Efficient Active Clustering of Mobile Ad-Hoc Networks

Damianos Gavalas[1], Grammati Pantziou[2], Charalampos Konstantopoulos[3], Basilis Mamalis[2]

[1] Department of Cultural Technology and Communication, University of the Aegean, Arionos & Sapfous Str., 81100, Mytilini, Lesvos Island, Greece
dgavalas@aegean.gr

[2] Department of Informatics, Technological Education Institute of Athens, Ag. Spyridonos Str., 12210, Egaleo-Athens, Greece
{pantziou, vmamalis}@teiath.gr

[3] Computer Technology Institute, 11 Aktaiou & Poulopoulou Str., 11851 Thiseio, Athens, Greece
konstant@cti.gr

Abstract. Mobile ad hoc networks comprise a collection of wireless nodes that dynamically create a network among themselves without using any pre-existing infrastructure. Clustering of mobile nodes among separate domains has been proposed as an efficient approach to answer the organization, scalability and routing problems of mobile ad hoc networks. In this work, we propose an efficient distributed clustering algorithm that uses both location and energy metrics for stable cluster formation. Unlike existing active clustering methods, out algorithm relieves the network from the unnecessary burden of control messages broadcasting. This is achieved through adapting broadcast period according to mobile nodes mobility pattern. For relative static network topologies, broadcast period is lengthened. In contrast, broadcast period is shortened to meet the requirements of highly dynamic networks for consistent cluster configurations.

1 Introduction

The field of wireless networking emerges from the integration of personal computing, cellular technology, and the Internet. This is due to the increasing interactions between communication and computing, which is changing information access from "anytime anywhere" into "all the time, everywhere". At present, a large variety of networks exists, ranging from the well-known infrastructure of cellular networks to infrastructureless mobile wireless ad-hoc networks (MANETs).

Unlike fixed wireless networks, MANETs are characterized by the lack of infrastructure offering fixed communication backbone to network users. Mobile Hosts (MHs) are free to move and organize themselves in an arbitrary fashion, while communication between peers is performed through multiple, multi-hop links. In the absence of a wired infrastructure, MHs are required to relay messages to other devices apart from solely transmitting and receiving packets [5].

Routing in ad hoc networks faces extreme challenges from node mobility/dynamics, potentially very large numbers of nodes, and limited communication resources (e.g. bandwidth and energy). The routing protocols for ad hoc wireless networks need to adapt quickly to frequent and unpredictable topology changes and must be parsimonious of communications and processing resources [2].

Several application fields have been identified for MANETs, including collaborative computing in convention centers, conferences, and electronic classrooms, on-the-fly message and file exchanges, crisis management services applications (e.g. disaster recovery); they are also expected to play an important role in the military and law enforcement.

Among the many challenges for ad hoc network designers and users, scalability is a critical issue. In particular, for topologies including large numbers of nodes, control overhead, such as routing packets, requires a large percentage of the limited wireless bandwidth. This problem becomes more emphatic due to the mobility feature of topology nodes and frequent wireless link failures. One promising approach to address the scalability issue is to build hierarchies among the nodes, such that the network topology can be abstracted. This process is commonly referred to as *clustering* and the substructures that are collapsed in higher levels are called *clusters* [1].

The concept of clustering in MANETs is not new, and there have been many algorithms that consider different metrics and focus on diverse objectives. However, existing algorithms fail to guarantee stable cluster formations. More importantly, they are all based on periodic broadcasting of control messages resulting in increased consumption of network traffic and MH energy. In this article, we describe a distributed algorithm for efficient and scalable clustering of MANETs that corrects the two aforementioned weaknesses.

The remainder of the paper is organized as follows: Section 2 provides an overview of clustering concepts and algorithms. Section 3 describes the details of our Adaptive Broadcast Period algorithm. Finally, Section 4 concludes the paper and draws directions for future work.

2 Clustering

In clustering procedure, a representative of each subdomain (cluster) is ‘elected’ as a *cluster head* (CH) and a node that belongs to more than two clusters at the same time is called a *gateway*. Remaining members are called *ordinary nodes*. A cluster is defined by the transmission area of its CH. With an underlying cluster structure, non-ordinary nodes can be the dominant forwarding nodes, as shown in Fig 1.

CHs hold routing and topology information, relaxing ordinary MHs from such requirement, however they represent network bottleneck points. In clusters without CHs, every MH has to store and exchange more topology information, yet, that eliminates the bottleneck of CHs. Yi et al. identified two approaches for cluster formation, *active* clustering and *passive* clustering [6]. In active clustering, MHs cooperate to elect CHs by periodically exchanging information, regardless of data transmission. On the other hand, passive clustering suspends clustering algorithm until the data traffic commences. It exploits on-going traffic to propagate “cluster-related information" (e.g., the state of a node in a cluster, the IP address of the node) and collects neighbor information through promiscuous packet receptions.

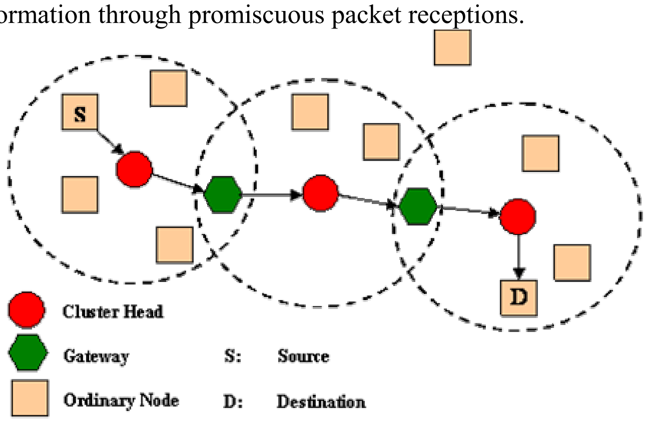

Fig. 1. Cluster heads, gateways and ordinary nodes in mobile ad hoc network clustering; demonstration of message forwarding through cluster heads and gateways

Passive clustering eliminates major control overhead of active clustering, still, it implies larger setup latency which might be important for time critical applications; this latency is experienced whenever data traffic exchange commences. On the other hand, in active clustering scheme, the MANET is flooded by control messages, even while data traffic is not exchanged thereby consuming valuable bandwidth and battery power resources.

A good clustering method should be able to partition a MANET quickly with little control overhead. Due to the dynamic nature of MANETs, optimal cluster formations are not easy to build. To this end, two distributed clustering algorithms have been proposed: Lowest ID algorithm (LID) and Highest Degree algorithm (HD) [6]. Both of them belong to active clustering scheme.

In LID algorithm, each node is assigned a distinct ID. Periodically, nodes broadcast the list of nodes located within their transmission range (including themselves) through a "Hello" control message. The lowest-ID node in a neighborhood is then elected as the CH; nodes which can 'hear' two or more CHs become gateways, while remaining MHs are considered as ordinary nodes. In HD algorithm, the highest degree node in a neighborhood, i.e. the node with the largest number of neighbors is elected as CH. Fig. 2 compares LID vs. HD algorithm approaches.

LID method is a quick clustering method, as it only takes two "Hello" message periods to decide upon cluster structure and also provides a more stable cluster formation than HD. In contrast, HD needs three "Hello" message periods to establish a clustered architecture [3]. In HD method, losing contact of a single node (due to MH movement), may cause failure of the current CH to be re-elected. However, HD method can get fewer clusters than LID, which is more advantageous in large-scale network environments.

In current clustering schemes, stability and cluster size are very important parameters; however, reducing the number of clusters does not necessarily result in more efficient architectures. A CH may end up dominating so many MHs that its computational, bandwidth and battery resources will rapidly exhaust. Therefore, effective control of cluster scale is another crucial factor.

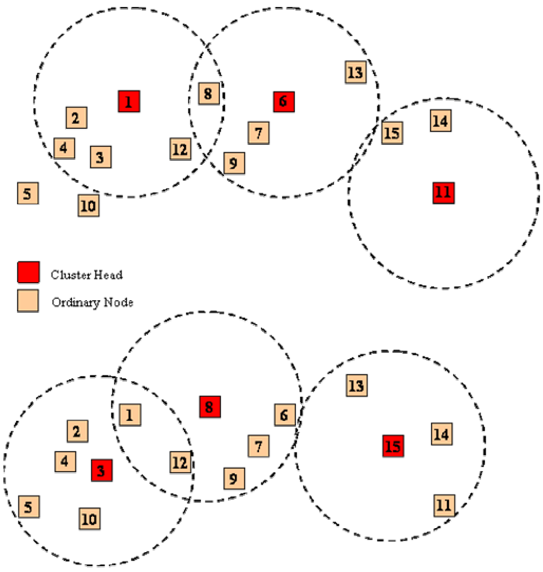

Fig. 2. LID vs. HD algorithms clustering

Summarizing, both LID and HD algorithms use exclusively location information to form clusters and elect CHs. In a more recent approach, Li et al proposed Vote-based Clustering (VC) algorithm, where CH elections are based not purely on location but also on the power level of MHs [3]. In particular, MHs with high degree (large number of neighbors) and sufficient battery power are elected as CHs. However, simulations have shown that the combination of position and power information in clustering procedure results in frequent CH changes, i.e. overall cluster structure instability [3].

In addition, LID, HD and VC algorithms share a common design characteristic which derives from their active clustering origin. Cluster formation is based on the periodic broadcast of 'Hello' signaling messages. In cases where MHs are relatively static (e.g. in collaborative computing, on-the-fly conferencing, etc), periodic 'storms' of control messages only occur to confirm that cluster structure established in previous periods should remain unchanged. These unnecessary message broadcasts not only consume network bandwidth, but valuable battery power as well.

3 Adaptive Broadcast Period (ABP) algorithm

Our Adaptive Broadcast Period (ABP) algorithm aspires to correct the inefficiencies of existing active clustering algorithms (LID, HD and VC). Emphasis is given on two directions:

- A quick method for cluster formation is needed; required speed though should not be achieved at the expense of instable cluster configurations. To meet this objective, we modify VC algorithm so as to avoid frequent CH 're-elections'.
- Control messages broadcast period should be dynamically adapted in order to avoid unnecessary message exchanges when the mobility pattern of MHs is such that network topology is relatively static.

The methodology chosen to achieve the three aforementioned objectives is detailed in the following sections.

3.1 Cluster Formation

Cluster formation strategy extends the ideas implemented in VC algorithm. Unlike LID and HD protocols, both position and battery power metrics are considered in CH election.

However, emphasis has been given to prevent frequent CH changes and prolong the average lifetime of CH and cluster membership, therefore, meeting the requirement for steadier cluster formations.

We assume that each MH has a unique identifier (MH_ID), which is a positive integer. MHs also hold information about the identity of their assigned CH (CH_ID). CHs are easily identified by their identical MH_ID and CH_ID values.

Control information is communicated through 'Hello' messages, transmitted on the common wireless channel. Every MH acquires topology information from incoming 'Hello' messages. Another attribute of MHs is their battery power level (percentage of remaining over full battery power), which is a positive integer, $0 \leq b \leq 100$, linearly decreased over time; naturally, battery energy of CHs exhausts faster than ordinary MHs as they serve a number of MHs, forwarding messages on their behalf.

Clustering algorithm. Our clustering algorithm considers both location and power information to partition a MANET into separate clusters. In this context, we introduce the concept of "cluster head competence" (CHC) which represents the competence of a MH to undertake the role of a CH.

The format of a typical 'Hello' message is shown in Fig. 3. Each 'Hello' message includes identifications of its sender (MH_ID) and sender's assigned CH (CH_ID). CCH represents a weighted sum of sender's degree (number of valid neighbors) and its battery power level. Finally, the BP field is used to adapt the broadcast period within a particular cluster (see Section 3.2).

MH_ID	CH_ID	CHC	BP
8 bit	8 bit	8 bit	8 bit

Fig. 3. 'Hello' packet format

CHC values are calculated according to the following equation:

$$\text{CHC} = (w_1 \times d + w_2 \times b) - p \quad \textbf{(1)}$$

- c_1, c_2: weighted coefficients of MH degree and battery availability, respectively;
- d: Number of neighbors (degree of MH);
- b: Remaining battery lifetime (percentage of remaining over full battery power);
- p: 'handover' penalty coefficient (explained below).

The algorithm involves the following steps:

1. Each MH sends a 'Hello' message randomly during a 'Hello' cycle. If a MH has just joined the MANET, it sets CH_ID value equal to a negative number. That signifies a MH is not a member of any cluster and has no knowledge of whether it is within transmission radius of another MH.
2. Each MH counts how many 'Hello' messages it received during a 'Hello' period, and considers that number as its own degree (d).
3. Each MH broadcasts another 'Hello' message, setting CHC field value equal to that calculated from Equation (1).
4. Recording received 'Hello' messages during two 'Hello' cycles, each MH identifies the sender with highest CHC value and thereafter considers it as its CH.

In the next 'Hello' cycle, CH_ID value will be set to elected CH's ID value. In the case of two or more MHs having the same lowest CHC value, the one with the lowest ID is 'elected' as CH. Following the aforementioned algorithm steps, clustering procedure is completed within two 'Hello' cycles.

The penalty coefficient p of Equation (1) is introduced to ensure that the algorithm provides stable cluster formations (cluster instability has been identified as the main weakness of HD and VC algorithms [3]), i.e. i.e. clusters insensitivity to hosts mobility. Clusters instability derives from frequent CH re-elections; according to the preceding algorithm description, such re-elections occur when an ordinary MH claims higher CHC value compared to current CH. For instance, when: a new MH moves within a cluster boundary or the current CH presents slightly lower power level than an ordinary MH. CH re-elections typically trigger a global cluster reconfiguration process and massive transfers of routing data among elected CHs.

To correct this inefficiency, we introduce a penalty coefficient p in the calculation of CHC value, as shown in Equation (1). The value of p is set to an integer value ($p > 0$) for ordinary MHs, while $p = 0$ for CHs. Assigning an appropriate value to p, we prevent MHs with slightly higher degree or lower battery power to that of current CHs to take up the role of CH, thereby avoiding unnecessary handovers.

3.2 Dynamically adaptive control messages broadcast period

A principal consideration of our Adaptive Broadcast Period (ABP) algorithm is to reduce the number of control messages circulated throughout the ad-hoc network. Minimization of message broadcasts would provide bandwidth savings and conserve computational resources and battery power on both CHs and ordinary nodes.

Ad-hoc networks have been proposed for many applications that do not involve highly mobile structures, e.g. in convention centers, conferences or electronic classrooms. Existing active clustering algorithms involve periodic broadcast of 'Hello' messages to sense potential topological differences between two successive 'Hello' periods. In relatively static MANET topologies though, such differences seldom occur; namely, bandwidth and power resources are unnecessarily consumed.

ABP algorithm corrects this clear inefficiency by adjusting 'Hello' messages broadcast period (BP). For highly mobile MHs, BP is shortened, i.e. message broadcasts are frequent enough to maintain consistent and accurate topology information. However, when mobility rate (MR) is low, i.e., MHs position on the plane does not considerably change over time relatively to their neighbors position, BP is lengthened, relaxing the MANET from unnecessary control message storms.

The main issue to be addressed is to accurately measure mobility pattern of MHs. In order to meet this objective, MR is measured by individual MHs through contrasting the topological information obtained during successive BPs; when MR increases, BP is shortened, otherwise it is prolonged.

In order to measure MR, CHs need to maintain vectors representing the IDs of the MHs dominated by the CH; each vector instance refers to a different 'Hello' BP. Calculated MR value is actually the 'distance' of vectors recorded during the two latest 'Hello' cycles. The latter is an integer value indicating not only the change of CH's degree but also changes in CH's network neighborhood (potential substitutions of its dominated MHs by other MHs). We assume that BP duration always lies between two limits: $BP_{\min} \leq BP \leq BP_{\max}$; at startup, BP is globally set to $BP_{\min}$. This default value changes over time reflecting different mobility patterns among separate clusters, according to Equation (2):

$$BP_{t+1} = \begin{cases} \max(BP_{\min}, BP_t - kMR_t), \text{if } MR_t > T \\ \min(BP_{\max}, BP_t + c), \text{if } MR_t = 0 \end{cases} \quad (2)$$

where k is a normalization factor, T is a threshold value, c is a constant and MR_T the mobility rate measured over the last BP . Modifications of BP values are announced by CHs to all their dominated MHs to achieve the required synchronization; this is done through setting appropriately the value of 'BP' field in 'Hello' message (see Fig. 3). Ordinary MHs set 'BP' field value equal to 0. MHs receiving a 'Hello' message from their assigned CH ('Hello' message CH_ID value coincides with their assigned CH's ID) adapt their BP duration accordingly.

An inviting side-effect of ABP algorithm is that it fits well in environments with high 'local' mobility: BP may differ among separate clusters, depending on the mobility pattern of their respective MHs. That way control traffic is localized only where needed, leaving unaffected clusters whose members exhibit low mobility.

4 Conclusions – Future Work

In this paper, we introduced a novel active clustering algorithm; its contributions, compared to existing solutions, are summarized in the following: (a) clustering procedure is completed within two ‘Hello’ cycles; (b) both location and battery power metrics are taken into account in clustering process; (c) derived cluster formations exhibit enhanced stability by preventing unnecessary CH re-elections; (d) for relatively static network topologies, control traffic volume is minimized; (f) fast packet forwarding and delivery is enabled, as clusters are pro-actively formed. The abovementioned contributions are achieved at the expense of slightly increased control packet sizes.

At the time this article was written, the proposed algorithm was under evaluation through simulations. Apart of our proposed algorithm, LID, HD and VC have also been implemented for demonstration and comparison purposes. Simulation results are evaluated to compare these algorithms in terms of: (a) overall control packet overhead, (b) CH changes, (c) average lifetime of CH and cluster membership. All these parameters are measured as a function of MHs density and mobility pattern.

Acknowledgements

The research work presented herein has been co-funded by 75% from EU and 25% from the Greek government under the framework of the Education and Initial Vocational Training II, Programme Archimedes.

References

1. Y. P. Chen, A. L. Liestman, J. Liu: Clustering Algorithms for Ad Hoc Wireless Networks. In Y. Pan and Y. Xiao (eds): Ad Hoc and Sensor Networks, Nova Science Publishers (2004).
2. X. Hong, K. Xu, M. Gerla: Scalable Routing Protocols for Mobile Ad Hoc Networks. IEEE Network, Vol. 16, No 4 (2002) 11-21.
3. Fei Li, Shile Zhang, Xin Wang, Xiangyang Xue, Hong Shen. Vote-Based Clustering Algorithm in Mobile Ad Hoc Networks. In: Proceedings of International Conference on Networking Technologies for Broadband and Mobile Networks (ICOIN’2004), LNCS vol. 3090. Springer-Verlag (2004), 13 – 23.
4. Li C. R., Gerla Mario. Adaptive Clustering for Mobile Wireless Networks. IEEE Journal of Selected Areas in Communications, Vol. 15, No 7 (1997) 1265-1275.
5. C. Perkins. Ad Hoc Networking. Addison-Wesley (2001).
6. Y. Yi, M. Gerla, T. Kwon. Efficient Flooding in Ad hoc Networks: a Comparative Performance Study. In: Proceedings of the IEEE International Conference on Communications (ICC’2003) (2003).