An Entropy-based Stability Algorithm for Regulating the Movement of MANET Nodes

Sang-Chul Kim
School of Computer Science, Kookmin University,
861-1, Chongnung-dong, Songbuk-gu, Seoul, 136-702 Korea
[e-mail: sckim7@kookmin.ac.kr]
*Corresponding author: Sang-Chul Kim

Received September 7, 2010; revised November 24, 2011; revised February 24, 2011; accepted May 4, 2011; published May 31, 201

Abstract

This paper proposes an algorithm that enables mobile nodes to implement self-regulated movements in mobile ad-hoc networks (*MANET*s). It is important for mobile nodes to maintain a certain level of network-based stability by harmonizing these nodes' movements autonomously due to their limited transmission range and dynamic topology. Entropy methods based on relative position are suggested, as a means for mobile nodes to regulate their local movements. Simulations show that an early warning mechanism is suitable to maintain movement-based stability. Isolation can be reduced by 99%, with an increased network cost of 12% higher power consumption, using the proposed algorithm.

Keywords: Mobile ad hoc nnetworks, relative position, topology control, entropy, stability, movement profile, isolation.

A preliminary version of this paper appeared in ICUFN 2011, June 15-17, Dalian, China. This research was supported by the Seoul R&BD program (No.10848) of Korea, by the 2011 research fund of Kookmin University, and by National Research Foundation of Korea (NRF) grants (No.2010-0028224, No.2011-0002519) funded by the Korea government (MEST).

DOI: 10.3837/tiis.2011.05.007

1. Introduction

Ad hoc wireless communication has emerged as a new research topic due to the demand for the self-organization property of mobile and pervasive computing. In this, the dynamic topology for the rapid deployment of independent mobile users is a new factor to be considered. Ad hoc wireless communication can be characterized by comparing them to a single-hop mobile cellular network that support wireless communication by installing fixed base stations as access points, on which mobile nodes rely to communicate with each other. Wireless sensor network (*WSN*), mobile ad hoc network (*MANET*), and vehicular ad hoc network (*VANET*) are the application areas of the wireless ad hoc networks. In such networks, each mobile host becomes a member of a self organizing wireless network, in which one another communicate over multi-hop wireless links without relying on any fixed communication infrastructure, such as a base station or an access point. Nodes need to be self-creating, self-administrating, self-organizing, and self-configuring by acquiring information exchanged among nodes due to the lack of infrastructure in wireless channels.

Topology control for ad hoc networks aims to maintain a specified topology by controlling which links should be included in the network, to achieve high network performance by reducing interference, energy consumption, and end-to-end delay, mobility models that dictate the movement behavior of a mobile terminal play an important role in the simulation or analytical based analysis of the impact of dynamically changing topology on the performance of these networks, since nodes in an ad hoc network can move according to many different mobility profiles. Random mobility models formulate the movement pattern of mobile hosts by consecutive random length termed movement epochs. During each epoch, mobile terminals move at a constant speed, and in a constant direction for a random duration. The speed and direction choice for each epoch may or may not be correlated with their values in the previous epochs, and mobility characteristics of other terminals [1]. Therefore, the network topology is autonomously formed based on the mobility profile of the node.

The control mechanism is based on the OSI-7 layer network model and the master for executing topology control. The control mechanism can be categorized into a logical-oriented approach and a physical-oriented approach in which the network layer (or the medium access control (*MAC*) layer) and the physical layer have the main responsibility of performing the logical and the physical-oriented approaches, respectively. Examples of the logical-oriented approach include the algorithm to select a percentage of nodes performing query routing [2] and the algorithm using network cartography [3]. Examples of the physical-oriented approach include the direct movement of a node to certain core nodes and adjusting the transmission power of the nodes, which is the primary method to accomplish topology control [4].

The following algorithms furnish a brief introduction to the logical-oriented approach. The authors in [2] focused on maximizing the amount of time that nodes are in sleep mode, and thus saving energy, by using selective querying in sensor networks, rather than flooding messages. In addition, they introduced several performance metrics to evaluate their algorithm, whereby only a relatively small percentage of the nodes participate in communication activities and as a result, a high resolution view of surveillance areas of interest can be achieved. In [3], an algorithm using network cartography was proposed to minimize the routing overhead and to maximize the routing pertinence in the situation where nodes have dynamic mobility. This algorithm is a logical-based approach, since the algorithm mainly focuses on finding the route correctness to avoid incorrect routes that cause severe traffic drifting in the network and over consumption of valuable network resources. Optimizing *MAC* protocol parameters to meet the targeted overall network utilization [5] can be categorized as a logical-oriented approach, since the *MAC* protocol parameter gives the decision information to the network layer and the network layer has the main responsibility to control network topology.

In contrast, examples of the physical-oriented approach follow. A mobility-aware topology control strategy has been proposed in [6], in which the system changes its behavior in response to, or by predicting, node movements. The key point of the physical-oriented approach is that nodes change their locations autonomously to optimize certain performance metrics. For example, a mobile sensor node can move about a sensor field to collect data from other sensor nodes, reducing energy consumption at those nodes. This strategy has been proposed as a means to recover a disconnected topology, increase sensor surveillance coverage, as well as reduce total energy consumption. Controlled mobility can reduce energy consumption up to 50 percent compared to the energy consumption of random mobility [6]. The failure of a critical node through intermediate relay nodes can result in a network partition, which can be termed as a network failure, since every node in an ad hoc network is responsible for forwarding data packets to other nodes [7]. In [6], after receiving a data packet, an intermediate node calculates its new target location according to the current mobility strategy and starts moving toward that location to provide seamless data packet flows. A flow node periodically broadcasts hello messages containing the residual energy and location information regarding itself and its flow neighbors to achieve this. It is possible to complement node mobility strategies with approaches that account for energy management, as ad hoc nodes are used in disaster releif areas (search and rescue) or battlefields, as in a fleet of microrobots; some or all of the nodes in the networks are mobile. The particle swarm optimization based algorithm (*PSOA*) [8] is a physical-oriented approach that regards a node in the network as a swarm, and recognizes the nodes by the particle swarm optimization algorithm.

This paper proposes an entropy-based movement regulation algorithm as one of methods of the physical-oriented approach, whereby mobile nodes can easily implement a stable autonomous physical movement using *MANETs*. An entropy metric from information theory is introduced to define and measure the stability of node movements. The difference change of relative node positions will strongly affect the service quality of *MANETs*, where it is important for a node to be consistently located within the transmission range of another node to provide the best quality network service. The entropy metric is extended to regulate the relative positions of nodes that vary every second.

2. Related Work

In [7], the authors introduced a movement control algorithm to realize fault-tolerant ad hoc sensor networks, in which every sensor node includes its location information (*GPS* coordinates or indoor relative location information) whenever this node broadcasts a link status update (*LSU*) to the rest of the network. When *LSUs* arrived at node *R* from other nodes, *R* extracts the location information from each *LSU* and calculates the geographic center *C* for the entire network. After calculating *C*, each node moves toward *C*.

[9] expresses that the most commonly used mobility model in the ad hoc networking research is the random waypoint (*RWP*) model, in which each node chooses uniformly at random a destination point (waypoint) in a rectangular deployment region. A node moves to the latter destination with a velocity v chosen uniformly at random in the interval $[v_{min}, v_{max}]$.

After reaching the destination, the node pauses for a random amount of time, and the entire cycle is repeated by selecting a new destination. **Fig. 1** shows the movement of ten nodes with respect to *RWP* in a 1 km^2 network. Marks, such as the triangle, circle and dot, illustrate the nodes' positions at time Δt, 2Δt, and *x*Δt, respectively.

The ability to analytically characterize the spatial topology distribution of nodes plays a key role in understanding fundamental network QoS measure, such as throughput, probability of successful transmission, and connectivity. Accordingly, Alparslan *et al* proposed a generalized random mobility model sufficiently flexible to capture different topology scenarios, and provide its long-run location and speed distibutions by a closed form expression for one-dimensional mobility terrains. In addition, they classified the existing mobility models for wireless ad hoc networks, briefly summarizing their assumptions [1].

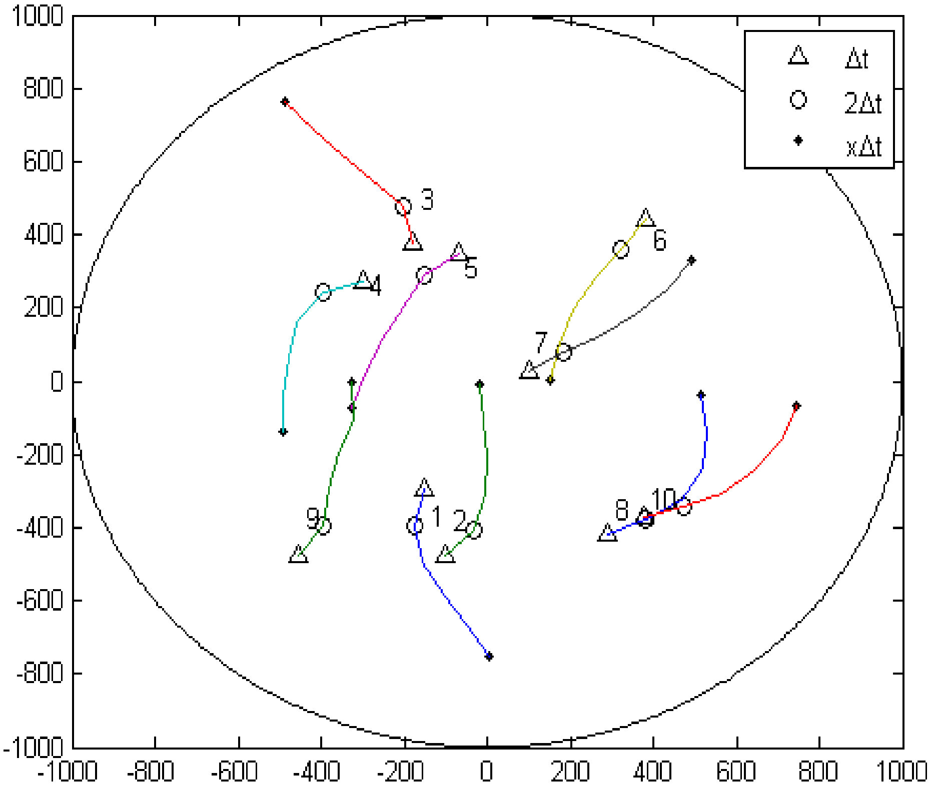

Fig. 1. *RWP* movement of nodes

Tang *et al.* studied the energy optimization problem that accounts for energy costs associated with both communication and physical node movement. They reviewed the theoretical foundations on how to reduce total communication energy consumption, as well as increase system lifetime, by combining node movement and transmission power [6]. The authors of [10] introduced a logical connectivity algorithm, in which a strong partitionable group with safe distance can guarantee the prevention of the occurrence of unannounced disconnection of ad hoc applications, to have a consistent application view among mobile ad hoc nodes.

MANETs are used as the method of the wireless communication network to control the movement of mobile robots in [7]. This indicates that traditionally researchers provided the use of server-client based network, in which all robot nodes communicate with a central controller (base station) over a wireless medium. However, in most application scenarios, it is difficult to guarantee the presence of a base station that coordinates the flow of information between any two nodes. Moreover, node movement can be severely restricted to stay in communication range of the base station, which ultimately causes the failure of wireless communication. They implemented the block movement algorithm (*BMA*) using the *MANET* characteristic of self-forming, self-organizing multi-hop communication, where the nodes' minimal possible movement to new locations yields the desired topography.

In *PSOA*, all nodes have fitness values, evaluated by the fitness function to be optimized. It is initialized with a group of random solutions and then searches for optima by updating generations. In every iteration, each node is updated by following the two best factors. The first one is the best fitness it has achieved so far; thus, it is stored in memory. Another best value obtained so far by any particle in the network is a global best. The best fitness and the global best are used as the parameters of the fitness function. Therefore, the current best position can be determined by recalling the previous best position [8]. However, when a leader, followed by hundreds or thousands of nodes in a group, loses a destination, all the following nodes may lose the same destination.

The literature review reveals that it is important for mobile nodes to obtain and keep a certain level of movement-oriented stability, when nodes are moving and relaying their packets due to limited transmission range and dynamic topology, since the multi-hop wireless connection can be easily broken in the middle of control and data packet transmission. Therefore, network-based communication might not perform well in a self-organized network. Therefore, this paper proposes an entropy-based stability algorithm to enable mobile nodes to auto-regulate their own movements, and eventually to maintain a stable wireless connection that results in high network performance, even when there is an external harmful environment, such as wireless interference.

3. Proposed Algorithm

Let's consider an ad hoc network, in which the nodes' positions are represented as a form of matrix based on a time sequence. $x(i, j)$ and $y(i, j)$ indicate the x and y coordinates of the i^{th} node at j^{th} time. Therefore, the matrix forms of $[x]_{M\times N}$ and $[y]_{M\times N}$ represent the x and y positions of all nodes, where M indicates the number of nodes in the network and N indicates the observed time. For example, the position matrix below represents $[x]_{3\times 3}$ where the first row consists of $x(1,1)$, $x(1,2)$, and $x(1, 3)$ with values of 79.88, 13.16, and 105.24, respectively, indicates a change of the x position of the first node at the first, second and third observed times.

$$P_x = [x]_{3\times 3} = \begin{bmatrix} 79.88 & 13.16 & 105.24 \\ 64.21 & 94.65 & 124.56 \\ 120.99 & 212.69 & 288.64 \end{bmatrix}$$

The position difference ($P(n, m, t)$) at t seconds of nodes n and m can be expressed as follows.

$$P(n,m,t) = (P_x, P_y)_n - (P_x, P_y)_m \quad (1)$$

The distance difference ($D(n, m, t)= D_{n,m}$) at t seconds of nodes n and m can be expressed as follows.

$$D(n,m,t)=\sqrt{\left|\left(P_x\big|_n - P_x\big|_m\right)^2 - \left(P_y\big|_n - P_y\big|_m\right)^2\right|} \tag{2}$$

It can be expected that if nodes n and m move in the same direction at the same speed, then the separation $D_{n,m}$ is not great. However, if they move in different or opposite directions, then great separation $D_{n,m}$ will occur. If the separation between a specific node and the surrounding nodes continues to be investigated for a certain period, then nodes are found that leave and move out of the transmission range, as a result of the great difference in speed with respect to a specific node, or there are nodes that can play the role of neighboring (one-hop) nodes by maintaining a small separation with respect to a specific node.

The average shift of the x, y coordinates up to t seconds of the i^{th} node is shown as $\left(\overline{P_x}, \overline{P_y}\right)_i$, where the average shift of the x coordinate up to t seconds of the i^{th} node can be shown as $\overline{P_x}\big|_i = \frac{\sum_{j=0}^{t} x(i,j)}{t}$, and the average shift of the y coordinate up to t seconds of the i^{th} node can be shown as $\overline{P_y}\big|_i = \frac{\sum_{j=0}^{t} y(i,j)}{t}$ [11].

The variable $k_{n,m}$, which is termed the friendly relationship constant (*FRC*), is introduced to show the distance relationship with respect to a specific node. *FRC* plays an important role in specifying the friendly relationship between nodes, since the distance between nodes varies with time due to node mobility. Nodes that are separated show a relatively high *FRC* value, while nodes maintaining little separation show a relatively low *FRC* value.

During N seconds, $k_{n,m}$ indicates the accumulated average change in distance with the n and m nodes, as is shown below. The node being separated shows a high $k_{n,m}$ value, while the friendly node shows a low $k_{n,m}$ value.

$$k_{n,m} = \frac{1}{N}\sum_{t=1}^{N} D(n,m,t) \tag{3}$$

When $k_{n,m}$ is divided by the transmission range (T) of nodes, $k_{n,m}$ assumes a value between 0 to 1; it is normalized and can be marked as $FRC(n,m)$, as is shown below.

$$FRC(n,m) = \frac{1}{T}k_{n,m} = \frac{1}{NT}\sum_{t=1}^{N} D(n,m,t) \tag{4}$$

$FRC(n,m)$ shows the normalized average value of the change of distance with n and m nodes during N accumulated seconds. When node n has neighboring nodes l, u, and v, the F_n (Family of node n) is shown as l, u, v, while the $FRC(n, l)$, $FRC(n, u)$, and $FRC(n, v)$ values can be calculated.

Therefore, when all *FRC* values are added and divided by $C(F_n)$ (the number of F_n element nodes), the entropy showing the position stability of the node using relative position is defined as $H_n(t)$, where $0 \le H_n(t) \le 1$, as shown below.

$$H_n(t) = \frac{1}{C(F_n)}\sum_{i \in F_n} FRC(n,i) \tag{5}$$

Fig. 2 illustrates an example of $FRC(i, x)$ value relation in each number of F_i element nodes when the network is initially constructed. Since $H_i(t)$ is composed of the $FRC(i, x)$ values of the number of F_i element nodes, denoted by x, each magnitude of $FRC(i, x)$ value can be compared, as shown in **Fig. 2**. The figure depicts the transmission range of each node colored by red, blue, black, and etc. and it is shown that the transmission range of two-tier case plotted with a double dotted circle is a little bit larger than the one of one-tier case.

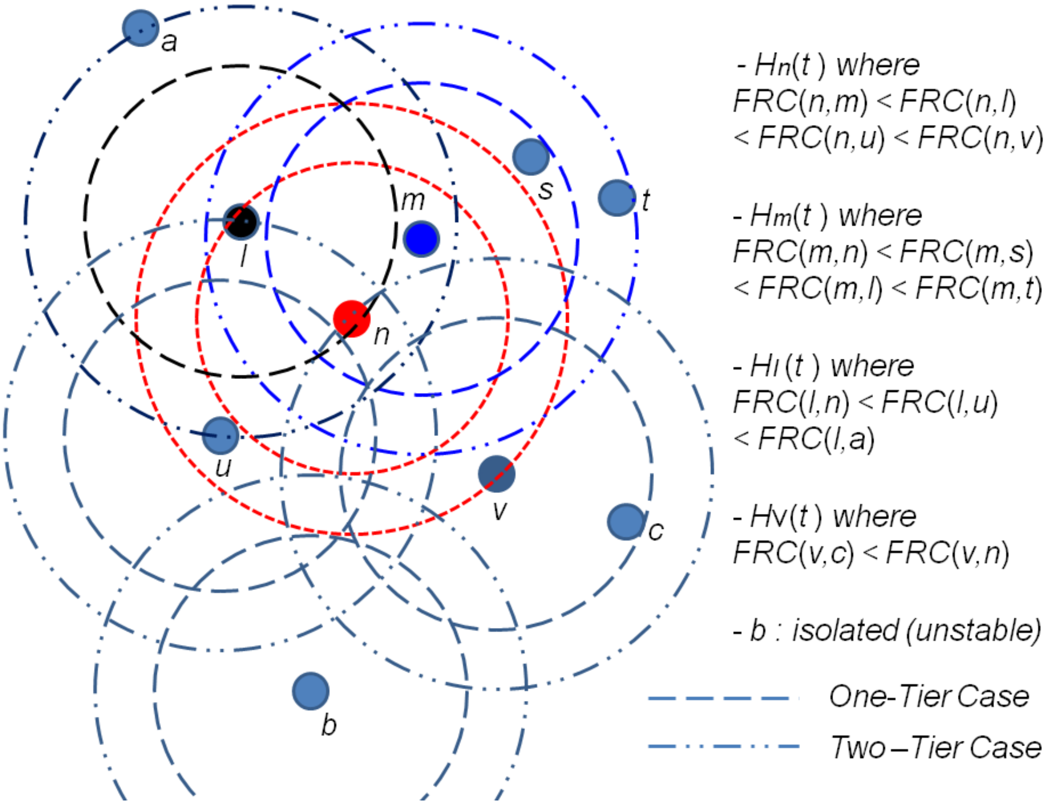

Fig. 2. An example of $FRC(i, x)$ magnitude comparison in F_i

4. Numerical Results and Observation

The discrete-event simulator was developed in *Matlab* to verify the performance of the proposed algorithm. The adaptive dynamic backbone (*ADB*) algorithm proposed in [4] has been implemented to verify the performance of the developed simulator. *ADB* is a clustering algorithm, where the entire network is covered by master nodes and where each master consists of several one-hop member nodes. The random node generator and simulator performance were verified when the number of nodes was 100, 125, 150 and 175. The average number of nodes per cluster, as well as several specifications of the *ADB* algorithm, matched the results in [4], using *QualNet*, with less than 1% difference in almost all cases.

Let's consider a scenario where ten nodes that are arbitrarily connected move towards destinations during 100 *secs* of simulation time, while constructing the family members of nodes. It is also assumed that nodes know their current positions via the global positioning system (*GPS*). There is no centralized control server to regulate their speeds and directions, since nodes are decentralized and autonomous. The random walk mobility model [8] has been used as the movement profile where every 0.5 *secs*, nodes choose their next speeds with a randomly chosen speed ranging from zero to four *m*/*s*, travel to their destinations with randomly chosen directions in the range (0, π), and construct clusters with a 100 *m* transmission range within a network size of 1 km^2.

The generation time of the *Hello* message, including the value of $H_n(t)$ and the current position is set to 1 and 2 *secs*. The threshold values of $H_n(t)$ to warn node n of the separation from the family members of node n (F_n) are set to 0.75 and 0.93. Whenever the $H_n(t)$ value of node n is above the threshold value, such as 0.75 or 0.93, the proposed warning system is activated among F_n members and gives nodes guidance in terms of the next direction. The power consumption of transmitting the message to F_n members is calculated when the $H_n(t)$ value of node n is above the threshold value, since the *Hello* message includes the guidance information.

Fig. 3 shows the architecture of *Hello* message generation with two different times. When node n is in a warning status informed by the *Hello* message, node n can select its next direction to reduce the $H_n(t)$ value. In **Fig. 2**, node movement occurs every 0.5 *secs* with random speed, and the direction is guided by *Hello* message generated every 1 *sec* and 2 *secs*. Therefore, it can be expected that the *Hello* in the 1 *sec* case is faster than *Hello* in the 2 *secs* case, when the node returns to the stable status. However, since *Hello* in the 1 *sec* case generates a message every second, it uses more power compared to *Hello* in the 2 *secs* case.

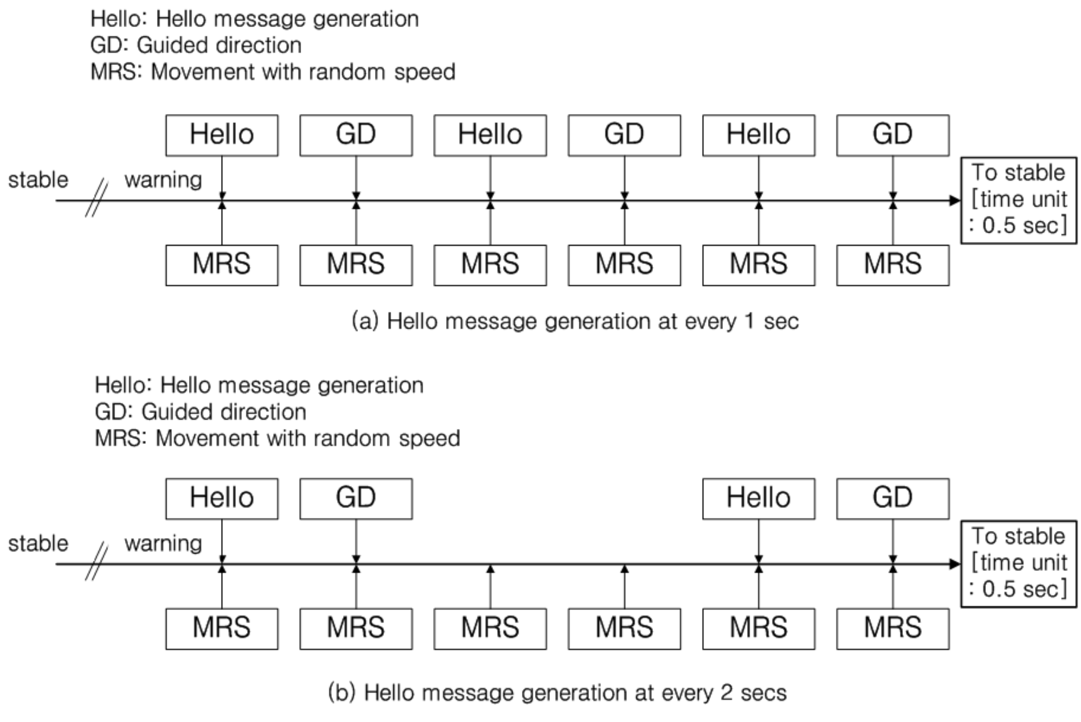

Fig. 3. *Hello* Message Generation

Nodes in the random walk mobility model moving towards the destination undergo not only many instances of separation from and re-joins to F_n members, but also many instances of partitions from and re-merges to a node cluster due to the wireless channel condition, such as multipath fading and multiple interferences. This kind of movement profile is common in ad-hoc networks. This occurs until they arrive at the destination, which will increase the number of communication breaks and decrease the movement stability. The number of separation instances from F_n members and the number of partition instances from a cluster are counted as the number of isolation instances, named as a failure in [12], that represents the performance metric used in this paper to show the entropy-based stability of the node.

An unstable status, where a node has been located outside the transmission range of F_n members and has been isolated from them, can result in the packet transmission error in ad-hoc networks. Therefore, in the entropy ($H_n(t)$) based-simulation, when $H_n(t)$ of node n is above the threshold value, with a randomly chosen speed, it is guided to change its direction towards the center of its family members (F_n). This is based on the theory of the center of gravity method. This results in a reduction of $H_n(t)$ and increases the movement stability compared to the stability of the block movement algorithm (*BMA*) proposed in [7], where the directions of the temporary failure nodes are guided to a geographic center of an entire network. In *PSOA*[8], even though $H_n(t)$ of node n is below the threshold value, it is guided to change its direction based on the best fitness and the global best. Therefore, it is regarded that all the nodes are in an unstable status from the start of travel, regardless of checking the location with F_n members. Conversely, in the entropy ($H_n(t)$) based-approach, the nodes may undergo the unstable status in the middle of travel.

Four cases (*BMA*, *PSOA*, one-tier with entropy, and two-tier with entropy) can be used to explain the benefits of the proposed algorithms. The *BMA* scenario has two cases, as below. The first case is that every 0.5 *s*, the nodes select their speeds and directions randomly without guidance, if they are not in a failure situation. The second case is that the nodes select their speeds and directions with guidance, if they are in a failure situation. In the *PSOA* scenario, every 1 *sec*, the nodes select their speeds and directions with guidance, even though they are in a stable status. The details of *PSOA* scenario follow the parameters of [8].

In the one-tier (1-*Tier*) entropy case, every 1 *sec*, the nodes randomly use selected speeds and follow the directions guided by the center of the family members when $H_n(t)$ of node n is above the threshold value. Subsequently, a node may undergo an isolation instance even though nodes follow directions due to the fact that the speed is random. In such a case, the isolation instance is counted, and a link break occurs in the one-tier entropy case.

The two-tier (2-*Tier*) entropy case, where nodes use two-times increased power as one of the members of F_n or a cluster, is adopted to avoid the situation of an isolated node occurring in the 1-*Tier* entropy case. When node n can contact one of F_n or cluster members with the 2-*Tier* power mechanism, we can say that there is no isolation of it with a cost of a two-times increased transmission power. This will increase movement stability.

Fig. 4 and **5** show the number of isolation (unstable) instances when entropy (warning) values of 0.75 and 0.93 are used as the respective threshold values, where the *BMA* (1 *sec*) case has the largest number of isolation instances, while the *PSOA* (1 *sec*) case has the smallest number of isolation instances, since the nodes select their speeds and directions with guidance every 1 *sec*, even though they are not in a warning status, and the 2-*Tier* (1 *sec*) case has the second smallest number of isolation instances, since the nodes select their speeds and directions with guidance every 1 *sec*, only if they are in a warning status.

Thus, in **Fig. 4** and **5**, the number of isolation instances in 1- or 2-*Tier* (1 *sec*) is much lower than in 1- or 2-*Tier* (2 *secs*), since 1- or 2-*Tier* (1 *sec*) case guides the nodes in a warning status towards a stable status every 1 *sec*, while 1- or 2-*Tier* (2 *secs*) case generates a *Hello* message to guide the nodes in a warning status towards a stable status every 2 *secs*, using the proposed entropy algorithm.

Compared to cases with a threshold of 0.75, cases with a threshold of 0.93 undergo a much higher number of isolation instances, since, when $H_n(t)$ of the node n is above the threshold value of 0.75, the node runs the proposed algorithm and changes its direction towards the center of the family members. This plays an early warning compared to the cases with a threshold value of 0.93. Therefore, a threshold of 0.75 is an early warning system, and a

threshold of 0.93 is a delayed warning system. In addition, a threshold of 0.75 is more likely to be stable than one of 0.93. Thus, 1- and 2 -*Tier* with a threshold of 0.75 undergo a lower number of isolation instances in **Fig. 4** than in **Fig. 4**. For example, the isolation of 1-*Tier* (1 *sec*) with a threshold of 0.75 can be reduced 3,238 times compared to the isolation of 1-*Tier* (1 *sec*) with a threshold of 0.93. The isolation of *PSOA* (1 *sec*) case has been reduced 522 times compared to the isolation of 2-*Tier* (1 *sec*) with a threshold of 0.75.
Since 2-*Tier* cases have more chances of reaching unstable nodes with a two-times power increase, they have a little lower number of isolation instances compared to 1-*Tier* cases in **Fig. 4** and **5**.

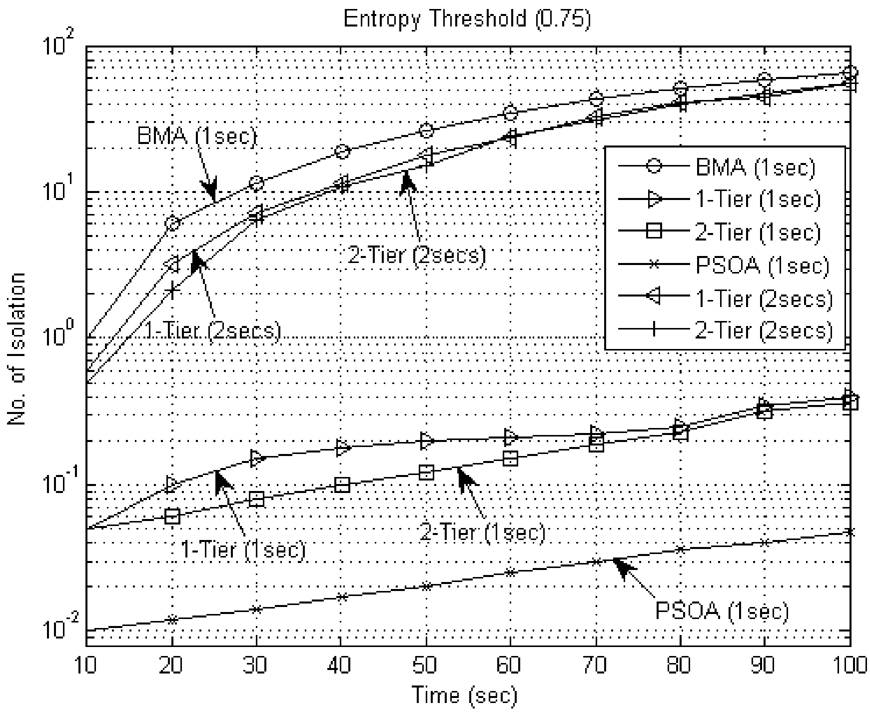

Fig. 4. Number of Isolation (Entropy Threshold=0.75)

In **Fig. 6** and **7**, the equation defined in (11) indicates the amount of power consumed (*P*) in the above cases using the *Hello* control message, where *D*(*n*, *m*, *t*) represents the distance of nodes *n* and *m* defined in (2), and P_r represents the receiver sensitivity of *Bluetooth*, which is -70 *dBm*, as specified in the *Bluetooth* specifications [13].

$$P = D(n,n,t)^4 \cdot P_r \tag{6}$$

In **Fig. 6** and **7**, 2-*Tier* (2 *secs*) consumes the highest power to reach the unstable nodes with a twofold increase in transmission power. However, 1-*Tier* (2 *secs*) consumes the least power during the simulation, since the *Hello* message is generated every 2 *secs*. The power consumption, representing no use of entropy, is marked as random.

Two graphs, 2-*Tier* (1 *sec*) and 1-*Tier* (2 *secs*) are compared below to show that the 0.75 entropy threshold warning system is better than the 0.93 entropy threshold warning system. Based on **Fig. 4** with a 0.75 entropy threshold system, 2-*Tier* (1 *sec*) has a 12,054% reduction in isolation instances with 1-*Tier* (2 *secs*), while, based on **Fig. 5** with a 0.93 entropy threshold system, 2-*Tier* (1 *sec*) with a 0.93 entropy threshold has a 357% reduction in isolation instances with 1-*Tier* (2 *secs*). Moreover, based on **Fig. 6** with a 0.75 entropy threshold, 2-*Tier* (1 *sec*) consumes 12% more power than 1-*Tier* (2 *secs*). In addition, based on **Fig. 7** with a 0.93 entropy threshold, 2-*Tier* (1 *sec*) consumes 49.4% more power than 1-*Tier* (2 *secs*).

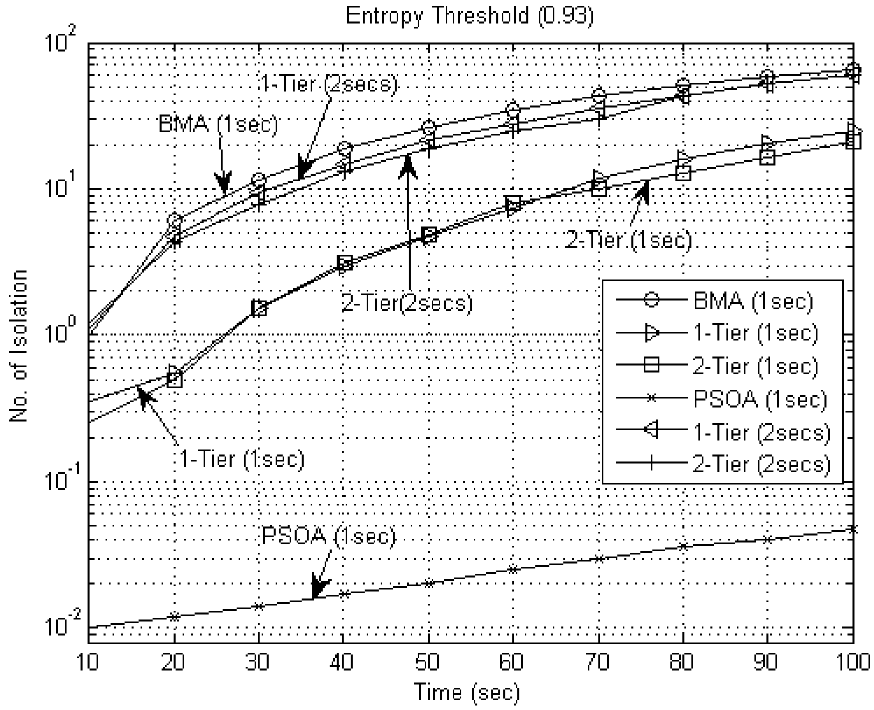

Fig. 5. Number of Isolation (Entropy Threshold=0.93)

Based on **Fig. 4** and **6**, *PSOA* and 2-*Tier* (1 *sec*) have 522% and 12,054% reduction in isolation instances with 270% and 12% more power consumption compared to the power consumption of 1-*Tier* (2 *secs*) respectively. It can be said that 2-*Tier* (1 *sec*) is better than *PSOA*, since 2-*Tier* (1 *sec*) has higher reduction rate with less power consumption than *PSOA*.

Based on **Fig. 6** and **7**, 2-*Tier* (1 *sec*) with a 0.75 entropy threshold consumes 12% more power than 1-*Tier* (2 *secs*), while 2-*Tier* (1 *sec*) with a 0.93 entropy threshold consumes 49.48% more power than 1-*Tier* (2 *secs*).

Compared to *BMA* where *BMA* follows the power consumption of 1-*Tier* (1 *sec*), since it generates a *Hello* message every 1 *sec*, when the failure node is detected, we can say that with a 12% increase in power, isolation can be reduced by 99% with 2-*Tier* (1 *sec*) with a 0.75 entropy threshold, while with a 49.48% increase in power, isolation can be reduced by 78.72% with 2-*Tier* (1 *sec*) with a 0.93 entropy threshold.

Therefore, it is shown that a 0.75 entropy threshold is a better choice than a 0.93 entropy threshold to reduce isolation, with less power consumption. It is also known that to reduce isolation instances in both cases of 0.75 and 0.93 entropy thresholds, more network resource, such as power, should be allocated.

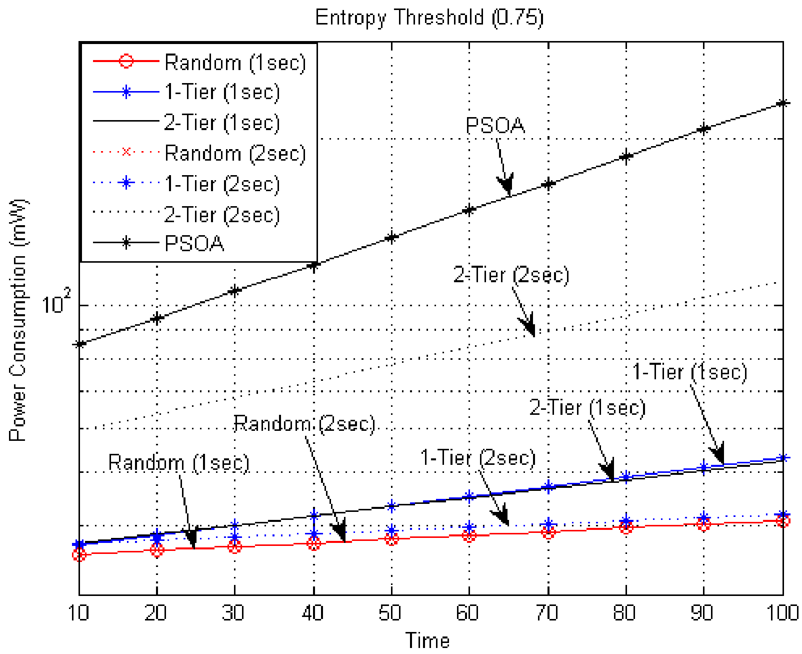

Fig. 6. Power Consumption (Entropy Threshold=0.75)

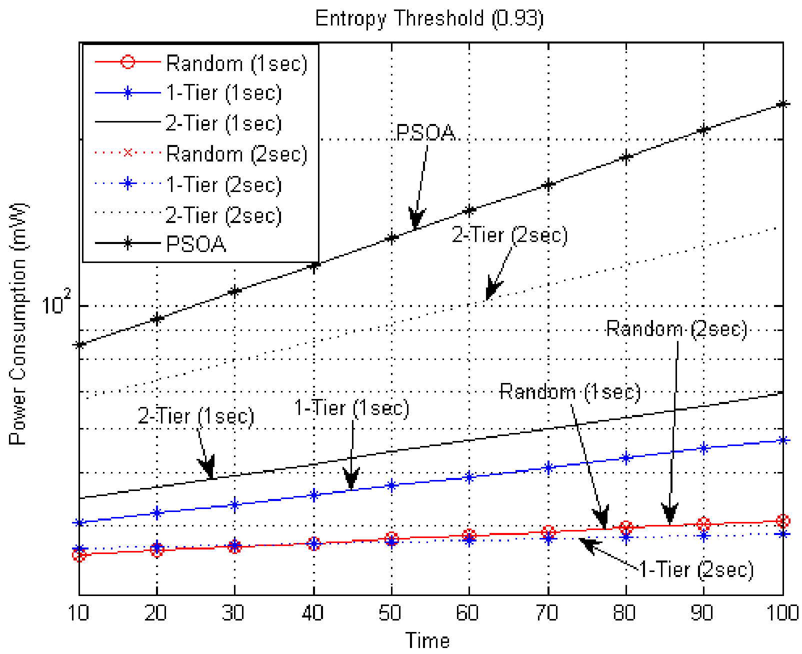

Fig. 7. Power Consumption (Entropy Threshold=0.93)

4. Conclusion

This paper proposed an algorithm for network-based mobile nodes to implement a decentralized self-regulated stable movement system. In addition, the mathematical background to achieve node movement stability was presented.

An entropy metric is used to regulate the future movements of mobile nodes based on the current entropy. It has a decision metric to determine whether a node is stable based on past accumulated entropy. It was shown that the effect by the guidance controlled by the direction change of the entire network (global) does not strongly impact each node to revert to a stable status compared to the effect by the guidance controlled by the direction change of F_n members (local). An early warning system is an effcient method to obtain stable movements. The proposed algorithm can reduce isolation by 99% with an increased network cost of 12% greater power consumption compared to that of the conventional *BMA* algorithm.

Based on the guided movement regulations among nodes, as well as the construction method of stable node groups, the self-administrating, self-organizing and self-configuring characteristics of mobile nodes can be easily deployed in a network-based multi-node area. They can ultimately support cooperative tasks, such as rescue missions in an emergent disaster situation.

References

[1] D. N. Alparslan and K. Sohraby, “A generalized random mobility model for wireless ad hoc networks and its analysis one-dimensional case,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 3, pp. 602-615, Jun. 2007. Article (CrossRef Link)

[2] F. Mili and J. Meyer, “Self-regulating sensor network,” in *Proc. of IEEE International Midwest Symposium on Circuits and Systems*, pp. 165-168, Aug. 2010. Article (CrossRef Link)

[3] A. Belghith and M. Abid, “Cartography based self regulating proactive routing protocols in MANETs,” in *Proc. of IEEE Global Information Infrastructure Symposium*, pp. 1-8, Jun. 2009. Article (CrossRef Link)

[4] C-. C. Shen, C. Srisathapornphat, R. L. Z. Huang, C. Jaikaeo and E. L. Lloyd, “CLTC: A cluster-based topology control framework for ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 18-32, Jan./Feb. 2004. Article (CrossRef Link)

[5] S. Ci, M. Guizani, H. Chen and H. Sharif, “Self-regulating network utilization in mobile ad hoc wireless networks,” *IEEE Transactions on Vehicular Technology*, vol. 55, no. 4, pp. 1302-1310, Jul. 2006. Article (CrossRef Link)

[6] C. Tang and P. K. McKinley, “Energy optimization under informed mobility,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 9, pp. 947-962, Sep. 2006. Article (CrossRef Link)

[7] P. Basu and J. Redi, “Movement control algorithms for realization of fault-tolerant ad hoc robot networks,” *IEEE Network*, pp. 36-44, Jul./Aug. 2004. Article (CrossRef Link)

[8] X. Wu, J. Cho, B. J. D’auriol and S. Lee, “Mobility-assisted relocation for self-deployment in wireless sensor networks,” *IEICE Transactions on Communications*, vol. E90–B, no. 8, pp 2056~2069, Aug. 2007. Article (CrossRef Link)

[9] C. Bettstetter, G. Resta and P. Santi, “The node distribution of the random waypoint mobility model for wireless ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, pp. 257-269, Jul./Sep. 2003. Article (CrossRef Link)

[10] Q. Huang, C. Julien and G.-C. Roman, “Relying on safe distance to achieve strong partitionable group membership in ad hoc networks,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 2, pp. 192-205, Apr./Jun. 2004. Article (CrossRef Link)

[11] B. An and S. Papavassiliou, “An entropy-based model for supporting and evaluating route stability in mobile ad hoc wireless networks,” *IEEE Communications Letters*, vol. 6, no. 8, pp. 328-330, Aug. 2002. Article (CrossRef Link)

[12] D. N. Alparslan and K. Sohraby, “Two-dimensional modeling and analysis of generalized random

mobility models for wireless ad hoc networks," *IEEE Transactions on Networking*, vol. 15, no. 3, pp. 616-629, Jun. 2007. Article (CrossRef Link)
[13] Bluetooth SIG, "Specification of the Bluetooth system," ver. 2.1+EDR, p. 41, Jul. 2007.

Sang-Chul Kim received the B.S. and M.S. degrees in Electrical Engineering from Kyungpook National University and Computer Science from Changwon National University in 1994 and 1998, respectively. He received the Ph.D. degree in Electrical & Computer Engineering from Oklahoma State University, U.S.A. in 2005. During 1994–1999, he worked in Samsung SDS and Aerospace as a System Engineer. He is now an Associate Professor in the School of Computer Science at Kookmin University, Seoul, Korea. His research areas are wireless communications and real-time operating systems.