# Effective Management Through Prediction-Based Clustering Approach in the Next-Generation Ad Hoc Networks

S. Sivavakeesar, G. Pavlou, C. Bohoris and A. Liotta
Center for Communication Systems Research
University of Surrey
Guildford GU2 7XH, United Kingdom
{S.Sivavakeesar, G.Pavlou, C.Bohoris, A.Liotta}@eim.surrey.ac.uk

***Abstract*—A framework for a proactive network management with the eventual aim to support Quality of Service (QoS) provisioning in ad hoc networks is proposed in this paper. This process is facilitated through our novel hierarchical clustering approach. This clustering approach is dynamic and distributed, and enables each mobile node (MN) to anticipate the availability of its neighbors through a scalable intelligent mobility prediction algorithm. With the formation of stable clusters, our clustering algorithm enables adaptability, autonomy, economy, scalability and survivability requirements in managing ad hoc networks by adopting policy-based management technique with mobile agent concepts. Initial results demonstrate the stability improvement of our approach.**

***Keywords – Ad-hoc networking; Mobility prediction; Hierarchical clusteringe; Ad-hoc management; QoS provisioning.***

## I. Introduction

Mobile ad hoc networks (MANETs), consisting of a collection of wireless nodes – all of which may be mobile – dynamically create a wireless network among themselves without using any pre-existing infrastructure or administrative support [2]. Although it offers unique benefits and versatility in variety of applications and situations, there are a still a number of problems that are open to research till to date due to its unique nature. With the eminent introduction of real-time applications, quality-of-service (QoS) provisioning and management in ad hoc networks are underway and would be challenging. Any approach to solve the above tasks needs to take the problems that are unique in the MANET environment such as its more dynamic topology, energy and bandwidth-constraint operation and wireless vulnerabilities into consideration. Since it is expected that MANETs in the future may comprise a large number of mobile nodes (MNs), a hierarchical as opposed to a flat structure will scale better [9][11][12]. This clustering technique facilitates hierarchical management in MANETs, and brings in a number of benefits as stated in [1][5][6][7]. Cluster-based mechanisms in ad hoc networks not only make a large network appear smaller, but more importantly, they make a highly dynamic topology to appear less dynamic [11][12]. This feature is important here, as one mission of any network management protocol is to present the network topology of the network to the management entity in a scalable manner. The rest of the paper explains how this can be performed with our (p,d,t)-clustering approach [1]. The primary step in clustering is the election of cluster heads (CHs) and the formation of clusters around them [1][11][12][12]. It is expected that future generation wireless networks will evolve towards non-authority based, self-organized, large-scale MANETs, which will have a significant impact on future communication models and m-business. In this work we envisage large-scale deployment of long-term multihop MANETs, which is similar but complementary to mobile telephony systems. The network model considered in this work is, thus, similar to that assumed in the 'Terminodes' project [10]. We adopt a hierarchical clustering approach, which is fully distributed and dynamic in nature.

One approach that has gained considerable attention in the networking community to meet QoS provisioning and management is Policy-based network management (PBNM). To date, however, its applications have been limited mainly to fixed high-bandwidth networks, although there are a few preliminary works on adopting it into MANETs [7]. This paper describes as to how an efficient network management system for MANETs can be realized through the use of our (p,d,t)-clustering, policy-based network managements and mobile-agent technology. In this paper, our primary focus is on performance management with much emphasis on (p,d,t)-clustering [1], as opposed to other four areas of network management as defined by international standard organization (ISO) [5]. The rest of this paper is organized as follows. Section II examines related previous work, and presents our motivation. Our novel clustering technique to be used in the network-management is briefly described in section III. Section IV presents the proposed management framework. The applicability of (p,d,t)-clustering algorithm is evaluated through simulations in section V. Section VI presents our conclusions and future work.

## II. Previous Work and Our Motivations

The management architectures and protocols that are used in conventional fixed networks cannot be easily adopted in MANETs without any modification. In addition to normal network information, data related to nodes' location, velocity, mobility and available battery-power needs to be considered in

This work was undertaken in the context of the UK Engineering and Physical Sciences Research Council (EPSRC) Programmable Ad-hoc Networks (PAN - GR/S02129/01) research project.

the data-collection process, while conserving the scarce wireless bandwidth and transmission power. In MANETs, the chances for different configuration changes to occur are very high, and any management architecture should allow the management entities to detect and react to such changes in a flexible way.

There have been only a few works that consider network management in MANETs. A management protocol known as ad hoc network management protocol (ANMP) based on simple network management protocol (SNMP) was proposed in [5]. This ANMP is built on Lowest-ID based clustering approach, and hence the performance of ANMP is limited by the clustering stability provided by the latter. Moreover, in a more dynamic and resource-constrained environments like MANETs, the conventional manager-agent management paradigm will not be efficient. Reference [7] discusses as to how the policy-based approach can be extended and applied to manage ad hoc networks. This work is again built on k-hop clustering algorithm where cluster head (CH) election process uses the same mechanism as that used in [5]. As a result, clustering stability resulting from the clustering algorithm used plays a vital role in the overall management performance. A guerrilla management architecture that employs a two-tier infrastructure to facilitate adaptive management was proposed in [6]. The higher tier consists of groups of peer-to-peer nomadic managers, while the lower tier consists of active probes (programmable scripts). Although the last two works are relatively new, they lack an important aspect of proactiveness in a more dynamic environment like MANETs. Proactiveness is essential in order to have continuous management operation without any interruption.

In our approach, the proactiveness results from our novel (p,d,t)-clustering algorithm and protocol [1]. The concept of clustering in MANETs is not new, and there have been many algorithms that consider different metrics and have different purposes in mind. However, almost none of them consider node mobility as a criterion in the clustering process effectively [1][12]. As a result, they fail to guarantee a stable cluster formation. In a MANET that uses cluster-based services, network performance metrics such as throughput, delay and effective management are tightly coupled with the frequency of cluster reorganization. Therefore, stable cluster formation is essential for better management and QoS support. The next section describes how (p,d,t)-model results in stable clustering, and enables proactive management in our management architecture.

## III. (p, t, d)- MODEL

The (p, t, d)-model has the following necessary ingredients: 1) The concept of *virtual clusters*, 2) Mobility prediction model, 3) Clustering algorithm and protocol, 4) Management architecture, which are briefly explained below. Since we are eventually seeking a QoS solution and proper network management, MNs should be able to predict the availability of network resources in order to manage as well as to support QoS. This is achieved with our scalable mobility prediction scheme based on the accumulated past behavior history of a specific MN [1].

### A. *The Concept of Virtual Clusters*

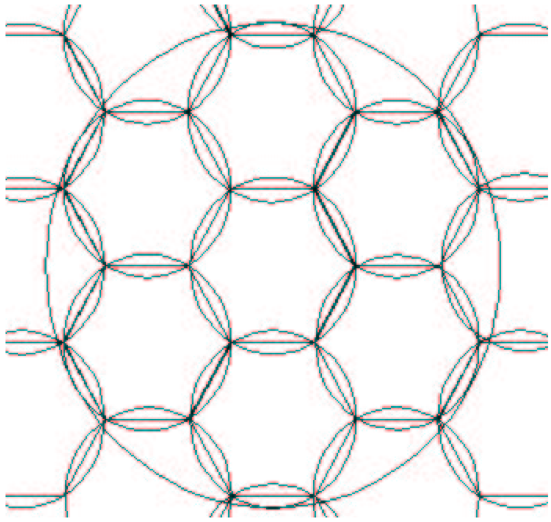

Figure 1. Concept of *Virtual Cluster* and *Tracking Zones*

We introduced the notion of *virtual clusters*, in order to make our mobility prediction viable, and our clustering mechanism scalable [1]. For correct operation of the (p,d,t)-model, each MN is supposed to have a complete picture of the locations of the centers of such *virtual clusters* (VCCs). If greater mobility prediction accuracy is required, each *virtual cluster* can be further split into a number of equal *tracking zones* (TZ) as shown in Fig. 1. These TZs are again circular in shape, and can handle the MNs' intra-*virtual cluster* movement patterns. Fig. 1 depicts a *virtual cluster* (big circle) that consists of seven TZs (smaller circular region). A *virtual cluster* can contain any number (N) of such TZs depending on factors such as the mobility prediction accuracy required and maximum control overhead that is allowable. However, 'N' should satisfy the following: $N = i^2 + j^2 + i*j$, where 'i' and 'j' are integers. Again each TZ has its own unique *tracking zone* identifier (TZI), which can be determined given the location information. Similarly, each MN is supposed to know all the *tracking zones*, and their corresponding identifiers within a particular *virtual cluster*. However, in our initial work we consider only the inter-*virtual cluster* movement pattern of MNs, and greater accuracy in this case is obtained by optimizing the value of virtual cluster radius (R). Micro mobility pattern (intra-*virtual cluster* movement) will be considered in our future work.

### B. *Mobility Prediction Model*

As node mobility is the main cause of uncertainty in MANETs, we developed a mobility-prediction model based on the Ziv-Lempel algorithms for data compression [1][3][11]. This prediction model derives a probabilistic prediction of user mobility based on the accumulated behavior history of a specific MN. It is also ensured that the prediction is made without too much complexity and waste of bandwidth and transmission power. In our approach, each MN is responsible for generating the strings of *virtual cluster* identifiers (VIDs) and maintaining its respective dictionary in its memory [1]. In addition to making predictions as to future movements of a particular MN, our model is used by each MN to predict its approximate residence-times of the *virtual clusters* it visits. For this purpose, each MN maintains its mobility database at a specific time in terms of a *Mobility Trie* [1][4]. This trie is a probabilistic model corresponding to the dictionary of the LZ78 algorithm. Each leaf except the root in the trie preserves the relevant statistics that can be used to predict the probability of following events.

The time-interval ($T_e$) at which update events are triggered based on the time-based updating, and the radius (R) of *virtual cluster* are two important parameters, and thus determines the accuracy of prediction, and hence the performance of our clustering algorithm. The shorter the '$T_e$', greater the accuracy of the residence-time, but higher the tracking overhead would be. Similarly, the smaller the 'R', the better the prediction capability, but the higher the tracking overhead. Therefore, a compromise decision is necessary, when selecting values for these two parameters. As mentioned in section III.A, if greater prediction accuracy is required, *Mobility Tries* can be constructed with respect to *tracking zones*. This attempt is to minimize the prediction faults that might occur if a MN has an extensive micro (intra-*virtual cluster*) mobility patterns. Accordingly, residence-time of a MN in a TZ can be used in the cluster head election process of a particular *virtual cluster* as described in section III.C. Again in this approach, the higher the number of TZs that a *virtual cluster* consists of, the greater the accuracy of the residence-time, but higher the tracking overhead would be.

### *C. Clustering Algorithm and Protocol*

The unique functionalities of our novel clustering algorithm and protocol are mainly due to our mobility prediction model as explained in [1]. The *Mobility Trie* that each MN constructs plays an important role to bring in proactiveness in our (p,d,t)-clustering approach. This is possible by enabling each MN 'X' to determine its residence-time ($t_{xk}$) in *virtual cluster* 'k' probabilistically from each leaf (or set of leaves) of its *Mobility Trie*, if the movement history context is known. For this purpose, each MN should maintain its past history in terms of a *Mobility Trie*. If however, a MN is unable to construct its trie, it can still use its distance in the criterion calculation, since $t_{xk} = 0$ (see equation (2) of [1]) in this case. The MN that has the highest $\Omega$ can become the primary CH, and the MNs that have the second and third highest values become assistant (secondary) CHs. In forming clusters, the CH has to make sure it can cover the whole area of the *virtual cluster*. Therefore, the CH makes a k-hop cluster where value 'k' is not necessarily uniform within the cluster in terms of distance between any border MN and the CH. However, efforts have been made to limit the value of 'k', as it is better if every MN is only a fewer hops (maximum of 2) away from its respective CHs for proper management.

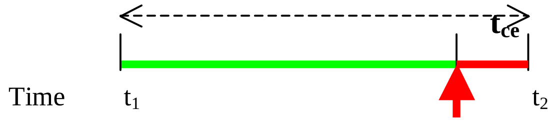


Figure 2. The Changeover Event Phenomenon.

Four different packet types such as *JOIN*, *HELLO_CH*, *HELLO_MN* and *SUCCESSOR* have been defined for the operation of our clustering protocol. Each cluster head broadcasts a *HELLO_CH* packet periodically – every CH_HELLO_INTERVAL – within its *virtual cluster* and each cluster member unicasts *HELLO_MN* to its respective CH periodically – every MN_HELLO_INTERVAL. On the other hand, if a new MN has not become a member in any cluster, it has to unicast a *JOIN* packet to its respective CHs. Each MN is supposed to calculate and include its present $\Omega$-value (see equation (2) of [1]) its approximate residence-time ($t_{xk}$) within its current *virtual cluster* 'k' (by calculating it from its own *Mobility Trie*), its location information, and its *Mobility Trie* corresponding to the next '$T_{mt}$' minutes in the protocol packet it transmits. The system parameter '$T_{mt}$' should take an appropriate optimal value. Since all mobile nodes except CHs use unicating as opposed to broadcasting, our clustering algorithm conserves the scarce bandwidth and transmission power. In addition to other information regarding each *virtual cluster*, each periodic *HELLO_CH* carries the neighbor-table. Neighbor-table is a set of MNs in the cluster and their latest $\Omega$-values. From the neighbor-table of *HELLO_CH*, each MN that resides within the same *virtual cluster* constructs its own neighbor-table, and hence becomes aware of its neighbors. If, however, any member node has not received a *HELLO_CH* packet from any CH during the last three consecutive CH_HELLO_INTERVAL periods, each node of a particular *virtual cluster* has to broadcast (instead of unicasting) its protocol packet in order for other neighbor nodes to know about its existence. This enables the MNs within the *virtual cluster* to elect one as their CH in a distributed manner. These control packets are relayed by intermediate MNs only within the *virtual cluster*. On the other hand, periodic *HELLO_CH* packets by CHs are unicast by gateways between CHs of adjacent *virtual clusters* to an extent that can be limited for scalability. This is to enable CHs to get the topology information of adjacent clusters. Given that each CH knows the predicted residence-time of each MN within its cluster, it deletes the entry associated with a particular MN exactly '$t_{to}$' (system parameter) seconds after its residence-time expires. This effect will be reflected in every *HELLO_CH* packet a CH broadcasts periodically. Also after having become a member of a cluster, each MN can dynamically increase its MN_HELLO_INTERVAL until the new CH election process is triggered, given that it knows the predicted residence-time of the current CH. This is economical with respect to both bandwidth and transmission power.

Another unique aspect of our protocol is that, before a particular CH becomes unavailable, it has to determine its successor and inform its members using the SUCCESSOR packet after triggering the "*CH changeover event*". This happens exactly '$t_{ce}$' seconds (see Fig. 2) before the time ($t_2$) at which the present CH has been predicted to leave the serving *virtual cluster*. According to Fig. 2, the present $CH_i$ can serve the present cluster for the maximum time duration of $t_2 - t_1$, unless $CH_i$ fails abruptly. This time period is actually its residence-time within a specific *virtual cluster* and is determined from its own *Mobility Trie*. For this purpose, the present CH consults its neighbor-table, and selects the primary and secondary CHs based on their **Ω**-values.

### *D. Management Architecture*

The clustering protocol is used to simplify the task of proper management in MANETs. By definition, "network management is a process of controlling a complex data network in order to maximize its efficiency and productivity" [5]. This requires that a network management protocol present the topology of the network to the network manager. This process generally involves data collection, data processing, data analysis, and problem fixing. However, these management related tasks have to be performed in an efficient and scalable way. To facilitate this, any management architecture should enable adaptability, autonomy, economy, scalability and survivability requirements in managing ad hoc networks [6].

Based upon the information collection and communication strategy, there are three types of network management architectures: centralized, distributed, and hierarchical. In MANETs, a centralized architecture would suffer from a high message overhead in data collection. Although the other two architectures are suitable for MANET, hierarchical is preferred in order to employ policy-based architectures. In order to have a protocol that is message efficient, a hierarchical model for data collection is appropriate, since intermediate levels of the hierarchy can collate data (possible producing a digest) before forwarding it to upper layers of the hierarchy. However, there would be cost of maintaining a hierarchy in the face of node mobility. A good trade-off is to use a three-level hierarchical architecture as depicted in Fig. 3. Our (p,t,d)-clustering approach facilitates this hierarchical management architecture that makes use of policy-based management technique together with mobile agents [5][6][7]. The lowest level of this architecture consists of individual managed MNs. Several MNs are grouped into clusters and managed by a cluster head. The cluster heads in turn are managed by the network manger.

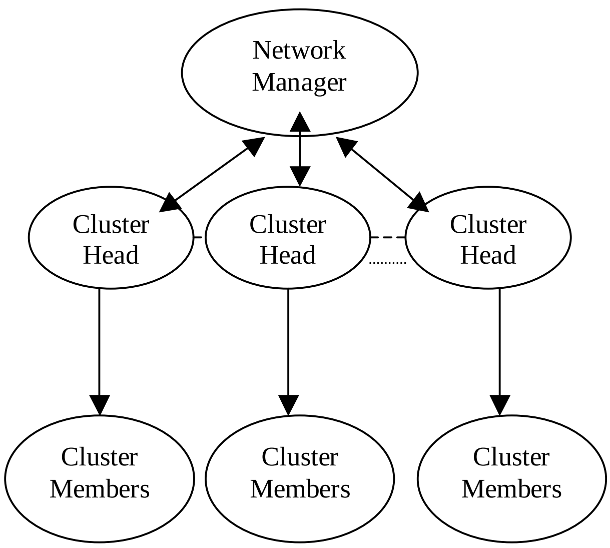


Figure 3. Hierarchical Management Architecture

The cluster thus formed should take an optimal size, and should be stable most of the time in order to enable the efficient data collection. The network manager, serving as a top-level manager, regulates and distributes management policies to a group of cluster heads that collaboratively carry out management operations [6]. These policies would include high-level management activities inside the ad hoc network (of a particular region), such as QoS parameters and management rules. As expected in hierarchical management, each cluster head provides functionality that is transparent to its network manager, and hence cluster heads are autonomous in this respect. In our approach, it is assumed that all nodes possess equal management intelligence (management software modules and states), and any node can become either a managing (cluster head) or a managed node (cluster member) depending on a situation as described by our clustering mechanism. The cluster heads collaborate autonomously to manage the entire ad hoc network with minimal help from external entities and the network manager. When the cluster head changeover occurs the management intelligence and data should be able to migrate or spawn itself to the new cluster head. Adaptive software, which integrates decision theory and intelligent agent techniques, dictates such management behavior of cluster heads. The reason why mobile agent technology is used here is that it can spawn (dynamically deploy) itself in an appropriate node and execute managerial functions locally. This aspect is desirable in more dynamic environments like an ad hoc network, and more importantly such technique could relieve the cluster heads by freeing the latter from periodically polling other managed nodes – if the conventional manager-agent paradigm were used. This again conserves scarce resources in ad hoc networks.

As a bootstrapping process, these cluster heads are autonomously elected through our (p,t,d)-clustering algorithm. In our clustering algorithm, since each cluster head knows the predicted lifetime of each of its members, the cluster heads and the mobile code that they originate can perform managerial operations in a proactive manner with minimal management interruption. Moreover, using *Mobility Tries*, the cluster head would know the predicted mobility patterns of its member nodes. In case a node moves from one *virtual cluster* to another, its respective cluster head knows this from the *Mobility Trie* of the former. This fact enables the corresponding cluster head to perform hand off process proactively by collaborating with the cluster head of the *virtual cluster* that the node is going to visit. Unlike in other cluster algorithms available in the literature, our (p,t,d)-algorithm enables proactive cluster head election process, and hence leads to continuous management operation without any interruption.

The PBNM ensures that a node that enjoys a certain QoS within a particular *virtual cluster* to get the same guarantee in a different *virtual cluster* [7]. In our three-level hierarchy, the network manager is responsible for the PBNM operations, and can function as a policy server or policy decision point [7]. The cluster head can function as either policy decision point or policy enforcement point. This PBNM can facilitate the management of QoS in MANET by supporting dynamic admission control, and bandwidth allocation based on factors namely the bandwidth availability, owner of the traffic and time of day. Unlike the scheme proposed in [7], mobile agent technology is used again in our approach for policy distribution and provisioning, and even for policy monitoring. To dictate an effective use of network resources, the PBNM should be aware of the available network resources. In MANET, network resources are mainly provided by collaborating nodes. Our (p,t,d)-clustering model facilitates each node to know about the availability pattern of neighboring nodes, and thus enables the PBNM to be aware of the possible network resources through prediction. It is here assumed that almost all nodes of the MANET contain the high-level policies at the time of initial deployment.

The network manager is elected exactly the same way as the cluster head is elected. Network manager is responsible for a particular region, and might contain a number of *virtual clusters* under it. In the network manager election process, only the cluster heads take part in a distributive manner, and the cluster head 'X' that has the maximum Ω-value (equation (2) of [1]) can become the network manager of that particular region.

## IV. INITIAL EVALUATION THROUGH SIMULATION

Our initial simulation work attempts to compare the performance of our clustering algorithm with the Lowest-ID, maximum-connectivity (Max-Connect) and LDV clustering algorithms, in terms of the stability of clusters formed [1][12]. We performed our simulations using the GloMoSim simulation package in which we implemented and compared the Lowest-ID, Max-Connect, LDV and our algorithm [8]. The cluster (in)stability can be measured by determining the number of times each MN either attempts to become a CH or gives up its role as a CH. In Fig. 4 it is measured by determining the number of MNs that have attempted to become the CH at least once in 100s time period (low values are better in stability terms). As can be seen, in all the other three algorithms the numbers tend to increase linearly with the node degree. On the other hand, our model has only a few numbers of MNs that have attempted to become the CH, and hence improves stability.

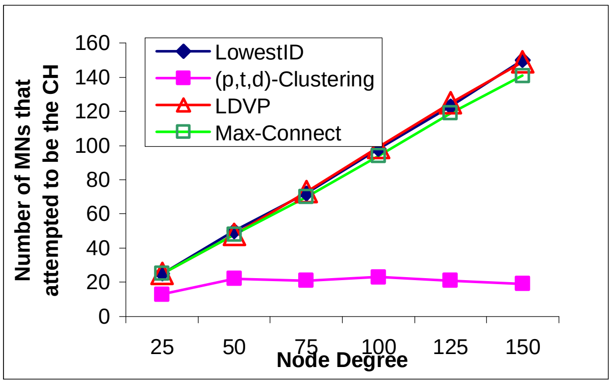


Figure 4. Number of MNs that Attempted to become the CH as a function of Node Degree

## V. CONCLUSIONS & FUTURE WORK

A new clustering approach that makes use of intelligent mobility prediction and location information in new long-term MANETs was proposed in this paper. This clustering approach is to enable the development of an automated, intelligent, efficient and robust management architecture. To facilitate this, we introduced the *virtual cluster* concept. This way of associating dynamic clusters to geographic locations results in many benefits as described in [1]. In addition, in the network management perspective, it results in the following unique benefit: We could predict a specific MN's future positions and continue managerial operations without interruptions in a proactive way. In our future work we will perform a complete evaluation of our proposed management architecture. We will also consider QoS routing, where the construction of longevity routes with sufficient resources is necessary. Work on QoS routing and resource reservation mechanisms will be built on this clustering scheme. We plan to report such findings in future papers.

**Acknowledgement:** This work was undertaken in the context of the UK Engineering and Physical Sciences Research Council (EPSRC) Programmable Ad-hoc Networks (PAN - GR/S02129/01) research project.

## REFERENCES

[1] S.Sivavakeesar, G.Pavlou, and A.Liotta, "Stable Clustering Through Mobility Prediction for Large-Scale Multihop Intelligent Ad Hoc Networks|", IEEE WCNC 2004, in press.

[2] S.Chakrabarthi, and A.Mishra, "QoS Issues in Ad Hoc Wireless Networks", IEEE Communications Magazine, Feb. 2001, pp 142 –148.

[3] A.Bhattacharya, and S.K.Das, "LeZi-Update : An Information-Theoretic Approach to Track Mobile Users in PCS Networks", 5th Annual ACM Int'l Conference on Mobile Computing and Networking (MobiCom'99), August 1999 , pp. 1 – 12.

[4] F.Yu, and V.C.M.Leung, "Mobility-Based Predictive Call Admission Control and Bandwidth Reservation in Wireless Cellular Networks", IEEE INFOCOM, 2001, pp. 518 – 526.

[5] W.Chen, N.Jain, and S.Singh, "ANMP: Ad Hoc Network Management Protocol", IEEE Journal on Selected Areas in Communications, vol. 17, no. 8, August 1999, pp. 1506 – 1530.

[6] C-C.Shen, C.Srisathapornphat, and C.Jaikaeo, "An Adaptive Management Architecture for Ad Hoc Networks", IEEE Communications Magazine, Feb. 2003, pp. 108 – 115.

[7] K.S.Phanse, L.A.DaSilva, and S.F.Midkiff, "Extending Policy-Based Management to Ad Hoc Networks", Technical Report, Brandley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Alexandia, USA.

[8] X.Zengu, R.Bagrodia, and M.Gerla, "GloMoSim: A Library for Parallel Simulations of Large-scale Wireless Networks", Proceedings of the 12th Workshop on Parallel and Distributed Simulations, May 1998.

[9] S.H.Lee and D-H.Cho, "A new adaptive routing scheme based on the traffic characteristics in mobile ad hoc networks", IEEE Vehicular Technology Conference, vol. 6, 2000, pp. 2911 – 2915 .

[10] L.Blazevic, L.Buttyan, S.Capkun, S.Giordono, J-P.Hubaux and J-Y.Boudec, "Self-Organisation in Mobile Ad Hoc Networks: The Approach to Terminodes", IEEE Communications Magazine, June 2001, pp 166 –173.

[11] B.McDonald, and F.Znati, "A Mobility-Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks", IEEE Journal on Selected Areas in Communications, vol. 17, August 1999, pp 1466 –1487.

[12] M.Jiang, J.Li, and Y.C.Tay, "Cluster Based Routing Protocol (CBRP) Function Specifications", IETF Draft, August 1999.