

## Performance Analysis of Clustering Protocols in Mobile Ad hoc Networks

Sharmila John Francis and Elijah Blessing Rajasingh  
School of Computer Science and Technology, Karunya University,  
Coimbatore, Tamil Nadu, India

---

**Abstract:** Network clustering is an important technique widely used in efficient MANETs network management, hierarchical routing protocol design, network modeling, Quality of Service, etc. Recently many researchers are focusing on clustering which is one of the fundamental problems in mobile ad hoc networks. This article presents the descriptions of recently proposed clustering algorithms and is categorized into different approaches that support similar features. Based on the comparison of different performance metrics of different clustering algorithms, the most suitable one is recommended that adapt it to various application scenarios. This study provides adequate information for the researchers that facilitate to analyze many avenues and to offer more effective and efficient clustering protocols for MANETs.

**Key words:** Network clustering, cluster head, cluster-gateway

---

### INTRODUCTION

The tremendous and continuous proliferation of the market of wireless-enabled portable devices facilitates the major impetus towards the development of large Mobile Ad hoc NETWORKS (MANET)<sup>[1]</sup> that bears great applications in variety of fields

In a clustered network organization, the clusters are formed by grouping the mobile nodes with one special role node termed as cluster head, vested with special supervising functions and acts as a local coordinator by performing information aggregation of topology and managing intra-cluster communication between the nodes in their proximity or ordinary nodes, without any inter-cluster links. Gateway nodes are the non-cluster head nodes situated at the fringe of a cluster facilitating inter-cluster cluster communication with its neighborhood clusters<sup>[2]</sup>.

Efficient clustering imposes minimum clustering overheads, limiting the number of clusters and their overlapping. Clustering is always of significance importance for network management, hierarchical address assignment, medium access control, Quality of Service (QoS), radio resource management, topology update etc. It has the following benefits.

- It improves the system capacity<sup>[3]</sup> by implementing spatial reuse of resources
- It reduces the amount of routing information propagated in the network and routing delay<sup>[4-7]</sup>

- A clustered architecture limits the amount of information to represent the state of network and reduces the amount of topology update messages<sup>[8]</sup>

A considerable amount of research has been done on clustering algorithm for mobile ad hoc networks. This article comprises of survey of various clustering protocols and their performances are analyzed using various performance metrics and its comparative study is performed

### CATEGORIZATION OF CLUSTERING PROTOCOLS

In general, clustering protocols can be broadly categorized into cluster head-based clustering protocols and non-cluster head based clustering protocols<sup>[8]</sup>. Cluster head-based clustering protocols outperform non-cluster head based clustering protocols in terms of traffic overhead<sup>[8]</sup>. Based on the hop distance of the cluster members from its associated cluster head, the cluster head based clustering can be further divided into 1-hop clustering<sup>[3, 7, and 8]</sup> and multihop clustering<sup>[9, 10]</sup>. In 1-hop clustering, the maximum distance between two cluster members is 2-hops, thereby maintaining one hop distance between the cluster members and its associated cluster head. In multihop clustering, cluster head can reach its farther member nodes by taking multiple hops through intermediate cluster members and relax its restriction of having direct connection with its associated members.

---

**Corresponding Author:** Sharmila John Francis, School of Computer Science and Technology, Karunya University, Coimbatore, Tamil Nadu, India

Table 1: Summary of different clustering approaches

Clustering approach	Prominent features
Low-maintenance clustering	Provides stable cluster architecture with minimized cluster maintenance cost
Mobility-based clustering	Mobility feature of the mobile nodes is taken into consideration for cluster formation that achieves maximum stability by grouping the mobile nodes of similar moving patterns
Weight-based clustering	The weight of the mobile nodes that provides the suitability of a node being a cluster head is taken into consideration for the choice of the cluster head, for forming stable clusters
Flooding-based clustering	Clustering is based on the distinguished feature of delivering data or control messages to all nodes within the network
Channel-based clustering	Utilizes the channel effectively by scheduling the transmissions of the mobile nodes

Normally cluster formation and cluster maintenance are the two phases that exists in clustering that facilitates smooth operation of the network throughout its lifetime. Cluster formation refers to the building of cluster structure for the MANET at the initial stage of the network organization. Due to the unpredictable mobility of the nodes, it leads to the arbitrary changes of network topology over time. Since the mobile nodes may not be aware of changes in their neighborhood, cluster maintenance is initiated to have frequent updating of clusters and cluster heads to maintain the accurate network topology. In this article, clustering protocols are categorized into different approaches based on its distinguished features and are shown in Table 1. Low-Maintenance Clustering approach [3, 11-13] provides a stable cluster structure incurring less maintenance cost.

Mobility-Based Clustering approach [10, 11, and 14] considers mobility feature of the mobile nodes for cluster formation. It achieves maximum cluster stability by grouping mobile nodes of similar patterns into a single cluster. Weight-Based Clustering approach [15] takes weight of the mobile nodes into consideration for the choice of the cluster head. Flooding-Based Clustering approach [6] forms the cluster by disseminating information over the whole network and the Channel Based Clustering [16] facilitates efficient utilization of channels by scheduling transmissions of the mobile nodes.

Finding good performance metrics to evaluate various clustering protocols is not a trivial task. The various performance metrics whose descriptions are self-explanatory are presented in Table 2. Based on these performance metrics, each clustering algorithm is rated to these parameters and its performance is compared with other clustering algorithms. Finally, the best clustering algorithm is recommended that is more suitable for many application scenarios by performing the analysis of various different clustering protocols.

Table 2: Description of performance metrics

Performance metrics	Description
Average cluster size	The average number of nodes catered by the cluster head
Average cluster head lifetime	The average time period for a mobile node to act as a cluster head
Average cluster membership lifetime	The average time period for a mobile node associated with its corresponding cluster head
Average number of cluster head changes	The average number of cluster head change
Average number of clusters	The average number of clusters formed in the network
Number of dominant set updates	A set of cluster heads is called a dominant set, if all the nodes in the network are either in the set or neighbors of nodes in the set. It gets updated when a node can no longer be a neighbor of any of the existing cluster heads
Number of re-affiliations	The re-affiliation count is incremented when a node gets dissociated from its cluster head and gets associated as a member of another cluster within the current dominant set.
Time complexity per topology change (T)	The time taken to accomplish valid cluster reorganization after a change in the network topology
Message complexity per topology change (M)	The number of messages exchanged between the nodes to accomplish valid cluster reorganization after a change in the network topology
Cluster stability	Maintaining a stable structure during network topology changes
Total overhead	Overhead incurred during the network operation time

**Low-maintenance clustering:** Low-Maintenance clustering aims at providing stable cluster structure architecture without the excessive consumption of network resources for cluster maintenance, thereby incurring less maintenance cost. The re-affiliation and re-clustering are the two major events that influence the cost of cluster maintenance. The re-affiliation refers to the disassociation of cluster member from its cluster head and associating itself to another cluster without affecting the corresponding cluster head(s). Re-clustering is an event that completely rebuilds the cluster topology over the whole network. These two events change the network topology more frequently with the drastic increase of clustering maintenance overheads. The following three protocols can be categorized under Low-Maintenance clustering approach.

**Lowest-ID:** The Lowest-ID [14] clustering is one of the most popular clustering schemes used in the old [18] as well as recent [3, 7] ad hoc networks literature. The Lowest-ID algorithm proceeds as follows.

**Step 1:** Assign unique ID to each node.

**Step 2:** Broadcast the ID of each node to the list of its neighbors (including itself).

**Step 3:** Elect the node as a cluster head that has the lowest ID relative to its neighbors.

**Step 4:** Assign a node as cluster-gateway if it can hear from two or more cluster heads and as cluster-member if it is a neighbor to a cluster head.

**Maximum Connectivity Clustering (MCC):** The MCC<sup>[14]</sup> uses the degree of connectivity instead of the node ID in the cluster head election.

**Step 1:** Find the number of neighbors for each node.

**Step 2:** Broadcast the number of neighbors of each node to all its neighbors.

**Step 3:** Elect a node as cluster head that has maximum number of neighbors. If there is any tie, elect the node with the lowest node ID as the cluster head.

**Step 4:** Assign a node as cluster-gateway if it can hear from two or more cluster heads and as cluster-member if it is a neighbor to a cluster head.

**Efficient Clustering Scheme (ECS):** ECS<sup>[12]</sup> is more suitable for large and dense MANETs. It eliminates the frozen period requirement by using random claim mechanism<sup>[17]</sup> in cluster formation and reduces cluster overlapping and prolongs the cluster-lifetime without producing excessive clustering overheads, thereby resulting in less maintenance cost. Mechanisms involved in cluster formation and cluster maintenance are as follows.

**Cluster formation:** It builds a cluster structure for the mobile nodes in the network at the very beginning. The Fig. 1 shows initial cluster construction with different status of nodes.

**Step 1:** Set all the mobile nodes as unspecified.

**Step 2:** Any unspecified status node can initiate cluster formation, by broadcasting a cluster-head-claim message after the random period of time to claim itself as a cluster head.

**Step 3:** During the random period of time  $T_{\text{backoff}}$ , if a mobile node hears a cluster head claim from any other neighbor, it joins that neighbor's cluster as a cluster-member. Otherwise, it claims itself as a cluster head in the neighborhood.

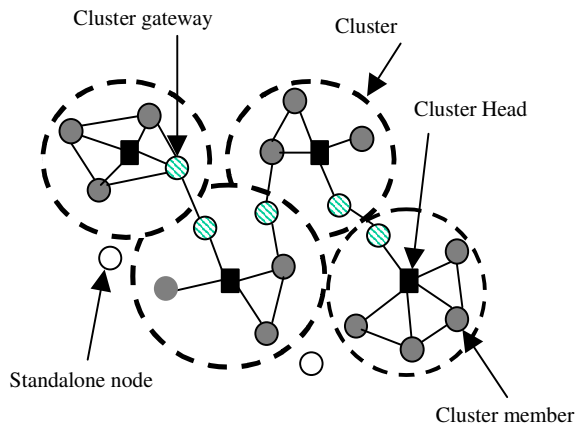


Fig. 1: Cluster formation phase

**Step 4:** Change the status of member node as cluster-gateway, if it belongs to two or more clusters.

**Step 5:** Change the status of cluster head as standalone, if it does not receive any messages from its neighbors, to indicate their joining as member nodes within a predefined period of time after it sends out its cluster head claim.

**Step 6:** Each cluster head builds its information table, which is the storage space for recording its entire neighbor's clustering related information.

**Cluster maintenance:** It updates the cluster structure according to the change of network topology during the operation. Each node periodically broadcasts hello messages to indicate its presence in its area. The two major events that will be involved in cluster maintenance are Membership Update and Cluster Head Change. They are explained in the following subsections.

**Membership update:** This event updates the status of the mobile nodes whenever there is change in the network topology.

- Change of status from cluster-member to cluster-gateway is similar to that of 1-hop clustering
- Set the status of a member node (cluster-member or cluster-gateway) as cluster-guest, if it is disconnected from all its cluster head(s) and can still connect to neighboring cluster(s) by choosing one of its members that can bridge with its cluster head(s). The chosen member nodes are termed as access points

- Set the status of cluster-guest node as cluster-member, if it hears cluster head claim from any mobile node or hello message from any existing cluster head
- Update the status of standalone node as cluster-guest, if it hears hello message from any existing cluster-member/cluster-gateway

**Cluster head change:** A mobile node broadcasts specific clustering message cluster-head-change, if a cluster head resigns from its cluster head role or changes from a non-cluster head to a cluster head. The change of cluster head is event-driven and invoked in following three scenarios.

- When two nodes incidentally pass by each other, the cluster head that has fewer member nodes resigns from its cluster head role and changes its state to cluster-member
- When a mobile node finds all its neighbors as either cluster-guests or with standalone status, it changes its state to cluster head. This mechanism moderates the cluster density by maintaining the distance between the newly claimed cluster head and its neighboring cluster heads as 3-hops
- When the cluster head finds all its member nodes are with cluster-gateway and are covered by at least two other cluster heads, the cluster deletion mechanism is invoked and changes its state to cluster-guest by choosing randomly one of the neighbors as its access point

The standalone status nodes that are present in cluster formation phase can be associated to the neighboring clusters by selecting their few nodes that lie within its transmission range as access points in Fig. 2. The status of the standalone node is changed to cluster guest to avoid formation of small and unnecessary clusters.

**Mobility aware clustering:** Mobility is a distinguished characteristic of MANETs that causes clusters to get disrupted, triggers frequent re-clustering and route invalidation<sup>[14]</sup>. Mobility-Aware Clustering takes mobility metric into consideration to form stable cluster architecture by grouping mobile nodes with similar speed into the same cluster.

**MOBIC:** A Lowest Relative Mobility Clustering Algorithm-MOBIC<sup>[14]</sup> is similar to the Lowest-ID algorithm, but it uses the mobility metric as a basis of cluster formation and cluster head selection. The algorithm is as follows:

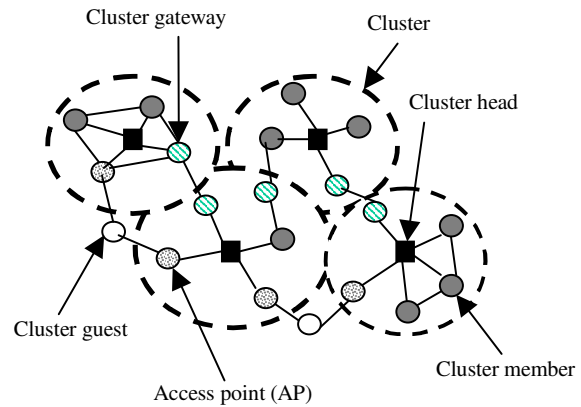


Fig. 2: Cluster maintenance phase

**Step 1:** Set all the mobile nodes as cluster-undecided state.

**Step 2:** Calculate the pair wise relative mobility metric<sup>[14]</sup> for each node by measuring the received power levels of two successive hello message transmissions from its every neighbor.

**Step 3:** Calculate the aggregate relative mobility metric  $M^{[14]}$  for each node by taking the variance of the entire set of the pair wise relative mobility values of its neighbors.

**Step 4:** Broadcast the mobility metric  $M$  of each node to its 1-hop neighbors.

**Step 5:** Compare the received aggregate mobility value of each node with that of its neighbors and elect the node as a cluster head that has the lowest value of  $M$  amongst all its neighbors.

**Step 6:** Change the status of node as cluster-gateway, if it is a neighbor to two cluster heads.

**Step 7:** Resort to comparison of node IDs and follow Lowest-ID algorithm, if two cluster-undecided state nodes or cluster heads that are in contention to retain the cluster head state have the same value of  $M$ .

**Step 8:** Utilize the Cluster-Contention-Interval (CCI) timer to defer re-clustering, if two cluster heads move into each other transmission range. The re-clustering is triggered and the node with the lower mobility metric declares its state as cluster head, if two cluster heads remains in the same reach range even after the CCI timer has expired.

**Mobility-based d-hop Clustering Algorithm (MobDHop):** MobDHop<sup>[13]</sup> dynamically forms variable diameter clusters that are adaptive to node mobility patterns. It allows the cluster members at most d-hops away from their cluster heads to control the cluster size and manage cluster head density. The cluster head election is dependent on three mobility metrics, namely (a) Variation of estimated distance between nodes over time (VD) which estimates the relative mobility of two nodes. (b) Local Variability (LV) which calculates the mean of VD of all the neighbors for a node (c) Group variability (GV) which calculates the mean of LV of its 1-hop neighbors. A node that has the lowest variability value (LV) assumes the role of cluster head. The following procedures explain the cluster setup and cluster maintenance phase.

**Role assignment:** This procedure assigns the role/status of each node that are explained in the following steps

**Step 1:** Declare the mobile node as cluster head, if its neighbor list is empty.

**Step 2:** Compare the local variability of a mobile node with all its non-clustered neighbors, if its neighbor list is not empty.

**Step 3:** Elect the mobile node as cluster head that has lowest local variability value relative to its neighbors

**Merge procedure:** This procedure is initiated when a non-clustered node requests to join its neighboring cluster. A node may become non-clustered when it is newly activated or it loses its cluster head due to node mobility. The following steps explain the procedure

**Step 1:** Choose a neighbor who has the lowest VD value to a non-clustered node so that it is most stably connected to its chosen neighbor.

**Step 2:** Evaluate whether the following criteria are met for a merging node.

- Hop count from a merging node to its new cluster head should be less than the parameter d, this condition is irrelevant if no restriction has been set for d (i.e., d is infinity)
- The VD between the non-clustered node and its chosen neighbor should be lower than the group variability (GV) value

**Step 3:** Merge the non-clustered node to the neighboring cluster, if the above criteria are met.

Otherwise, non-clustered node chooses another neighbor node, which has the second lowest VD value and repeats the evaluation.

**Step 4:** Declare the non-clustered node as cluster head, if there is no node that can fulfill the merging criteria.

**Update status:** This procedure updates the status of all the mobile nodes against the changes in surrounding topology.

**Step 1:** Resolve the contention of two cluster heads that move into the transmission range of each other by selecting the cluster head with the lower local variability value.

**Step 2:** Determine the status of a node using the role assignment procedure, if all its cluster members leave its cluster.

**Step 3:** Update the role of an ordinary node to that of a gateway node, if it can hear from one or more cluster heads.

**Step 4:** Execute merge procedure, if a mobile node finds that it has lost its cluster head.

**Step 5:** Update the role of the gateway node to that of an ordinary node, if it can only hear hello messages from only one cluster head.

**Weight-based clustering:** This weight-based clustering approach can dynamically adapt itself with the ever changing topology of ad hoc networks. In this approach, each node is assigned with an appropriate weight based on its suitability of being a cluster head. Depending upon the application scenarios, different clustering algorithms use different methods for assigning weights to the mobile nodes and the cluster head election is done on the basis of these node weights. Amongst the different clustering algorithms, the most suitable one that takes multiple metrics into account for cluster configuration and can effectively adapt itself to different kinds of networks is, On-Demand Weighted Clustering Algorithm

**On-Demand Weighted clustering algorithm (On-Demand WCA):** The On-Demand WCA<sup>[15]</sup> aims to maintain stability of the network and facilitates the optimal operation of medium access control protocol. The assignment of weight to a mobile node is the combined effect of several system parameters like ideal node degree, degree difference, transmission power,

cluster head serving time and mobility. The advantage of this clustering scheme is the flexibility of adjusting the weighting factors for each system parameter to make it suitable for different scenarios. The procedure for cluster head election is as follows:

**Step 1:** Calculate the degree difference  $D_u$ ,  $D_u = n_u - N$  where  $n_u$  is the number of neighbors of a mobile node  $u$  and  $N$  is the number of nodes that a cluster head can handle ideally. It is done to ensure efficient MAC functioning, as it is desirable for a cluster head to handle up to a predefined threshold.

**Step 2:** Compute the sum of the distances  $S_u$ , with all the neighbors for every mobile node.

**Step 3:** Measure the mobility  $M_u$  of every node by computing its average speed until the current time  $T$  as:

$$M_u = 1/T \sum_{i=1}^T \sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2}$$

Where,  $(X_t, Y_t)$  and  $(X_{t-1}, Y_{t-1})$  are the coordinates of the node  $u$  at time  $t$  and  $t-1$  respectively.

**Step 4:** Compute the cluster head serving time  $T_u$ , which is the total time that a node  $u$  acts as a cluster head and it is measured by taking into the consideration of the drainage of node's battery power.

**Step 5:** Calculate the combined weight  $C_u$ , for each node  $u$  as:

$$C_u = w_1 D_u + w_2 S_u + w_3 M_u + w_4 T_u$$

Where  $w_1, w_2, w_3$  and  $w_4$  are arbitrarily chosen weighting factors for the corresponding system parameters that satisfies the following condition:

$$\sum_{i=1}^4 w_i = 1$$

**Step 6:** Elect the smallest  $C_u$  as cluster head and restricts its neighbors to participate in further election procedure.

**Step 7:** Repeat the above steps for the remaining nodes that are neither cluster head nor a cluster member.

**Flooding-based clustering:** As the mobile ad hoc networks are characterized by scarce bandwidth, radio interferences and lack of fixed infrastructure, it

circumvents the need for more efficient flooding techniques that serves as a building block for complex protocols. Flooding is the information dissemination covering the entire nodes in the network without the requirement of computation and maintenance of routing tables and avoids network delay. Each node redistributes the information to all its neighbors until the entire network is inundated. Based on different criteria, the flooding can be limited to a set of nodes rather covering the whole network.

**Max min D-cluster algorithm** <sup>[6]</sup>: This allows the control and flexibility in the determination of the cluster head density by generalizing the distance of a mobile node from its cluster head to be  $d$  hops. One of the features of the algorithm is to reflect the existing cluster heads even when the network configuration changes, thereby reducing communication overheads that are incurred during the transition from old cluster head to new cluster head. This clustering scheme has the capability to evenly distribute the mobile nodes among the cluster heads and evenly distribute the responsibility of acting as cluster heads among all nodes, thereby sustaining fair and stable cluster architecture. The algorithm runs for  $2d$  rounds of flooding and each node maintains two arrays WINNER, which records the winning node id of a particular round to determine the cluster head and SENDER, which sends the winning node id for a particular round and determines the shortest path back to the cluster head after the selection of cluster head. The operation of the algorithm is explained in the following steps

**Step 1:** Set each node's WINNER to be equal to its own node id and initiate  $2d$  rounds of flooding.

**Step 2:** Segment the 1st  $d$  rounds as floodmax that broadcasts the WINNER value to all of its 1-hop neighbors and chooses the largest value among its own WINNER value as its new WINNER and this process continues for  $d$  rounds. The 2nd  $d$  rounds are referred as floodmin that is same as floodmax but it chooses smallest value as its new WINNER.

**Step 3:** Restore the results of the flooding as logged entries and are maintained in each node's database.

**Step 4:** Identify the nodes whose node id occurs at least once as a WINNER in both the 1st and 2nd  $d$  rounds of flooding and refer it as node pairs.

**Step 5:** Look at the logged entries of each node after the completion of 2nd  $d$  rounds of flooding and declare

the node as cluster head, if it has received its own original node id in the 2nd d rounds of flooding. Otherwise, proceed to the following step 6.

**Step 6:** Select the minimum node pair as cluster head amongst all the identified node pairs to accomplish load balancing, if node pairs exist for each node. Otherwise, proceed to step 7, if node pair does not exist for a node.

**Step 7:** Elect the node as cluster head that has maximum node id in the 1st d rounds of flooding.

**Step 8:** Broadcasts the elected cluster head of each node to all of its neighbors after the cluster head selection.

**Step 9:** Determine the node as gateway node, if all of its neighboring nodes have different cluster head selections.

**Channel-based clustering:** The Channel based clustering helps the mobile node to schedule transmissions and facilitates efficient usage of channel by reducing collisions and overhead. Due to lack of centralized control in ad hoc networks, a separate out-of-band signaling is more suitable for control information dissemination <sup>[15]</sup> by adopting bichannel structure. The two channels are control channels that are used to exchange the control information and another is the data channel for data transmission.

**Channel access based clustering:** It sustains a stable cluster structure by limiting time and overhead to cluster the given network. Based on the randomized control channel broadcast access method that maximizes the worst-case control channel efficiency, a distributed clustering algorithm for multihop ad hoc networks is employed. There is a main focus on control channel as all the clustering operations are done through the control channel

**Control channel access algorithm** <sup>[16]</sup>: The time axis of the control channel is divided into equal length slots and the channel is assumed to be error free with busy tone detection capability. Each slot of the control channel is divided into Sensing Period (SP), the Packet Period (PP) and the Acknowledgement Period (AP). During the communication of mobile nodes, two types of conflicts occur. The primary conflict occurs when a mobile node transmits in a given slot and it cannot receive a packet in the same slot and vice versa. The secondary conflict occurs when a mobile node cannot receive more than one packet in one slot. To avoid

these conflicts, the algorithm uses two busy tones, primary and secondary busy tones. The primary busy tone avoids the primary conflict and the secondary conflict by the secondary busy tone. The following steps explain the operation of the algorithm.

**Step 1:** Generate a back off value  $\delta$  following a predefined probability density function, if a mobile node wishes to transmit the packet in a certain slot.

**Step 2:** Listen to the channel at time  $B\delta$ , which is the normalized value with respect to the packet length.

- If no busy tone is detected, transmit the primary busy tone until the end of the SP and control packet in the PP
- If primary busy tone is detected, transmit the secondary busy tone until the end of the SP and cancel the packet transmission in the slot
- If both busy tones and only the secondary busy tone are detected, cancel the packet transmission in the slot

**Step 3:** Transmit the primary busy tone in the AP, if no packet is received successfully after listening to PP, when the mobile node does not transmit the control packet during the slot.

**Step 4:** If no busy tone is detected after listening to AP, conclude that a packet is broadcast successfully and delete the packet from the control packet buffer, when a mobile has transmitted the control packet during the slot and otherwise, keep the packet in the control packet buffer for retransmission in later slots.

**Distributed clustering algorithm:** This algorithm uses only one control message LCM, Leader Contending Message for clustering. Each mobile node maintains two local variables, myCID variable refers to the ID of the cluster head to which the mobile node is associated and myID variable represents the ID of the mobile node. This clustering scheme has two types of beacon signal, leader beacon signal transmitted by the cluster head and the normal beacon signal transmitted by the cluster members. The following steps describe the operation of the algorithm.

**Step 1:** Set the entire mobile node's myCID variable to NULL after the power is on.

**Step 2:** Broadcast the Leader Contending Message (LCM) message along with the ID of the mobile node to its neighbors, if the mobile node does not belong to any cluster.

**Step 3:** Elect the mobile node as cluster head that can successfully broadcast the LCMs to its neighbors.

**Step 4:** Listen to the beacon signal for a duration  $t_b$ , if a mobile node enters the network.

- Contend to be cluster head, if a mobile node does not receive any beacon signal from any of the cluster heads
- Associate the mobile node as a member of a cluster, if it receives leader beacon signal from the cluster head and then assign myCID of the associated mobile node as myCID of the received signal
- Cancel the LCM transmission attempts, if leader beacon signal is received during the attempt to be the cluster head and then assign myCID field of a mobile node as the myCID field of the received signal

**Step 5:** Delete the mobile node from the member list of a cluster head, if the cluster head does not receive the beacon signal from its cluster member for duration longer than  $t_b$ .

**Step 6:** Change the status of a cluster head to a normal mobile node (that does not belong to any cluster) and determine its new cluster head, if its member list is NULL.

**Step 7:** Change the status of a cluster member to a normal mobile node and determine its new cluster head, if it does not receive any beacon signal from its cluster head for duration longer than  $t_b$ .

**PERFORMANCE ANALYSIS OF CLUSTERING PROTOCOLS**

The performance of each clustering protocol is analyzed against various crucial performance metrics and it is compared with other clustering protocols. As the clustering protocols are grouped together based on its unique features, the performance of various groups of clustering protocols are compared in the following Table 3. Based on the performance comparison of various clustering protocols, the best clustering protocol is recommended that is more suitable for many application scenarios.

**Discussions on low-maintenance clustering:** Table 3 summarizes the performance of three protocols. The lowest-ID and MCC builds 1-hop clusters with the participation of cluster heads, but ECS outperforms them by extending to k-hop ( $k > 1$ ) by relaxing the requisite that cluster members must be directly connected with their associated 1-hop cluster head. Both Lowest-ID and MCC need frozen period of motion of mobile nodes for initial cluster formation to guarantee the accurate information exchange between the neighborhood nodes, but ECS effectively eliminates it by using random claim mechanism.

ECS outperforms both Lowest-ID and MCC by forming large size clusters since it is extended to k-hop clustering. On comparison of Lowest-ID and MCC with respect to the max\_group distance deviation,  $G_{max}$ , the former forms larger<sup>[13]</sup> size clusters than MCC.

The average cluster head lifetime is lower for MCC than Lowest-ID since the degree of connectivity changes very rapidly for the mobile nodes and its cluster head change also occurs very frequently, thereby reducing its lifetime. But, in ECS, since cluster head

Table 3: Performance analysis of clustering protocols using performance metrics

Clustering approach	Performance metrics	Average cluster size	Average cluster head lifetime	Average member-ship lifetime	Average No. of cluster head change	No. of re-affiliations	No. of dominant set	Average No. of clusters	Cluster stability	Time complexity	Message complexity	Total over head
Low-maintenance protocols	Lowest-ID	Low	Low	Low	High	2WCA, higher	No	High	Low	$O(1)$	$O(1)$	$O(N)$
	MCC	Very low	Lower	Lower	Higher	High	No	Very high	Very low	$O(1)$	$O(1)$	$O(N)$
	ECS	Very high	High	Higher	Very less	Very less	No	Very less	Higher	$\leq (T_{backoff} + 1)$	$O(1)$	$O(N)$
Mobility-based protocols	MOBIC	8-12nodes	Low	High	High	Higher	No	Relatively high	High	$\leq (T_{sample} + 1)$	$O((2\Delta + 1 + m(N)))$	$O(N)$
	MobDHop	12 nodes	Relatively high	Higher	Very less	High	No	Less	Higher	$\leq (T_{sample} + H_{max}) + H_{max}$	$\leq mH_{max}$	$O(N)$
Weight-Based protocols	WCA	Predefined threshold value	Low	Lower	Lower	½lowest-ID, less	Low	Less	Lower	Unbounded	$O((\Delta + m(N)))$	$O(N)$
Flooding-based protocols	Max-min D-cluster	High	Moderate, stable	High	Low	Very less	Lower	Almost stable	Low	$O(d)$	$O(d)$	$O(N)$
Channel-based protocols	Channel-based	Recommend 10	Moderate	Low	Moderate	Less	No	Less	Moderate	Less	$O(1)$	Very less



change is event driven and occurs only in few situations, its cluster head lifetime is very high than the other two protocols.

The average membership lifetime metrics is analyzed with respect to the node speed,  $N_s$ . With increase of  $N_s$ , MCC performs lower by gradual decrease<sup>[13]</sup> of average membership lifetime than Lowest-ID. ECS also decreases<sup>[12]</sup> with increase in  $N_s$  but its lifetime is higher than Lowest-ID and MCC since it is k-hop clustering. The nodes tend to stay inside its cluster for a longer time as it can be k-hops from its cluster head, thereby increasing its membership lifetime and reducing the number of re-affiliations to a great extent. The number of re-affiliations is high for Lowest-ID and MCC as they are 1-hop clusters. The number of re-affiliations of Lowest-ID is twice<sup>[18]</sup> that of WCA and it is higher than MCC, since in MCC the degree of connectivity of the mobile node changes very rapidly and thereby its cluster head change occurs more frequently but in the case of Lowest-ID cluster head change is not as frequent as in MCC. Thus the number of re-affiliations is higher than MCC.

In ECS, with the introduction of access point and cluster guest status, the average number of clusters formed is very less, since the nodes that move out of its cluster could still be connected to any other cluster(s) by using cluster guest status. Thus, the cluster head handles more number of nodes and reduces the number of clusters to a great extent than the other two protocols. On comparison of Lowest-ID and MCC with respect to the max\_group distance deviation,  $G_{max}$ , MCC forms higher<sup>[13]</sup> number of clusters than Lowest-ID.

The time complexities and message complexities of both Lowest-ID and MCC are of  $O(1)$  because each mobile node takes one time step to broadcast its node ID or degree of connectivity to reach its 1-hop neighbors. In ECS, each unspecified mobile node backs off a random period of time  $T_{backoff}$  before it broadcast the cluster head claim message. After the clustering decision is made, each node broadcast a new hello message with its latest decision. If a node wishes to declare itself as a cluster head or joins as a member to a neighboring cluster, it takes one time step to reach its cluster head or to its 1-hop neighbors since cluster guest is not formed during the cluster formation. Thus its time complexity is  $\leq(T_{backoff} + 1)$  and its message complexity is  $O(1)$ . Cluster stability for Lowest-ID and MCC increases only when the pause time of the nodes increases which is impossible in real life scenarios as its mobility is unpredictable. But, ECS greatly improves the stability, as the cluster head change is event driven and invoked in only few situations. It incurs relatively

less overhead than the other two algorithms since re-clustering and re-affiliation events occurs more frequently causing additional overhead in the case of Lowest-ID and MCC.

**Discussions on mobility-based clustering:** Table 3 summarizes the performance of two mobility based clustering protocols. Both MOBIC and MobDHop are cluster head based clustering protocols that takes into account the mobility metrics, to decide the relative speed or path availability thereby eliminating frozen period of motion of mobile nodes during the initial cluster formation. The MOBIC forms 1-hop clusters, whereas MobDHop can be extended to multihop clusters. MOBIC outperforms lowest-ID by forming larger size clusters for medium to high values of transmission range and also in reduction of as much as 33%<sup>[14]</sup> in number of cluster head changes. MobDHop outperforms Lowest-ID and MCC by forming less number of clusters with the less occurrence of number of re-affiliations.

The average cluster head lifetime of MobDHop is relatively high than MOBIC since the cluster head election is based on powerful three mobility metrics. As the mobility criterion is taken into consideration for cluster formation, the average membership lifetime is high for both the protocols. But MobDHop outperforms MOBIC since the cluster members are at most d-hops away from its cluster head, the members tend to stay in their cluster for a long time than MOBIC.

The average number of clusters formed in MobDHop is comparatively less than MOBIC as the cluster head covers large number of nodes that are at most d-hops away from it, thus forming less number of clusters. In MOBIC, the number of re-affiliations is comparatively higher than MobDHop since the nodes tends to move out the clusters very rapidly as it is 1-hop from its cluster head.

In MOBIC, each mobile node takes at least  $T_{sample}$  number of time steps to decide whether to declare itself as a cluster head or to join as a member to the neighboring cluster and broadcasts its latest cluster decision in the subsequent hello message. When a node opts to join as member or to declare itself as a cluster head, it takes one time step to reach its 1-hop cluster head or 1-hop neighbors. Thus, the time complexity of MOBIC is  $\leq(T_{sample}+1)$ . Each mobile node sends  $2\Delta$  messages to measure the relative speed with its neighbors. Then each mobile node calculates aggregate local mobility and broadcasts to its neighbors. Due to the overlapping cluster structure in MOBIC, each mobile node may have a chance of broadcasting more than one cluster-related message during the cluster

formation phase, thereby sending  $m$  such messages on average. Hence the total messages required sending by  $N$  nodes to re-achieve the valid cluster organization after the occurrence of change of topology is  $O((2\Delta+1+m)N)$ . In MobDHop,  $T_{\text{sample}}$  time steps are taken for local variability computation and decides its state either to be a cluster head or as a member to the neighboring cluster. Also, time needed to propagate this latest decision takes almost  $H_{\text{max}}$  time steps to reach its associated  $H_{\text{max}}$  hops members or to its associated  $H_{\text{max}}$  hops cluster head. When a new link is established between two cluster heads or when a link failure occurs between an ordinary node and its cluster head or its upstream ordinary node, the cluster re-organization process is triggered. In both the cases, only the downstream member nodes react to this topology change and this is a chain reaction that comes to an end when the effect reaches the border node of the clusters. Since each member is at most  $H_{\text{max}}$  hops away from its cluster head, the chain reaction will almost extend to  $H_{\text{max}}$  hops and involve all cluster members. When  $m$  is the average number of members in a cluster, the time complexity and message complexity is  $\leq (T_{\text{sample}}+H_{\text{max}}) + H_{\text{max}}$  and  $\leq mH_{\text{max}}$  respectively.

Due to the unpredictable mobility nature of the mobile nodes, it is prone to the arbitrary changes in network topology, thereby leading to cluster instability. However, the MOBIC and MobDHop clustering protocols take mobility feature into consideration that adaptively forms stable clusters incurring the total overhead of  $O(N)$ . MobDHop provides higher stability than MOBIC as it outperforms in terms of cluster head lifetime, membership lifetime and cluster head change metrics.

**Discussions of weight-based, flooding-based and channel based approaches:** Table 3 shows three different kinds of protocols. In WCA clustering protocol, a predefined threshold that each cluster head can ideally handle, is fixed for each cluster to ensure efficient medium access control functioning, thereby achieving high system throughput and balanced load among the cluster heads. The Max-Min D-cluster and Channel Access Based clustering protocols produces larger clusters as  $N$  increases, since no restriction is imposed on the cluster size.

When a new node is added to the existing cluster whose weight is lower than its associated cluster head weight, then the existing cluster head has to delegate its duties to the newly joining lower weight node. Thus, the role of the cluster head in WCA is retained as long as its weight is lower than the neighborhood member nodes and its lifetime is low than the other two

protocols. In Max-Min clustering protocol, the cluster head lifetime increases<sup>[6]</sup> as the network becomes dense and it is almost stable as the network becomes denser. Whereas in channel access based clustering the mobile nodes that do not belong to any cluster contend to be a leader and its lifetime is based on its successful contention. Thus, both Max-Min and channel access based protocols provides moderate cluster head lifetime.

The average membership lifetime of WCA is lower than other two protocols since it forms 1-hop clusters, cluster members tends to move out of its cluster very rapidly and gets attached to other neighborhood cluster. The Max-Min clustering protocol has its cluster members  $d$ -hops away from their cluster head, its members stay inside the cluster for a long time and also it remains almost constant<sup>[6]</sup> as the number of nodes increases. Thus, its membership lifetime is comparatively high than the other two algorithms.

It is desirable to elect a cluster head that does not move very quickly to avoid frequent change of cluster head. When a cluster head moves fast, it leads to re-affiliation as the nodes get detached from the cluster head. The frequent change of cluster head incurs additional control overhead leading to instability of the cluster structure. In WCA, average number of cluster head change increases when the mobility of the cluster head increases. In Max-Min D cluster, it decreases as the network becomes denser whereas in Channel Access Based clustering, the cluster head changes are not frequent and its change rate (change rate is defined as average number of cluster heads disappeared or generated per second) decreases with decrease in the average moving speed of the mobile nodes, producing stable clusters. Thus, the cluster head change is moderate than the other two protocols.

The number of re-affiliation in WCA increases<sup>[4]</sup> with transmission range and reaches a peak for certain values of transmission range and it decreases with further increase of  $T_r$  as the nodes tends to stay inside the  $T_r$  covered by the cluster head despite of their random motion resulting in half the number obtained to that of protocol of Lowest-ID.

The number of re-affiliations and the number of dominant set updates in Max-Min clustering protocol is very less when compared to other algorithms as its cluster members are at  $d$ -hops from their cluster head, nodes tends to stay inside the clusters for a long time, thereby reducing the number of re-affiliations and dominant set updates. In the case of Channel Access Based clustering, it decreases with decrease of average  $N_s$ , because the nodes tend to stay in the same cluster for a long time.

The average number of clusters in WCA decreases with increase in transmission range and it almost remains same for different values of max-displacement  $D_{max}$ , particularly for larger values of  $N$  as it simply results in a different configuration irrespective of its mobility. As more nodes are added, the number of clusters formed remains almost unchanged in Max-Min clustering protocol, but it decreases quickly with increase of  $N_{av}$  in the case of Channel Access Based clustering

The clustering protocols that rely on frozen period of motion consider constant computation rounds metric since more rounds are needed for its cluster formation. The WCA clustering algorithm also assume frozen period of motion but unfortunately it requires non-constant number of rounds to complete its cluster formation phase. Thus, the time complexity cannot be bounded and may vary noticeably for different network topologies. In addition, each mobile node has  $\Delta$  maximum number of direct neighbors, each mobile sends up to  $\Delta$  messages to calculate the distance between itself and its neighbors by utilizing the receiving and transmission power. Therefore, each node broadcasts at least one message to claim its own information like degree difference, average speed etc to its neighbors. Each node may broadcast cluster related message more than once because of overlapping cluster structure. Hence each mobile node sends  $m$  such messages on average and thus, the message complexity of On-Demand WCA is  $O((\Delta+1+m)N)$  forming stable cluster topology with  $N$  number of nodes in the network with the overhead of  $O(N)$ .

In Max-Min D-cluster, the convergecast of  $O(d)$  rounds of messages is initiated to inform the cluster head of its children which is not more than  $d$  hops from its cluster head and each node propagates node ids for  $2d$  rounds to elect cluster heads, thus the time complexity is  $O(2d+d) = O(d)$  rounds and message complexity is  $O(d)$ . It provides stability as it has the load balancing scheme and has the tendency of re-electing existing cluster heads by promoting minimal transferal of databases.

In Channel-Based Clustering, time taken to cluster the network is very small and it is not sensitive to the network size. Since only one control message is responsible for valid cluster re-organization, the message complexity is  $O(1)$ . It provides stable clusters incurring very small overhead even on high  $N_s$  during the cluster maintenance phase and the overhead decreases quickly with increase of  $N_{av}$  of mobile node during the cluster formation. Thus, the cluster stability is moderate than the other two protocols. The cluster stability of WCA is lower than Max-Min as Max-Min

outperforms the former by providing moderate and stable cluster head lifetime and with low cluster head change.

## CONCLUSION

Clustering can provide large scale MANETs with a hierarchical network structure to solve the critical scalability issue that is prone to the flat structure of MANETs. It is always of significant importance for routing operations, network management, mobility management, quality of service support etc.

In this article, the fundamental concepts of clustering are presented and its associated clustering algorithms are classified into different categories based on its distinguished features. The mechanisms and descriptions of each clustering protocol are presented and their performances are rated against the very critical performance metrics.

A major target of any clustering protocol is to maintain stable cluster structure adapting itself to the ever-changing network topology by promoting minimal cluster structure updating and clustering overheads. This paves the way to evaluate the cluster stability using various significant metrics like average cluster head change, cluster head lifetime, time complexity etc to accomplish the target successfully. The evaluation with respect to cluster head change has immense impact on stability of any clustering protocol. The frequent cluster head change leads to many drawbacks like route invalidation, frequent occurrence of re-affiliation, incurring additional overhead etc that paves way to rebuilding the whole network causing instability of the network topology. With the detail analysis of all the protocols, we can conclude that only ECS greatly improves the stability of the cluster structure, as the cluster head change is event driven and invoked in only few situations. Though the Channel Access Based clustering maintains a stable structure with decrease of cluster head change with decrease of node speed, it is not possible to decrease the  $N_s$  as the mobility of the nodes is unpredictable.

Another practical problem that is prone to many clustering protocol is the stationary assumption of mobile nodes for initial cluster formation. Due to the mobility nature of the nodes, it is impossible to assume them static during the cluster formation. This critical issue is faced by Lowest-ID, MCC, Max-Min and WCA. Though mobility based and Channel-Based clustering avoids this issue, they are more suitable for specific scenarios. But, ECS outperforms all other algorithms by effectively eliminating stationary assumption of nodes by using random claim mechanism that grants the privilege for any non-clustered node to

send its cluster head claim at random time after a back off period and the first successful claim in its local neighborhood becomes a cluster head.

A good clustering protocol should meet the optimum number of clusters as another unique criterion. A too large cluster may vest heavy load for the cluster heads that becomes the bottleneck of the whole system whereas too small clusters is not feasible as it increases the length of hierarchical routes resulting in longer end-to-end delay. Thus, it is preferable to have optimum number of clusters to facilitate effective and efficient operation of the network. The ECS effectively reduces the small and unnecessary clusters with the introduction of new clustering status called cluster guest.

The time complexity and message complexity, is small in ECS when compared to MCC, MOBIC, MobDHop, WCA and Max-Min. The Channel Access Algorithm also takes less time for large N, but its applicability is more specific to channel access based cluster topology. As the cluster head change is event driven in ECS, clustering overhead incurred is also comparatively less than the other clustering protocols.

It is impossible to guarantee a protocol to be best suited for all application scenarios, as each clustering protocol is more suitable for certain specific scenarios. But, still we recommend a clustering protocol that is more suitable for many application scenarios. With the extensive performance analysis of all the recently proposed clustering protocols, we can recommend ECS as the best suited one that effectively fulfills its major target of cluster stability by promoting relatively moderate overhead and with effective time and message complexity after a change in topology.

### NOTATIONS

$N_s$	= Node Speed
$T_r$	= Transmission Range
$N_d$	= Node density
$N_{av}$	= Average number of neighbors for a mobile node
$N$	= Total number of nodes in a network
$T_{backoff}$	= Time taken for a mobile node to back off a random period of time before it broadcasts its cluster related message
$T_{sample}$	= Number of time steps taken by a node to collect stability information from its neighbors.
$H_{max}$	= Maximum hop count from cluster head
$\Delta$	= Maximum number of neighbors
$D_{max}$	= The maximum displacement allowed for the nodes to move in all possible directions

uniformly between 0 to maximum value per unit time

$G_{max}$  = The maximum distance allowed for the group members deviation to deviate from the group leader when Reference Point Mobility Model is used

### REFERENCES

1. Chlamtac, I., M. Conti and J.J.N. Liu, 2003. Mobile ad hoc networking: Imperatives and challenges. Elsevier Ad Hoc Networks, Vol.1, No.1, pp: 13-64.
2. Nocetti, F.G., J. Solano-Gonzales and I. Stojmenovic, 2003. Connectivity- based k-hop clustering in Wireless Networks. Kluwer Telecommun. Syst., Vol.22, No.1-4, pp: 205-220.
3. Lin, C.R. and M. Gerla, 1997. Adaptive clustering for mobile wireless networks. IEEE J.Select. Areas Comm., Vol.15, No.7, pp: 1265-1275. doi: 10.1109/49.622910.
4. Jiang, M., J. Li and Y.C. Tay, 1999. IETF Internet-Draft. Cluster-based Routing Protocol. National University of Singapore, July. <https://datatracker.ietf.org/drafts/draft-ietf-manet-cbrp-spec/>
5. Joa-Ng, M. and I. Lu, 1999. A peer-to-peer zone based two-level link state routing for mobile ad hoc networks. IEEE J. Select. Areas Comm., 17: 1415-1425. doi: 10.1109/49.779923.
6. Alan D. Amis, Ravi Prakash, Thai H.P. Vuong, Dung T. Huynh, 2000. Max-min D-cluster formation in wireless ad hoc networks. In: Proceedings of IEEE INFOCOM, Vol. 1, pp: 32-41, Tel Aviv, Israel March 26-30. doi: 10.1109/INFOCOM.2000.832171
7. Chiang, C.C., M. Gerla and L. Zhang, 1998. Forwarding Group Multicast Protocol (FGMP) for multihop, mobile wireless networks. Cluster computing. ACM/Baltzer J., Special Issue on Mobile Computing, Vol.1, No.2.pp: 1187-96, November.
8. Hou, T.C. and T.J. Tsai, 2001. An access-based clustering protocol for multihop wireless ad hoc networks. IEEE J. Select. Areas Comm., Vol.19, No.7, pp: 1201-1210, July. doi: 10.1109/49.932689.
9. Ohta, T., S. Inoue, Y. Kakula, 2003. An adaptive multihop clustering scheme for highly mobile ad hoc networks. In: Proceedings of the 6th International Symposium on Autonomous Decentralized Systems (ISADS'03). IEEE Computer Society Washington, DC, USA. 9-11 April. pp:293-300. doi:10.1109/ISADS.2003.1193960.

10. McDonald, A.B. and T.F. Znati, 2001. Design and performance of a distributed dynamic clustering algorithm for ad-hoc networks. In: Proceedings of the 35th Annual Simulation Symposium, IEEE Computer Society Washington, DC, USA. pp: 27-35, April 22-26. doi: 10.1109/SIMSYM.2001.922111.
11. McDonald, A.B. and T.F. Znati, 1999. A mobility-based framework for adaptive clustering in wireless ad hoc networks. *IEEE J. Select. Areas Comm.*, 17: 1466-1487. doi. 10.1109/49.780353.
12. Jane Yang Yu and Peter Han Joo Chong, 2006. An efficient clustering scheme for large and dense mobile ad hoc networks (MANETs). *Elsevier Comput. Commun.*, 30: 5-16.
13. Inn Inn Er and Winston K.G. Seah, 2006. Performance Analysis of mobility-based d-hop (MobDHop) clustering algorithm for mobile ad hoc networks. *Elsevier Comput. Networks*, 50: 3375-3339.
14. Basu, P., N. Khan and T.D.C. Little, 2001. A mobility based metric for clustering in mobile ad hoc networks. In: Proceedings of 21<sup>st</sup> International Conference on Distributed Computing Systems Workshops. *IEEE ICDCS' 01*, pp: 413-418, IEEE Computer Society, Phoenix, AZ, USA. 16-19 April. doi: 10.1109/ICDCS.2001.918738.
15. Chatterjee, M., S.K. Das and D. Turgut, 2000. On demand Weighted Clustering Algorithm (WCA) for Ad hoc Networks. In: Proceedings of IEEE Globecom'00, pp: 1697-701.
16. Zhijun Cai, Mi Lu and Xiaodong Wang, 2003. Channel Access-Based Self-Organized Clustering in Ad Hoc Networks. *IEEE Trans. Mobile Comput.* Vol. 2, No 2. pp: 102-113, April. doi: 10.1109/TMC.2003.1217231
17. Xu, X., X.Y. Hong and M. Gerla, 2002. An ad hoc network with mobile backbones. In: Proceedings of IEEE International Conference on Communications. *ICC2002*, 2-2 May, Vol. 5, pp: 3138-3143. doi: 10.1109/ICC.2002.997415.
18. Chatterjee, M., S.K. Das and D. Turgut, 2002. WCA: A Weighted Clustering Algorithm (WCA) for mobile ad hoc networks. *Cluster Computing*, Vol.5, pp: 193-204, Kluwer Academic Publishers.