An Energy Efficient Adaptive HELLO Algorithm for Mobile Ad Hoc Networks

Danping He, Nathalie Mitton, David Simplot-Ryl

An Energy Efficient Adaptive HELLO Algorithm for Mobile Ad Hoc Networks

Danping He
Centro de Electronica Industrial
Universidad Politecnica de Madrid
danping.he@upm.es

Nathalie Mitton and David Simplot-Ryl
Inria Lille – Nord Europe
{firstname.lastname}@inria.fr

ABSTRACT

HELLO protocol or neighborhood discovery is essential in wireless ad hoc networks. It makes the rules for nodes to claim their existence/aliveness. In the presence of node mobility, no fix optimal HELLO frequency and optimal transmission range exist to maintain accurate neighborhood tables while reducing the energy consumption and bandwidth occupation. Thus a Turnover based Frequency and transmission Power Adaptation algorithm (TFPA) is presented in this paper. The method enables nodes in mobile networks to dynamically adjust both their HELLO frequency and transmission range depending on the relative speed. In TFPA, each node monitors its neighborhood table to count new neighbors and calculate the turnover ratio. The relationship between relative speed and turnover ratio is formulated and optimal transmission range is derived according to battery consumption model to minimize the overall transmission energy. By taking advantage of the theoretical analysis, the HELLO frequency is adapted dynamically in conjunction with the transmission range to maintain accurate neighborhood table and to allow important energy savings. The algorithm is simulated and compared to other state-of-the-art algorithms. The experimental results demonstrate that the TFPA algorithm obtains high neighborhood accuracy with low HELLO frequency (at least 11% average reduction) and with the lowest energy consumption. Besides, the TFPA algorithm does not require any additional GPS-like device to estimate the relative speed for each node, hence the hardware cost is reduced.

Categories and Subject Descriptors

C2.1 [**Network Architecture and Design**]: Network topology

General Terms

Algorithms

MSWiM'13, November 3–8, 2013, Barcelona, Spain.
Copyright 2013 ACM 978-1-4503-2353-6/13/11
Enter the DOI string/url from the ACM e-form confirmation ...$15.00.

Keywords

Adaptive Hello Algorithm; Energy Efficient; Mobile Ad Hoc Network; Neighborhood Discovery; Transmission Range; Wireless Sensor Networks

1. INTRODUCTION

Wireless Sensor Networks (WSNs) are composed of sensor nodes which have the ability to detect surrounding environments and are able to communicate with each other through specific wireless protocols. Mobility of the sensor nodes is an emerging issue and nowadays it becomes more and more under the attention of scientific community. It arises new questions such as optimization of energy consumption, connectivity of the WSN, routing in mobile networks and a lot more. Sensor nodes can be attached to animals with the purpose to track animals' habits and their natural habitat. In this case, mobility pattern is very often hard to estimate and protocols built upon this issue must take account of possible losses of connectivity or delays in the transmission of messages. Other possibility would include mobile agents (robots) which may have been given mobile pattern to follow to improve some of the parameters in the WSN [7]. In this case, overall energy efficiency of transmission can be significantly augmented using controlled mobility based smart placement of nodes related to the routing path. Due to the specific nature of mobile networks, some of the characteristic mechanisms used in static WSNs need to be redefined and adapted to the specific types of node mobility.

Neighborhood discovery is one of the most important protocols in WSN. The mechanism behind this protocol is rather simple, it includes periodically sending a specific type of message, called HELLO messages (also known as beacon messages) and gathering data from the received HELLO messages. HELLO messages contain the data of the sender id, unique identification number for the node in the WSN-usually MAC address in practical applications. Each node, usually, acquires data from all HELLO messages that it has received and organizes them into the neighborhood table which can be further used for some kind of topology control [10] or proactive routing [4].

However, finding the proper HELLO frequency is not obvious: if the HELLO frequency is too low, nodes may not be detected by their neighbors, leading to deprecated neighborhood tables, and protocol failures are likely to occur. On the contrary, if the frequency is too high, neighborhood tables are up to date, but the energy and bandwidth are wasted.

There exist many algorithms that utilize the information gathered from additional device such as GPS. The ARH algorithm presented in [6] adapts HELLO frequency when there is obvious error between the estimated speed and actual speed read from GPS. In [5], the authors deduced the relationship between the HELLO frequency and the link failure possibility for a given speed which again is obtained from GPS. The proper HELLO frequency is then calculated to guarantee that the link failure possibility is not greater than the predefined threshold. However, the speed obtained from GPS can not represent the actual changes in the surrounding environment. When a node is static or moving slowly but its neighbors are changing frequently, the HELLO frequency calculated by the previous two algorithms is lower than the required value to maintain the expected accuracy. On the contrary, when a node is moving fast with a relatively static neighborhood table, the computed HELLO frequency will be higher than the actual required value and will consume more energy.

The authors in [3] proposed a Turnover based Adaptive hello Protocol (TAP) to reduce the hardware cost and to consider the relative changes among nodes. It calculates the number of changes that appears in the neighborhood table, called *turnover*. The optimal value of *turnover* is derived and each node adapts the HELLO frequency by increasing or decreasing the value within a fixed range to keep the *turnover* as close as possible to the optimal value. In [9], the authors proposed two algorithms, one is run with GPS called *Cost*, the other called *NoTAP* relies on the TAP algorithm. Both algorithms adapt the HELLO transmission range to reduce the energy consumption. *Cost* risks sending HELLO messages with a very low power. Although the neighborhood table of *Cost* is accurate and power consumption is low, the average number of neighbors is too small to guarantee the connectivity of the network.

In this paper, a Turnover based Frequency and transmission Power Adaptation algorithm (TFPA) is presented. It is based on a previous work and allows to **dynamically adapt the HELLO frequency and its transmission power based on the estimated relative speed for each node**. The contributions of this work are:

- a theoretical analysis allowing the computing of the relative speed from the *turnover* in neighborhood,
- a theoretical analysis of the global energy optimal transmission range of HELLO message. The analysis highlights that the optimal range is independent of the speed, and hence can be applied to all nodes in the network to reduce the overall energy consumption,
- an adaptive neighborhood discovery mechanism, TFPA, that dynamically adapts jointly node transmission range and HELLO frequency, allowing accuracy and energy efficiency.

In TFPA, every node regularly checks its neighborhood and based on the observed changes and the theoretical analysis results, every node dynamically and periodically adapts both of its transmission range and HELLO frequency, allowing low energy consumption while maintaining reliable neighborhood tables. The features of TFPA are as follows:

- **Local**: in TFPA, every node only watch its neighborhood to adapt range and frequency.
- **Distributed**: every node computes the same algorithm.
- **GPS-free**: TFPA is not constraint by GPS-like devices and can be applied to general mobile ad hoc network.
- **Energy-efficient**: Results show that applying TFPA allows up to 11% energy savings.
- **Reliable**: Results show that neighborhood tables achieved through TFPA present the same error and accuracy ratios than alternative neighborhood discovery protocols.

The remainder of the paper is organized as follows: Section 2 sets the models, notations and preliminaries to this work. Section 3 provides the theoretical analysis related to this work. Section 4 presents the TFPA algorithm. Simulation results are detailed in Section 5. Section 6 gives conclusion and directions of future work.

2. MODELS AND PRELIMINARIES

2.1 Models and Notations

Wireless networks are represented by a graph $G = (V, E)$ where V is the set of nodes and $E \subseteq V^2$ is the set of edges: $(u, v) \in E$ means that u and v are neighbors (i.e., close enough to communicate). The neighborhood set $N(u)$ of a vertex u is equal to

$$v : (u, v) \in E \vee (v, u) \in E.$$

Each node is assigned a unique identifier (e.g., a MAC address). Wireless links are determined by the physical model. The most frequent one is the unit disk graph model [1]:

$$E = \{(u, v) \in V^2 | u \neq v \wedge |uv| \leq R_{max}\}$$

with $|uv|$ being the Euclidean distance between nodes u and v, and R_{max} the maximum communication range.

The basic HELLO protocol, first described in the OSPF [8], works as follows: Nodes regularly send HELLO messages to signal their presence to nodes near by, and maintain a neighborhood table. The frequency of these messages is noted f_{HELLO} and the delay between them d_{HELLO} (i.e., $d_{HELLO} = 1/f_{HELLO}$). When a node u receives such a message from a node v, u adds v to its table, or updates the time stamp of the entry if v was already there. We do not make assumptions about the content of HELLO messages, but they must contain the identifier of the sender.

The relation between HELLO frequency and relative speed is formulated (named Fopt for simplicity) in our previous work [11]. The idea of Fopt is that a node which strides a given distance in the communication area of another node has to be detected with a certain chance. If two nodes with transmission range R, move with a relative speed S, to make them discover each other with a chance of $1 - \alpha$, the optimal HELLO frequency is:

$$f_{opt} = \frac{2S}{\alpha R} \tag{1}$$

where $\alpha < 1$ predefines an expectation of accuracy of neighborhood table, e.g. the expected accuracy is 90% when $\alpha = 0.1$.

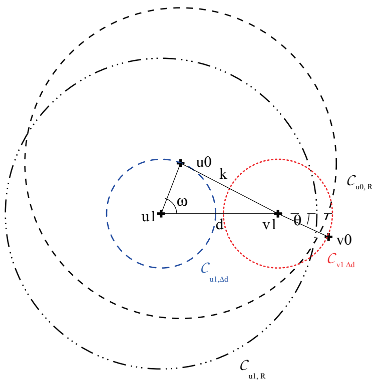

Figure 1: A new neighbor v of a node u - Global view [3].

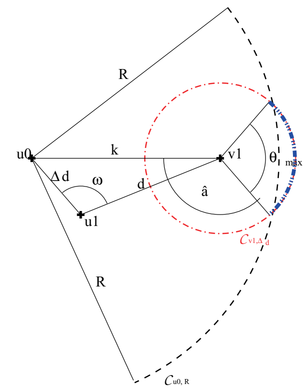

Figure 2: A new neighbor v of a node u - Zoom [3].

2.2 Turnover

The concept of *turnover* is first defined by [3], which is expressed by the ratio of new neighbors of a sensor and the number of all its neighbors between two HELLO packets. We re-establish part of the analysis provided in [3] that will be useful for the follow-up of this paper. In particular, we return the analysis of the optimal turnover ratio.

Let u_0 (resp. v_0) be the position of node u (resp. v) at time t_0 and u_1 (resp. $v1$) be its position at time t_1, the probability $P(d)$ that node v is a *new* neighbor of u is calculated through finding the probability that $|u_0v_0| > R$ knowing that $|u_1v_1| < R$. Fig. 1 illustrates the mobility model. Given the new position of a node and its traveling distance $\Delta d = S \times \Delta t$, the blue dashed circle $C_{u_1,\Delta d}$ (resp. red dotted circle $C_{v_1,\Delta d}$) represents the possible position of u_0 (resp. v_0). There exists a minimum distance $d_{min} = min(0, R{-}2\Delta d)$ that if $d < d_{min}$, v was an existing neighbor of u and $P(d < d_{min}) = 0$. Only the node v comes from the dotted blue angular sector of Fig. 2 can become a new neighbor of u. The $P(d)$ is expressed as:

$$P(d) = \begin{cases} \frac{1}{\pi^2}\int_{\omega_{min}}^{\pi}\theta_{max}d\omega & \text{if } d_{min} < d < R \\ 0 & otherwise \end{cases} \tag{2}$$

where $\theta_{max} = 2\arccos(\frac{R^2-\Delta d^2-k^2}{2k\Delta d})$ is the maximum angle of the dotted blue sector, $k = \sqrt{\Delta d^2 + d^2 - 2d\Delta d\cos\omega}$ and $k > R - \Delta d$ which leads to $\omega_{min} = \arccos(\frac{d^2+2R\Delta d-R^2}{2d\Delta d})$ according to the law of cosines.

In this analysis, it is assumed that nodes are randomly deployed using a Poisson Point Process (node positions are independent) with a density $\lambda > 0$, λ being the mean number of nodes per surface unit. The expected number of new neighbors that node u encounters after a time period Δt is simply equal to:

$$E[n]_{\Delta t} = \int_{d=0}^{R} 2\lambda\pi P(d)dd \tag{3}$$

By substituting (2) into (3), we have:

$$E[n]_{\Delta t} = \frac{\lambda}{\pi}\int_{d_{min}}^{R}\int_{\omega_{min}}^{\pi} d * \theta_{max}d\omega dd \tag{4}$$

In this work, $E[n]_{\Delta d}$ is used as a synonym of $E[N]_{\Delta t}$. The *turnover* of node u is expressed by the ratio of the new neighbors and number of all neighbors between two HELLO packets:

$$r = \frac{E[n]_{\Delta d}}{\lambda\pi R^2} \tag{5}$$

The above procedure shows how to deduce r from the neighborhood observation in the previous work [3]. Afterwards, we approximate the integration expression of r and the relative speed S is derived.

3. THEORETICAL ANALYSIS OF HELLO FREQUENCY AND RANGE ADAPTATION

3.1 Relationship between Turnover Ratio and Relative Speed

Fig. 3 plots the variation of $E[n]_{\Delta d}$ with different R and Δd. As can be seen, for a given Δd, $E[n]$ increases as R increases. Note that the curves vary almost linearly with R thus they can be approximated as linear variation and we have $E[n]_{\Delta d}$ expressed as:

$$E[n]_{\Delta d} = \frac{\lambda}{\pi}l(\Delta d) * R$$

where $l(\Delta d)$ is the slope of a line – a function of Δd. By checking its value for different Δd, it has $l(5) = 13\pi$, $l(10) = 26\pi$, $l(20) = 51\pi$, thus $l(\Delta d) = 2.6\pi\Delta d$ and we have $\frac{\lambda}{\pi}l(\Delta d) \approx 2.6\lambda\Delta d$. Therefore, $E[n]_{\Delta d}$ can be expressed as:

$$E[n]_{\Delta d} = 2.6\lambda\Delta dR \tag{6}$$

By substituting (6) to (5), we obtain the approximated expression for turnover ratio:

$$r = \frac{2.6\Delta d}{\pi R} \tag{7}$$

If $\Delta t = \frac{1}{f_{opt}}$, which is the optimal HELLO period (1) defined by Fopt, the $\Delta d = S \times f_{opt} = \frac{\alpha R}{2}$, then we have

$$r_{opt} = \frac{1.3\alpha}{\pi}$$

As can be noticed, r_{opt} is only a function of α and is not dependent on S, R and λ. For different α:

$$r_{opt}(0.1) \approx 0.04,\ r_{opt}(0.2) \approx 0.08,\ r_{opt}(0.3) \approx 0.12$$

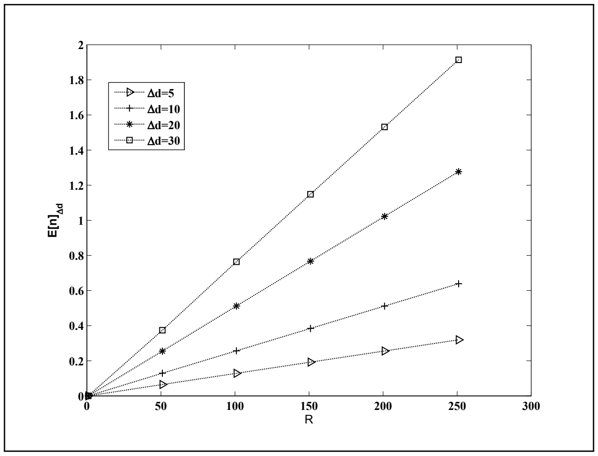

Figure 3: $E[n]_{\Delta d}$ varies with R for different Δd

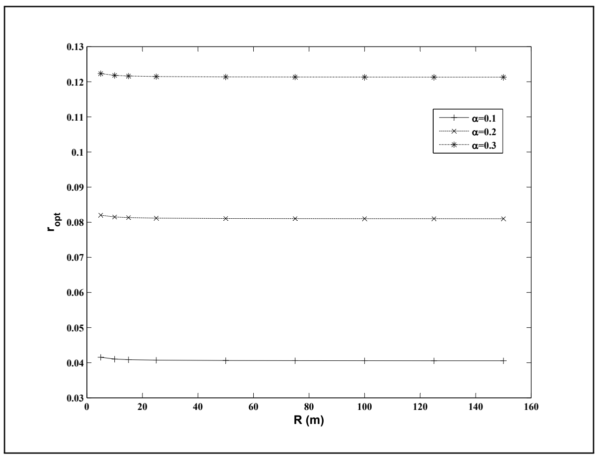

Figure 4: r_{opt} drawn by TAP [3]

When compared to the *turnover* curves drawn in TAP (see Fig. 4), the results of the simplified expression of r_{opt} correlate very well with the curves. Therefore the linear approximation of the complex integration expression in 4 is validated to be correct. The analysis will then be benefit from the linear approximated expression.

In practical application, the *turnover* r is obtained through analyzing the receiving HELLO messages for a time period Δt. Note that (7) can be represented as $r = \frac{2.6\Delta t \times S}{\pi R}$. Once r is provided, the relative speed S is calculated by

$$S = \frac{r\pi R}{2.6\Delta t} \tag{8}$$

Afterwards, the HELLO frequency f_{HELLO} and transmission range R are calculated by (1) and the adaptation solution are available to the each node.

We provide results that demonstrate the correctness of the theoretical analysis for speed estimation. In Fig. 5 the blue line represents the speed estimated through (8) by observing the *turnover* and the red line represents the speed when TAP is implemented. At the beginning of establishing the network, each sensor node is a new comer to their "neighbors", and many new neighbors are detected

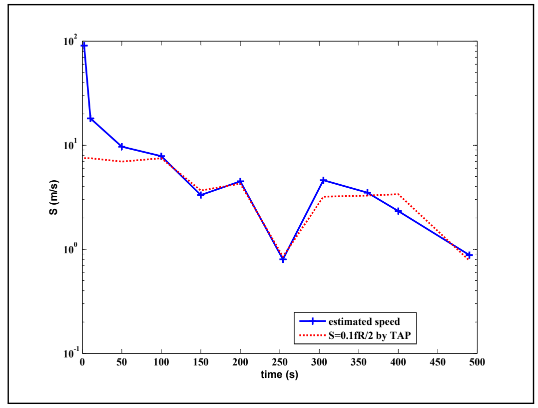

Figure 5: The comparison of speed estimated by 8 and the speed calculated by (1) in TAP

in a very short time which results in a steep rise of *turnover*, thus S is much higher than the following stable periods. The results correlate well between both methods when the network enters relative stable state. However the speed that calculated from TAP is much smaller than directly computed through (8) at the beginning, as there is a constraint to adapt the f_{HELLO} within a certain range which results TAP not as sensitive as (8) to the dynamic changes in the network.

3.2 Minimizing Energy Consumption

The energy spent in period of time Δt by a node u can be expressed as the number of sent messages sent in Δt multiplied by the energy cost of sending one message. The number of messages sent by u during Δt is equal to $\Delta t \cdot f_u(R_u, t)$ where $f_u(R_u, t)$ is the HELLO frequency of node u and R_u is the communication range of node u at time t. Energy consumption varies with both HELLO frequency and transmission range of sensor nodes. In this work, the energy consumption model of sending a HELLO message employs the one given by [2]:

$$E(R) = R(t)^A + C$$

where $A(> 1)$ represents the signal attenuation coefficient along the distance, C is the overhead due to signal processing. A and C are constant as long as the deployed environment is homogeneous and the lengths of HELLO messages are identical. Based on the model, the energy spent by a sensor node on sending HELLO message during Δt is:

$$cost_{\Delta t}(R) = (R^A + C)f\Delta t \tag{9}$$

by substituting (1) into (9), we have:

$$cost_{\Delta t}(R) = (R^A + C)\frac{2S}{\alpha R}\Delta t \tag{10}$$

The objective becomes finding R_{opt} for different S that gives the minimum value of $cost_{\Delta t}$, the solution is obtained by solving the partial derivative of (10) when it equals 0:

$$\frac{\partial cost_{\Delta t}(R)}{\partial R} = [(A-1)R^{A-2} - \frac{C}{R^2}]\frac{2S}{\alpha}\Delta t = 0 \tag{11}$$

therefore we have:

$$R_{opt} = \sqrt[A]{\frac{C}{A-1}} \tag{12}$$

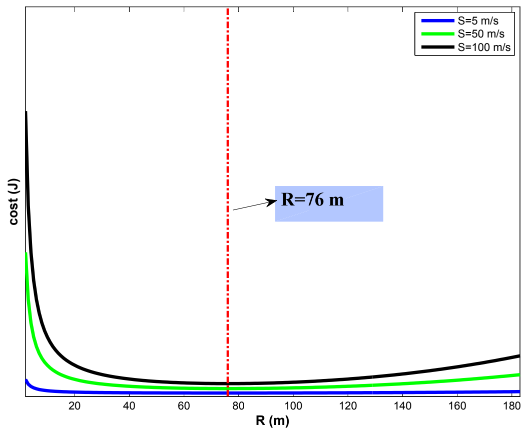

Figure 6: Curves of $cost(R)$ for different speeds

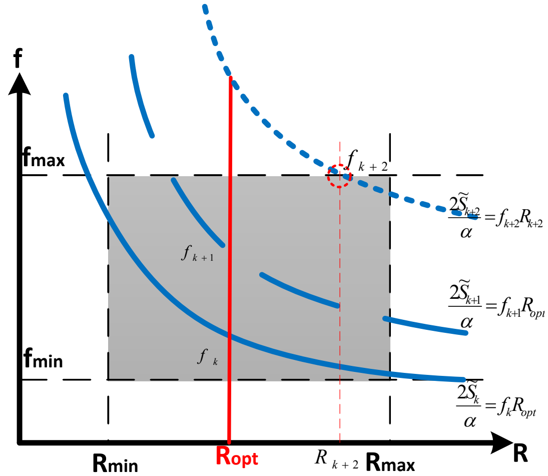

Figure 7: The example of valid region and curves of 1 for different S

Because the second derivative of $cost_{\Delta t}$ is

$$\frac{\partial^2 cost_{\Delta t}(R)}{\partial R^2} = [(A-1)(A-2)R^{A-3} + \frac{2C}{R^3}]\frac{2S}{\alpha}\Delta t \tag{13}$$

When $A \geq 0$, the value of $\frac{\partial^2 cost_{\Delta t}(R)}{\partial R^2}|_{R=R_{opt}} > 0$. Since A is always larger than 1, it is guaranteed that $cost_{\Delta t}(R)$ reaches the global minimum value at R_{opt}. Fig. 6 shows the variations of energy consumption with R under three different speed levels. With $C = 2.25 \times 10^4$ and $A = 2$, the R_{opt} is equal to 150 m, and as expected, the minimum energy cost of each speed level is obtained when $R = R_{opt}$.

We notice from 12 that R_{opt} only varies with C and A, and it is independent of S, which concludes that theoretically all the sensor nodes in the same network should keep transmission range as close as possible to R_{opt} in order to minimize the energy cost and maintain high accuracy of neighborhood table simultaneously.

4. THE TFPA ALGORITHM

As discussed in the previous section, all the sensor nodes are supposed to transmit the HELLO messages with communication range close to R_{opt} in order to minimize the energy cost. However, due to the constraint on hardware device, there exists a maximum transmission range (R_{max}), and for the sake of reliable communication there is a minimum transmission range (R_{min}). Therefore, before implementing the algorithm, the value of R_{opt} should be checked. If $R_{opt} \in [R_{min}, R_{max}]$ the TFPA algorithm can be implemented directly, otherwise, if $R_{opt} \notin [R_{min}, R_{max}]$ the global minimum *cost* is not achievable. When $R_{opt} < R_{min}$, the local minimum *cost* is obtained at R_{min}, we set $R_{opt} = R_{min}$. If $R_{opt} > R_{max}$, the local minimum *cost* is obtained at R_{max}, thus $R_{opt} = R_{max}$. After truncating R_{opt} to the valid region, the TFPA algorithm can be implemented.

It should be noticed that, the change of R can be realized by modifying transmission power of antenna. There are several models for wireless communications such as free-space model, log-normal model, indoor multi-wall model and so on. Through computing the minimum transmit power to reach the computed R, the adaptation of TFPA is easily obtained. Besides, the f_{HELLO} also has an adaptation range: $[f_{min}, f_{max}]$. Fig. 7 demonstrates the valid region (gray rectangle) in which the f_{HELLO} and R can be adjusted. The curves of (1) for different S are shown in the figure as well, and the adaptations from time t_k to t_{k+2} will be explained later.

In the previous section we indeed showed that the *turnover* may be very small (e.g., $r = 0.04$ when $\alpha = 0.1$), while it is nearly impossible for a node to practically observe such a small *turnover* between two successive HELLO messages. A solution to this problem is to let nodes archive more than one table into a history of size X: if X is sufficiently large, then a correct value may be expected. The turnover will then be computed by counting neighbors present in the most recent table that are not present in the older ones and by using the current HELLO delay as:

$$r = \frac{\text{nb of new neighbor}}{\text{total nb of neighbors}} \cdot \frac{d_{HELLO}}{\Delta t} \tag{14}$$

The implementation of TFPA algorithm is consist of two steps: *Training step* and *Adaptation step*.

In the training step, the procedure is run for $X = 10$ periods at each node to build up the historic data about neighbors which is similar as the TAP and NoTAP algorithms. It also converges the frequency and communication range to a certain level and then serves better for the adaptation step.

The adjustment of f_{HELLO} is calculated through the period between two HELLO messages d_{HELLO}, where $f_{HELLO} = \frac{1}{d_{HELLO}}$:

$$d_{HELLO} = \begin{cases} d_{HELLO} + \frac{d_{HELLO}}{factor} \cdot g(r) & \text{if } r < r_{opt} \\ d_{HELLO} - \frac{d_{HELLO}}{factor} \cdot g(r) & otherwise \end{cases} \tag{15}$$

Function $g(r)$ is retrieved using turnover to reflect the distance between r and r_{opt}:

$$g(r) = \begin{cases} (\frac{r - r_{opt}}{r_{opt}})^2 & \text{if } r < 2 \cdot r_{opt} \\ 1 & otherwise \end{cases} \tag{16}$$

Hence, the maximum change of d_{HELLO} at each adjustment is limited by the value of $\frac{d_{HELLO}}{factor}$, $factor > 0$. The pseudo code of *training step* is shown as follow:

TFPA: Training step

```
while X < 10 do
      Calculate r
      if r < r_opt
            Decrease f_HELLO by (15)
      else if r > r_opt then
            Augment f_HELLO by (15)
      end if
      X++
end while
```

The *Adaptation step* is run after finishing the *training step*. At each adaptation interval, each node computes the relative speed S from current turnover ratio r by using (8). The f_{HELLO} and R are then adjusted directly by (1) within the valid region. An example is shown in Fig. 7, at time t_{k+1}, the estimated speed $\widetilde{S}_{k+1}$ is higher than $\widetilde{S_k}$, the curve intersects with $R_{opt} = 150\ m$ in the valid region, thus R_{k+1} is unchanged and equals R_{opt}; while at t_{k+2}, the $\widetilde{S}_{k+2}$ is too high that the intersection with $R_{opt} = 150\ m$ is outside the valid region. However there is still a small part of the curve in the valid region, and the minimum cost will be obtained at R_{k+2} according to the theoretical analysis, therefore f_{k+2} is assigned with the maximum value f_{max}. The pseudo code of *Adaptation step* is shown below:

TFPA: Adaptation step

```
if time to send HELLO packet do
      Calculate r
      Calculate S by (8)
      if H~_optmin <= H~ = 2S/α <= H~_optmax do
            R = R_opt
            f_HELLO = H~/R
      else if H~ > H~_optmax do
            f_HELLO = f_max
            R = min(H~/f_HELLO, R_max)
      else if H~ < H~_optmax do
            f_HELLO = f_min
            R = max(H~/f_HELLO, R_min)
      end if
end if
```

Note that $\widetilde{H}_{optmax} = f_{max}R_{opt}$ and $\widetilde{H}_{optmin} = f_{min}R_{opt}$.

5. PERFORMANCE EVALUATION

In this section, the proposed TFPA algorithm is evaluated through simulation by using the WSNET[1] simulator. Because the purpose of having a HELLO algorithm is for neighborhood discovery, the algorithm must be able to keep

[1]http://wsnet.gforge.inria.fr/

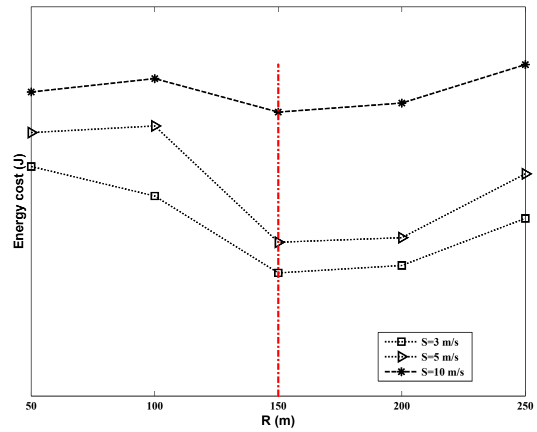

Figure 8: Validate the minimum cost curve

the consistency of neighborhood tables among nodes with minimum cost. Thus in addition to HELLO frequency and power consumption, we use two evaluation metrics: *neighborhood accuracy* and *neighborhood error*. Assuming that $N(u)$ is the set of actual neighbors of a node u, and $N'(u)$ the set of neighbors known to u (i.e. whose identifier is present in its neighborhood table), these two metrics are defined below. Notice that $acc(u)+err(u)$ is not necessarily equal to 1.

Definition 1. Neighborhood accuracy $acc(u)$ is the proportion of actual neighbors of node u that have been indeed detected by u.

$$acc(u) = \frac{|N(u) \bigcap N'(u)|}{|N'(U)|} \times 100.$$

Definition 2. Neighborhood error $err(u)$ measures both how many neighbors of node u have not been detected, and how many "false neighbors" remain in its neighborhood table (i.e. old neighbors that have not been removed).

$$err(u) = \frac{|N(u) \setminus N'(u)| + |N'(u) \setminus N(u)|}{|N(u)|} \times 100.$$

First of all, the minimum cost model stated in Sec.3.2 is validated. In the simulation, 100 nodes were randomly distributed in a square area of size $1000\ m \times 1000\ m$ and the maximum speed of nodes is varied with 3 levels: $0 \sim 3\ m/s,\ 0 \sim 5\ m/s\ and\ 0 \sim 10\ m/s$. The propagation employs the free-space model, and $A = 2$, $C = 2.25 \times 10^4$, therefore $R_{opt} = 150\ m$ by substituting A and C in (12) and $\lambda = \frac{100}{1000 \times 1000}$. Fig. 8 shows the energy consumption by the TFPA protocol. For each level, different $R'_{opt} \in [50\ m, 250\ m]$ are tested. Simulation lasts $100\ s$ for each R'_{opt}. As can be seen, the energy cost at $R_{opt} = 150\ m$ is the minimum for all the mobility level and the curve varies similarly as Fig. 6, therefore the theoretical analysis on the R_{opt} is proved.

To evaluate the performance of the TFPA algorithm, we chose to compare it to four other comparable schemes: TAP algorithm[3], NoTAP algorithm[9], Fopt algorithm[11] and ARH algorithm [6]. The TFPA, TAP and NoTAP do not

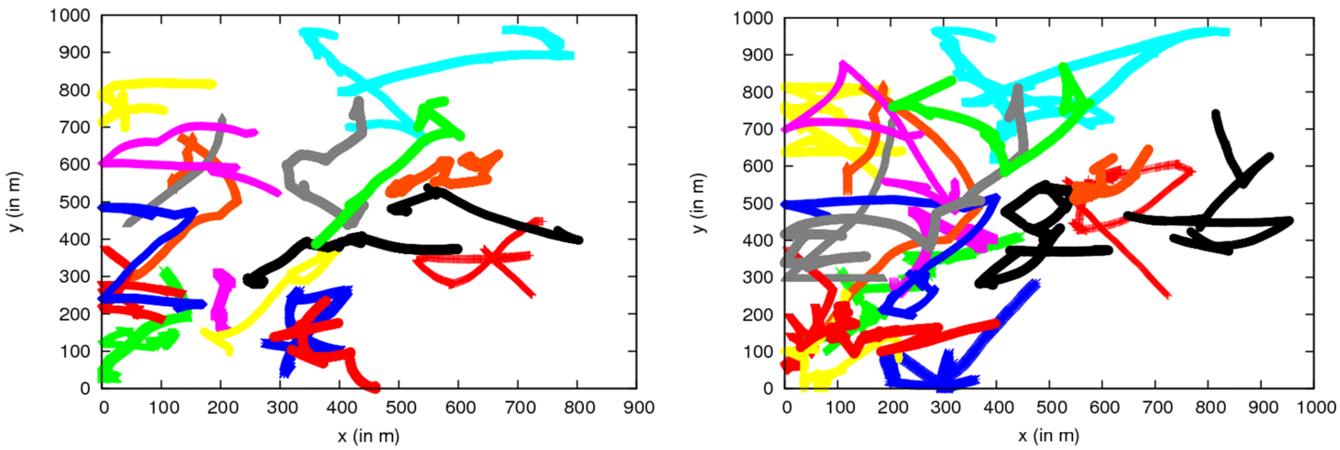

Figure 9: Real mobility trace from pedestrian runners (two examples)

require nodes equipped with GPS. While Fopt sets f_{HELLO} based on the current speed and the ARH sends HELLO message when the error between the prediction of location and the real location is greater than a threshold, thus GPS or a localization module is needed for both Fopt and ARH algorithms.

The nodal mobility trace are generated based on log files obtained from real experiments on pedestrian runners[2]. The moving speed of each node was spread around a mean value of 3 m/s. Two examples of the mobility trace are shown in Fig. 9, where one curve stands for one runner. A varying number of nodes (from 100 to 600) were deployed in a 1000 $m \times 1000$ m square region. These nodes have the same optimal transmission range $R_{opt} = 100$ m and $R_{min} = 50$ m, $R_{max} = 150$ m. For each network size, we conducted 100 simulation runs and obtain the average results with 95% confidence intervals.

Fig. 10(a) and Fig. 10(b) show the accuracy and error of the neighborhood table respectively. The TFPA algorithm performs slightly better than the TAP, Fopt and ARH, but slightly worse than NoTAP.

However, if take a look at Fig. 10(c) which compares the value of f_{HELLO} with regards to different number of nodes, we observe that the HELLO messages sent by TFPA and ARH algorithms are much less than the others. The TFPA performs better than ARH when node density is lower, while ARH performs slightly better as the number of node increases. However, in average, the f_{HELLO} of ARH is about 11% higher than TFPA. The f_{HELLO} of Fopt is 90% higher than that of TFPA, the NoTAP and the TAP are 80% and 60% higher than the f_{HELLO} of TFPA respectively. Bost of the Fopt and ARH algorithms perform consistently, but with different explanations: For Fopt algorithm, the average speed of each node is almost constant thus the average f_{HELLO} is almost constant; for ARH, the HELLO frequency depends on the error that a node detects between its position estimate and its real position, which are both independent from the number of neighbors and the total number of nodes. The curves of TFPA, TAP and NoTAP algorithms have similar variation trend and they climb slightly as the node density increases, which indicates that the dynamic of surrounded neighbors of each sensor node increases.

Fig. 10(d) demonstrates the *turnover* for all the methods except the ARH (which can not explain the *turnover*). As expected, all the turnover ratios are close to 0.04 which is in coincident with the theoretical analysis. The Fopt algorithm provides highest f_{HELLO}, thus the turnover is the lowest. The TFPA gives the lowest f_{HELLO}, thus the turnover is the highest. However, the turnover ratio of Fopt decreases as it only focuses on the absolute speed of each node and ignores the relative change on the neighbors. On the contrary, the turnovers of the TFPA, TAP and NoTAP are relatively stable because again they take into account the mobilities of the neighbors.

The remaining energy is shown in Fig. 10(e). Although the f_{HELLO} of NoTAP algorithm is less than that of Fopt, its energy consumption is the highest. Since Fopt sends HELLO messages always with R_{opt} and NoTAP intends to adapt R to achieve high accuracy, the reduction on the f_{HELLO} can not compensate the consumption on modifying the transmission range, hence the overall energy cost is at the end the highest compared to others. Meanwhile, we observe that the TFPA costs least energy among the algorithms. This can be explained by the factor that in addition to the lowest f_{HELLO}, the TFPA only adapts R when the computed f_{HELLO} reaches the boundary of valid region, while the NoTAP algorithm adapts R whenever f_{HELLO} changes. Therefore, with the lowest f_{HELLO} and quasi global optimum R, the TFPA algorithm is able to save more energy than the others.

The average speed of the WSN is also obtained, Fig. 10(f) shows that the estimated result is slightly higher than the real average speed with a mean error of 0.67 m/s. Therefore, without any GPS-like devices, the TFPA algorithm estimates properly the dynamic of the WSN which can be employed on other implementations that also require speed information.

6. CONCLUSION AND FUTURE WORKS

In this paper, a TFPA HELLO algorithm is proposed for mobile ad hoc networks. The algorithm relies on appropriate theoretical analysis in which the expression of turnover ratio

[2]http://researchers.lille.inria.fr/∼mitton/mobilitylog.html

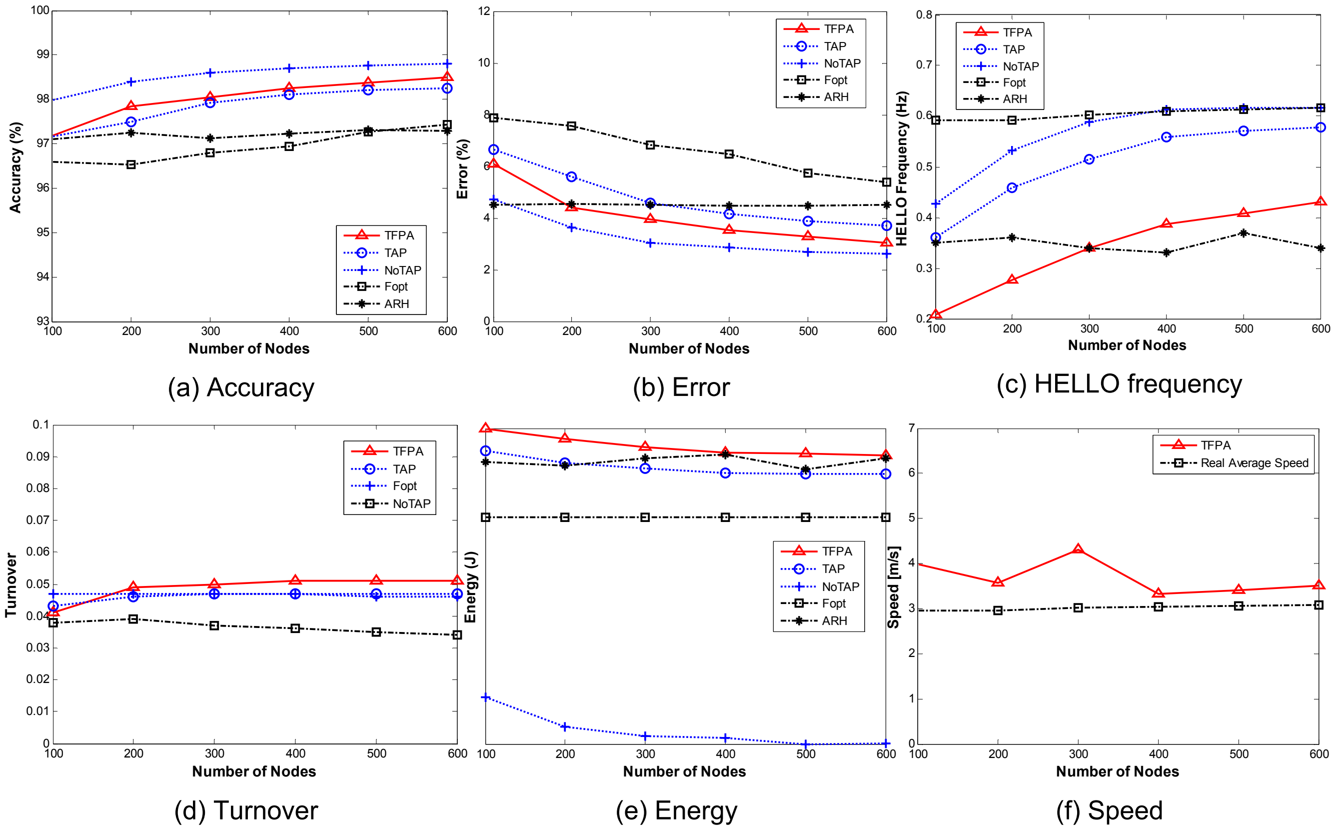

Figure 10: Different metrics of the WSN vary with number of nodes

r is approximated properly. By taking account of the relative change of neighborhoods, the relationship between r and relative speed S is derived which enables the sensor nodes be aware of the dynamic of their environment without using GPS-like devices and hardware cost is saved. Moreover, the optimal transmission range is deduced from the battery consumption model which makes the adaptation procedure energy efficient. The simulation results demonstrate that the algorithm is able to maintain the neighborhood table accurate with low HELLO frequency (comparable with and even lower than GPS-needed algorithm) and lowest energy consumption.

There are some open issues, such as the battery consumption modeling, as different models result in different R_{opt} with similar deduce method; the propagation model could be more practical to adapt the transmission power. We would like to further study the consequences of these more realistic assumptions and adapt TFPA protocol consequently.

7. REFERENCES

[1] B. Clark, C. Colbourn, and D. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1-3):165–177, 1990.

[2] E. Fleury and D. Simplot-Ryl. *Réseaux de capteurs*. Hermes Science - Lavoisier, 2009.

[3] F. Ingelrest, N. Mitton, and D. Simplot-Ryl. A turnover based adaptive hello protocol for mobile ad hoc and sensor networks. In *MASCOTS*, pages 9–14, 2007.

[4] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks, 2001.

[5] N. Li, J. C. Hou, and L. Sha. Design and analysis of an mst-based topology control algorithm. *IEEE Transactions on Wireless Communications*, 4(3):1195–1206, 2005.

[6] X. Li, N. Mitton, and D. Simplot-Ryl. Mobility prediction based neighborhood discovery in mobile ad hoc networks. In *Networking (1)*, pages 241–253, 2011.

[7] V. Loscrì, E. Natalizio, and C. Costanzo. Simulations of the impact of controlled mobility for routing protocols. *EURASIP J. Wireless Comm. and Networking*, 2010, 2010.

[8] J. Moy. *OSPF - Open Shortest Path First*. RFC 1583, March 1994.

[9] J. Radak and N. Mitton. Transmission range adaptation based energy efficient neighborhood discovery. In *MSWiM12*, 2012.

[10] J. Radak, N. Mitton, and D. Simplot-Ryl. Using battery level as metric for graph planarization. In *ADHOC-NOW*, pages 58–71, 2011.

[11] A. Troël. *Prise en compte de la mobilité dans les interactions de proximité entre terminaux à profils hétérogènes*. PhD thesis, Universite de Rennes, 2004.