Convergent SparseDT Topology Control Protocol in Dense Sensor Networks *

Chengdong Jiang
Department of Computer Science and Technology
University of Science and Technology of China
Hefei 230027, P.R.China
cdjiang@ustc.edu.cn

Guoliang Chen
Department of Computer Science and Technology
University of Science and Technology of China
Hefei 230027, P.R.China
glchen@ustc.edu.cn

ABSTRACT

When a sensor network is deployed for monitoring a protected region, topology control is usually used to save energy consumption, especially in a dense deployment. In this paper, we propose a new and simple topology control protocol, *Convergent SparseDT*, which controls the network density pretty well, and disposes of some important faults of classical topology control protocols while working with dense sensor network. It is a compromise between the energy consumption, network congestion and the area coverage. It guarantees the coverage of most of the monitored area with almost negligible communication and computing overhead. The performance of this protocol is thoroughly analyzed and simulated in NS2 simulator.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Network Topology, Wireless Communication, Distributed Networks*

General Terms

Design, Theory, Performance, Experimentation

Keywords

Wireless sensor network, Topology control, Gravitational field, Delaunay triangulation, Distributed algorithm

1. INTRODUCTION

Wireless sensor networks have been widely used in many applications nowadays, especially in the area of environmental monitoring. Energy efficiency is the most critical issue in wireless sensor networks since sensors are all battery powered and have limited resources. Thus how to prolong network lifetime attracts much our attentions. One of the most important way to save energy is based on the scheduling sensor activity so that only part of sensors remain on service with others sleeping. It has been recognized that the energy consumption in the sleep state is about tens times less than that in the active state [20]. One of the metrics of network service quality is coverage degree which has been handled in many works. In general, the more sensors are active, the better service this network could provide. But coming with it might be severe communication congestion problem and much higher energy consumption which is not preferred. We should work out carefully a compromise between the energy consumption and the service quality of sensor network in the working area.

Several important topology control protocol have been proposed in recent years. The authors in [21] proposed an algorithm called Coverage Configuration Protocol(CCP) for the connected k-coverage problem. This algorithm can dynamically configure the network to provide different coverage degrees requested by applications. The central part of CCP is a local algorithm which determines the coverage of all intersection points of the sensing range of all neighbors. This local algorithm has computing complexity of $O(n^3)$. It is also integrated with the SPAN protocol [6], which decides if a node should be working or sleeping based on the connectivity among its neighbors, to provide a 1-connected k-coverage network.

In [25], Zhang and Hou proposed an algorithm called Optimal Geographical Density Control(OGDC). It ensures balanced energy consumption over the whole set of vertices by constructing the active network from scratch periodically. A power-on message containing geographical location initiates a message passing procedure at each round to locate working nodes according to their battery levels. This procedure requires much communication overhead, and also a synchronization scheme to lower the package collision probability.

Ye et al. [24] present PEAS, a probing based density control algorithm. In this work, a sleeping node wakes up occasionally to check if there exist working nodes in its vicinity, and it decides to be active or sleep again. The probing range can be adjusted to achieve different levels of coverage redundancy. It does not ensure that the coverage area of a sleeping node is completely covered by other nodes, i.e., it does not guarantee complete coverage.

Another classical protocol is GAF algorithm [23] which

*This paper is supported by the National Grand Fundamental Research 973 Program of China under Grant No.2006CB303006.

INFOSCALE 2007, June 6-8, Suzhou, China

DOI 10.4108/infoscale.2007.228

is grid based. It is the only protocol which is adopted in the NS2 simulator because of its simplicity. GAF assumes the availability of GPS and conserves energy by dividing a region into rectangular grids, ensuring that the maximum distance between any pair of nodes in adjacent grids is within the transmission range of each other, and electing a leader in each grid to stay awake and relay packets. The leader election scheme in each grid takes into account of battery usage at each node.

The most related work is a greedy algorithm from Amitabha [13], which focuses on the minimum connected sensor cover problem based on the notions of maximal independent sets in static sensor network. In principle, the rule of letting part of sensors to judge that most others be dead or alive, is not fair. In other words, it could not balance the energy consumption during the lifetime of total network.

We also noticed that there is some dense region in simulation result of CCP/SPAN algorithm [21], and it is redundant to have so many sensors active in this area only to guarantee complete coverage. The verbose data it generated might jam the network casually. This is especially not preferred in a dense network. We suggest a new method in this paper to circumvent this problem. This is in fact a compromise between energy consumption and coverage degree just as mentioned earlier. So in this paper, we do not guarantee complete area coverage because only probabilistic coverage degree is concerned in randomly deployed sensor network.

We present a new topology control protocol with almost negligible communication and computing overhead. First, we construct the *Sparse Delaunay Triangulation (SparseDT)* topology structure [17] to balance the energy consumption of different part of the sensor network. To increase the area coverage along with the triangulation topology performance, we make the topology converge to potential field equilibrium under the sparseness constraint. This procedure densifies total network evenly, and still maintains the nice attributes of SparseDT graph.

The rest of this paper is organized as following. In section 2, we define the problem formally, describe heuristic potential field method from the celestial mechanics background, and propose an efficient distributed algorithm to maintain the SparseDT topology. In section 3, we give the upper bound of the number of active sensors, and the asymptotic coverage probability is also analyzed. Especially, we prove that the algorithm will converge to Nash equilibrium, and the run-time performance is analyzed thoroughly. In section 4, we provide some simulation results which agree with our analysis in section 3. Section 5 concludes this paper.

2. CONVERGENT SPARSEDT TOPOLOGY CONTROL

2.1 Preliminaries and Problem Formulation

We assume that all sensors are deployed in a two dimensional space with Poisson spatial distribution. Every node has the same sensing range R_s and communication range R_c with the relation of $R_c \geq 2R_s$. All sensors have been localized in advance. We assume a deterministic sensing model that a sensor node s can cover any point inside its sensing circle, $C_s^{R_s}$, which is centered at s and has a radius R_s. That is, any point x with distance to s less than R_s is covered by s, which is denoted by $x \in C_s^{R_s}$. We define circle C_s^r in the same way. We now give a detailed model of unit disk graph.

Definition 1. Given a set of nodes V in a two dimensional space, we model a wireless sensor network with *Unit Disk Graph with unit length* r, $G_r(V,E)$ or $G_r(V)$, which is comprised of all nodes in V and all edges connecting nodes of V whose distance is at most the unit length r, i.e., two nodes u and v are direct neighbors if and only if $|uv| < r$.

It's generally convenient for the analysis of coverage problems that we set the unit length in this model as $r = R_s$, and $r = R_c$ for the analysis of connectivity problems. With different unit length, the edge sets could be quite different, such as $G_{R_c}(V,E_1)$ and $G_{R_s}(V,E_2)$.

We now introduce the motivation of our distributed topology control method. First, the total sensor network needs to be distributed uniformly at every part of the region. We model it as a *dispersed network* which is a maximal independent set V' in $G_{R_s}(V)$, and its sparseness is denoted by the gap between sensors which equals R_s here. Second, we construct locally the SparseDT graph on node set V' with only 1-hop neighbor information. It has been proved in [17] that the SparseDT graph is a Delaunay triangulation with unit length $r = \sqrt{3}R_s$ as long as $R_c \geq 2R_s$. Precisely, it is $UDel_r(V') = Del(V') \cap G_r(V')$, where $Del(V')$ is the full Delaunay triangulation of V'. The asymptotic full coverage of the dispersed network, as well as the connectivity of SparseDT graph, is also guaranteed as long as $n = |V|$ approaches infinity.

But the dispersed network might not be dense enough to afford the connectivity and coverage requirement and triangulation perfectness. That is, there might be small regions which are not triangulated in SparseDT graph. This kind of region is called *hole*. We propose a potential field method in this paper to deal with the hole problem considering the inequable node density in different part of the monitoring area. Our approach is to increase the density of dispersed network as possible as we can under the sparseness constraint. By assigning each node value of special potential function and competition mechanism, the dispersed network could reach equilibrium, and the SparseDT graph constructed on it is almost a full Delaunay triangulation.

Definition 2. We define the weight of an arbitrary edge $e \in E_1$ in $G_{R_c}(V,E_1)$ according to its Euclidean measure.

$$w(e) = \begin{cases} 1/|e|, & R_s \leq |e| < R_c \\ -\infty. & 0 < |e| < R_s \end{cases}$$

With the edges weighted, we could formulate the optimization purpose of our algorithm as following.

Definition 3. We define the *Maximum Potential Problem* as the problem of finding the Maximum Induced Subgraph $G_{R_c}[V']$ of $G_{R_c}(V,E_1)$ with hereditary Property $\mathcal{P}$, where $V' \subseteq V$, and property $\mathcal{P}$ is that V' is an independent set of $G_{R_s}(V,E_2)$.

In another way, we need to find an independent set V' in $G_{R_s}(V,E_2)$, and the total edge weight of the induced subgraph $G_{R_c}[V']$ of $G_{R_c}(V,E_1)$ should be the maximum of all.

If $\mathcal{P}$ holds for arbitrarily large graphs, does not hold for all graphs, and is hereditary (It holds for all induced subgraphs of a graph whenever it holds for the graph), then it could be verified that the problem of finding a *Maximum Weight Induced Subgraph with Property* $\mathcal{P}$ is NP-complete(see GT21 in [12]) by reduction from maximum clique, and assigning unit weight to edges and $-\infty$ to non-edges. Examples of such properties $\mathcal{P}$ are "being an independent set", "being m-colorable", and "being a planar graph". This problem is closely related to the maximum edge subgraph [10] and the maximum dispersion problem [15] which are also NP-complete. With both node and edge weights, It has many interesting applications in statistical physics [9] and other areas [8].

If we sum the weights of the indent edges of node s, it approximates the *potential* [7] of node s in this gravitational field in which each node has the unit mass. That is the reason why we call it maximum potential problem, and an appropriate method will be proposed to approach it in following sections.

2.2 Potential Field Theory

First, we give a brief introduction to the Potential Field knowledge from celestial mechanics [7]. The solution of any classical mechanics problem is first determining the equations of motion which one could choose from the formalism of Lagrange or Hamilton. In the methods developed, the correct formulation of the Lagrangian required knowledge of the potential through which the system of particles moves. In this way the more complicated vector equations of motion can be obtained from the far simpler concept of the scalar field of the potential.

It is generally a good first approximation to assume that the potential of the sun and planets is that of a point mass. This greatly facilitates the solution of Laplace's equation and the determination of the potential.

We now describe the method whereby the potential can be calculated for an arbitrary collection of mass points to an arbitrary degree of accuracy.

The gravitational field $\vec{G}$ is defined as the gravitational force per unit mass so that

$$\vec{G} = \vec{F}/m \equiv \nabla\Phi \tag{1}$$

Here Φ is known as the gravitational potential, and $\nabla\Phi$ is the gradient of it. Now Newtonian gravity says that the gravitational force between any two objects is proportional to the product of their masses and inversely proportional to the square of the distance separating them and acts along the line joining them. The collective sum of the forces acting on a node s of mass m will be $\overrightarrow{F_s} = \sum GmM_i\overrightarrow{ss_i}/|\overrightarrow{ss_i}|^3$, where M_i is the mass of node s_i and G is the gravitational constant. The potential that will give rise to the force field resulting from such a configuration is

$$\Phi_s = \sum \frac{GM_i}{|\overrightarrow{ss_i}|} \tag{2}$$

The scalar sum of equation (2) is ready for insertion in the Lagrangian to determine the motion of nodes that compose the mechanical system.

2.3 Gravitational Field Equilibrium

Assuming all sensor nodes have the same unit mass, we regard the sensor network as a celestial mechanical system forming a gravitational field except that the only permitted motion of nodes is to sleep and wake up. It looks like a multiple star system. The gravitational force pulls them together, and the repulsion force keeps them apart from each other to avoid collapse, just as the universe is. We only consider the gravitational force from the neighbors in communication range, and a sensor node overcast its neighbors in sensing range just like a nebular devors nearby star. So is confined the push-pull action to local environment, and fluctuates through total network.

Here we mix up the concept of gravitational field and potential field. We know from potential field theory that a system in a potential field must have some properties, such as the energy dissipative property [7], to settle down or converge to an equilibrium state. With these configurations, everything else is trivial. Just let universal gravitation decides everything. There is no real vacuity in universe, and constellation will rearrange itself while new planet is born in "Vacuum".

Definition 4. We define the set of active neighbors of s nearby in r range as

$$N_s^r = \{s_i \mid 0 < |\overrightarrow{ss_i}| < r\}$$

Definition 5. With the assumption that all nodes have the same unit mass and only active neighbors are considered, we define the *potential function* δ_s of sensor node s in the gravitational field as the sum of inversions of distance between s and those distant neighbors, called *legal neighbors* of s, which are at least R_s distance away from s. Ignoring the gravitational constant G, it is also the total gravitation force that those legal neighbors act on s. According to equation (2), δ_s could be formalized as a scalar sum,

$$\delta_s = \sum_{s_i \in N_s} \frac{1}{|\overrightarrow{ss_i}|}, \qquad N_s = \{s_i \mid R_s \le |\overrightarrow{ss_i}| < R_c\} \tag{3}$$

Here, we only consider the active neighbors which could affect node's potential energy, and it is $N_s = N_s^{R_c} - N_s^{R_s}$ with definition 4. Those sleep or dead nodes are unaware of what is happening in this virtual world.

We define the potential energy of a set of sensors S as $\delta_S = \sum_{s\in S} \delta_s$, and energy of the total system is Φ, which is the sum of potentials of all active sensors. According to Definition 3, potential of the optimum solution to the maximum potential problem should be $\delta_{opt} = Max_{V' \subseteq V}\{\delta_{V'}\}$, where V' is an independent set of $G_{R_s}(V, E_2)$.

2.4 Distributed Algorithm

We propose a simple but efficient distributed algorithm to deal with the sensor network topology control. The following procedure only describes the action of a single node while it wakes up.

There are several points worthy of note in this protocol. First, sensors communicate with each other only through broadcasting messages, which is the most likely manner in sensor networks. The communication cost could be bounded as long as the sparseness of the network is guaranteed, and there is only negligible additional messages posted except for neighbor discovery. Second, sensors only need to compute the potential energy and local Delaunay triangulation. There are many algorithms to construct the Delaunay triangulation. One of the most commonly used is Randomized Incremental Method with complexity of $O(n \log n)$. We use CGAL routine [1] to construct the local Delaunay triangulation at step 13 in Proc.1. The computing complexity of

Proc. 1 *Convergent SparseDT Topology Control*

```
Require: Sensor s wakes up while its wakeup_timer expires.
Ensure: SparseDT topology is updated properly.
  {Neighbor Discovery}
  Sensor s broadcasts "Discovery" message
  Collecting one-hop neighbor information
  {Local Optimization}
  if |N_s^{R_s}| = 0 then
     s.state=active
  else if |N_s^{R_s}| = 1 and δ_s > δ_{s'}, s' ∈ N_s^{R_s} then
     s.state=active
     s'.state=sleep; s'.wakeup_timer=random
  else
     s.state=sleep; s.wakeup_timer=random
  end if
  {Update Delaunay Triangulation Topology}
  if s.state==active then
     Sensor s broadcasts "Update" message
     All s_i ∈ N_s ⋃ N_{s'} call CGAL routine to update their
     local Delaunay triangulation
  end if
```

this step is $O(|N_s| \log |N_s|)$ for each node, and it is $O(1)$ as long as $|N_s|$ could be bounded. Comparing with it, the CCP/SPAN protocol [21] uses a local algorithm with computing complexity of $O(n^3)$ to determine the coverage property of a node's surrounding area, and its space complexity is $O(n^2)$ in worst case which is unbearable in wireless sensor networks. Third, this sensor network topology control protocol does not need any time synchronization scheme which is another subtle problem to be solved. Every sensor has only timer triggered actions, and total network works asynchronously.

We could see that a newly active sensor maximizes potential energy among neighboring nodes within its sensing range. It is in fact a local optimization technique. As for the local optima problem, there is a little trick to circumvent it. We could schedule sensors to sleep periodically, and escape the local optima. Sensor's sleeping action is quite like the disappearance of an "attractor", and new sensors will wake up in this vacuum. We prove that this procedure converges to Nash equilibrium in next section.

3. THEORETICAL ANALYSIS

In this section, we give some analysis results of our algorithm. The number of active sensors and the coverage probability are bounded, and the run-time performance of this algorithm is analyzed thoroughly.

3.1 Coverage Probability Analysis

Suppose that all sensors are deployed in a rectangular region with area $A = L \times L$. We have the following bound of the number of active sensors.

THEOREM 1. *The number of active sensors in dispersed network is at most*

$$N_{DT} \leq \frac{2(L+R_s)^2}{\sqrt{3}R_s^2} \tag{4}$$

PROOF. We expand the region from four directions each with length $R_s/2$, and get an expanded region with area $A' = (L+R_s)^2$. Since all active sensors are at least R_s apart from each other, the maximum number of active nodes in dispersed network equals the maximum number of circles with equal radius $R_s/2$ inside the expanded region such that no two overlap and some (or all) of them are mutually tangent. There is a well-developed theory of this transformed problem which is called circle packing.

The densest packing of circles in the plane is the hexagonal lattice, which has a packing density of $\eta_h = \sqrt{3}\pi/6$. Gauss proved that the hexagonal lattice is the densest plane lattice packing, and in 1940, L. Fejes Tóth proved that the hexagonal lattice is indeed the densest of all possible plane packings. But it is still an open problem as for the problem of packing circles in a bounded region, which is known as the Hilbert's eighteenth problem presented in 1990. However, this value η_h can provide an upper bound for the solution of the problem here. So we have the following estimation.

$$N_{DT} \leq \frac{A'\eta_h}{\pi(R_s/2)^2} \Longrightarrow \lim_{L\to\infty} \frac{N_{DT}}{A} = \frac{2}{\sqrt{3}R_s}$$

□

THEOREM 2. *The probability that a sensor node s is active, P_{DT}, is bounded asymptotically.*

PROOF. We suppose that the deployment of n sensors in the area is subject to Poisson process. We set $\lambda_0 = n\pi R_s^2/A$. Mark symbol $x \equiv P_{DT}$ for short, and We know that $x \ll 1$ in a quite dense network. For any certain point X in the area, the probability that X is covered by exactly k sensors is $P_k = e^{-\lambda_0}\lambda_0^k/k!$. While sensor s is active, all sensors s_i within its sensing range must be asleep. It means that the following inequations should exist:

$$\begin{aligned}
&x \leq \sum_{k=1}^{n}(1-x)^{k-1}P_k \\
\Longrightarrow\, & x(1-x) \leq \sum_{k=1}^{\infty}(1-x)^k P_k = e^{-\lambda_0 x} - e^{-\lambda_0} \leq e^{-\lambda_0 x} \\
\Longrightarrow\, & xe^{\lambda_0 x} \leq \frac{1}{1-x} \approx 1+x \\
\Longrightarrow\, & \lambda_0 x e^{\lambda_0 x} \leq \lambda_0(1+x) \approx \lambda_0 \\
\Longrightarrow\, & x \leq \frac{W(\lambda_0)}{\lambda_0}
\end{aligned} \tag{5}$$

In the solution we found, W is the *Lambert W-Function* [22] which satisfies $W(y)e^{W(y)} = y$. □

As for the computing of the bound $z = \frac{W(\lambda_0)}{\lambda_0}$ in equation (5), We solve the equation $\lambda_0 z e^{\lambda_0 z} - \lambda_0 = 0$ numerically in Matlab [2].

Suppose that 900 sensors with sensing range of 5 meters are deployed in area of 50 by 50 meters according to our simulation configuration in section 4, we have $\lambda_0 = 9\pi$, and $P_{DT} \leq 0.0865$. It means that the expected number of active sensors in dispersed network is less than 78, which matches the simulation result perfectly. Theorem 1 also bounds the maximum number of active sensors as $N_{DT} \leq 139$.

Assume that the initial deployment of all sensor nodes in set V_0 takes the Poisson distribution with density λ_0. We bound the probability P_a that the dispersed network fully covers the whole monitoring area, and this probability approaches one as the total number n of all deployed sensors increases.

The probability that there is a point x which is not covered in the dispersed network, is $P_{hole} = 1 - P_a$. It means that no sensors are active nearby, that is, $\forall s \in V', x \notin C_s^{R_s}$, where V' is the set of active nodes. We have $P_{hole} = \sum_{k=0}^{n} e^{-\lambda_0}\lambda_0^k/k!(1-P_{DT})^k \leq e^{-\lambda_0 P_{DT}}$, where P_{DT} is the probability that a deployed node is active in the dispersed network. So that $P_a \geq 1 - e^{-\lambda}$, where λ is the node density of the dispersed network. It is also a result from stochastic geometry [14] that the area coverage of the dispersed network is $P_a = 1 - e^{-\lambda_0 P_{DT}}$, which is also confirmed in [19].

We have $\lim_{n\to\infty} P_a = 1$, so that the dispersed network is connected as long as $R_c \geq 2R_s$. In fact, if there is a hole with radius r which is not covered, according to the attributes of dispersed network, there must be no sensors deployed initially in this hole, which has the probability $P\{\forall s \in V_0, x \notin C_s^r\} = e^{-n\pi r^2/A}$ approaching zero as $n \to \infty$.

3.2 Convergence Analysis

3.2.1 *Nash Equilibrium*

We define a Nash equilibrium as a choice of action by each node so that no node can improve its potential by changing its state alone. For the maximum potential problem, a state is a Nash equilibrium if for all active node s, $\delta_s > \delta_{s'}$, $\forall s' \in N_s^{R_s}$ at this time. We will show that Nash equilibrium could be found via local optimization.

THEOREM 3. *The gravitational field algorithm we proposed in Proc.1 will converge at last.*

PROOF. From the algorithm, It could be easily verified that the energy increment of total system is $\Delta\Phi = 2(\delta_s - \delta_{s'})$, where s is the node that survived a single local exchange step. It is guaranteed that δ_s is positive by our gravitational field equilibrium algorithm. Since there are only a finite number of different values for the energy of total system, and we cannot cycle because Φ increases in each iteration, We will definitely find a Nash equilibrium. □

3.2.2 *Expectation of Potential Increment*

We only consider the case of $R_c = 2R_s$. Two prepositions are made here to simplify our analysis. First, the potential functions of two distinct sensor nodes are independent from each other, even though there are some intersections between their neighborhoods. Second, the active sensors are uniformly distributed in the deployed region, so that the number of one's legal neighbors, $|N_s|$, is subject to Poisson distribution with density of $N_{DT}\cdot\pi(R_c^2 - R_s^2)/A = 3\lambda$, where $\lambda = N_{DT}\cdot\pi R_s^2/A = \lambda_0 P_{DT}$. In fact, the second condition could be deduced from the random competition policy of our algorithm.

We consider an active node s and one of its legal neighbors $s_i \in N_s$. The distance of s and s_i is $d \in [R_s, R_c)$, and its contribution to the potential of s is $1/d \in (1/R_c, 1/R_s]$. The probability density function (p.d.f.) of d is $f_d(x) = \frac{2\pi x}{\pi(R_c^2 - R_s^2)} = \frac{2}{3R_s^2}x$, and the p.d.f. of $1/d$ could be deduced as a function of $f_d(x)$, i.e., $f_{\frac{1}{d}}(y) = f_d(\frac{1}{y})|(\frac{1}{y})'| = \frac{2}{3R_s^2 y^3}$. Its expectation is $E(\frac{1}{d}) = \frac{2}{3R_s}$, and the second moment is $E((\frac{1}{d})^2) = \frac{2}{3R_s^2} ln2$.

Now, we could get the mathematical expectation and variance of the random variable δ_s, which is subject to a compound Poisson distribution, $\mu = E(|N_s|)E(\frac{1}{d})$, and $\sigma^2 = E(|N_s|)E((\frac{1}{d})^2)$, making use of the law of total cumulance [5].

According to Laplace central limit theorem [16], the distribution of δ_s could be regarded as a bounded normal distribution $N(\mu, \sigma^2)$, with the bound of $\delta \in (a, b]$, the p.d.f. $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, accumulative function $\phi(y) = \int_{-\infty}^{y} f(x)\,dx$, and the variable $z = (x-\mu)/\sigma \sim N(0,1)$. We could estimate the average increase of potential value:

$$\begin{aligned}
E(\Delta\delta) &= E(\delta_s - \delta_{s'} \mid \delta_s > \delta_{s'}) \\
&= \int_a^b f(y)\,dy \int_a^y (y-x) f(x)\,dx \\
&= \int_a^b f(y) y \phi(y)\,dy - \int_a^b f(y)\,dy \int_a^y (\sigma z + \mu) f(x)\,dx \\
&= \int_a^b f(y)(y-\mu)\phi(y)\,dy + \int_a^b f(y) \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}\,dy \\
&= \int_a^b f(y)(y-\mu)\phi(y)\,dy + \frac{\sigma}{2\sqrt{\pi}} \\
&= \int_\mu^b f(y)(y-\mu)(2\phi(y)-1)\,dy + \frac{\sigma}{2\sqrt{\pi}} \\
&= \Delta_1 + \Delta_2
\end{aligned} \tag{6}$$

In equation (6), the first item Δ_1 is greater than zero. Suppose that $y_o > \mu$ and $\phi(y_0) > 1/2$, Δ_1 could further be concretized as

$$\begin{aligned}
\Delta_1 &> (2\phi(y_0) - 1) \int_{y_0}^b f(y)(y-\mu)\,dy \\
&= \sigma^2 f(y_0)(2\phi(y_0) - 1)
\end{aligned} \tag{7}$$

With $(y_0 - \mu)/\sigma = 0.9$, we'll get $\Delta_1 > \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{0.9^2}{2}} (2\times 0.816 - 1) \approx 0.168\sigma$. The second item Δ_2 in equation (6) could be estimated as 0.28σ. According to equation (6) and (7), we get

$$E(\Delta\delta) > 0.448\sigma. \tag{8}$$

which will be verified in simulation.

3.2.3 *Description of Algorithm Stages*

In our algorithm, each node s wakes up at uniformly scheduled time with period T, and checks if it could initiate a local exchange step. Supposing that the density of active nodes is $\lambda = N_{DT}\cdot\pi R_s^2/A$ while node s wakes up, and the number of sleeping nodes $n' = n - N_{DT}$ is also fixed.

We first model the collective scheduled wake-up events as the superposition of n' *Renewal Processes* [11]. According to the central limit theorem of point processes [18], the superposition of n' independent and identically distributed renewal processes each with rate $2/T$ tends to a Poisson process with rate $\lambda' = 2n'/T$ as n' gets large.

While s wakes up, three scenarios might occur depending on the local potential field of its neighborhood.

- *Idleness*: s has more than one active neighbors in R_s range, and goes back to sleep again.
- *Exchange*: s has only one active neighbor s' in R_s range, and decides to be active while $\delta_s > \delta_{s'}$, or go to sleep again while $\delta_s \leq \delta_{s'}$. From symmetry, probability of the first case could be calculated as $p_1 \approx P(|N_s^{R_s}| = 1)P(\delta_s > \delta_{s'}) = \frac{1}{2}P(|N_s^{R_s}| = 1) \approx \frac{1}{2}\lambda e^{-\lambda}$, with the approximation to Poisson distribution.

- *Increment*: s has no active neighbors in R_s range, and becomes active thereby. This probability p_0 could not be simply regarded as the Poisson probability of no neighbors, because every node here is covered by an active sensor initially according to our distributed algorithm. This event could be reworded that all active neighbors of s at the beginning of this stage have been replaced until now.

We define the *stage* here as a continuous time slice during which the density λ keeps unchanged. The expected duration of a single stage, T_{stage}, will be estimated below.

First, we estimate the expectation of number of wake-up events in a single stage, N_{stage}. Define $P_{s_i\downarrow}$ as the probability of s_i having been replaced during this stage, and it means that at least one local exchange step has happened at s_i's neighborhood since previous stage. Suppose that it has been m steps, we have

$$P_{s_i\downarrow} \approx \sum_{j=1}^{m} \binom{m}{j} P_e^j (1-P_e)^{m-j} \qquad where\, P_e = \frac{\pi R_s^2}{A} p_1$$
$$\approx 1 - e^{-mP_e}$$

While s wakes up, it finds that all neighbors have been replaced at previous steps.

$$P_{N_{stage}\le m} \ge P_{N_{stage}=m|N_{stage}\ge m} \approx \sum_{k=0}^{\infty} P_{|N_s|=k} \prod_{s_i\in N_s} P_{s_i\downarrow}$$

So we get

$$\begin{aligned} E(N_{stage}) &= lim_{M\to\infty} \sum_{m=0}^{M} (1 - P_{N_{stage}\le m}) \\ &\le lim_{M\to\infty} \sum_{m=0}^{M} (1 - \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} e^{-\lambda} (1-e^{-mP_e})^k) \\ &\approx lim_{M\to\infty} \sum_{m=0}^{M} \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} e^{-\lambda} \cdot k e^{-mP_e} \\ &\approx lim_{M\to\infty} \lambda \cdot \sum_{m=0}^{M} e^{-mP_e} \\ &= \frac{\lambda}{1-e^{-P_e}} \approx \frac{\lambda}{P_e} = \frac{2A}{\pi R_s^2} e^{\lambda} = ce^{\lambda} \end{aligned} \tag{9}$$

We could see that λ keeps steady until the *Increment* scenario happens, and then the evolution of total sensor network comes into another stage. Since the scheduled wake-up process we modeled has expected interval $1/\lambda'$, the duration of one stage could be estimated as $E(T_{stage}) \le ce^{\lambda}/\lambda'$, which increases exponentially with the density λ.

3.2.4 Metrics and Convergence Time Analysis

Suppose that the density of active sensors is λ_1 initially, and it becomes λ_2 when the system converges. The corresponding numbers of active nodes are n_1 and n_2 respectively. We only consider the time, T_c, to reach maximum density here. Given that the wake-up period $T = 2n$ according to the simulation configuration in section 4, T_c could be estimated as following.

$$E(T_c) \le \sum_{N_{DT}=n_1}^{n_2-1} \frac{ce^{\lambda}}{\lambda'} < \int_{n_1}^{n_2} \frac{ce^{\lambda}}{\lambda'}\, dx = \int_{n_1}^{n_2} \frac{ce^{2x/c}}{1-x/n}\, dx \tag{10}$$

Because N_{DT} is bounded as theorem 1 says, $\lim_{n\to\infty} E(T_c) < \frac{c^2}{2} e^{\lambda_2}$ which is also bounded exponentially with λ_2.

We have defined some metrics and symbols here to facilitate our analysis and simulation. Table 1 is a summary of them.

n	total number of all sensors
N_{DT}	number of active sensors
δ_s	potential of a single node
Φ	total potential of all active nodes
λ_0	density of poisson distribution of all sensors
λ	density of poisson distribution of active sensors
λ'	intensity of poisson process of the wake-up events
N_{stage}	number of wake-up events in a stage
T_{stage}	duration of one evolution stage
T_c	time to reach maximum density

Table 1: description of metrics

From previous potential analysis, we have the average potential value

$$E(\delta_s) = \mu = \frac{\pi R_c^2 - \pi R_s^2}{A} N_{DT} \times \frac{2}{3R_s} = \frac{2\pi R_s}{A} N_{DT} \sim \lambda \tag{11}$$

and the average total potential value

$$E(\Phi) = E(\delta_s) N_{DT} = \frac{2\pi R_s}{A} N_{DT}^2 \approx 0.0126 N_{DT}^2 \sim \lambda^2 \tag{12}$$

These models will be verified in our simulation in next section.

4. SIMULATION

We implement the *Convergent SparseDT Protocol* as an agent in NS2 simulator [4] of version 2.30. All sensor nodes use the IEEE 802.15.4 as the MAC layer protocol. It is quite appropriate for monitoring applications with sensor networks, and its dynamic performance is better than 802.11 and some other MAC protocols aiming at sensor networks, such as s-mac and d-mac. We select two-ray ground model as the radio propagation model.

We set the sensing radius R_s to be 5 meters, and communication radius R_c to be 10 meters. Hundreds of sensors are deployed uniformly in area of 50×50 meters to simulate a dense sensor network. The sensor sleeping duration submits the uniform distribution with expectation of n seconds. So that the collective scheduled wake-up events approximate a Poisson process with rate $\lambda' \approx 1$, which facilitates our analysis and simulation. Several variables are also fixed, such as $c = \frac{2A}{\pi R_s^2} \approx 63.7$. No matter how large the total number n of sensors is, we have $N_{DT} < 139$, and the relaxed bound of time to reach maximum density is $E(T_c) < \frac{c^2}{2} e^{\frac{2\pi}{\sqrt{3}}}$. We vary the total number of sensors, n, from 100 to 900, which represents the density of network from sparseness to denseness, and find that the convergence procedure always terminates.

At the beginning, the network will take on a legal SparseDT structure via a random selection scheme, which works like a series of speedy *Increment* scenarios. It skips over the trivial phase of initialization, and the convergence procedure follows thereafter until the network reaches equilibrium.

We illustrate a representative sample case here, where $n = 900$. Fig.1 is a random selected sparse Delaunay triangulation mentioned above, and Fig.2 is the SparseDT topology while Proc.1 converges. We can see that the not triangulated area of the region, i.e. the uncovered area, decreases markedly, while the triangulation topology be retained.

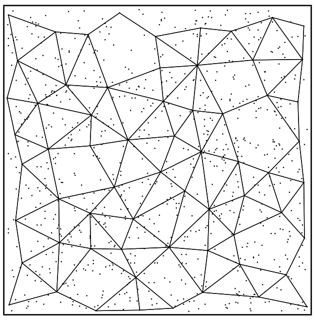

Figure 1: Delaunay triangulation at the beginning.

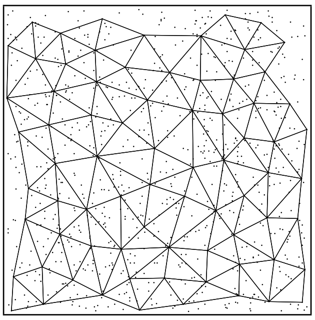

Figure 2: Convergent SparseDT topology

The following figures illustrate the run-time performance of our algorithm. In Fig.3, we could see that the total potential Φ increases almost linearly with the exchange steps, in that the average increments at different stages are almost the same. The proportion approaches 0.22, which matches the analysis result in equation (8). Where in my simulation scenario of $N_{DT} = 70$, we have $\lambda \approx 2.2$, $\mu \approx 0.3$, $\sigma \approx 0.2$, so that $E(\Delta\delta) > 0.448\sigma \approx 0.09$, and $E(\Delta\Phi) = 2E(\Delta\delta) > 0.18$ according to theorem 3. Furthermore, Δ_1 in equation (7) could be calculated approximately to be 0.4σ using Maxima software [3]. So the result value of $E(\Delta\delta) \approx 0.68\sigma \approx 0.136$, and $E(\Delta\Phi) \approx 0.27$.

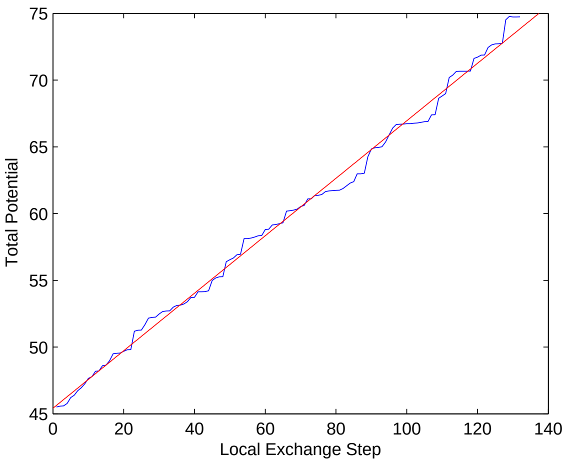

Figure 3: Plot of total potential with local exchange steps.

Similarly, the total potential Φ in Fig.4 is approximately proportional to N_{DT}. But Φ is modeled as quadratic approximation previously in equation (12), and the best quadratic fitting curve $y = 0.013x^2$ shown with dashed line in the figure matches this model perfectly. We could also use Maxima software to plot this model, and find that it matches the simulation result. So there is no reason to refer linear model.

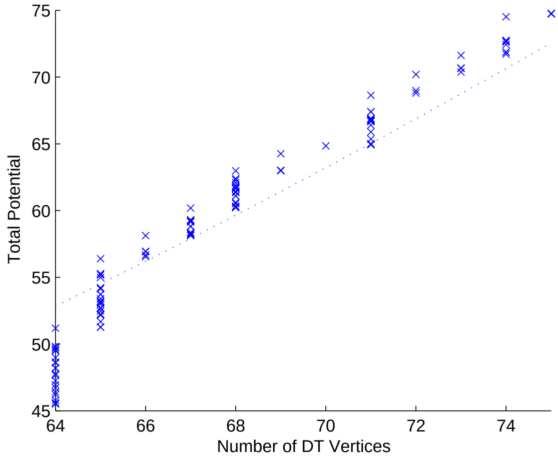

Figure 4: Scatter plot of total potential with number N_{DT} of active sensors.

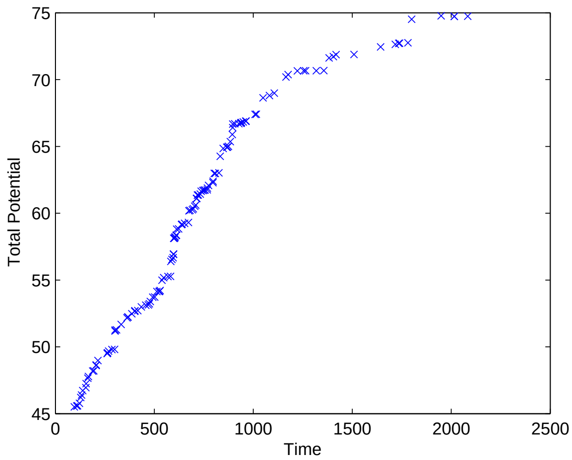

Figure 5: Increment of total potential with time

Fig.5 gives the diagram of run-time performance. Although the duration of single stage increases exponentially with the density according to equation (9), we could see that N_{DT} reaches maximum quickly and almost linearly, in that the density λ doesn't vary much in each stages. The tail in the convergence procedure shows that the SparseDT structure is settling down at the last stage with maximum density. In fact, this stage is not needed since what we want is just the densest Delaunay triangulation under the sparseness constraint, and that is it when N_{DT} reaches maximum. The convergence procedure could also be approximated with an inhomogeneous Poisson process, and this result could also be simulated with Monte-Carlo method which is omitted here.

5. CONCLUSION

In this paper, we propose a new and simple topology control protocol, Convergent SparseDT, which controls the network density pretty well. It is a compromise between network density and the area coverage. While guaranteeing the coverage of most of the monitored area, this protocol aims to optimize the topology of sensor network and find the most dense network topology under the sparseness constraint with almost negligible communication and computing overhead. In section 2, we formulate this purpose as the maximum weight induced subgraph problem with property of sparseness, and propose a potential field equilibrium method to approach it. The performance of Convergent SparseDT protocol is thoroughly analyzed in section 3, where the number of active sensors needed to maintain the SparseDT topology and the coverage probability are bounded. We further point out that the protocol we propose shall converge to Nash equilibrium with bounded speed, and convergence procedure be modeled with stochastic process. These models and verdicts are validated with NS2 simulator in section 4, and the result matches perfectly. In wireless environment, network topology has always taken the idea of cluster or back bone based form with data aggregating objective. On the other hand, we propose the flat and triangulation based topology with applications for environmental monitoring, and the elegant attributes of Delaunay triangulation will make it survive the stringent constraints of wireless communication.

6. REFERENCES

[1] Computational geometry algorithm library, July 2006.

[2] The mathworks - matlab and simulink for technical computing, 2006.

[3] Maxima, a computer algebra system, December 2006.

[4] The network simulator - ns2, September 2006.

[5] D. Brillinger. The calculation of cumulants via conditioning. *Journal Annals of the Institute of Statistical Mathematics*, 21(1), December 1969.

[6] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks*, 8(5):481–494, 2002.

[7] G. W. Collins, II. *The Foundations of Celestial Mechanics.* Pachart Foundation dba Pachart Publishing House, 1989/2004.

[8] A. Czumaj, M. M. Halldórsson, A. Lingas, and J. Nilsson. Approximation algorithms for optimization problems in graphs with superlogarithmic treewidth. *Inf. Process. Lett.*, 94(2):49–53, 2005.

[9] C. De Simone, M. Diehl, M. Jünger, P. Mutzel, G. Reinelt, and G. Rinaldi. Exact ground states of two-dimensional $\pm J$ Ising spin glasses. *Journal of Statistical Physics*, 84:1363–1371, 1996.

[10] U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[11] W. Feller. *An introduction to probability theory and its applications. - Vol. 1.* John Wiley & Sons, 1968.

[12] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Mathematical sciences series. W. H. Freeman, 1979.

[13] A. Ghosh and S. K. Das. A distributed greedy algorithm for connected sensor cover in dense sensor networks. In V. K. Prasanna, S. S. Iyengar, P. G. Spirakis, and M. Welsh, editors, *DCOSS*, volume 3560 of *Lecture Notes in Computer Science*, pages 340–353. Springer, 2005.

[14] P. Hall. *Introduction to the Theory of Coverage Processes.* John Wiley and Sons, New York, 1988.

[15] R. Hassin, S. Rubinstein, and A. Tamir. Approximation algorithms for maximum dispersion. *Oper. Res. Lett.*, 21(3):133–137, 1997.

[16] E. T. Jaynes. *Probability Theory - The Logic of Science.* Cambridge University Press, Cambridge, 2003.

[17] C.-D. Jiang and G.-L. Chen. Double barrier coverage in dense sensor networks. Technical report, University of Science and Technology of China, 2007. in communication.

[18] J. A. Lane. The central limit theorem for the poisson shot-noise process. *Journal of Applied Probability*, 21(2):287–301, 1984.

[19] B. Liu and D. Towsley. A study of the coverage of large-scale sensor networks. In *The First IEEE International Conference on Mobile Ad hoc and Sensor Systems(MASS04)*, 2004.

[20] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, pages 40–50, Mar. 2002.

[21] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. D. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In I. F. Akyildiz, D. Estrin, D. E. Culler, and M. B. Srivastava, editors, *SenSys*, pages 28–39. ACM, 2003.

[22] E. W. Weisstein. Lambert w-function. From MathWorld–A Wolfram Web Resource.

[23] Y. Xu, J. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 70–84, New York, NY, USA, 2001. ACM Press.

[24] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang. Peas: A robust energy conserving protocol for long-lived sensor networks. In *ICDCS '03: Proceedings of the 23rd International Conference on Distributed Computing Systems*, page 28, Washington, DC, USA, 2003. IEEE Computer Society.

[25] H. Zhang and J. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Journal of Wireless Ad Hoc and Sensor Networks*, 1(1):89–124, 2005.