# Accelerating Initialization for Sensor Networks

Linghe Kong, Luoyi Fu, Xuemei Liu and Min-You Wu
Shanghai Jiao Tong University, China
{linghe.kong, yiluofu, xuemeiliu, mwu}@sjtu.edu.cn

***Abstract*—Initialization is an indispensable process for wireless sensor networks since newly deployed sensors lack a reliable infrastructure for communication. Many efforts have been made to address the problem on initialization. However, most of them fall into the category of energy efficiency. In mission-critical scenarios, it is vital to reduce the initialization time in order to gather information as early as possible. In this paper, we study a new problem of accelerating initialization. First, we provide a novel parallel initialization mechanism that selects cluster heads before deployment and organizes single-hop clusters simultaneously after random deployment. Then, we prior-estimate that the number of cluster head candidates from $n$ sensors is $\Theta(\log n)$. Finally, simulation results show that these cluster head candidates can achieve the parallel initialization bounded by 16 time slots even for a large $n$.**

***Keywords*—Parallel initialization, Cluster, Prior-estimation**

## I. Introduction

Wireless Sensor Networks (WSN) [1] in real application has attracted lots of interest on researches due to its great potential. Generally, newly deployed sensors have no additional network infrastructure, so they cannot transmit data before establishing a communication scheme themselves. The process of setting up such a connected WSN infrastructure is called initialization. Our study focuses on the problem on how to accelerate the initialization process.

The rapid initialization plays a major role in practice due to various reasons. For instance, timely response is one of the most important factors in mission-critical networking, such as military surveillance and disaster relief. Obviously, the initialization process should be as fast as possible so that the WSN can start to carry out immediately. Moreover, energy waste is a critical issue. Existing study [4] shows that a large fraction of battery is already used during initialization. Current mechanisms tend to be too slow to solve these problems.

There is a multitude of excellent protocols on WSN in literature, which are designed for structuring networks. Kuhn *et al.* [4] provide an algorithm that can efficiently compute an asymptotically optimal clustering. McGlynn *et al.* put forward Birthday protocol [5] for discovering complete neighbors' information during the process. NoSE protocol [8] is proposed to take link quality estimation into consideration by Meier *et al.* Ribeiro [11] gives a robust self-initialization using whispering to avoid intruders. Nakano *et al.* [10] propose the protocol for single-hop radio networks with no collision detection. Moscibroda *et al.* [9] discussed the division of a WSN lifetime into phases. But the topic of accelerating initialization in WSN is still a vacancy.

Although state-of-the-art studies do not focus on accelerating initialization, clustering [13] notion in them is also tailor-made for the purpose of establishing a WSN structure rapidly. Thus, we propose to divide sensors into clusters and make the clusters self-initialize in a parallel pattern. Intuitively, this proposal can reduce the duration intuitionally, but there are still some challenges remained.

First, sensors have no topology information before the network is initialized. However, traditional clustering algorithms usually execute based on neighbors' information. This dilemma makes it difficult to cluster sensors in absence of an established structure. Second, transmission collision is another common problem. Plentiful messages exchange is required during the initialization, but this, to a certain extent, brings about frequent collisions, which will then cause severe delay of initialization. So it is essential to design a scheduling scheme for reducing collisions. Third, compared to the lifetime of sensors, the initialization phase is relatively minor. The realization of a complex algorithm is at the cost of redundant time. Hence, a simple and distributed initialization mechanism is preferred in WSN.

As is discussed above, we propose a novel mechanism called parallel initialization (PI) to accelerate the initialization phase. In order to save the time for electing cluster heads (CH), PI selects CH candidates before deployment. We prior-estimate that the number of CHs is $C\frac{(\log n)\cdot a^2}{r^2}$. Then we introduce a redundancy coefficient $\beta$ for selecting $\beta C\frac{(\log n)\cdot a^2}{r^2}$ CH candidates to increase the connection probability. After stochastic deployment, CH candidates organize single-hop clusters according to PI. Performance evaluation shows that PI can finish the initialization in 8-16 time slots even $n$ becomes large.

To our best knowledge, the concept of accelerating initialization for WSN has not been studied before. The main contributions of this paper are summarized as follows:

- We first motivate and formulate the accelerating initialization problem in WSN.
- We introduce the method of parallel initializing the single-hop clustering WSN.
- We derive the number of CHs using prior-estimation and provide the corresponding explicit formulas.
- We design a simple PI mechanism that can sharply accelerate the initialization duration.

The rest of the paper is organized as follows. In Section II, we define the accelerating initialization problem and present a sensor model. Detail PI mechanism is described in Section III. In Section IV, we present the PI algorithm. In Section V, we evaluate the PI performance. We conclude and discuss our future work in Section VI.

## II. Initialization Issues

### A. Definition

The lifetime of a WSN is normally divided into four phases as shown in Fig.1. Before a WSN can start its operation

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

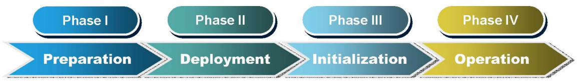
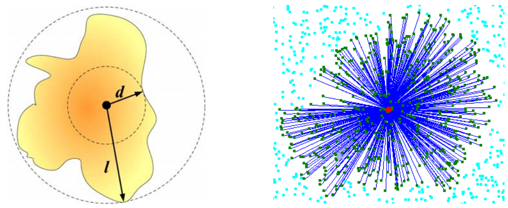



Fig. 1. The lifetime phases of WSN




(a) (b)

Fig. 2. Quasi Disk Model: $r$ varying between $d$ and $l$

phase, sensors must establish an efficient structure that provides reliable point-to-point connections. We define that the initialization phase begins when all sensors are installed and a wake-up beacon is sent, and ends when sensors gather enough topology information to apply the routing protocol.

Accelerating initialization aims to shorten the duration of phase III. The objective can be formulated as min($T_{Init}$).

Actually, the initialization phase is a process of message exchange. Sensors are required to transmit their node identifier (ID). We assume that transmitting one ID message needs one time slot $t$. So $T_{Init}$ can be combined with $x$ time slot $t$:

$$T_{Init} = \Sigma_{i=1}^{x} t_i, x \in N^+.$$

Then the accelerating initialization min($T_{Init}$) can be treated as reducing the quantity of time slots min($x$) and shortening the duration of each time slot min($t$).

### *B. Model and Assumption*

Disk model [12] is widely employed for the study of sensor communication, which assumes that all the sensors have the same transmission range in the plane. This model is a simplification of reality. Since even for homogeneous sensors, the transmission range is affected by the factors such as obstruct signal, reception power, etc. However, Quasi Disk model [5] is significantly closer to reality. In this model, two sensors are connected if their Euclidean distance is less than $d$; in the range between $d$ and $l$, the existence of a connection is unspecified as shown in Fig. 2(a). For the remainder of this paper, we will adopt Disk model in analysis for readability and Quasi Disk model in simulation for approaching reality as Fig. 2(b).

We assume that the total number of sensors used in the network is denoted by $n$. Homogenous $n$ sensors are deployed randomly into the monitoring field and they follow a uniform distribution. Newly released sensors have no knowledge of their neighbors' information. All sensors stay in the sleep state for the sake of low power until they receive the wake-up beacon. Sensors have only one communication channel. The monitoring area is assumed to be a square area in the plane $\mathcal{R}^2$, whose length of side is denoted by $a$.

## III. Parallel Initialization Mechanism

In this Section, we analyze some problems of accelerating initialization first. Then a PI mechanism is put forward. And we provide theoretical derivation about PI finally.

### *A. Problems Analysis*

**Clustering:** The reason for dividing sensors into clusters is that parallel pattern can accelerate initialization efficiently. In order to enhance the parallelism as much as possible, we adopt the single-hop clustering scheme [6] for min($x$). Compared to multi-hops cluster, formatting each single-hop cluster is apparently faster. As shown in Fig. 3, CHs are connected through Gateways (GW), so the distance between any two CHs is $d(CH_1, CH_2) < 2r$. In addition, if $d(CH_1, CH_2) \leq r$, two CHs interfere mutually when transmitting messages at the same $t$. So we adopt single-hop cluster manner with $r < d(CHs) < 2r$.

Existing algorithms usually cluster a WSN by voting CHs after the topology is established. But how can we cluster without neighbors' information? We determine CH candidates in preparation phase, which also saves the voting time. Hence, the $x$ is decreased further.

An important question arisen from the above proposal is how to estimate CH candidates in advance? Since we know some information such as $n$, $a$ and $r$, we can compute the quantity of CHs that will be used to organize the single-hop clusters. Then we prior-estimate the redundancy of CH candidates due to uncertain position after deployment. Then we upload the CH program to this quantity of sensors. Detail derivation of the quantity and redundancy is given in Section D and E.

**Initialization:** In traditional protocols, huge amounts of time is wasted on redundant messages exchange. WSN can achieve initialization under single-hop cluster scheme on some basis: a sensor knows whether it is a CH, a cluster member (CM) or a GW. As a CM/GW, it must know additionally which CHs it links to. In other words, when sensors have such information, at least one path between any sensors can be found in a connected graph using routing protocol. Since the CH candidates have known their attributes in the preparation phase, the basic demand can be satisfied when ID messages broadcasted by CHs are received by their CMs and GWs.

If all ID information can exchange at the same time, the initialization can be achieved in one time slot. But a sensor in GW position cannot receive different CHs messages simultaneously. Hence, we introduce a few more time slots into the initialization phase. CH candidates randomly and independently choose one of these time slots to broadcast. How many time slots are the best for rapid initializing and few collisions? We will discuss in Section C.

Since CHs are selected in advance, it is possible that some CHs are released close to others. An abdication mechanism is necessary for a guarantee on $r < d(CHs) < 2r$. We propose that when a CH candidate hears other CHs' message before broadcasting its own ID, it abdicate from CH to be a CM/GW.

It is possible that a few sensors are not located in any CHs single-hop range. To increase the robustness of whole networks, a compensation mechanism is essential.

**Time slot size:** In order to obtain min($t$), we define that the size of $t$ is equal to the duration of transmitting a message containing ID information only.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

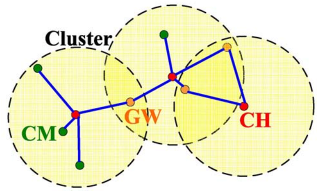



Fig. 3. Single-hop clusters

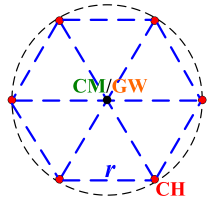



Fig. 4. 6 CHs connectable

### B. PI Mechanism

In this part, we provide the Parallel Initialization Mechanism is provided in this part. The procedures of PI are as follow:

- **Information collection**: In the preparation phase, PI need to collect the information such as the length of the surveillance area $a$; the total number of using sensors $n$; the transmission range $r$ and so on.
- **CH candidates selection**: This procedure includes prior-estimating the quantity of CH candidates, uploading CH program to this quantity sensors from $n$.
- **Deployment**: Sensors are randomly put on the monitoring area. We assume it follows a uniform distribution.
- **Activation**: The wake-up beacon is broadcasted once all over the monitoring area so that sensors can be active synchronously [15].
- **Self-organization:** In $T$ time slots, each CH candidate either organizes single-hop cluster by broadcasting ID information, or becomes a CM/GW by abdication mechanism. Each CM/GW is assigned into different clusters.
- **Compensation mechanism:** If the isolate sensors exist after $T$, they will upgrade into CH candidates and self initialize in additional $\Delta$ time slots.

The initialization phase ends. The $x$ in PI is $x = T + \Delta$. Hence the total time that PI needs to initialize a WSN is:

$$T_{Init} = \Sigma_{i=1}^{x} t_i = (T + \Delta) \times t$$

### C. PI Time Slots

The quantity of PI Time slots is determined by two factors. First, the number should be the fewer the better for accelerating the duration. Second, we should guarantee sensors that are not in the isolated location are connected.

Since the distance between any two CH candidates cannot be smaller than $r$, a CM/GW sensor can at most connect 6 CHs as shown in Fig. 4.A CM/GW is connected to WSN when it receives at least one ID message in a time slot without collisions. With the two conditions, we can compute the connection probability of a CM/GW having all cases (numbered from 2 to 6) of connectable CHs.

Fig. 5 shows the results of probability distribution as the quantity of time slots varies from 2 to 10. It can be seen from the figure that when $x > 3$, the connection probability in any of the curves will converge. When $x > 8$, the probability has little change with the increasing of the time slot. When $x < 3$, most probabilities are less than 80%, which makes CM/GW easily disconnected. In order to guarantee a high connection probability and accelerate the initialization phase, we set $T = 8$ and $\Delta$ can be chosen from 3 to 8.

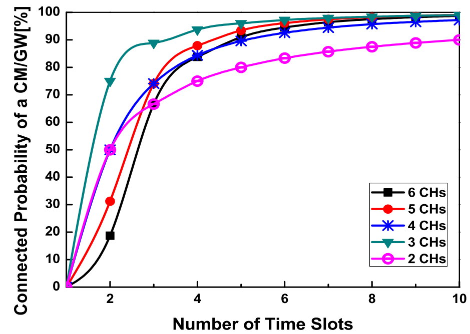



Fig. 5. Connection PDF of a CM/GW in different cases

### D. The Number of CHs

In this section, we will prove how many CHs is sufficient to make a WSN connected by single-hop clustering. We assume that the network consists of $n$ sensors, in which there are $m$ CHs. All sensors are placed randomly, uniformly and independently in the $a^2$ square area. Let $\mathcal{G}(n, m)$ denote the graph of this WSN. And two nodes are connected if their Euclidean distance is at most $r$, then we have:

*Theorem 3.1:* Graph $\mathcal{G}(n, m)$ with $m = C\frac{(\log n)\cdot a^2}{r^2}$ is connected with probability one as $n \to \infty$, where $C$ is a tunable parameter.

*1) Necessary condition on $m$:* In proving the necessary part of Theorem 3.1, we assume that the $a^2$ area is denoted by $\mathcal{D}$ in $\mathcal{R}^2$. We neglect edge effects that arise when a node is close to the boundary of $\mathcal{D}$ for the sake of simplicity.And we define $\gamma = \frac{r}{a}$ as the ratio of $r$ to $a$.

We start with two technical lemmas:

*Lemma 3.1 (Lemma 2.1 in [3]):* For any $p \in [0, 1]$,

$$(1 - p) \leq e^{-p}. \tag{1}$$

*Lemma 3.2:* If $m = \frac{\log n + \varpi}{\pi\gamma^2}$ for any fixed $\alpha < 1$ and for all sufficiently large $n$,

$$n(1 - \pi\gamma^2)^m \geq \alpha e^{-\varepsilon}. \tag{2}$$

*Proof:* Taking the logarithm of the left hand side of Eqn. 2, we get

$$\log(\text{L.H.S of (2)}) = \log n + m\log(1 - \pi\gamma^2). \tag{3}$$

Using the power series expansion for $\log(1 - x)$,

$$\begin{aligned}\log(\text{L.H.S of (2)}) &= \log n - m\Sigma_{i=1}^{\infty}\frac{(\pi\gamma^2)^i}{i} \\ &= \log n - m(\Sigma_{i=1}^{2}\frac{(\pi\gamma^2)^i}{i} + H(n)),\end{aligned} \tag{4}$$

where

$$H(n) = \Sigma_{i=3}^{\infty}\frac{(\pi\gamma^2)^i}{i} \leq \frac{1}{3(\pi\gamma^2)}(\pi\gamma^2)^x|_3^{\infty} = \frac{1}{3}(\pi\gamma^2)^2 \tag{5}$$

for all large $n$. Substituting Eqn. 5 in Eqn. 4, we get

$$\begin{aligned}\log(\text{L.H.S of (2)}) &\geq \log n - m(\pi\gamma^2 + \frac{5}{6}\frac{(\pi\gamma^2)^2}{m^2}) \\ &= -\varpi - \frac{5}{6}\frac{(\pi\gamma^2)^2}{m} = -\varpi - \frac{5}{6}\frac{(\pi\gamma^2)^3}{\log n + \varpi} \\ &\geq -\varpi - \delta\end{aligned} \tag{6}$$

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

for all sufficiently large $n$. Taking the exponent of both sides and using $\alpha = e^{-\delta}$, the results follows. ■

*Theorem 3.2:* If $m = \frac{\log n + \varpi(n)}{\pi\gamma^2}$, then

$$\lim_{n\to\infty} \inf P_d(n,m) \geq e^{-\varpi}(1 - e^{-\varpi}) \qquad (7)$$

where $\varpi = \lim_{n\to\infty} \varpi(n)$.

*Proof:* We first study the case where $m = \frac{\log n + \varpi}{\pi\gamma^2}$ for a fixed $\varpi$. Let $P^{(k)}$, $k = 1, 2, \ldots$ denote the probability that a graph $\mathcal{G}(n,m)$ has at least one order-$k$ component. We have

$$\begin{aligned} &P_d(n,m) \\ &= P^{(1)}(n,m) \\ &\geq \Sigma_{i=1}^{n} P(\{i \text{ is the only isolated CM in } \mathcal{G}(n,m)\}) \\ &\geq \Sigma_{i=1}^{n} (P(\{i \text{ is an isolated CM in } \mathcal{G}(n,m)\}) \\ &\quad - \Sigma_{j\neq i} P(\{i, j \text{ are isolated CMs in } \mathcal{G}(n,m)\})) \\ &\geq \Sigma_{i=1}^{n} (P(\{i \text{ is an isolated CM in } \mathcal{G}(n,m)\}) \\ &\quad - \Sigma_{i=1}^{n} \Sigma_{j\neq i} P(\{i, j \text{ are isolated CMs in } \mathcal{G}(n,m)\}). \end{aligned} \qquad (8)$$

Neglecting edge effects, we get

$$P(\{i \text{ is an isolated CM in } \mathcal{G}(n,m)\}) = (1 - \pi\gamma^2)^m. \qquad (9)$$

Whether a cluster member is isolated is independent from other member nodes. Thus we obtain

$$P(\{i, j \text{ are isolated CMs in } \mathcal{G}(n,m)\}) = (1 - \pi\gamma^2)^{2m}. \qquad (10)$$

Substituting Eqn. 9 and Eqn. 10 in Eqn. 8, we obtain

$$P_d(n,m) \geq n(1-\pi\gamma^2)^m - n(n-1)(1-\pi\gamma^2)^{2m}. \qquad (11)$$

Thus, using Lemma 3.1 and 3.2, for any fixed $\alpha < 1$,

$$\begin{aligned} P_d(n,m) &\geq \alpha e^{-\varpi} - n(n-1)e^{-2m\pi\gamma^2} \\ &\geq \alpha e^{-\varpi} - (1+\delta)e^{-2\varpi} \end{aligned} \qquad (12)$$

for all $n > N(\delta, \alpha, \varpi)$.

Now we consider the case where $\varpi$ is a function of $\varpi(n)$ with $\lim_{n\to\infty} \varpi(n) = \hat{\varpi}$. Then for $n \geq N'(\delta)$ and any $\delta > 0$, $\varpi(n) \geq \hat{\varpi} + \delta$. Considering the probability of disconnectedness is monotone decreasing in $\varpi$, then we have

$$P_d(n,m) \geq \alpha e^{-\hat{\varpi}+\delta} - (1+\delta)e^{-2(-\hat{\varpi}+\delta)}. \qquad (13)$$

Since this holds for all $\delta > 0$and $\alpha < 1$, take limits and then the results follows. ■

As an obvious consequence of Theorem 3.2, we have:

*Corollary 3.1:* Graph $\mathcal{G}(n,m)$ is asymptotically disconnected with positive probability if $m = \frac{\log n + \varpi(n)}{\pi\gamma^2}$ and $\lim_{n\to\infty} \varpi(n) < +\infty$.

Hence we have proved the necessary part of Theorem 3.1.

*2) Sufficient condition on $m$:* Suppose we connect $\kappa\frac{\log n}{\gamma^2}$ CMs/GWs in $\mathcal{G}$ to nodes which are within the range $r$. Denote the resulting graph by $\mathcal{G}(n, \kappa\frac{\log n}{\gamma^2})$. Then it suffices to show that for some $\kappa > 0$,

$$\lim_{n\to\infty} P(\{\mathcal{G}(n, \kappa\frac{\log n}{\gamma^2}) \text{ is connected}\}) = 1.$$

*Tessellation*: For the simplicity of proof of the sufficient condition, we assume in this section that the area where all the nodes are placed, is a square $\mathcal{S}$ in $\mathcal{R}^2$. Then we divide it

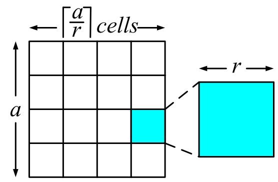



Fig. 6. Tessellation of the square into cells

equally into $S_n = \lceil\frac{a}{r}\rceil^2$ square cells as shown in Fig. 6, where $\lceil x \rceil$ is the cell function that returns the smallest integer larger than or equal to $x$. This tessellation of the unit square will be denoted by $\mathcal{T}_n^s$. Here we treat $\frac{a}{r}$ as an integer for the sake of clarity of presentation. The error of this approximation can be ignored since when $a$ is sufficiently large compared to $r$, the fraction of $\frac{a}{r}$ is can be neglected.

*Proof of sufficient part of Theorem 3.1*: First, we consider the number of CHs in each cell.

*Lemma 3.3:* Suppose we tessellate the square by $\mathcal{T}_n^s$. Then each cell holds at least one CH node with probability one as $n \to \infty$.

*Proof:* Let $E_i$, $i = 1, 2, \ldots, S_n$, be the event that a particular CH node $N_i$, $1 \leq i \leq m$ falls into a particular cell with probability $P(E_i) = \frac{1}{S_n} = (\frac{r}{a})^2 = \gamma^2$. So the probability that a particular cell has no CH is $(1 - P(E_i))^m$. Then we consider the probability that at least one cell is empty, and by the union bound, we have

$$P(\cup_{i=1}^{S_n} E_i) \leq \Sigma_{i=1}^{S_n} P(E_i) = \frac{1}{\gamma^2}(1-\gamma^2)^m. \qquad (14)$$

Applying $(1-p) \leq e^{-p}$, we obtain

$$P(\cup_{i=1}^{S_n} E_i) \leq \frac{1}{\gamma^2} e^{-\gamma^2 m} = \frac{1}{\gamma^2} e^{-\kappa \log n} = \frac{1}{\gamma^2 n^\kappa}. \qquad (15)$$

Then the result follows. ■

*3) Estimation on the bound of $C$:* Since we have proved that given a specific $r$, the number of CH is $m = C\frac{(\log n)\cdot a^2}{r^2}$, where $C$ is a tunable parameter, we will then proceed to estimate the upper bound and lower bound of $C$.

There exist such $C_1$ and $C_2$ , where $0 < C_1 < C_2$, that the optimal number of CH will be no less than $C_1\frac{(\log n)\cdot a^2}{r^2}$ and no more than $C_2\frac{(\log n)\cdot a^2}{r^2}$. According to the criteria that $r < d(CHs) < 2r$, so $\lceil\frac{a}{r}\rceil^2$ CHs will be needed in the worst case and $\lceil\frac{a}{2r}\rceil^2$ CHs will be needed in the best case.

We estimate that the optimal number of CHs yields to Guassian distribution where $\lceil\frac{a}{2r}\rceil^2$ and $\lceil\frac{a}{r}\rceil^2$ act as two 3-delta threholds. Thus, there exists a value which maximizes the distribution. It is the average of the two 3-delta thresholds.

Substituting these values into $C\frac{r^2}{(\log n)\cdot a^2}$, we get

$$C_1 = \frac{r^2}{(\log n)\cdot a^2}\lceil\frac{a}{2r}\rceil^2, \quad C_2 = \frac{r^2}{(\log n)\cdot a^2}\lceil\frac{a}{r}\rceil^2.$$

### E. Prior-Estimate the Redundancy of CH Candidates

We have derived that $C\frac{(\log n)\cdot a^2}{r^2}$ CHs can make WSN connected by single-hop clustering. However, if we only select such sensors to be CH candidates in the preparation phase, we cannot guarantee that they can be placed in the right position and meet no abdication actions. In such a case, $C\frac{(\log n)\cdot a^2}{r^2}$ CH candidates are not sufficient to make all sensors connected in a randomly deployed WSN.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

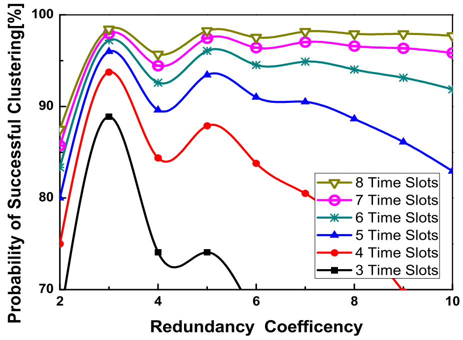



Fig. 7. PDF of achieving clusters by different $\beta$

Hence, we introduce a coefficient of redundancy $\beta(\beta > 1)$ into CH candidates selecting. More CHs can increase the connection probability of the whole WSN but meanwhile bring about more collisions. Thus the value of $\beta$ depends on both the requirement of connection probability and the quantity of time slots in PI. Hence the prior-estimation of the number of CH candidates is:

$$m = \beta C \frac{(\log n) \cdot a^2}{r^2}$$

The redundancy coefficient $\beta$ can be regarded as the number of CH candidates in a cell in Fig. 6. So the probability of clustering a whole WSN by all CH candidates can be simply treated as the probability of clustering a cell by $\beta$ CH candidates. It is equivalent to the probability that at least one of the $\beta$ CH candidates can broadcast its ID in a certain time slot without collisions.

The result is shown in Fig. 7. We can find that the highest probability is always in the $\beta = 3$ as time slots varying from 3 to 8. Thus, we set $\beta = 3$ in our PI algorithm.

## IV. PI Algorithm

In this section we will present our PI algorithm. We denote $x$ or $y$ as the ID number of a sensor; $A \rightarrow B$ means that a sensor changes its attribute from A to B. The part below the dashed line in the algorithm is the compensation mechanism.

**Procedure 1** PI_CH($x$)_Algorithm $(T, \Delta)$

1: Set $k$=random$(0, T)$
2: Listening during $k$ time slots{
3: **If** CH($x$) hears an ID messages from CH($i$) ($i \in N, i \neq x$)
4: CH($x$)$\rightarrow$ CM($x$) in cluster $i$
5: **If** CH($x$) hears more ID messages from CHs($i_1, i_2, i_3, \ldots$)
6: CM($x$), GW($x$)$\rightarrow$GW($x$) among cluster $i, i_1, i_2, i_3, \ldots$
7: }
8: **If** CH($x$) hears no message in $k$
9: CH($x$) broadcasts ID($x$) message and **Quit** this algorithm
10:Listening $(T - k)$ time slots
11: execute line 5 to 6
– – – – – – – – – – – – – – – – – –
12:Listening $(\Delta)$ time slots
13: execute line 5 to 6

When a CH candidate is active by the wake-up beacon, it executes the algorithm as shown in Procedure1. It randomly listens $k$ $(k < T)$ time slots first. If no message is received, it then broadcasts its ID message and exits this program directly; if it receives messages from other CHs during $k$ time slots, it abdicates to be a CM/GW. As a CM/GW, it should work at the state of listening messages in the rest of the $(T - k)$ time slots and the compensation $\Delta$ time slots as well.

**Procedure 2** PI_CM/GW($y$)_Algorithm $(T, \Delta)$

1: Listening during $T$ time slots{
2: **If** CM/GW($y$) hears an ID messages from CH($j$)
3: CM/GW($y$)$\rightarrow$ CM($y$) in cluster $j$
4: **If** CM($y$) hears more ID messages from CHs($j_1, j_2, j_3, \ldots$)
5: CM($x$), GW($x$)$\rightarrow$GW($x$) among cluster $j, j_1, j_2, j_3, \ldots$
6: }
– – – – – – – – – – – – – – – – – –
7: **If** CM/GW($y$) hears no message in $T$
8: CM/GW($y$)$\rightarrow$ CM($y$)
9: execute line 1 to 11 of PI_CH($y$)_Algorithm $(\Delta, 0)$

A CM/GW starts the algorithm as shown in Procedure 2. It keeps listening in $T$ time slots. When it receives one ID message, it is determined as a CM in the cluster of this CH. When it receives more messages from different CHs, it becomes a GW connecting the clusters of these CHs. If a CM/GW hears no message in $T$ time slots, it means that this sensor is still an isolated one in WSN. Then it should be promoted to become a CH candidate in compensation mechanism and execute the PI_CH($y$)_Algorithm in $\Delta$ time slots.

## V. Performance Evaluation

In this section, we will give an indication of the PI mechanism on a variety of test settings. In our simulation, $n$ sensors are randomly distributed in an $800 \times 800$ square. The transmission range of each sensor is varying from 120 to 200. In PI algorithm, we use the result of time slots in Section III C, so $T = 8$ and $\Delta = 8$.

Fig. 8 shows a topology graph of $n = 1000$ WSN by PI with $CHs = 21$ and $\beta = 3$, which is computed as Section III D and E. CHs (Red) connects CMs(Green) and GWs(Yellow) by blue line. There are no isolated sensors in the field.

Fig. 9 shows the probability distribution of the number of clusters generated by PI through statistic of 1000 times'

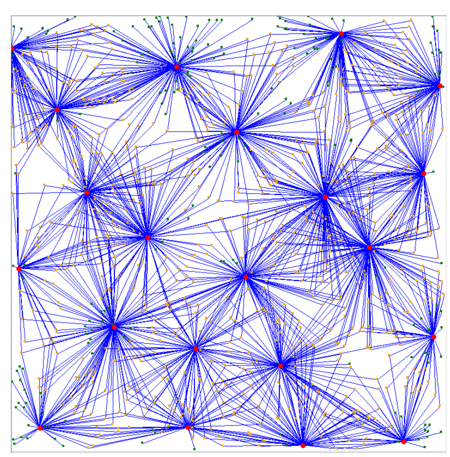


Fig. 8. WSN topology using PI Mechanism


This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

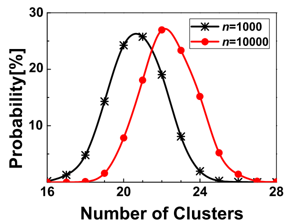



Fig. 9. PDF of the number of clusters by PI through 1000 times iteration

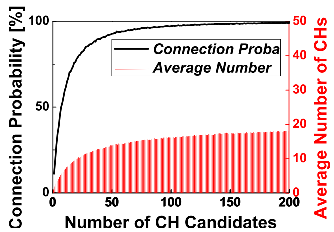



Fig. 10. PI results by different prior-selection number of CH candidates

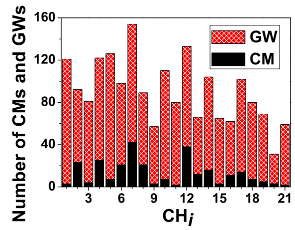



Fig. 11. The number of CMs and GWs of every CH in Fig.8

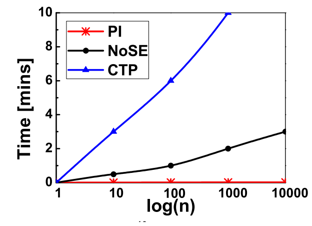



Fig. 12. Initialization time with $n$ sensors by PI, NoSE and CTP

iteration when $n = 1000$ and 10000 respectively. We can find that both the probabilities follow the Guassian distribution. It can also be seen that the number of CHs is $\Theta(\log n)$ and bounded by $C1$ and $C2$, which is the same as the result computed in Section III D. And the variation of $n$ brings no change on the initialization duration.

Fig. 10 shows the PI results in 8 time slots without the compensation mechanism when $n = 1000$ by changing the prior-selection number of CH candidates from 1 to 200. We can find when CH candidates $> 63$, the WSN connection probability does not increase a lot. We can also see that under abdication mechanism, the average number of final CHs is near 19. From Fig. 9 we know that the CHs is from 16 to 25. Hence, according to the result in Section III E, the redundancy coefficient $\beta = 3$ is sufficient to guarantee few isolate sensors and few collisions.

Fig. 11 shows the domination situation of every CH in Fig. 8. Every CH has more GWs than CMs, so in this WSN there are several connected paths between any two sensors. This feature provides more room available for the higher layer routing protocol. The dominating situation are similar to those in the other single-hop clustering algorithms [14].

Under 802.15.4 protocol (20k-250kbps), an ID message ($<$ 10 Bytes) can be transmitted in 1 ms. Hence, we set one time slot to be $t = 1$ ms. Fig. 12 shows that PI can initialize a WSN in no more than 16 ms. In contrast, NoSE [8] costs 0-3 mins as $n$ varies from 1 to 10000; the duration of CTP in TinyOS 2.0 depends on $n$ and the initialization duration is more than 10 mins when $n > 1000$. Compared to the other protocols, PI sharply reduces the duration of WSN initialization and thus has an obvious advantage of time over those protocols.

**Discussion:** With the maturing of multiple packets reception (MPR) technology [2], a sensor can have 8-MPR ability. Then PI can achieve initialization in 1 time slot in our simulation.

## VI. Conclusion and Future Work

Although many existing works focus on initialization phase in WSN, most of them fall into the energy efficient mechanisms. In this paper, we address the issue of accelerating initialization. The PI mechanism is provided to reduce the initialization duration, which prior selects CH candidates before deployment and parallel initializes into single-hop clusters by these CHs after random deployment.

Since the accelerating initialization problem is a relatively new concept, several respects still remain to be improved. Tradeoff between the knowledge of topology information and the acceleration initialization is one of our future work. A rapid mechanism combined with initialization and routing protocols is a practical future work.

## Acknowledgment

This research was partially supported by NSF of China under grant No. 60773091, 973 Program of China under grant No. 2006CB303000, 863 Program of China under grant No. 2006AA01Z247, the Key Project of China NSFC Grant 60533110. Our shepherd, Zhi Li, Jialiang Lu, Yanmin Zhu and Xinbing Wang gave us highly valuable comments to improve the paper.

## References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey", *Computer Networks Journal*, 2002.
[2] J. J. Garcia-Luna-Aceves, H. R. Sadjadpour, and Z. Wang, Challenges: "Towards truly scalable ad hoc networks",*MobiCom 2007*, Montreal, Quebec, Canada, 2007.
[3] P. Gupta and P.R. Kumar,"Critical Power for Asymptotic Connectivity in Wireless Networks", *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*, 1998
[4] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "Initializing Newly Deployed Ad Hoc and Sensor Networks", *MobiCom2004*, New York, 2004
[5] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Ad-hoc Networks Beyond Unit Disk Graphs", in *Proceedings of the 2003 Joint Workshop on Foundations of Mobile Computing*, 2003.
[6] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks", *Journal on Selected Areas in Communication*, Sept, 1997.
[7] M. J. McGlynn and S. A. Borbash, "Birthday Protocols for Low Energy Deployment and Flexible Neighbor Discovery in Ad Hoc Wireless Networks", *MobiHoc 2001*, New York, 2001.
[8] A. Meier, M. Weise, J. Beutel and L. Thiele, "NoSE: Neighbor Search and Link Estimation for a Fast and Energy Efficient Initialization of WSNs", *TIK-Report*, No. 285, 2008.
[9] T. Moscibroda, P. von Rickenbach and R. Wattenhofer, "Analyzing the Energy-Latency Trade-off during the Deployment of Sensor Networks", *INFOCOM 2006*, Barcelona, Spain, 2006.
[10] K. Nakano and S. Olariu, "Energy-Efficient Initialization Protocols for Single-Hop Radio Networks with no Collision Detection", *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 11(8), 2000.
[11] C. Ribeiro, "Robust Sensor Self-Initialization: Whispering to Avoid Intruders", *International Conference on Emerging Security Information, Systems and Technologies*, Valencia, Spain, 2007.
[12] P. Wan, K. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks", *INFOCOM 2002*, New York, 2002
[13] O. Younis and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach", *INFOCOM 2004*, Hong Kong, China, 2004.
[14] J. Yu, P. Chong, "A Survey of Clustering Schemes for Mobile Ad Hoc Networks", *IEEE Communications Surveys*, 7(1), 2005
[15] R.Zheng, J. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks", *MobiHoc 2003*, Anapolis, Maryland, 2003.