Research Article

Exploiting Mobile Ad Hoc Networking and Knowledge Generation to Achieve Ambient Intelligence

Anna Lekova

Institute of System Engineering and Robotics, Bulgarian Academy of Sciences, Acad. G. Bonchev Street, Block 2, 1113 Sofia, Bulgaria

Correspondence should be addressed to Anna Lekova, alekova@icsr.bas.bg

Received 10 October 2011; Accepted 27 December 2011

Academic Editor: Tzung P. Hong

Copyright © 2012 Anna Lekova. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ambient Intelligence (AmI) joins together the fields of ubiquitous computing and communications, context awareness, and intelligent user interfaces. Energy, fault-tolerance, and mobility are newly added dimensions of AmI. Within the context of AmI the concept of mobile ad hoc networks (MANETs) for “anytime and anywhere” is likely to play larger roles in the future in which people are surrounded and supported by small context-aware, cooperative, and nonobtrusive devices that will aid our everyday life. The connection between knowledge generation and communication ad hoc networking is symbiotic—knowledge generation utilizes ad hoc networking to perform their communication needs, and MANETs will utilize the knowledge generation to enhance their network services. The contribution of the present study is a distributed evolving fuzzy modeling framework (EFMF) to observe and categorize relationships and activities in the user and application level and based on that social context to take intelligent decisions about MANETs service management. EFMF employs unsupervised online one-pass fuzzy clustering method to recognize nodes’ mobility context from social scenario traces and ubiquitously learn “friends” and “strangers” indirectly and anonymously.

1. Introduction

Ambient intelligence (AmI) is emerging as a new research discipline joining the fields of ubiquitous computing and communications, context-awareness, and intelligent user interfaces. The paradigm is also known as “pervasive computing”, “things that think”, “ubiquitous computing”, and so forth. Energy, fault-tolerance, and mobility are newly added dimensions of the AmI [1]. AmI places people and social contexts at the centre, while the information and communication technologies as well as network context go to the background. The new AmI paradigm is made possible by the convergence of low-cost sensors, embedded processors, and wireless ad hoc networks in new generation industrial digital products and services. Mobile ad hoc networks (MANETs) are multihop wireless networks without fixed infrastructure, formed by mobile nodes. The connection between knowledge generation and mobile ad hoc networks will be symbiotic—knowledge generation will utilize the wireless ad hoc networking to perform their communication needs, and MANETs will utilize knowledge generation to enhance their network services. Current mobile devices, which go together with us anywhere and at anytime, are the most convenient tools to help us in ubiquitous computing, that is, to intermediate between us and our surroundings in an unobtrusive fashion. In other terms, the data processing and communication go to the background and must adjust to the user’s personality. Some of the decisions related to communication have to be completely made at run time by learning the users’ mobility patterns based on personal actions, roles, and social networks. The routing services need local neighbor view over time (who are friends and who are strangers) and which friends can be a source of support to convey messages. The presented routing heuristic takes the idea of today social networks. Our society is divided into groups of interest, and in groups of moving people and vehicles, entities with similar mobility patterns tend to cluster together and form support groups with certain mobility and social characteristics. Members belonging to one group interact in several ways with members within that group, and the mobility patterns of specific groups are often predictable [2]. Moreover, the social factor impacts the effectiveness of a particular MANET, that is, whether users will allow their phones to participate in this kind of
networks. We assume that users from specific social networks are supposed to accept that power for the batteries of their own phones will be drawn, and their phones will convey messages between third parties without any benefit for them.

MANETs concept for “anytime and anywhere” is supposed to support network services independent on the application scenarios. Context is any information that can be used to characterize the network environment. Context awareness is the ability of the system to infer contextual knowledge in order to recognize nodes’ situations. Dynamism in MANETs and major concerns, such as unpredictable bandwidth and topology changes, make questionable the usage of analytical modeling. At the same time, data-driven artificial intelligence (AI) approaches are capable of modeling nonlinear relationships, adapting and managing uncertainties by learning from empirical data without human participation. By “learning in MANETs”, the researchers most often understand how a node constantly updates its view of the local network and, given the same inputs, the node may respond differently later than it did earlier. Usually, learning mechanisms use interaction with the neighboring nodes to extract information about the local topology and broadcast learning parameters in data packets or in control packets. Much of the focus of machine learning in MANETs has been on supervised learning (SL) and reinforcement learning (RL). In both approaches, the learner receives feedback from the environment about the appropriateness of its response as an output or reward. However, active broadcast of learning parameters or feedbacks could easily congest the network and degrade throughput when network topology changes frequently. SL main drawback is the need to know exactly what the learner should do a priori, which cannot meet MANETs unpredictable topology changes. RL main drawbacks are assumptions for finite set of state actions in learner’s environment and its slow convergence speed. At the same time, unsupervised learning (UL) is based on the similarities and differences among data without a feedback from the environment. UL represents the input data in a more efficient way and the system context is the outcome of some cognitive process, for example, data mining. Therefore, we apply an unsupervised approach to learn context from data in a passive (nonintrusive) mode without a priori knowledge and focus our study on context-awareness for routing services. Thus, embedded intelligence in MANETs is supposed to support network services independently on application scenarios by managing uncertainties and inferring contextual knowledge by learning from empirical data without human participation. The contribution of the present study is ubiquitous computing by observing and analyzing fluctuations in local mobility context. The heuristic “care of myself by leaning on friends for support” is applied in a local scope (direct communication in one hop) for packets transmission, while the heuristic “the friends of my friends are my friends too” —in a global scope.

2. Problem Definition and Our Solutions

In this section, we present the framework for evolving fuzzy modeling, as well as what type of data mining technique we use for knowledge discovery. We introduce in brief a metric using the concept of binary-coded trend in fluctuations for context that we later use to analyze the network connections and service reliability.

In the present work, we studied multicast routing services for group-oriented application in disaster area (DA) scenario. The maneuver, took place in May 2005 in Cologne, Germany, has been simulated [3]. Communication is a challenge for all MANETs protocols, since nodes have heterogeneous area-based movements, pass by obstacles, and join/leave subareas. For instance, in the disaster area (DA) scenario, people or vehicles behavior changes over time and “neighbor relations” are not random, since particular group of rescue agents reacts in a similar way. Events act as motivations for mobile nodes to react and move according to the high-level context of their roles. The average density of nodes increases in response to the events due to gathering around the “incident location”. Roles aid the learning of mobility, node density, and traffic patterns. For instance, ambulance teams are immobile when rescue patients in the casualties’ treatment areas and oscillate for a certain time before reaching the next treatment area. Police and firefighters approach the “incident location”, while patients flee it. Neighbors with similar roles have similar mobility and are supposed to remain connected for a certain time. Therefore, we can lean on these nodes for support in local communication (in vicinity) for information propagation. As a consequence, if a node logs scenario traces in vicinity over time and categorize them using a relevant clustering algorithm it might learn its behavior dynamically and its local network view over time. “Friends” are all nodes that show up similar mobility behavior. To comply to MANETs concerns, clustering should be a passive, fast, and one-pass algorithm.

First, we observed and analyzed MANETs DA scenario traces in structural, special, and temporal characteristics and according to metrics for context, described below, we further analyze the network connections and service reliability. We concluded that (1) the nodes in a local talk group are positioned inside one area and throughput is high since the number of hops, the packets are transmitted, is small (a lot of one-hop connections); (2) higher density of nodes in DA subareas [3] near the “incident location” and entry/exit points might cause packet collisions and calls for reducing the flood of control packets; (3) reaction to events is repetitive over time. It is our belief that if we utilize hidden knowledge and dependencies from high-level context of DA scenario, we will increase the intelligence of service protocols’ algorithms. Thus, we tailor the evolving fuzzy modeling,
proposed by Lekova in [4], to enhance low-level services in mobile ad hoc networking by observing and categorizing social activities in user and application level.

Key factors for reliable performance of services in MANETs are protocols’ adaptation to mobility and traffic patterns. The problem complexity and the lack of knowledge about the functional dependence of the movement patterns and node density over time to future mobility and density impose the necessity to perform clustering analysis to extract knowledge from data and search a model determined by IF-THEN fuzzy rule base. Fuzzy logic [5] is applied because there are uncertainties associated with estimating node mobility and link crash, as well as handling missing data as result of missing data observation when nodes enter and exit subareas. Fuzzy clustering is a well-known technique for unsupervised learning when data are not labeled and clusters we are searching for are not well defined and possess smooth boundaries. Moreover, the fuzzy mathematical model allows extending control factors into consideration to accomplish multicriteria evaluation. As a consequence, we designed unsupervised evolving fuzzy method in order to categorize node context and to adapt services to the current scenario for improving the network performance in nonintrusive way.

We exploit the concept of binary-coded trend in fluctuations for mobility (BTFM) that is applied to manage the routing services. BTFM describes the dynamic nature of movement context. By exchanging it among neighbors, the mobility in a local scope could be analyzed. We studied ordered sequence of scenario traces in consecutive data periods, since we are interested in the trace dynamism. One of the reasons for not using any statistical metrics is that all samples in n consecutive time steps that show equal mean and standard deviation have different trend in fluctuations. Additional benefit of using BTFM is that it is computationally less intensive than the standard deviation. Thus, we use base-2 number system to code with “1” the state of moving and with “0”—immobility and obtain a digit in a binary format. BTFM decimal equivalent is denoted as decimal-coded trend in mobility (DTFM). It facilitates BTFM broadcasting and categorizing of movement context. For instance, if after eight steps the BTFM for an “ambulance agent” is 11011000, the corresponding DTFM is 216. All nodes in vicinity with DTFM = 216 are indexed as friends, other—as strangers. The key idea of the proposed by us heuristics “lean on my friends for support” is to enhance destination-initiated mesh-based multicast protocol, such as ODMRP, nodes that will stay connected in n consecutive time steps. These nodes show similar mobility patterns as the destination nodes in the next n-hop scope. The value of DTFM is carried and broadcasted by the join query message in the ODMRP protocol. After converting the DTFM into binary-number system, bitwise operation “AND” for BTFM of the node with BTFM of the destination node in the join query is performed to decide which node is a “friend” to forward data packets.

3. Overview of Evolving Fuzzy Modeling Framework

In this section, the evolving fuzzy modeling framework and LEFCM algorithm are described and explained by an example in 2D space.

The fuzzy modeling task demands in an adaptive, lifetime, and distributed way to partition the input data space in order to identify the rules and determine of the model parameters according to a fuzzy rule of inference. Evolving fuzzy clustering and reasoning on each node are performed to ensure a system to evolve its structure and functionality gradually from a continuous input data stream. Evolving clustering approaches do not need the number of clusters to be prespecified. Their algorithms are one-pass than batch-mode, fast, and do not keep any information of passed examples; therefore, they are applicable for mobile applications without wasting resources during the run of MANETs. LEFCM is a distance-based and computationally less intensive algorithm. The data space to be clustered consists of DTFM and mean of the actual values of deviation in coordinates for n consecutive time steps, which each mobile node extracts from its movement traces (relative coordinates). After quantifying DTFM, the trend in fluctuations for mobility over time is derived to predict the upcoming reliability and according to relevant heuristic specific to the implemented routing protocol, the routing services are managed.

Figure 1 represents the corresponding clustering partition for a node located near the “incident site”. Data are given by dots and cluster centers—by stars. The repetitive categories of node movement patterns have three feature values—the mean and the trend in a node movement for the past n steps, and the trend in a node movement for the next n steps. The model for prediction is determined by fuzzy rules that are given by representative values of the cluster-centered coordinates. On each node, the membership degree of ith cluster for the input data vector k is matched online, where k is an index related to the time instant. The crisp output values for mobility—O^k+\tau is returned by fuzzy reasoning model. The specifics of the used metric, that is, conversion of numbers in base-2 arithmetic into base-10, and the difference of how the numbers in these two numeric systems are comparable to each other, causes a big value for the mean threshold [4], such as 0.8, to be assigned. This results in a model with more rules but more correct estimation.

3.1. Overview of the Lightweight Evolving Fuzzy Clustering Method

In the online clustering process, the given data set consists of input vectors (examples) X = {x_1, …, x_p} which are p points in q-dimensional space, x ∈ R^q. Examples come from a data stream one by one, and the algorithm starts with an empty set of clusters. When a new cluster C_k is created (k is an index related to the time instant), the current input vector is assigned to be a cluster center (C_c) and its cluster radius (R_k) is initially set to zero. Then, this cluster is updated or new cluster is created depending on a threshold value (D_{thr}) that affects the number of clusters. The maximum distance from any cluster center to the examples that belong to this cluster is not greater than the threshold
the number of clusters online means online optimization of and increasing their radii or creating new clusters. Estimating updating existing clusters changing their centers’ positions
do nothing; cluster $C_i$ with center $C_{c j}$ and radius $R_{u i}$ = 0; $x_j$ update cluster $C_{c j} \rightarrow C_{c j}'$ with center $C_{c j}$; $x_j$; create a new cluster $C_{c j}$; $x_4$ and $x_7$—do nothing; $x_5$ and $x_6$ update cluster $C_{c j} \rightarrow C_{c j}''$; $x_0$ update cluster $C_{c j} \rightarrow C_{c j}'''$ with center $C_{c j}$.

Figure 2: A brief clustering process using LEFCM with input vectors $x_1$ to $x_6$ in a 2D space. $x_1$ causes the LEFCM to create a new cluster $C_{c 1}$ with center $C_{c 1}$ and radius $R_{u 1}$ = 0; $x_2$ update cluster $C_{c 1} \rightarrow C_{c 1}'$ with center $C_{c 1}$; $x_3$; create a new cluster $C_{c 1}$; $x_4$ and $x_7$—do nothing; $x_5$ and $x_6$ update cluster $C_{c 1} \rightarrow C_{c 1}''$; $x_0$ update cluster $C_{c 1} \rightarrow C_{c 1}'''$ with center $C_{c 1}$.

value $D_{thr}$. Therefore, $D_{thr}$ affects the number of clusters by updating existing clusters changing their centers’ positions and increasing their radii or creating new clusters. Estimating the number of clusters online means online optimization of threshold value $D_{thr}$. We incorporate the fuzzy membership degrees and distance to reference center for each cluster into the clustering process (solid squares denote reference centers for the two clusters in Figure 2). Input vector is bound to each cluster by means of a membership degree (Md), which is a number between 0 and 1, and makes clustering more accurate in case of overlapping clusters. A novel multidimensional membership function equation (1) is proposed for obtaining the degree of membership between data and centers of clusters:

$$\text{Md}_{ij} = 1 - \left( \frac{||x_i - C_{c j}||}{\sum_{j=1}^{m} ||x_i - C_{c j}||} \right)^{(2/m) - 1}, \quad x \in R^q, \quad (1)$$

where $||x_i - C_{c j}|| = 1$, when $||x_i - C_{c j}|| > 1$, $m \in [1, \infty)$ is the weighted exponent coefficient which determines how much clusters may overlap. We set $m$ to be equal to 2, since in the fuzzy clustering literature, a value of $m = 2$ is the commonly used value.

A reference center (the arithmetic mean) $\bar{M_{c j}}$ for each cluster $j$, described by its center coordinates in $q$ space $\bar{M_{c j}}$.

$$\bar{M_{c j}} = \frac{\sum_{i=1}^{c p} x_{i j}}{c p}, \quad (2)$$

where $x \in R^q$ and $c p$ is the number of the input data in cluster $j$.

When the current input vector $x_i$ comes, the distances between this new input and mean center for each cluster $D_{MC_{ij}} = ||x_i - \bar{M_{c j}}||$, $i = 1, \ldots, p_i$ is calculated. We introduce two new thresholds: (1) Md threshold ($D_{thr}$) in the range [0.4–0.6]; (2) mean threshold ($M_{thr}$) in the range [0.5–0.7]. We establish that for $D_{thr} = 0.3$ and $M_{thr} = 0.6$ we obtain optimal (or at least feasible) cluster centers and radii. These numbers are used in Figure 2. When $D_{MC_{ij}}$ is less or equal to $M_{thr}$, we can increase the radius of this cluster, otherwise we make new cluster. In other words, a cluster will not be updated any more if $D_{MC_{ij}}$ is bigger than the acceptable tolerance to the mean center. When data are not well partitioned in the input space, that is, sparse data exist, the $M_{thr}$ should be bigger. The increasing of $D_{thr}$ to 0.6 or decreasing $M_{thr}$ to 0.5 results in more clusters.

The LEFCM algorithm is described as follows.

Step 1. Set up a small value for $D_{thr}$, that is, 0.1. Normalize the input vector in [0–1] using (3):

$$x_{i,norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}}, \quad (3)$$

where the range of $x_i$ is $[x_{min} - x_{max}]$ and $x \in R^q$.

Create the first cluster $C_{c 1}$ by taking the position of the first example from the input stream as the first cluster center $C_{c 1}$, then setting a value for its cluster radius $R_{u 1}$ = 0. Add the value of $x_1$ to $\bar{M_{c 1}}$ and set up the number for inputs in this cluster to 1.

Step 2. If all examples of the data stream have been processed, the algorithm is finished. Else, the current input example, $x_i$, is taken and the distances between this example and all of the $n$ already created cluster centers $C_{c j}$, $D_{ij} = ||x_i - C_{c j}||$, $j = 1, \ldots, n$, are calculated.

Step 3. If there is any distance value, $D_{ij} = ||x_i - C_{c j}||$ equal to, or less than, at least one of the radii, $R_{u j}$, it means that the current example belongs to a cluster $C_{c m}$ with the minimum distance $D_{m} = ||x_i - C_{c m}|| = \min(||x_i - C_{c j}||)$ subjected to the constraint $D_{ij} < R_{u j}, j = 1, \ldots, n$. 

Applied Computational Intelligence and Soft Computing
Add the value of \( x_i \) to \( \text{MC}_j \) and increase the number of inputs in cluster \( j \) by one.

In this case, neither a new cluster is created nor any existing cluster is updated, and the algorithm returns to Step 2.

**Step 4.** Else, find cluster \( C^k_a \) (with center \( C_{C^k_a} \) and cluster radius \( R_{U_{C^k_a}} \)) from all existing cluster centers through calculating the values \( S^k_{ij} = D_{ij} + R_{U_{C^k_a}} \), \( j = 1, \ldots, n \), and then choosing the cluster center \( C_{C^k_a} \) with the minimum value \( S^k_{ij} \):

\[
S^k_{ij} = D_{ij} + R_{U_{C^k_a}} = \min \left( S^k_{ij} \right), \quad j = 1, \ldots, n. \tag{4}
\]

**Step 5.** Calculate Md according to (1).

**Step 6.** Calculate the distances between the new input and mean center for each cluster—\( \text{MC}_j \), \( \text{DMC}_{ij} = ||x_i - \text{MC}_j|| \), \( i = 1, \ldots, p \).

**Step 7.** If \( \text{Md} \) is greater than \( \text{Mthr} \) and \( \text{DMC}_{ij} \leq \text{Mthr} \), then \( D_{\text{thr}} = S_{\text{md}}/2 \). This means that the input example is very close to that cluster and its mean center and \( D_{\text{thr}} \) is set up to produce cluster updating instead of new cluster creation.

**Step 8.** If \( S_{\text{md}} \) is greater than \( 2 \times D_{\text{thr}} \), the example does not belong to any existing clusters. A new cluster is created in the same way as described in Step 1, and the algorithm returns to Step 2.

**Step 9.** If \( S^k_{ij} \) is not greater than \( 2 \times D_{\text{thr}} \), the cluster \( C^k_a \) is updated by moving its center, \( C_{C^k_a} \), and increasing the value of its radius, \( R_{U_{C^k_a}} \). The updated new radius \( R_{U_{C^k_a}}^{k+1} \) is set to be equal to \( S_{\text{md}}/2 \), and the new center is located at the point on the line connecting the \( x_i \) and \( C_{C^k_a} \) and the distance from the new center \( C_{C^k_a}^{k+1} \) to the point \( x_i \) is equal to \( R_{U_{C^k_a}}^{k+1} \). Add the value of \( x_i \) to \( \text{MC}_j \) and increment the number of inputs in cluster \( j \). The algorithm returns to Step 2.

Let us explain the clustering process by a 2D example shown on Figure 2. The input vectors \( x_1 \) causes the LEFCM to create a new cluster \( C_{C^1} \) with center \( C_{C^1} \) and radius \( R_{U_{C^1}} = 0 \) according to Step 1. Correspondingly, \( x_2 \) updates cluster \( C_{C^1} \) with center \( C_{C^1} \). \( x_3 \) creates a new cluster \( C_{C^2} \); \( x_4 \) and \( x_7 \)—do nothing; \( x_5 \) and \( x_6 \) update cluster \( C_{C^1} \) to \( C_{C^2} \). When \( x_8 \) comes, if \( D_{\text{thr}} = 0.1 \) \( x_8 \) will create a new cluster, if \( D_{\text{thr}} = 0.6 \), two clusters are candidates to be updated—cluster \( C_{C^1} \) or cluster \( C_{C^2} \)—since the distances to the first and second clusters are almost equal (\( D_{C^1} = 0.57 \) and \( D_{C^2} = 0.6 \)). Applying the LEFCM algorithm, where the threshold \( D_{\text{thr}} \) is tuning online, the degree of membership between data and centers of clusters as well as reference centers are evaluated. According to Step 6, \( \text{DMC}_{i1} = 0.5 \) and \( \text{DMC}_{i2} = 0.41 \). According to Steps 7 and 9, cluster \( C_{C^1} \) has to be updated to \( C_{C^1} \) with center \( C_{C^1} \) and radius \( R_{U_{C^1}} \).

**4. Related Work**

In this section, the related works are briefly described. They present general models for monitoring and estimating the quality of links among nodes in MANETs.

Related works for efficient routing management based on quality of links are in the research community. Prediction of stable routes could be based on simple heuristics [7]; “graph theory” algorithms [8]; AI techniques: fuzzy-logic approaches [9–12]; multicriteria decision making, other theories [13, 14]. The mentioned algorithms rely on frequently update and broadcasting the information about ad hoc topological parameters acquired from the underlying routing protocols, such as number of neighbors, rate of changing the neighbors, receiving power of consecutive packets, and GPS location information. Routing table of each node or control packets are extended with new fields about this extra information, which increase the burden of these data structures and the overhead, since extra information is disseminated. In contrast, we summarize and predict mobility context as one digit, which is easy to be disseminated or compared to these of neighbors.

Only few studies learn node mobile motion behavior. A classification of the existing mobility prediction methods can be found in [15]. Authors determine a mobility prediction process as the future location of a mobile node. Their prediction scheme applies the evidence theory of Dempster-Shafer in order to predict the future position of the mobile node by evaluating itself on relevant criteria. In [7], authors propose a mobility prediction scheme based on GPS tracking to help select stable routes and to perform rerouting in anticipation of topology changes. In [16], mobile nodes learn using a hidden Markov model to predict the future state of host’s movement. Fuzzy applicability to express the matching degree among neighboring nodes is used in [17]. In [13], the authors introduce an adaptive mobility prediction that uses learning automation to estimate the coefficients of a simple adaptive filter to predict the future distance of two neighboring nodes. Distributed bioinspired routing algorithms explore the network and learn good routes, using variations of reinforcement learning (RL). These algorithms are adaptive to MANETs topology dynamism and changes in link costs; however, they inherit the above-mentioned RL drawbacks. GA-ODMRP [18] optimizes the route selection using particle swarm optimization. PIDIS [19] exploit swarm intelligence to keep track of the changing topology by a combination of positive/negative reinforcement and amplification of local network fluctuations.

Only few studies deal with social network routing [2, 20]. In [20], the authors use “social similarity” to destination node for information propagation in wireless networks. To the best of our knowledge, our framework is more inherit, generalized, and involves more abstract social level. It can be used for enhancing the service protocol algorithms for sending data packet from source to destination in an unsupervised and passive mode. It is reactive to environmental events in a role-based mode after deriving knowledge from the user and application high-level context.

**5. Conclusions**

The main and novel contribution of this paper is the distributed evolving fuzzy modeling framework (EFMF) to observe and categorize relationships and activities in the...
user and application level and based on that social context to take intelligent decisions about ambient intelligence mobile ad hoc networks (MANETs) service management. Especially, the EFMF employs unsupervised online one-pass fuzzy clustering method to recognize nodes mobility context from social scenario traces and ubiquitously learn “friends” and “strangers” indirectly and anonymously. The proposed evolving fuzzy modeling framework (EFMF) is applied for mobility context-awareness and routing services; however, EFMF is flexible and can be extended to learn ubiquitously traffic patterns over time or to aid security services—misbehaving nodes can be detected and reported to the upper layer and other nodes. The selfish or malicious nodes can be discovered by their social reaction to environmental events in a diverse role-based mode. The other nodes can isolate the misbehaving nodes, resulting in MANETs composed of only well-behaving nodes.

References


Submit your manuscripts at http://www.hindawi.com