMC brings no ripple effect of re-clustering and maintains good cluster stability. Considered as a 2-hop clustering scheme, DLBC requires that neighboring clusterheads should be at least three hops away. Thus, a new clusterhead election likely influences neighboring clusters' structures and invokes global clusterhead re-elections. Also, DLBC requires the stationary period in the cluster formation phase for each mobile node to receive the node degree information of neighbors within two hops.

### Combined-metrics-based Clustering

Combined-metrics-based clustering takes a number of metrics into account for cluster configuration, including node degree, residual energy capacity, moving speed, and so on. This category aims at electing the most suitable clusterhead in a local area, and does not give preference to mobile nodes with certain attributes, such as lowest ID or highest node degree. One advantage of this clustering scheme is that it can flexibly adjust the weighting factors for each metric to adjust to different scenarios. For example, in a system where battery energy is more important, the weighting factor associated with energy capacity can be set higher [28]. However, not all of these parameters are always available and accurate, and the information inaccuracy may affect clustering performance.

***On-Demand WCA (Weighted Clustering Algorithm)*** — **On-Demand WCA** [28] considers four parameters for each mobile node in the clusterhead election procedure. They are degree-difference $D_v$, sum of the distance with all neighbors, $P_v$, average moving speed, $M_v$, and clusterhead serving time, $T_v$. Here, $D_v$ is given by $D_v = |d_v - M|$ where $d_v$ is the number of neighbors of a mobile node $v$, and $M$ is the number of nodes a clusterhead can handle ideally. But how to select $M$ is not addressed explicitly. Besides, it is not easy to estimate the distance between two mobile nodes in a practical environment. Also, the clusterhead serving time $T_v$ alone cannot guarantee a good assessment of energy consumption because data communication consumes a large amount of energy and varies greatly from node to node.

The combined weight factor, $I_v$, is calculated as $I_v = c_1D_v + c_2P_v + c_3M_v + c_4T_v$, where $c_1$, $c_2$, $c_3$, and $c_4$ are the weighting factors and

$$\sum_{i=1}^{4} c_i = 1.$$

All the parameter values are normalized according to some pre-defined values. On-Demand WCA chooses mobile nodes with minimum $I_v$ in the local area to be clusterheads. All mobile nodes covered by elected clusterheads cannot participate in further clusterhead selection. This procedure is repeated until each mobile node is assigned to a cluster. Since it is necessary for each mobile node to obtain so much infor-

| | With clusterhead? | 1-hop or multi-hop | Specific objectives |
|---|---|---|---|
| On-demand WCA | Yes | 1-hop | Electing the most suitable clusterheads in local areas by considering several metrics and keeping a stable cluster structure by reducing re-clustering situations. |

■ Table 13. *Summary of on-demand WCA scheme.*

| | Explicit control message for clustering | Ripple effect of re-clustering | Stationary assumption for cluster formation | Constant computation round | Communication complexity |
|---|---|---|---|---|---|
| On-demand WCA | Yes | Yes | Yes | No | $O((\Delta + 1 + m)\|V\|)$ |

■ Table 14. *Cost of on-demand WCA scheme.*

mation and to calculate its combined weight for the initial clusterhead election, the cluster formation procedure requires a longer frozen period of motion for all mobile nodes.

The clusterhead election algorithm is invoked at the very beginning of cluster formation or when the current clusterheads are not able to cover all mobile nodes. WCA does not re-cluster when a member node changes its attaching cluster. Even though this mechanism can maximize the stability of cluster topology, this also indicates clusterheads keep their status without considering the attribute of minimum $I_v$ for some situations in later cluster maintenance. When a mobile node goes into a region not covered by any clusterhead, clusterhead election will be performed again. The new clusterheads are elected to ensure that the all new clusterheads are selected based on the minimum $I_v$. However, this kind of re-clustering completely destroys the current cluster architecture and may require a significant number of messages to be exchanged to build a new one. So the effectiveness of WCA's maintenance mechanism is in doubt.

The summary and cost analysis of On-Demand WCA are shown in Tables 13 and 14, respectively. Since on-demand WCA relies on the stationary assumption for the initial cluster construction, the "constant computation round" metrics should be taken into consideration. Unfortunately, on-demand WCA requires a non-constant number of rounds to complete its cluster formation, which means the time complexity is unbounded for the cluster formation phase. By sending a message to each direct neighbor, each mobile node can help each of its neighbors to decide the distance between them (by measuring the ratio of receiving power and transmission power). Supposing $\Delta$ is the maximum number of direct neighbors a mobile node can have, each mobile node then needs to send up to $\Delta$ messages for the distance confirmation for all its neighbors. Then each mobile node broadcasts at least one message to claim its own information, including the degree difference, average speed, distance sum to all direct neighbors, and clusterhead serving time, to its neighbors. As an overlapping cluster structure, a mobile node may broadcast more than one message about its cluster-related status. On average, each mobile node sends out $m$ such messages. Hence, each mobile node needs to send out at least $(\Delta + 1 + m)$ messages for cluster formation, and then the communication complexity of On-Demand WCA can be specified as $O((\Delta + 1 + m)|V|)$.

## Conclusions

As MANETs have attracted more attention in recent years, much research has been addressing all kinds of issues related to them. Dynamic routing plays an important role in the performance of a MANET, and research associated with routing is always a focus. One related topic is the scalability of a routing protocol. Since a large-scale MANET cannot guarantee performance with a flat structure, many cluster hierarchy algorithms have been proposed to solve the scalability issue.

In this article, we first provided fundamental concepts about clustering, including the definition of cluster and clustering, the necessity of clustering for a large dynamic MANET, and the side effects and cost of clustering. Then we classified 14 proposed clustering schemes into six categories based on their main objectives. We discussed each clustering scheme in terms of objective, mechanism, performance, and application scenario, and discussed the similarities and differences between schemes of the same clustering category.

With this survey we see that a cluster-based MANET has many important issues to examine, such as the cluster structure stability, the control overhead of cluster construction and maintenance, the energy consumption of mobile nodes with different cluster-related status, the traffic load distribution in clusters, and the fairness of serving as clusterheads for a mobile node. Also, different types of clustering schemes may have a different focus and objectives. However, clustering cost always needs to be considered when discussing a clustering scheme, because clustering cost is important to evaluate the performance and scalability improvement of a clustering scheme no matter which specific objectives it bears.

Through the detailed study of these 14 clustering schemes, we can conclude that PC can perform best in terms of communication complexity because it completely eliminates explicit clustering-related control message but piggybacks clustering information in upper-layer data packets for forming and maintaining a cluster structure in a MANET. However, all the other 13 schemes exchange explicit control messages between node pairs for clustering.

The ripple effect of re-clustering is an important metric to determine the communication cost of a clustering scheme in the cluster maintenance phase since the ripple effect might completely destroy the current cluster topology and require all mobile nodes to update their cluster status. Normally multi-hop clustering schemes, such as DDCA and AMC, do not cause ripple effect in re-clustering because their cluster structures are more flexible and normally they do not have a hop limit between two neighboring clusterheads. But most 1-hop clustering algorithms, such as Wu's CDS scheme, LCC, PC, MOBIC, and On-Demand WCA, may cause the ripple effect because they always require a clusterhead to be directly connected with its member nodes and any two clusterheads are separated by at least one clustermember node. The exception is 3hBAC, which is a 1-hop clustering scheme but has no ripple effect because the introduction of a new node status, clusterguest, relaxes the requirement that an ordinary mobile node must always be connected with at least one clusterhead.

Some schemes, including Wu's CDS algorithm, Chen's WCDS algorithm, LCC, 3hBAC, Lin's algorithm, IDLBC, Wu's energy-efficient clustering, and DLBC, require the assumption of frozen period of motion for mobile nodes when cluster formation is in progress because normally in these schemes a mobile node can declare a clusterhead claim only after it exchanges some attribute information with its neighbors and it can ensure that it bears some specific attribute in its neighborhood, such as lowest ID or highest connectivity. However, this assumption may not be applicable for a real scenario, where mobile nodes may move randomly all the time from the very beginning of an operation. PC does not depend on this stationary assumption because it is based on a "first claim wins" solution for the initial cluster formation which means an unclustered mobile node sends out a clusterhead claim at a random time and if its claim is the first successful one in its local neighborhood, it becomes a clusterhead. DDCA also does not need a frozen period of motion because a mobile node always actively seeks a cluster to join from the time it is powered-on, and creates a single-node cluster to cover itself when there is no proper cluster to join.

With this survey, readers can have a more comprehensive understanding of MANET clustering, especially those schemes discussed in this article. Although each scheme is well suited for certain scenarios, it is not guaranteed that any one of them is the best for all situations. We hope that this survey article can facilitate researchers to offer more efficient and effective clustering schemes for MANETs.

## ACKNOWLEDGMENT

The authors would like to thank our anonymous reviewers for their helpful comments.

## REFERENCES

[1] C. E. Perkins, *Ad Hoc Networking*, Addison-Wesley, 2001.

[2] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Mobile Networking for 'Smart Dust'," *Proc. 5th IEEE/ACM MOBICOM*, Aug. 1999, pp. 271–78.

[3] P. Guptar and P. R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Info. Theory*, vol.-IT 46.2, Mar. 2000, pp. 388–404.

[4] X. Y. Hong, K. X. Xu, and M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks," *IEEE Network*, July-Aug. 2002, pp. 11–21.

[5] K. X. Xu, X. Y. Hong, and M. Gerla, "An Ad Hoc Network with Mobile Backbones," *Proc. IEEE ICC'2002*, vol. 5, Apr.–May 2002, pp. 3138–43.

[6] E. M. Belding-Royer, "Hierarchical Routing in Ad Hoc Mobile Networks," *Wireless Commun. and Mobile Comp.*, vol. 2, no. 5, 2002, pp. 515–32.

[7] R. Rajaraman, "Topology Control and Routing in Ad Hoc Networks: A Survey," *ACM SIGACT News*, vol. 33, no. 2, June 2002, pp. 66–73.

[8] A. B. MaDonald and T. F. Znati, "A Mobility-based Frame Work for Adaptive Clustering in Wireless Ad Hoc Networks," *IEEE JSAC*, vol. 17, Aug. 1999., pp. 1466–87.

[9] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE JSAC*, vol. 15, Sept. 1997, pp. 1265–75.

[10] T.-C. Hou and T. J. Tsai, "An Access-Based Clustering Protocol for Multihop Wireless Ad Hoc Networks," *IEEE JSAC*, vol. 19, no. 7, July., 2001, pp. 1201–10.

[11] C. R. Lin and M. Gerla, " A Distributed Control Scheme in Multi-hop Packet Radio Networks for Voice/Data Traffic Support," *Proc. IEEE ICC'95*, June 1995, pp. 1238–42.

[12] U. C. Kozat *et al.*, "Virtual Dynamic Backbone for Mobile Ad Hoc Networks," *Proc. IEEE ICC'01*, vol. 1, June 2001, pp. 250–55.

[13] M. R. Pearlman and Z. J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol," *IEEE JSAC*, vol. 17, Aug. 1999, pp. 1395–414.

[14] A. Iwata *et al.*, "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE JSAC*, vol. 17, Aug. 1999, pp. 1369–79.

[15] W. Chen, N. Jain, and S. Singh, "ANMP: Ad Hoc Network Management Protocol," vol. 17, Aug. 1999, pp. 1506–31.

[16] J. Wu and H. L. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," *Proc. 3rd Int'l. Wksp. Discrete Algorithms and Methods for Mobile Comp. and Commun.*, 1999, pp. 7–14.

[17] Y.-Z. P. Chen and A. L. Liestman, "Approximating Minimum Size Weakly-Connected Dominating Sets for Clustering Mobile Ad Hoc Networks," in *Proc. 3rd ACM Int'l. Symp. Mobile Ad Hoc Net. & Comp.*, June 2002, pp. 165–72.

[18] C.-C. Chiang *et al.*, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel," in *Proc. IEEE SICON'97*, 1997.

[19] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," in *Proc. IEEE*, vol. 75, 1987, pp. 56–73.

[20] M. Gerla and J. T. Tsai, "Multiuser, Mobile, Multimedia Radio Network," *Wireless Networks*, vol. 1, Oct. 1995, pp. 255–65.

[21] J. Y. Yu and P. H. J. Chong, "3hBAC (3-hop between Adjacent Clusterheads): a Novel Non-overlapping Clustering Algorithm for Mobile Ad Hoc Networks," in *Proc. IEEE Pacrim'03*, vol. 1, Aug. 2003, pp. 318–21.

[22] P. Basu, N. Khan, and T. D. C. Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks," in *Proc. IEEE ICDCSW'01*, Apr. 2001, pp. 413–18.

[23] A. B. McDonald and T. F. Znati, "Design and Performance of a Distributed Dynamic Clustering Algorithm for Ad-Hoc Networks," in *Proc. 34th Annual Simulation Symp.*, Apr. 2001, pp. 27–35.

[24] A. D. Amis and R. Prakash, "Load-Balancing Clusters in Wireless Ad Hoc Networks," in *Proc. 3rd IEEE ASSET'00*, Mar. 2000, pp. 25–32.

[25] J. Wu *et al.*, "On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks," *J. Commun. and Networks*, vol. 4, no. 1, Mar. 2002, pp. 59–70.

[26] J.-H. Ryu, S. Song, and D.-H. Cho, "New Clustering Schemes for Energy Conservation in Two-Tiered Mobile Ad-Hoc Networks," in *Proc. IEEE ICC'01*, vo1. 3, June 2001, pp. 862–66.

[27] T. Ohta, S. Inoue, and Y. Kakuda, "An Adaptive Multihop Clustering Scheme for Highly Mobile Ad Hoc Networks," in *Proc. 6th ISADS'03*, Apr. 2003.

[28] M. Chatterjee, S. K. Das, and D. Turgut, "An On-Demand Weighted Clustering Algorithm (WCA) for Ad hoc Networks," in *Proc. IEEE Globecom'00*, 2000, pp. 1697–701.

[29] T. J. Kwon *et al.*, "Efficient Flooding with Passive Clustering — an Overhead-Free Selective Forward Mechanism for Ad Hoc/Sensor Networks," in *Proc. IEEE*, vol. 91, no. 8, Aug. 2003, pp. 1210–20.

[30] S. Basagni, "Distributed and Mobility-Adaptive Clustering for Multimedia Support in Multi-Hop Wireless Networks," in *Proc. IEEE VTS 50th*, vol. 2, Sept. 1999, pp. 889–93.

[31] B. Das and V. Bharghavan, "Routing in Ad Hoc Networks Using Minimum Connected Dominating Sets," in *Proc. IEEE ICC'97*, June 1997, pp. 376–80.

[32] B. Das, R. Sivakumar, and V. Bharghavan, "Routing in Ad Hoc Networks Using a Spine," in *Proc. IEEE Int'l. Comp. and Commun. Net. '97*, Sept. 1997., pp. 1–20.

[33] R. Zheng and R. Kravets, "On-demand Power Management for Ad Hoc Networks," in *Proc. IEEE Infocom'03*, vol. 1, Mar.–Apr. 2003, pp. 481–91.

[34] J. Gomez *et al.*, "PARO: Supporting Dynamic Power Controlled Routing in Wireless Ad Hoc Networks," *Wireless Networks*, vol. 9, 2003, pp. 443–60.

[35] C.-K. Toh, "Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks," *IEEE Commun. Mag.*, vol. 39, no. 6, June 2001, pp. 138–47.

[36] D. Kim *et al.*, "Routing Mechanisms for Mobile Ad Hoc Networks Based on the Energy Drain Rate," *IEEE Trans. Mobile Comp.*, vol. 2, no. 2, Apr.-June 2003, pp. 161–73.

## BIOGRAPHIES

JANE YANG YU received her B. Eng degree in electrical engineering from Tianjing University, China, in 2000. Between July 2000 and July 2002 she worked as a software and system engineer in the Data Communication Product Department, Nanjing Research and Development Center, ZTE Corporation, China. In July 2002 she joined the Network Technology Research Centre, Nanyang Technological University, as a Ph.D. student. Her research interests include wireless/mobile communications and ad hoc networks.

PETER H. J. CHONG (ehjchong@ntu.edu.sg) received the B.Eng. (with distinction) in electrical engineering from Dalhousie University, Halifax, NS, Canada, in 1993, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of British Columbia, Vancouver, BC, Canada, in 1996 and 2000, respectively. From July 2000 to January 2001 he worked with the Advanced Networks Division at Agilent Technologies Canada Inc., Vancouver, BC, Canada. From February 2001 to May 2002 he was a research engineer in the Radio Communications Laboratory at Nokia Research Center, Helsinki, Finland. Since May 2002 he has been with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, as an assistant professor. He is a member of the Network Technology Research Centre, Nanyang Technological University. His research interests include channel assignment schemes, radio resource management, multiple access, and mobile ad hoc networks.