

Research Article

AMHC: Adaptive Multi-Hop Clustering based Resource Discovery Architecture for Large Scale MANETs

Saad Al-Ahmadi and Abdullah Al-Dhelaan

Department of Computer Science, College of Computer and Information Sciences,
King Saud University, P.O. Box 51187, Riyadh 11543, Saudi Arabia

Abstract: In this study we propose an efficient clustering protocol called AMHC used for resource discovery in large scale Mobile Ad hoc Networks (MANETs). AMHC is an Adaptive Multi-Hop Clustering generating several non-overlapping network localities (clusters) with explicit elected cluster-heads. Every cluster member is on average d hops away from its cluster-head, where d is an integer parameter for the protocol. The generated set of clusters are highly stable and has low restructuring frequency that takes into consideration the dynamic network topology due to nodes mobility and depleted energy. The head election process is a distributed process based on a node's weight formula calculated by every node independently. The node's weight involves the current energy level, the current neighborhood degree and distance (in number of hops) between the nominated head and the voting node. The cluster-head is responsible of coordinating intra-cluster and inter-cluster resource discovery activities. Inter-cluster communication is handled through gateway nodes which hear from more than one cluster and able to connect clusters with each other. The aim of AMHC is to identify all the possible gateways for creating highly fault-tolerant architecture. AMHC is an asynchronous, scalable and robust architecture capable of handling large amount of resource queries with high degree of power and communication efficiency. We conducted a comparative study using simulation to demonstrate AMHC's efficiency and superiority against other recently proposed clustering algorithms in the literature. The comparison is based on: number of generated clusters, average cluster size, cluster stability and nodes re-affiliation. These results show a lot of promise for AMHC as efficient, energy-aware, load-balance and fault tolerant resource discovery architecture for large-scale MANETs.

Keywords: Distributed algorithms, head-based non-overlapping clustering, mobile ad hoc networks, multi-hop clustering, network resource discovery

INTRODUCTION

Large proliferation of mobile computing devices like smart phones, tablets, pads, note books, PDA, sensors and wearable computers caused revolutionary changes in our society and the need to be connected all the time. These wireless devices are the basis of new generation of computer networks called MANETs which make information available any time and everywhere for pervasive and ubiquitous computing (Edwards, 2006). A MANETs is a multi-hop, self organizing, self configured and rapidly deployed wireless network that has neither fixed infrastructure nor centralized control. MANET nodes are heterogeneous mobile nodes with limited power and short wireless transmission range. Topology may change rapidly and unpredictably due to node mobility, depleted energy, or intermittent wireless communication. Every mobile device is required to act as a router to forward messages in a multi-hop way

without relying on a fixed infrastructure and despite transmission failures and limited available power.

MANETs find application in instantly deployed network requirement when network infrastructure is unavailable, difficult to install, or economically infeasible. Examples of MANET applications include tactical mission scenarios such as battle fields, emergency response operations, disaster relief and short-term scenarios such as public events electronic classroom, etc. There are many challenging issues that need to be considered in designing effective large scale MANET protocols. Some of these issues are routing, MAC, deployment, self organization, security, scalability, energy management, QoS and Resource Discovery (Mian *et al.*, 2009).

MANETs have been the subject of rigorous research to create fully decentralized multi-hop wireless network, but little research has been given to resource management in large scale MANETs compared to routing (Cheng, 2002; Das and Bharghavan, 1997). Resource discovery is an integral part of any computer

Corresponding Author: Saad Al-Ahmadi, Department of Computer Science, College of Computer and Information Sciences, King Saud University, P.O. Box 51187, Riyadh 11543, Saudi Arabia

This work is licensed under a Creative Commons Attribution 4.0 International License (URL: <http://creativecommons.org/licenses/by/4.0/>).

network. The main aim of both routing and resource discovery is to find paths with minimum overhead and quick reconfiguration of broken paths in MANETs. Routing deal with the problem of finding suitable (optimal) paths between nodes for various types of communication (text, voice, multimedia, etc.,) which are, in general, long data flows. Resource discovery queries (registration and lookup) are small transactions with short data flows. In resource discovery, finding suitable paths is usually the dominant factor in communication cost (Sailhan and Issarny, 2005). Reducing the overhead involved in finding the route to the required resource for short data transfer is the main design goal of resource discovery protocols. The two terms resource and service are used interchangeably in the literature. A resource can be a tangible (hardware) or intangible (software) entity. Hardware resources include processors, memories, printers and communication links, to name a few. Software entities include files, database management systems, hyperlinks, security authentication, etc. Resource discovery is the mechanism by which network resources are registered and looked up. Resources are registered by fully descriptive information tags such as resource name, host node location, input and output parameters and various additional resource related attributes. Resources can be categorized based on different objectives such as functionality, content, semantic, characteristics, performance utilization, QoS, etc. The design and performance of resource discovery protocols are influenced by network characteristics as network size, transmission speed, topology dynamics and type of service. Resource discovery performance is measured by network scalability, load balancing, query efficiency and fault tolerance. The increased number of mobile heterogeneous nodes with various types of resources increases the size of MANETs and imposes great challenges for effective exploitation of the available resources with scalable capabilities. Resource discovery process should be tolerant to communication faults between the wireless nodes due to depleted energy or mobility reasons (Lenders *et al.*, 2005).

There are various protocols for resource discovery in MANETs particularly tailored to specific sets of objectives (Mian *et al.*, 2009). The two main tasks in resource discovery are: registration (advertisement) and search (lookup). Every resource discovery protocol consists of at least two basic participating entities: client and server. The client is the entity interested in finding and using the resource offered by the server entity. A third entity called the directory or broker may or may not be used. Resource discovery protocols can be classified into two categories: directory-less protocols and directory-based protocols. In directory-less structure, servers broadcast their advertisements and clients broadcast their requests. Both broadcasting processes may take place at the same time in the network. A major problem in these protocols is how to determine the frequency of resource advertisements in order to reduce network load and avoid redundant

transmissions. In directory-based structures, resources are registered using central or distributed resource repositories to minimize resource search messages. Centralized directory approaches eliminate the need for extensive resource lookup, however, it is not good strategy for MANETs, because the central directory: not always reachable, presents a single point of failure, not scalable and exhaust the limited node's resources. Thus, centralized directories are not well suited for MANETs and distributed directories are more efficient where a few nodes act as resource directories. Distributed directories may coordinate their activities to relief resource providers from replicated publishing for more than one directory node. Distributed directories may be implemented using many techniques such as connected dominating sets and clusters. There are some hybrid structure that consists of directory-based and directory-less architectures where the directory is used to hold information about existing resources only in the node's vicinity. The discovery task can be placed in the application layer of the network structure or coupled with the network layer. In the latter case, the overhead of message traffic is reduced and limited energy resources are saved at the expense of flattening the route discovery task.

Clustering provides an effective method for establishing a hierarchical structure in MANETs (Yu and Chong, 2005). Clustering is an important research topic for MANETs because it guarantees basic levels of resource discovery architecture performance, such as throughput and delay, in the presence of both mobility and a large number of mobile nodes. A large variety of approaches for MANET clustering have been presented, whereby different approaches typically focus on different performance metrics. In MANET, the movement of the network nodes may quickly change the network topology resulting in the increase of the overhead message in topology maintenance; the clustering schemes for mobile ad hoc networks therefore aim at handling topology maintenance, managing node movement or reducing overhead. In clustering, a MANET of n nodes at a specific point of time t is represented by a unit-disk graph $G = (V, E)$, where V is the set of n wireless nodes and the E is the set of edge connecting these n devices at t . Two wireless nodes $u, v \in V$ has an edge $e = (u, v) \in E$ if and only if u and v can communicate directly with each other; i.e., inside the transmission region of each other. Clustering aggregates nodes into groups that share nearby physical proximity. Some clustering approaches select one node in every cluster and assign it the task of cluster-head. The cluster-head act as local coordinator and resolve resource registration and lookup queries. Some clustering approaches do not have explicit cluster-heads; they are assumed implicitly. In this study, we are interested in the first type of clustering algorithms with designated set of cluster-heads.

A dominating set D of G is a subset of V such that each node $u \notin D$ has at least one neighbor $v \in D$. Each

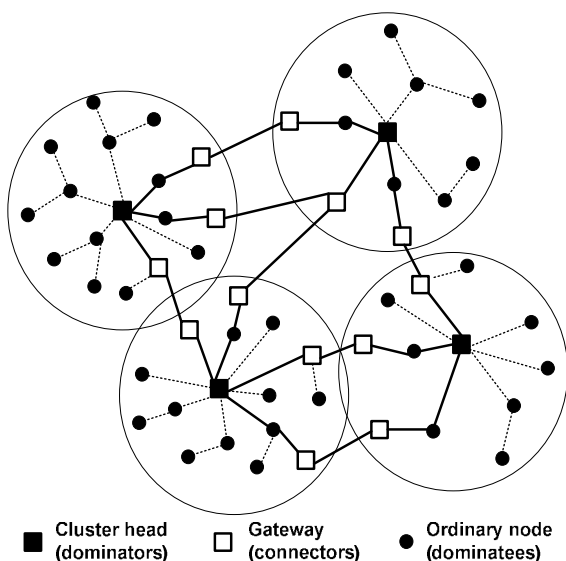


Fig. 1: Backbone of four clusters each of 2-hops

node in D is called a dominator, while a node not in D is called a dominee. The set D is called a one-hop dominating set where the neighborhood definition is expressed as a direct edge between two nodes, i.e., a path length $l = 1$. If the neighborhood is extended to a path length $l \leq d$ for some $d \in \mathbb{N}^+$ we have the d -hop dominating set D_d to be a subset of V such that for each node $u \notin D_d$ there exists a simple path of length at most d to node $v \in D_d$ (Wan *et al.*, 2002). In this study, we deal with the problem of dividing the network into set of multi-hop clusters where the cluster-heads form a graph dominating set that can be used as a virtual backbone to register or lookup network resource. Usually, a cluster is described either by the shortest path to the cluster-head or by its diameter (the longest shortest path between any two nodes in the graph). A one-hop cluster can also called two-hop cluster using the diameter definition. We adopt using the first definition and describe the d -hop cluster as a cluster with members that are at most d hops from their cluster-head. A one hop clusters is an instance of the generic d -hop cluster.

Clustering algorithms that create group of clusters where every node in the network is no more than one-hop, away from their cluster-head generate large number of cluster-heads in large networks (Lin and Gerla, 1997). This type of clustering leads to many problems caused by significant communication responsibility on individual nodes. Extensive communication drains the battery, limits the useful life of the nodes and decreases the overall network performance among many other side effects. As the network size increases, number of messages exchange needed to maintain resource tables may cause congestion in the network and generate significant delays in message propagation from one node to

another. Therefore, it is desirable to have control over the cluster-head density in the network. Cluster members can be classified as: cluster-head (exactly one node), gateway nodes, or ordinary nodes (Fig. 1). A cluster-head is a node responsible for managing resource discovery tasks within its proximity; i.e., the d -neighborhood of the cluster-head. A gateway is a node that has a neighboring node in another cluster. All the other nodes that are non cluster-head and non gateways are ordinary members.

Clustering algorithm in MANETs should be able to maintain its cluster structure as stable as possible while the topology changes to avoid prohibitive overhead incurred during cluster-head changes (Xu *et al.*, 2002). Also, from resource discovery point of view, the communication tasks for resource registration and search can be categorized as: inter-cluster traffic and intra-cluster traffic. Intra-cluster traffic within a single cluster is limited to routes of maximum length of $2d$. Inter-cluster traffic between clusters involves routes of longer path length and hence more overhead than intra-cluster traffic. Inter-cluster communication passes through the gateways from one cluster to another until it establish a path with the required resource while intra-cluster communication is limited to the current cluster locality and involves a short path through the current cluster-head (Chen *et al.*, 2004).

In this study, we propose an Adaptive Multi-Hop Clustering algorithm (AMHC) that forms clusters with members that are on average d -hops away from their elected cluster-head. Cluster-heads are elected based on their weights. Node's weight is calculated based on current energy level, neighborhood degree and path length. The algorithm ensures maximum cluster stability, low communication overhead, small storage requirements, low cluster maintenance, fault tolerance and load balance. AMHC is a distributed algorithm, dynamically forms stable clusters which can serve as underlying resource discovery architecture. AMHC represented MANET as a set of clusters that allows network protocols scalability and permits the design of efficient resource discovery schemes. AMHC reduces the complexity of the resource discovery tasks and routing algorithms, optimizes network shared resources through distributed local management, stabilize the network topology when the cluster sizes becomes large compared to the speeds of mobile nodes.

LITERATURE REVIEW

In wireless networks, a node is equipped with a battery of limited energy that needs to be used carefully to increase its node's lifetime and increase the overall network throughput. Wireless communication consumes large amount of this scarce energy, thus routing resource queries should be efficient and minimal. Fault tolerance is needed to face the mobility results on topology changes and need to rapidly adapt to these changes. Resource discovery usually placed on

upper layers of network protocols, but we see that applying cross-layer design will reduce the infrastructure and protocol overhead and improve service accessibility. Integrated with the network routing layer, the proposed resource discovery mechanism will be more efficient (Kawadia and Kumar, 2003). A very simple and commonly used energy dissipation model is the first order radio model hardware energy dissipation model. It gives $E_{Tx}(k, d)$, the energy consumed by the transmitter to send a k -bit long packet over distance d , as $E_e \times k + E_a \times k \times d^\alpha$, where and gives $E_{Rx}(k)$, the energy consumed by the receiver in receiving a k -bit long packet, as $E_e \times k$. The value E_e is the energy used by the electronics of the transmitter or the receiver, E_a is the energy expended by the transmitter amplifier and $\alpha = 2$ for free space model & $\alpha = 4$ for multi-path fading model. It can be seen that receiving packets is not a low cost operation. Any proposed protocols should consider not only number of transmitting messages but also number of receiving messages. Also, the battery discharge behavior is sensitive to the discharge rate, temperature and the number of charge-recharge cycles. Battery discharge does not have linear behavior. Lower efficiency means less battery energy remains and thus the battery lifetime is proportionally lower. Example, if the battery efficiency is 60% and its rated capacity is 100 mAh at 1 V, then the battery would be drained in 12 min at an average DC-DC current of 300 mA while the same battery with efficiency of 100% would last 1 h (Al-Ahmadi *et al.*, 2011).

Recently, cross layer design become the new trend in wireless communication systems. To fully optimize MANETs, both the challenges from the physical medium up to the applications demand have to be taken into account. Rate, power, mobility and coding at the physical layer can be adapted to meet the requirements of the applications given the current channel and network conditions. The channel knowledge has to be shared between (all) layers to obtain the highest possible adaptively. Careful utilization of some layer interactions that cross the normal boundaries can improve the performance of the communication and hence better application-layer performance. MANETs are based on cooperation between nodes and future MANETs should also be based on collaborative design between network stack layers.

A number of clustering algorithms have been proposed in literature such as Linked Cluster Algorithm (LCA) (Baker and Ephremides, 1981), Lowest-ID Algorithm (Ephremides *et al.*, 1987), Maximum Connectivity Clustering (MCC) (Parekh, 1994), Max-Min Clustering Algorithm (Amis *et al.*, 2000), Mobility-Based metric for Clustering (MOBIC) (Basu *et al.*, 2001) and Mobility-based d-Hop Clustering Algorithm (MobDhop) (Er and Seah, 2006).

The LCA algorithm Baker and Ephremides (1981) was developed and intended to be used with small

networks of less than 100 nodes. LCA organizes nodes into clusters on the basis of node proximity. Each cluster has a cluster-head and all nodes within a cluster are within direct transmission range of the cluster-head. Gateways are nodes that are located in the overlapping region between clusters. Two clusters communicate with each other via gateways. Pair of nodes can act as gateways if there are no nodes in the overlapping region. In this scenario, the delay between node transmissions is minimal and may be tolerated. LCA relied on synchronous clocks for exchange of messages between nodes as TDMA frames. In TDMA, each frame has a slot for each node in the network to communicate without collisions. For every node to have knowledge of all nodes in its neighborhood it requires $2n$ TDMA time slots, where n is the number of nodes in the network. However, as the number of nodes in the network grows larger, LCA will impose greater delays between node transmissions in the TDMA communication scheme and may be unacceptable. The Lowest-ID algorithm (Ephremides *et al.*, 1987) is a revised Linked Cluster Algorithm (or LCA2). In the Lowest-ID algorithm, number of cluster-heads is reduced where a node is said to be covered if it is in the one hop neighborhood of a declared cluster-head node. Starting from the lowest id node to the highest id node, a node declares itself to be a cluster-head if among the non-covered nodes in its one hop neighborhood, it has the lowest id. The Lowest-ID is biased and the cluster-heads with smaller IDs suffer from battery drainage resulting in renumbering of the nodes. It suffers from frequent change of the neighbor list, short system lifetime and low network stability.

In MCC Parekh (1994), cluster-head election is based on degree of connectivity instead of node id. A node is elected as a cluster-head if it is the highest connected node in all of the uncovered neighboring nodes. A node which has not elected its cluster-head yet is an uncovered node otherwise it is a covered node. A node, which has already elected another node as its cluster-head, gives up its role as a cluster-head. In this algorithm, a cluster-head may not be able to handle a large number of nodes due to resource limitations even if these nodes are its immediate neighbors and lie well within its transmission range. Further, since the cluster-head is the most heavily loaded node, having most connections will deteriorate the power of the node and result in more number of cluster-head changes leading to instability in the cluster. This algorithm suffers from dynamic network topology, which triggers frequent changes of cluster-heads. Frequent cluster reconfiguration and cluster-head reselection incur prohibitive overhead.

All of the above algorithms create one-hop clusters in MANETs. They are more suitable for small to moderate dense MANETs in which most of the nodes are within direct transmission range of cluster-heads. However, these algorithms may form a large number of

clusters in relatively large and sparse MANETs. Max-Min algorithm (Amis *et al.*, 2000) generalized the clustering heuristics so that an ordinary node can be at most d hops away from its cluster-head. Nodes participate in the leader election algorithm based on their node id. Each node has two arrays of size $2d$ named WINNER and SENDER records the winner node id and the node id that sent it during a particular round of $2d$ rounds of flooding. The WINNER array helps in identifying the elected cluster-head while the SENDER array is used to trace back the path to it. The $2d$ rounds of flooding where the nodes exchange their ids with their neighborhood are divided into two phases of d rounds called Floodmax and Floodmin. During a particular round of the Floodmax, each node exchanges its WINNER id with the received node id from a one-hop neighboring node. The received node id replaces the stored WINNER node id only if it is greater than it. The Floodmin is similar to Floodmax except it chooses the smallest node id in order to give chance to small node ids to regain back some of their lost territories. After the flooding process terminates, leader election process follows four rules to elect the leader. This heuristic does not take into account mobility and assumes the network is static during the algorithm execution. It also does not take into consideration energy or neighbor degree parameters. Although it is executed in $O(d)$ but the hidden constants that reflect number of messages exchanged is high.

MOBIC Basu *et al.* (2001) considers mobility in the cluster-head election process. The cluster-head is 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 cluster-heads. The mobile node's speed variance is calculated relative to each of its neighbors. A low variance value indicates that this mobile node is relatively less mobile to its neighbors. Consequently, mobile nodes with low variance values in their neighborhoods take the cluster-head responsibility. The aggregate local speed of a mobile node can be estimated. For cluster maintenance, MOBIC uses the least cluster change heuristic with a timer to avoid unnecessary cluster-head relinquishing when two cluster-heads incidentally pass by each other in a short period. That means if two cluster-heads are in the reach range of each other longer than time period, one gives up its cluster-head role. Otherwise, they both keep the cluster-head status. This mechanism reduces the cluster-head change rate by avoiding re-clustering for incidental contacts of two passing cluster-heads. However, the mobility behavior of mobile nodes is not always considered in cluster maintenance, so a cluster-head 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.

The MobDhop (Er and Seah, 2006) clustering algorithm partitions the ad hoc network into d -hop clusters based on mobility metric. The clustering diameter is adaptable with respect to node mobility. It assumes that each node can measure the neighbor RSS (Received Signal Strength) to determine the closeness of its neighbors and measure the distance between them. The RSS value is used as approximation of the distance rather than the actual physical distance between two nodes. Sometimes nodes transmit at lower energy if their power resources are about to run out. The algorithm MobDhop algorithm requires the following parameters: the estimated distance between nodes, the estimated distance mean and variation over time, the relative mobility between nodes and the local stability. The relative mobility is measured at two consecutive time slots as the difference of the estimated distance of one node with respect to another. It shows if their mobility pattern if they are becoming closer to each other or moving away. The local stability is used to elect the most stable cluster-heads. The MobDhop is executed in three successive stages called the discovery stage, the merging stage and the cluster maintenance stage. The discovery stage creates on-hop non-overlapping clusters. The merging stage merges the created clusters during the discovery stage by requests from two adjacent gateway nodes or non cluster-head nodes. The cluster maintenance stage is initiated when the network topology change due to node mobility and/or energy reasons.

MATERIALS AND METHODS

The proposed AMHC resource discovery architecture identifies a subset of nodes within the network (called cluster-heads) where they are given the extra responsibility of being leaders of some other nodes in their proximity. The cluster-heads are responsible for managing resources within their own cluster as well as forwarding resource queries to other cluster-heads in different localities. AMHC creates network backbone that can handle resource queries efficiently. The backbone architecture is resilient to changes in network topology due to nodes mobility and/or limited resources. The main components of the network are the cluster-head nodes and the gateway nodes in every MANET locality. The algorithm typically employs a so called cluster diameter factor d as the parameter serves as an average bound on the cluster diameter during the cluster formation stage. The AMHC includes the cluster formation stage and cluster-maintenance stage.

Cluster-formation stage: During this stage, mobile nodes periodically broadcast their status information using the HELLO message and maintain the status information received from the nodes in the neighborhood list. The AMHC algorithm selects in

every locality a cluster-head not greater than d hops from any node in a cluster within a MANET, wherein $d > 1$, comprising the steps of: determining at least one largest node weight for each node within a d -neighborhood of at least one node; selecting a cluster-head for the d -neighborhood. In every cluster, some nodes are identified as gateways to link the clusters. The AMHC algorithm runs for d rounds of information exchange and in every round it elect a node that has the highest weight based on: current available energy, connectivity degree and path length. If a node has the largest weight in its d -neighborhood then it becomes a cluster-head. The weight of a node is decided by the following weight formula:

$$Node\ Weight = \frac{(d-\gamma \times i+1)}{d} \times (\alpha \times E + \beta \times N)$$

where,

- E : The current battery level normalized to 100% as the fully charged battery (Lithium Ion)
- N : One-hop neighborhood list size
- d : An algorithm parameter indicating number of rounds
- i : The current round where $i \leq d$
- α, β, γ : These constants represent the ratio of energy contribution, neighborhood degree and path length to the voting cluster-head, respectively. The parameter γ involves the cluster-head calculated weight. These constants can be fixed or changed adaptively during the life of MANET. The part of the formula $\alpha \times E$ can be calculated locally by a cross-layer algorithm (Varshavsky *et al.*, 2005), while the part $\beta \times N$ needs neighborhood discovery

The AMHC parameter d represents the average path length between the cluster-head and its members. In every round of the d rounds the propagated cluster-head weight is recomputed based on the current hop count, that's the current path length from the currently elected cluster-head at round i to the current node. Figure 2 shows the affect of using AMCH weight formula. The propagated current head weight is W_H decremented by a value $(x, y, \dots$ where $x < y < \dots$) to reflect the current degradation of the path as its length increases or some weak nodes lies on the path. At the end of d round, each node broadcasts its elected cluster-head to all of its neighbors. Only after hearing from all neighbors can a node determine if it is a gateway node. If a node can hear from other nodes with different cluster-head, then this node is considered a gateway node. Also, all the nodes (members and gateways) report their identities as well as resources to their elected cluster-head.

AMHC does not assume global node identification system. Node address can be from any viable identification system such as: IP addressing, MAC addressing, or any other ad hoc addressing. Node

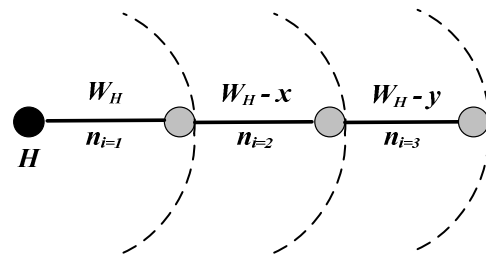


Fig. 2: The effect of cluster parameters d and i on head H weight value W_H ($0 < x < y$)

addresses are used for tables indexing such as the neighborhood list, cluster member list in the cluster head, resource table, etc. For efficient table indexing, all node addresses are converted (locally) to integers using a simple hashing function. The AMHC algorithm ensures the integrity of the neighborhood list of every node by periodic beacon messages. In beacon messages, every node includes three important field:

Node ID	CH ID	CH Weight
---------	-------	-----------	-------

where,

- $Node\ ID$: The sender node calculated weight value
- $CH\ ID$: The current cluster-head ID where the sender node belongs to. If the node is newly arrived one or reawaken node, then it does not belong to any cluster and $CH\ ID = Node\ ID$
- $CH\ Weight$: The current cluster-head weight, if the node is not attached to any cluster, then $CH\ Weight = Node\ Weight$

The mechanics of the algorithm are quite simple. After the neighborhood discovery stage and at some common epoch, each node initiates d rounds of flooding. Each node maintains a logged entry of the results of each flooding round. Initially, all nodes are assumed to be cluster-heads. After every round of the d rounds, every node selects its cluster-head as the node with the largest weight value among currently stored node weight values. This process continues for d rounds and the final stored cluster-head node id is the node with the winner weight is the elected cluster-head. Figure 3 shows sample run of AMHC to generate four clusters a given MANET with $d = 3$. The integer value assigned to the nodes is the corresponding node weight. The average path length of the generated clusters is: 1.3, 1.7, 1.2 and 1.3 hops which gives an overall intra-cluster communication path length of 1.5 hops (i.e., between 1 and 2 hops only).

During the cluster formation stage there may be cases where a cluster-head is disjoint from its members as a result of being overtaken by another cluster-head. In Fig. 4 cluster-head A grows on the areas of 3 other clusters (cluster E is not affected). Cluster B is

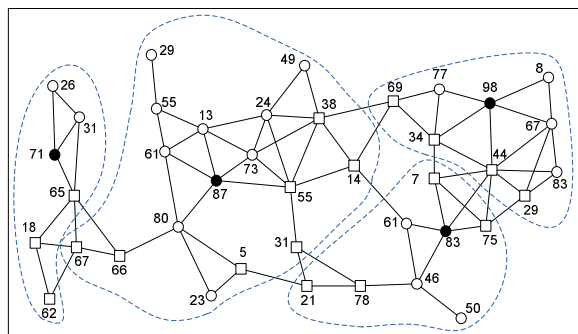


Fig. 3: Example of AMHC with $d = 3$

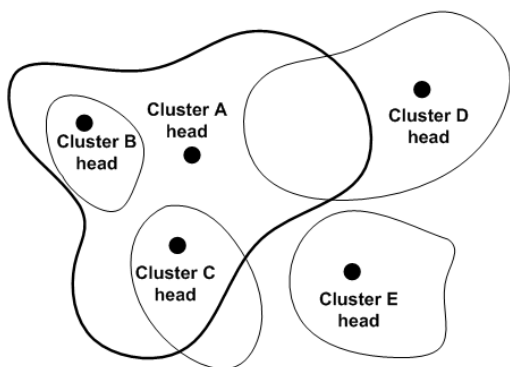


Fig. 4: Different types clustering takeover during the creation of multi-hop clusters

completely swallowed by A while clusters B and C are partially swallowed. Only the case of cluster C causes the problem of separating the leader from its members. AMCH solves this problem by expanding cluster A (at the end of the d round) to overtake all the remaining of cluster C. This case can create a cluster with some members are on $2d$ hops on maximum from their leader. This case is rare and still the members are on average d hops away from their cluster-head. An absent neighbor due to mobility or energy reasons is discovered after some number of successive beacons. We consider a random mobility model where changes in network topology over time are arbitrary and node's neighborhood changes with time. Since nodes may not be aware of changes in their neighborhood, generated clusters and their heads must be updated frequently to maintain accurate network topology.

Cluster-update stage: The cluster-head elected during the cluster-formation stage may not remain the best choice for long in a MANET. Hence we have to monitor the performance of the cluster-head by periodically checking the current weight value of the cluster if it dropped below a threshold ρ . The AMHC favors reelecting cluster-heads in future rounds, thereby reducing transition overheads when old cluster-heads give way all the resource details to the new cluster-heads. Also, AMHC utilizes all the generated gateway nodes (nodes that can hear from more than one cluster)

to generate a backbone with multiple paths connecting adjacent clusters. This property what makes AMHC generates highly tolerant network backbone against node failures and network congestion. Also, it optimizes both stability and load-balance which are mutually conflicting goals. In high stable clustering, cluster-heads consume their resources rapidly since they incur high load for long period of time. In low stable clustering, cluster-heads pass current network resources from the one old leader to the new leader. This transfer of resource tables consumes excessive communication causing performance degradation and energy drainage.

AMHC is initiated by each node and continues to run for the entire lifetime of the node. As a node moves around, it decides which cluster it currently belongs to and what role it currently plays based solely on the local stored information. Each node reacts to the changes in the surrounding topology and changes its status or cluster membership accordingly. A stable and valid cluster structure can be re-established after a certain convergence period. If a node finds itself in a non-clustered state, it will attempt to merge with neighboring clusters. Otherwise, it will declare itself to be a cluster-head of a one-node cluster and periodically tries to merge with a neighboring cluster. The AMHC is a simple distributed asynchronous algorithm that uses a small set of variables and tables in every node. Some of the algorithm variables and data structures are:

- **Cluster exist:** Boolean flag to indicate CREATION PHASE (false) or MAINTENANCE PHASE (true)
- **Is cluster head:** Boolean flag to indicate the current node is a cluster-head or not
If the current node is not a cluster head, then we have the extra fields:
 - Current head node id
 - Current head node weight
 - **Sender node id:** A neighborhood node id that sent the elected cluster head id. The sender node is called the parent node. It plays very important role in building the correct shortest path between any node and its elected cluster-head
 - **Path length:** Current Hop Count
- **Head weight:** Battery Level, Neighborhood list degree
- **One hop list:** One-hop neighbors list
- **T_n :** Time interval to hear from all the neighbors
- **T_r :** Time interval to go for the next round where $r \leq d$

When the link of an ordinary node to its parent node fails, the ordinary node will first try to merge into neighboring clusters after ensuring that the stability can be preserved. If this fails, it will determine its new role in the same way as it does during initialization phase. Its children nodes will be notified of the cluster-head loss and react similarly. The algorithm utilizes several

types of messages where the main components of the message are:

- **Type:** Main types are: HELLO, ACK-HELLO, JOIN, CH-ELECTION, DELEGATE, CREATE
- **ID:** Any valid addressing can be used to identify nodes such as IP address, MAC address, or ad hoc address
- **Weight:** Calculated according to the partial formula $\alpha \times E + \beta \times N$
- **CH ID:** The currently elected cluster-head id
- **CH weight:** The currently elected cluster-head weight
- **Hop count:** Distance from the currently elected cluster-head (number of hops)
- **Round:** Current round $i < d$

The main message types of the AMHC algorithm are:

Message type: HELLO

Function: Discover nearby area and build one-hop neighbors list.

Sender: Request to join the network by a node; it can be new, re-affiliated, or awaken node.

Receiver: Add the sender node id to its one-hop neighbors list after making sure it is not already there. If the receiving node is not part of a cluster (CREATION PHASE) then transmit ACK-HELLO message; else transmit JOIN message (MAINTENANCE PHASE).

Message type: ACK-HELLO

Function: Reply to recently heard HELLO messages.

Sender: If one or more HELLO messages received during time interval $t < T$ then reply to acknowledge all of them.

Receiver: Add the received node id to the one-hop neighbor list. When $t = T_n$, starts the 1st round of CREATION PHASE through CH-ELECTION message.

Message type: CH-ELECTION

Function: This message is sent during every round (of d rounds) of the leader election process. It is used to publish the elected cluster-head id and weight. Before the first round, every node is assumed to be a cluster-head that calculate its own weight using formula-2.

Sender: Sends the current elected cluster-head

Receiver: Upon reception of this message, if the current time $t < T_r$ then the current node re-computes

the received clusterhead's weight according to the received hop count using the weight formula. If the calculated weight is higher than the currently stored cluster-head weight, the received cluster-head information in the CH-ELECTION message is saved and the hop count is incremented.

Message type: RECREATE

Function: When the cluster-head energy is below a threshold ϕ or overwhelmed by large number of members.

Sender: Cluster-head asking his members to initiate a new election process to elect a new cluster-head.

Receiver: Send HELLO message to the neighbors.

Message type: JOIN

Function: A message sent asking a new node to join a nearby existing cluster.

Sender: A node that received HELLO message and the Cluster Exist flag is ON.

Receiver: Store the nearby cluster information and send REGISTER message to the cluster head. If more than one cluster JOIN message received, the new node calculates the received weights and chooses the highest weighted cluster-head.

Message type: REGSITER

Function: Register a node to be a member of a cluster.

Sender: All non cluster-heads nodes should register themselves as members in the table of their elected cluster head. The message is sent upon the completion of the d rounds of election process or when a new node request joining already existing cluster.

Receiver: A cluster-head that records the member information in its table.

RESULTS AND DISCUSSION

The first step in AMHC analysis is to investigate the impact of network size and wireless transmission range on the average number of generated clusters. Two nodes are connected if they are within the wireless communication range of each other. We simulated AMHC using NS-2 with the communication range of the network nodes incremented 50 m/every run. The clustering parameter d was set to 3 for a network of 200, 400, 600 and 800 nodes, respectively. The simulation ran for 2000 sec and the network was sampled every 2 sec. The mobile nodes are moving continuously at 20 m/sec throughout the entire network simulation period. Figure 5 shows the performance of

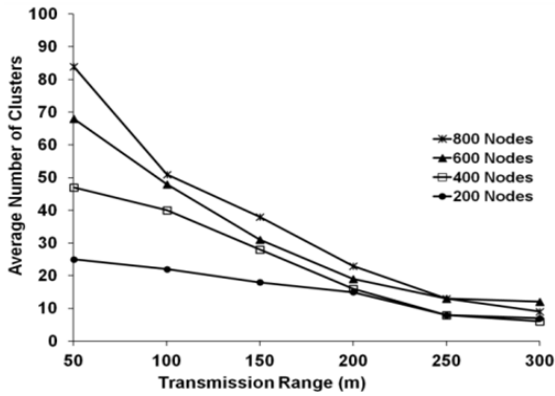


Fig. 5: Impact of network size and transmission range on average number of clusters

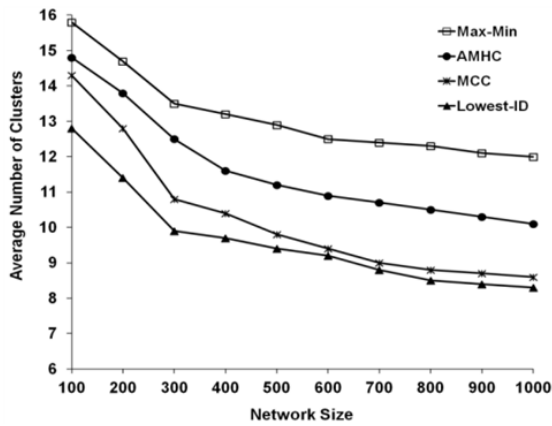


Fig. 6: Average number of clusters

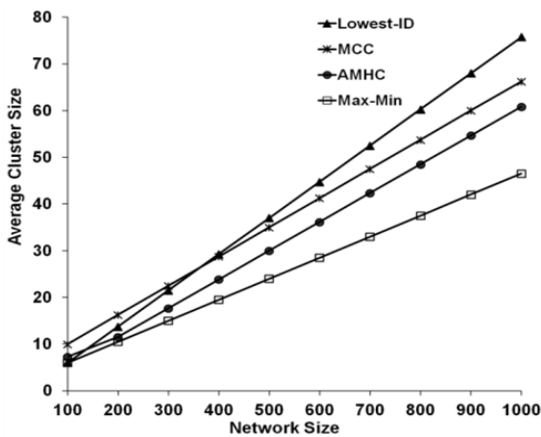


Fig. 7: Average cluster size

AMHC for various MANETs (200, 400, 600 and 800 nodes, respectively) with different transmission ranges. The figure shows that the average number of generated clusters is relatively high when the transmission range is small (10-20 m). When transmission range increases, more nodes can hear each other and hence the average number of clusters decreases and the clusters become larger in size. When the transmission range is large,

mobile nodes tend to remain in the range of their neighbors. Therefore, clusters are less dynamic and the number of cluster-heads changes also decreases.

We evaluated the performance of the resource discovery architecture AMHC against some well known algorithms using NS-2 simulation with the following parameters: network size between 100 and 1000 nodes, network coverage area 350x350 m, node speed between 1 and 50 km/h using random mobility model, node transmission range from 30 to 300 m and simulation duration 15 min without pause. The AMHC performance is measured against three clustering algorithms, Lowest-ID Algorithm (Ephremides *et al.*, 1987), Maximum Connectivity Clustering (MCC) (Parekh, 1994) and the Max-Min Clustering Algorithm (Amis *et al.*, 2000).

We study the performance of the clustering algorithm based on average number of generated clusters, average cluster size and the average duration of the cluster-head as a leader. From the resource discovery point of view, we measure the total message overhead as the network size changes and the current load for every directory agent (cluster-head). Since the two clustering algorithms MCC and Lowest-ID generate one-hop clusters while Max-Min and AMHC generate multi-hop clustering with $d > 1$, we perform d -closure on the connectivity topology of the network before running Max-Min or AMHC. The d -closure graph is a modified graph in which two nodes are directly connected (i.e., one-hop neighbors) if they are at most d -hops away in the current actual network graph.

Both Fig. 6 and 7 show that AMHC generates moderate number of clusters with moderate cluster size compared to the other three clustering algorithms. There are tradeoffs between number of clusters and cluster size which can be stated as number of cluster-heads versus the cluster-head load. If the average number of clusters is low, then cluster size is large and hence the cost of intra-cluster communication is high and high overhead placed on the cluster-head. When the algorithm generates large number of clusters, the average cluster size is small and hence low load on the cluster-head and high inter-cluster communication cost. When inter-cluster communication increases, resource request will incur larger delays. Large number of cluster-heads will increase substantially the number of routing hops causing higher communication overheads and extra energy consumption. Moderate number of clusters with moderate cluster size is necessary for efficient resource discovery protocol.

Any MANET faces the challenge of network topology change due to node mobility, link failure, or depleted energy. Clustering algorithms should cope with such difficult changes in seamless manner. Highly stable clustering algorithm allows the neighbors of a certain node to spend a longer time in its transmission range. The stability property is used to address movements and adjacencies of nodes. Figure 8 shows the cluster-head durations of the clustering algorithms

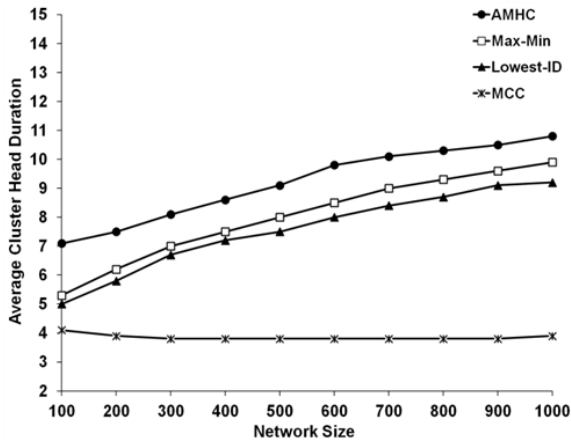


Fig. 8: Average cluster-head duration

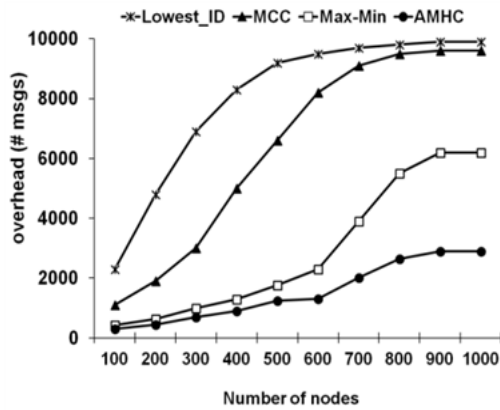


Fig. 9: Message overhead versus number of nodes

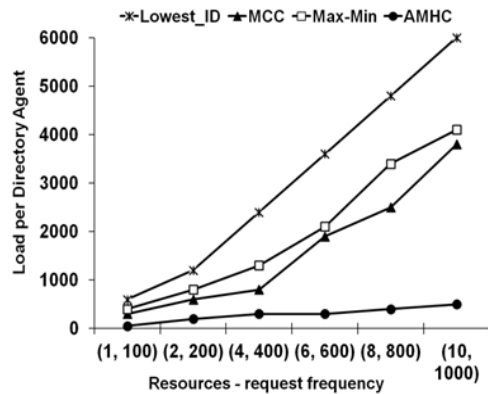


Fig. 10: Load per directory agent

under investigation. AMHC has the highest average cluster-head duration since it generates highly stable clusters that can perform resource discovery tasks for longer period of time.

The overhead of a resource discovery protocol is measured by the number of generated messages (advertisement and search messages) for certain network size. Figure 9 shows the overhead involved in

the AMHC along with the selected resource discovery protocols. Lowest_ID and MCC generate large number of resource discovery messages as the network size increase until certain point when the network is saturated and number of resource discovery messages becomes stable. The generated overhead of Max-Min is higher than AMHC because the created clusters has higher diameter. Moreover, AMHC is highly tolerant because it provides multi-paths property. This important property comes from considering all the available gateways in every cluster.

Overload avoidance is very important characteristic of a resource discovery protocol. Figure 10 shows the load per directory agent (cluster-head) participating in the resource discovery process. The load is measured by a pair of two numbers (x, y) where x is the rate of the client lookup messages per minute and y is the total number of network resources. During our simulation, every node with a resource (server) advertises at the rate of 5 messages/min. During the simulation the client search rate (x) is varied from one to ten search message per minute. The number of resource in the network (y) is chosen to be 10 times the client search rate which is normal value in comparing resource discovery protocols. In all these protocols, all local resource publishing and lookup queries are directed to the cluster head. The cluster head load is very high compared to other node because it serves as local central directory for all nodes that exists within a diameter of $2d$. The Lowest_ID protocol suffers from high load per directory agent as the network size grow. The performance of MCC is comparable to the performance of Max-Min protocol with $d = 1$. The resource discovery architecture AMHC has the lowest load per directory agent.

Fault tolerance is an important characteristic of any resource discovery protocol. If the failed node is a directory agent (cluster-head) that holds information about some resources, then the overall resource discovery system performance is affected and resource utilization is degraded. The ability of AMHC to utilize all the available gateways within a cluster makes the clusters connected the overall network backbone robust.

These results show a lot of promise for AMHC as efficient, energy-aware, load-balance and fault tolerant resource discovery architecture for large-scale MANETs. As the network scale increase, the performance of MAHC becomes more attractive as it based solely on local communications between nodes.

CONCLUSION

Resource discovery is an integral part of any computer network. It is considered an essential base ground for various pervasive and ubiquitous applications. In this study we proposed a new resource discovery architecture based on Adaptive Clustering

algorithm (AMHC) for large scale MANETs that can be used for efficient resource publishing and search operations. AMHC is adaptive multi-hop based on a weighted formula to create several non-overlapping clusters. Every cluster has a leader that acts as a local resource directory. Every node in a cluster is on average d -hops away from its cluster-head for some integer d . The proposed scheme is scalable, fault tolerant and generates evenly distributed clusters. It offers good optimization among load balance, stability and algorithm costs in terms of communication and storage. We have evaluated the performance of our clustering algorithm as well as some of the well known algorithms in the literature to demonstrate its superiority. In conclusion, AMHC can be used to provide an underlying resource discovery based on hierarchical structure to address the scalability of resource discovery protocol in large scale MANETs.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this study through research group no. RGP-VPP-264.

REFERENCES

- Al-Ahmadi, S., A. Al-Dhelaan and N. Al-Hosini, 2011. New data gathering scheme for large scale wireless sensor networks. Proceeding of the 15th WSEAS International Conference on Computers, pp: 111-117.
- Amis, A.D., R. Prakash, T.H.P. Vuong and D.T. Huynh, 2000. Max-min d-cluster formation in wireless ad hoc networks. Proceeding of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM, 2000). Tel Aviv, Israel, Mar. 26-30, pp: 32-41.
- Baker, D. and A. Ephremides, 1981. The architectural organization of a mobile radio network via a distributed algorithm. *IEEE T. Commun.*, 29(11): 1694-1701.
- Basu, P., N. Khan and T.D.C. Little, 2001. A mobility based metric for clustering in mobile ad hoc networks. Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS 2001). Phoenix, Arizona, USA, pp: 413-418.
- Chen, Y.P., A.L. Liestman and J. Liu, 2004. Clustering algorithms for ad hoc wireless networks. In: Pan, Y. and Y. Xiao (Eds.), *Ad Hoc and Sensor Networks*. Nova Science Publishers, Vol. 28.
- Cheng, L., 2002. Service advertisement and discovery in mobile ad hoc networks. Proceeding of the studyshop on Ad hoc Communications and Collaboration in Ubiquitous Computing Environments (CSCW 2002). New Orleans, Louisiana, USA, Nov. 16-20.
- Das, B. and V. Bharghavan, 1997. Routing in ad-hoc networks using minimum connected dominating sets. Proceeding of the IEEE International Conference on Communications (ICC'97), Towards the Knowledge Millennium. Montreal, Que, June 8-12, pp: 376-380.
- Edwards, W.K., 2006. Discovery systems in ubiquitous computing. *IEEE Pervas. Comput.*, 5(2): 70-77.
- Ephremides, A., J.E. Wieselthier and D.J. Baker, 1987. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proc. IEEE*, 75(1): 56-73.
- Er, I.I. and W.K.G. Seah, 2006. Performance analysis of mobility-based d-hop (MobDHop) clustering algorithm for mobile ad hoc networks. *Comput. Networks*, 50(17): 3375-3399.
- Kawadia, V. and P. Kumar, 2003. Power control and clustering in ad hoc networks. Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications, IEEE Societies (INFOCOM 2003). San Francisco California, USA, 30 March-3 April, pp: 459-469.
- Lenders, V., M. May and B. Plattner, 2005. Service discovery in mobile ad hoc networks: A field theoretic approach. *Perv. Mob. Comput.*, 1(3): 343-370.
- Lin, C.R. and M. Gerla, 1997. Adaptive clustering for mobile wireless networks. *IEEE J. Sel. Area. Comm.*, 15(7): 1265-1275.
- Mian, A.N., R. Baldoni and R. Beraldi, 2009. A survey of service discovery protocols in multihop mobile ad hoc networks. *IEEE Pervas. Comput.*, 8(1): 66-74.
- Parekh, A.K., 1994. Selecting routers in ad-hoc wireless networks. Proceedings of the SBT/IEEE Intl Telecommunications Symposium, pp: 420-424.
- Sailhan, F. and V. Issarny, 2005. Scalable service discovery for MANET. Proceeding of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom, 2005), pp: 235-24.
- Varshavsky, A., B. Reid and E. De Lara, 2005. A cross-layer approach to service discovery and selection in MANETs. Proceeding of the IEEE International Conference on Mobile Ad hoc and Sensor Systems Conference, pp: 446-454.
- Wan, P.J., K.M. Alzoubi and O. Frieder, 2002. Distributed construction of connected dominating set in wireless ad hoc networks. Proceeding of the IEEE 21st Annual Joint Conference of the IEEE Computer and Communications Societies, (INFOCOM 2002), 3: 1597-1604.
- Xu, K., X. Hong and M. Gerla, 2002. An ad hoc network with mobile backbones. Proceeding of the IEEE International Conference on Communications (ICC, 2002), 5: 3138-3143.
- Yu, J.Y. and P.H.J. Chong, 2005. A survey of clustering schemes for mobile ad hoc networks. *IEEE Commun. Surv. Tutorials*, 7(1): 32-48.