Authentication based Hop Count Clustering Algorithm in Mobile Adhoc Network

Pranita M. Potey
Department of Electronics &Telecommunication
Vivekanand Education Society's Institute of
Technology, Chembur, India.

Naveeta Kant
Department of Electronics Engineering
Vivekanand Education Society's Institute of
Technology, Chembur, India.

ABSTRACT

Recently, extensive research efforts have been devoted to the design of clustering algorithms to organize all the hosts in a mobile ad hoc network into a clustering architecture. Clustering is an important research topic because clustering makes it possible to guarantee basic levels of system performance, such as throughput and delay, in the presence of both mobility and a large number of mobile. MANETS have a limitation of battery power, cluster formation is expensive in terms of power depletion of nodes. This is due to the large number of messages passed during the process of cluster formation. A large variety of approaches for ad hoc clustering have been presented, whereby different approaches typically focus on different performance metrics. In this paper, we use the hop count based approach for binding a node to a cluster. We minimize the explicit message passing in cluster formation. We also used the route message of a proactive routing protocol for keeping track of nodes in cluster. Our scheme also involves low latency in the cluster formation phase. In addition, we choose the cluster gateway during cluster formation avoiding the need to explicitly discover the gateways, thus reducing further the transmission overheads. In this paper, we will propose an efficient clustering algorithm that can establish a stable clustering architecture by keeping a host with weak battery power from being elected as a cluster head. Addition to this we will focus authentication of node. Computer simulations show that clustering architectures generated by our clustering algorithm are more stable than those generated by other clustering algorithms.

General Terms

Algorithm, Authentication, Clustering, Hello message.

Keywords

Hop count, clustering, cluster Maintenance.

1. INTRODUCTION

Mobile ad hoc network (MANET) is a collection of wireless nodes that can dynamically form a network to exchange information without using any pre-existing fixed network infrastructure.

Wireless nodes in mobile ad hoc networks can be dynamically set up anywhere & anytime without using any pre-existing network infrastructure. It is an autonomous system in which, mobile hosts connected by wireless links are free to move randomly. A mobile ad hoc network is a network that results from the cooperative engagement of a collection of (mobile) hosts without any centralized access point. The mobile hosts volunteer to act as a router that forwards packets to the destination. Therefore, the routing protocols in MANET are not performed by routers, but performed by normal hosts & hence often mobile nodes act as routers at the same time.

The previous research on mobile ad-hoc network has heavily stressed the use of clustering algorithm because clustering simplifies routing and can improve the performance of flexibility and scalability, improved bandwidth utilization, and reduce delays for route strategies. Addition to this node authentication is also an important issue. Each node possesses a pair of public and private keys based on an asymmetric algorithm like RSA. Based on this key pair each node can

Perform authentication and message integrity operations or further exchange pair wise symmetric keys used for efficient authentication and encryption operations. There are three different scenarios where authentication needs to be performed. First when a node joins a network for the first time, secondly when a node leaves a cluster and joins another cluster and last when a node from a cluster wishes to communicate with a node belonging to another cluster In this paper, we try to present a scheme that leads to cluster formation and node authentication which efficiently uses the resources of the MANETs. We define below, some of the terminology used in the remaining sections.

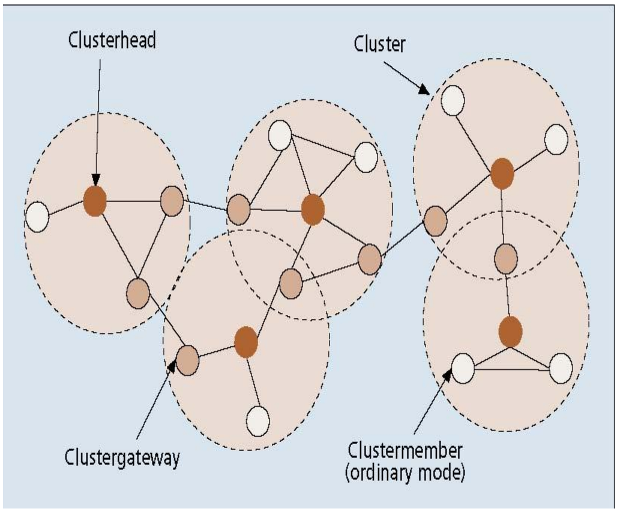
Cluster Head: If a clusterhead detects that it has a bi-directional link to another clusterhead for a time period, it changes its state to member if the other clusterhead has lower ID. Otherwise it stays the clusterhead and the other node has to change its state. This is a special case which may result in cluster re-organization.

Undecided: If a member losses its clusterhead, it looks for bi-directional links to other nodes. If it detects any, it changes its state to clusterhead if it has the lowest ID, otherwise it switches to the undecided state. Each member node belongs at least to one cluster.

Cluster Gateway: A cluster gateway is a non-cluster head node with inter-cluster links, so it can access neighboring clusters and forward information between clusters.

Cluster Member: A cluster member is a node that is neither a cluster head nor a cluster gateway.

Fig. 1. Cluster structure



In this paper we propose Hop Count Based Approach for Clustering in Mobile Ad-Hoc Networking, Which maintains stable clusters. The following part of the paper is divided as follows: Section 1 reviews some of the related work and their drawbacks; Section 2 discusses the importance of self-organization principles in MANETs; Section 3 describes our proposal; we conclude with Section 4.

2. RELATED WORK

In this section, we describe some of the most important clustering schemes. Several approaches to cluster formation have been proposed and surveyed in [9]. Here we briefly review the salient features of a few major approaches.

2.1 DS Based Clustering

In this scheme, routing is done based on a set of dominating nodes [1] which function as the cluster heads and relay routing information and data packets. The vertices of a Dominating Set (DS) act as cluster heads and each node in a MANET is assigned to one cluster head that dominates it. A DS is called a Connected Dominating Set (CDS) if all the dominating nodes are directly connected to each other. Wu's CDS Algorithm [1] gives details for the formation of CDS. Later, Chen's Weakly Connected Dominating Set (WCDS) algorithm [2] was proposed which relaxed some of the rules of Wu's Algorithm to form a Weakly Connected Dominating Set. There are many disadvantages with the CDS algorithm. The cluster head in CDS algorithm dissipates more power as compared to other nodes in the cluster since all inter-cluster routing and forwarding happen through it alone. Hence it has a shorter lifespan than the other nodes in the cluster. The cluster head re-election is done after the cluster head dies or moves out of the range of the cluster. This re-clustering incurs a large communication (and power dissipation) overhead.

2.2 The distributed and mobility-adaptive clustering algorithm

The DMACA is a generalization of the lowest-ID clustering algorithm. In the DMACA, each node is assigned a weight, which becomes the criterion of electing a cluster head. The weight of a node can be given according to its some qualities such as its mobility, its processing power, and so on. Obviously, if the weight of each node is set to its ID (degree), the DMACA will become the lowest-ID (the highest-connectivity) clustering algorithm. At first glance, the DMACA seems to be able to successfully solve the stability problem. This is because if the battery power is taken as the node weight, nodes with larger battery power will have a higher probability of becoming cluster heads. Thus, the stability of the whole clustering architecture seems to become higher. However, this is not always true.

2.3 Power aware clustering

Vikas and Kumar [3] proposed cluster power algorithm in which dynamic and implicit clustering is done on the basis of transmit power level. The transmit power level is the power level required to transmit each packet. The transmit power level to a node inside the cluster is less s compared to the level required to send a node outside the cluster. So here the clustering is done keeping the nodes with lower transmit power level together. The primary drawback of their scheme is that there is no cluster head or cluster gateway. Each node here has routing tables corresponding to different transmit power levels. The routing table for a power level ***Pi*** in a node. This leads to the overhead of collecting the state information and building many routing tables for each power level in a node. There were also other algorithms such as Wu's Algorithm [4] which try to build the DS keeping power as criteria to choose the cluster head. But this scheme also does not overcome the basic drawback of DS based clustering algorithm.

2.4 Mobility based and weighted clustering

Some clustering schemes have been proposed keeping mobility as a metric for cluster construction. In mobility aware clustering, cluster architecture is determined by the mobility behavior of mobile nodes. In such schemes, a cluster is formed by grouping mobile nodes moving with the same velocity. This results in the formation of highly connected intra-cluster links. MOBIC [5] was proposed which takes aggregate local mobility as the metric for cluster formation. Each node broadcasts two hello packets, separated by a time interval, to its neighbors. Every node calculates the relative mobility of each of its neighbors using the signal strength of the hello packets received from each of them. Each node then calculates its aggregate mobility as the average of the relative mobility of its neighbors and broadcasts it to the other nodes. The node with the lowest aggregate mobility is chosen as the cluster head. This requires larger communication overhead and a higher latency in cluster formation. There also exist other approaches like combined metric based clustering such as On Demand Weighted Clustering Algorithm [6]. This approach calculates a combined weight factor and uses this metric for the cluster formation. These metric based clustering schemes require explicit control messages for cluster formation to exchange the metric information, thus leading to more communication overhead.

2.5 Energy-efficient clustering

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

2.6 Load balancing clustering

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

3. PROPOSED ALGORITHM

In this section, we will propose a new clustering algorithm ***"Authentication Based Hop Count Clustering Algorithm in Mobile Adhoc Network"*** to form a stable clustering architecture. Before describing our clustering algorithm in detail, we will discuss Scenarios when Authentication is needed and make some assumptions, which are common in designing clustering algorithms for MANETs.
There are three different scenarios where authentication needs to be performed they are as follows:

When a node joins a network for the first time:

This is a trivial case where a strong authentication is done by sending a challenge and receiving a response. The system key pair is used for mutual authentication between the joining node and a existing member of the network. When a new node joins the network and is detected by a cluster head (by means of hello messages), it gets the cluster key and also the table containing the cluster ids and head public keys.

When a node leaves a cluster and joins another cluster:

This situation arises due to the movement of nodes. When a node moves from a cluster to new one, the new cluster head treats it as any new node joining its cluster. A mutual authentication is performed between the moved node and its new cluster head using the system key pair. The cluster head then gives the node the cluster key for the new cluster. The old cluster purges the entry for this node when it doesn't receive hello message for a certain predefined time interval.

When a node from a cluster wishes to communicate with a node belonging to another cluster: This is a complex scenario and our scheme tries to minimize the overhead involved here. For complete confidentiality of the message, the entire packet has to be encrypted with a session key. The session key is shared solely by the two parties involved in the communication and therefore serves as authentication. But, in cases where the emphasis is on authentication alone and confidentiality is not very critical, it is unnecessary to encrypt the whole packet. A small encrypted tag appended to each packet, is sufficient to achieve authentication. In order to prevent the replay problems we need to perform strong authentication for each packet, i.e. a series of challenge and response back and forth. It is not feasible to do this for each packet as the delays and packet overhead would be too high. Therefore an algorithm that offers reliability similar to that achieved by performing strong authentication for each packet is implemented.

Following assumption we made before final algorithm,

1. The network topology is stagnant during the implementation of the clustering algorithm.

2. A packet broadcasted by a node can be received correctly by all its one-hop neighbors within a predetermined time.

3. Each node has an irreplaceable ID and knows its degree (the number of its neighbors hop). At the same time, each node knows the ID and the degree of its every one-hop neighbor.

4. The precondition also includes the use of a preemptive routing protocol such as DSDV within the cluster. We define a parameter D that limits the number of hops the node can be away from its cluster head. We assume that the parameter D is known to each node participating in the cluster formation. This hop limit, D, can be tuned based on empirical results and/or dynamically, keeping the mobility into consideration. If the nodes in a MANET are highly mobile, then, the value of D for the cluster can be relatively small as compared to a scenario where mobile nodes in a MANET are stable.

The cluster formation starts when a node boots up and broadcasts a cluster solicitation message to its immediate neighbors. If it does not get any reply within the maximum attempts, it declares itself a cluster head. If it receives a cluster advertisement, in response to its solicitation it examines the hop count value and if it is less than D then, it joins the cluster with the minimum hop count to the cluster head. However if the hop count advertised is D, then it declares itself as a cluster head. We describe, below, the steps involved in this process:

3.1 Cluster formation:

Step 1: As shown in Fig 2 when a node i does not belong to any cluster and wants to join a cluster it announces a Hello message to its immediate neighbors.

Step 2: Node j and node l which receive the Hello message send out a cluster advertisement message. The cluster advertisement of the node j and the node l contains information such as the cluster head ID of the corresponding cluster. It also contains information regarding the number of hops the new node i will be away from the cluster head. Each node maintains its approximate hop count. As shown in Fig 2 the hop count sent by node j to node i in the cluster advertisement is having the value 2 that is its own hop count incremented by one. Similarly the hop count value in the cluster advertisement sent from node l to node i is 3.

Step 3: The node i, after receiving the cluster advertisement(s), first check whether the hop count value in cluster advertisement message is less than D value. Then it chooses the cluster head of the node with the minimum hop count in its advertisement, as its cluster head. Then it sends a cluster acceptance message to the nodes whose cluster advertisements have been received. It sets the A bit to indicate acceptance of advertisement. If the hop count value is the same in two or more cluster advertisements then one of them can be selected randomly.

Step 4: When the new node i receives two or more cluster advertisements from nodes that belong to different clusters, it declares itself as a cluster gateway. It sets the G bit, in the cluster acceptance message. This is shown in Fig 2 with message labeled 3.

Step 5: If the new node i does not receive any cluster advertisement after sending the Hello message multiple times or it receives all advertisements with maximum hop count, it declares itself as a cluster head.

3.2 Authentication:

Step 6: Nodes l and node j and their respective cluster heads, CH1 and CH2, are marked. The cluster head acts as the certification authority for all its members. If l wishes to communicate with j, the following steps are to be performed for data authentication.

Step 7: The two communicating parties, i and j, exchange a session key that is only valid for one TCP session. This is exchanged after mutual authentication for which their corresponding heads act as CAs. The head's keys are used for secretly exchanging session keys. The Cluster Heads then decrypt and transmit the session key to their corresponding members who are involved in the session.

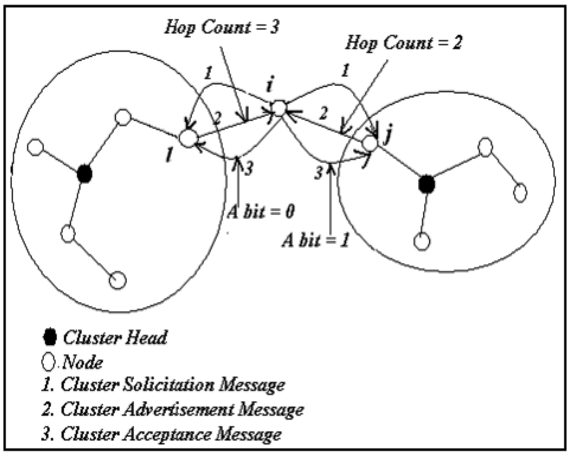

Hop Count = 3
Hop Count = 2
1
1
i
2
2
l
j
3
3
A bit = 0
A bit = 1
● Cluster Head
○ Node
1. Cluster Solicitation Message
2. Cluster Advertisement Message
3. Cluster Acceptance Message

Fig 2. Cluster Formation as described

4. EVALUATION RESULTS

Simulation Setup: The scenario of two clusters and one central system has been simulated with proper authentication using the JAVA coding (jdk1.5.0 Version). Its provides an easy way to create clusters and a non-cluster node joining a new cluster. To join any new cluster that node should be authenticated so fig # show simulation set up of authentication.

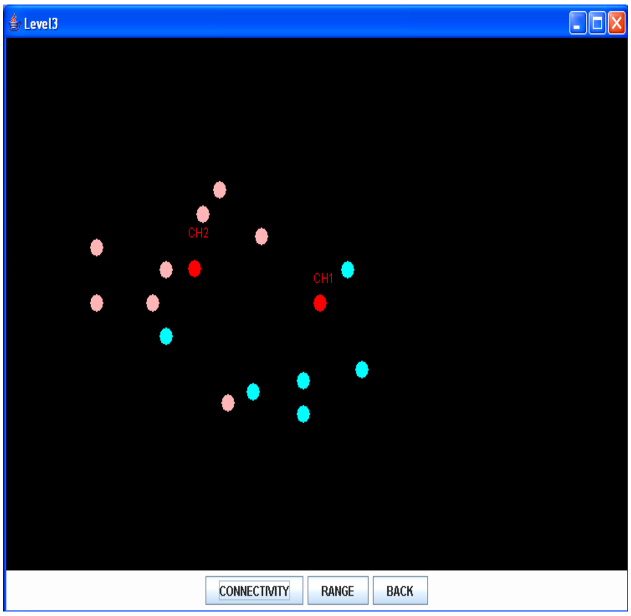
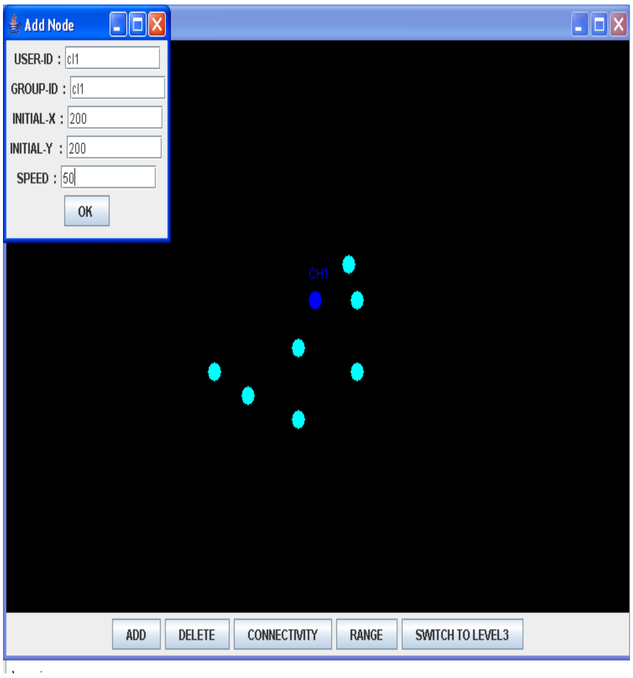


Fig 3. Cluster Formation using Hop Count based Algorithm


Add Node

USER-ID : cl1

GROUP-ID : cl1

INITIAL-X : 200

INITIAL-Y : 200

SPEED : 50

Fig 3. Authentication of when new (non-cluster) node join a cluster

5. CLUSTER MAINTAINANCE

After the new node has chosen its cluster head, the new node is included in the route table of the neighboring node. If the new node declares itself as the cluster gateway then there is a column in the route table of mobile nodes in the cluster which will be marked as 1. Within each cluster, a proactive routing protocol such as DSDV is used. Thus, every node in the cluster knows about every other node in its own cluster. When a new node joins the cluster, it starts advertising itself and after a short time, all nodes in its cluster will have an entry for this node in their routing table.

When a node moves out of the range of the cluster, it becomes unreachable to the nodes in the cluster. Thus the entry for this node is deleted from each node's route table within the cluster. Hence the mobility of the node does not cause any ripple effect of re-clustering in cluster maintenance as it occurs in DS based clustering scheme. When a node becomes unreachable to a cluster it can join another cluster by following the cluster formation steps as discussed above.

6. CONCLUSION & FUTUREWORK

The scheme proposed in this paper, is feasible in the practical world. As our scheme is self-organized it does not require any central control to start the clustering. It also does not require knowledge of the entire MANET and its topology to cluster the nodes. Our clustering scheme does not involve latency in cluster formation. The clustering parameter ***k*** helps in adapting the formation of the cluster to its environment. When the nodes in a MANET are having high mobility the ***k*** value can be smaller as compared to the case when the nodes are stable. Simulation experiments can be done with different mobility to arrive at an approximate empirical value of ***k*** under different conditions. Thus our future work aims to simulate the proposed scheme. We will also try to find solutions to the scenarios of the cluster maintenance phase as part of our future work.

6. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the this algorithm.

7. REFERENCES

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

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

[3] Vikas Kawadia and P.R.Kumar, "Power Control and Clustering in Adhoc Networks", *IEEE Proc. INFOCOM conference* 2003, pp 459 - 469.

[4] J.Wuetal, **"**On Calculating Power Aware Connected Dominating Sets for Efficient Routing in Adhoc Wireless Networks**",** *J. Commun. and Networks vol 4, no . 1,* Mar 2002, pp 59 – 70.

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

[6] M. Chatterjee, S. K Das, and D. Turgut, "An On-Demand Weighted Clustering Algorithm (WCA) for Adhoc Networks", *in Proc 6th ISADS'03*, Apr 2003.

[7] "Markov Chain Trust Model for Trust-Value Analysis and Key Management in Distributed Multicast MANETs",Ben-Jye Chang; Szu-Liang Kuo; Vehicular Technology, IEEE Transactions on Volume: 58 , Issue: 4 , 2009 , Page(s): 1846 - 1863IEEE JOURNALS

[8] "Cluster-Based Cross-Layer Design for Cooperative Caching in Mobile Ad Hoc Networks", Denko, M.K., Jun Tian , Nkwe, T. ,Obaidat, M.S. Dept. of Comput. & Inf. Sci., Univ. of Guelph, Guelph, ON, Canada,Systems Journal, IEEE Volume: 3 , Issue: 4 , 2009.

[9] Jane Y. Yu and Peter H.J. Chong, "A Survey of Clustering Schemes For Mobile Ad Hoc Networks" *IEEE Commun. Survey & Tutorial*, First quarter 2005, Vol 7 No. 1.

[10] Distributed Combined Authentication and Intrusion Detection With Data Fusion in High-Security Mobile Ad Hoc Networks" Shengrong Bu; Yu, F.R.; Liu, X.P.; Mason, P.; Tang, H.Vehicular Technology, IEEE Transactions , 2011 , IEEE JOURNAL