

Implementing Information paths in a Dense Wireless Sensor Network

Masoumeh Haghpanahi, Mehdi Kalantari and Mark Shayman
Department of Electrical & Computer Engineering, University of Maryland
{masoumeh,mehkalan,shayman}@eng.umd.edu

Abstract—In a sensor network with an information source and a sink, we assume a single path between every location in the network to the destination. These abstract paths represent the desired paths for transmission of information. If the number of sensor nodes are large enough to approximate information paths with a certain precision, we call the network implementable. In this paper, we investigate on the relation between density of nodes and implementability of information paths. We show that a quadratic relation between the density of nodes and the magnitude of information flow does not guaranty implementability. On the other hand we show that in a rectangular geometry with a line source and a line sink, and a total load of θ bps, $N = O(\theta^2 \log \theta)$ number of nodes is sufficient for implementing the information paths. Furthermore, we show via simulation that adding the log factor is also a necessary condition for implementing the paths.

I. INTRODUCTION

Dense sensor networks has recently become a subject of interest to many researchers, many of who have analyzed different aspects of such networks with approaches inspired by physics [1], [2], [3], [4], [5], [6]. An underlying assumption of many of these works is that the network is dense or in other words the number of nodes is very large, technically reaching infinity (see the survey [14] for more information in this regard).

Discrete models are not good candidates for modeling the flow of information in a wireless sensor network as the number of sensor nodes grow. A main shortcoming of discrete models is that the flow model becomes intractable when the number of nodes grows very large. To overcome this problem, we use a continuous space model by defining the information flow vector field in formulating the flow of information in a dense wireless network. The vector field has two components at every point of the network: magnitude which is proportional to the density of communication, and direction which is toward the flow of information. This model was first introduced in [1] and [2] with an inspiration from electrostatics, where a quadratic cost function was proposed to find the optimal flow of information from a source to data sink. Approaches based on electrostatics have since become popular. [7], [8], [9], and [10] are among the works based on this analogy. In a recent work [11], we generalized the quadratic cost function to a p -norm flow optimization problem with $p > 1$ by minimizing the p -norm of the information flow vector field subject to the basic flow constraints. We proposed an iterative method to

numerically solve the p -norm flow problem. This was done using Sequential Quadratic Programming (SQP). It was shown in the paper that the p -norm problem has load balancing property; for p close to 1, the information flow tends to pass through geometric shortest paths. By increasing the value of p , the flow spreads in the network using resources more evenly, and it reaches maximum load balancing when $p \rightarrow \infty$. In [12], we discussed more about the challenges of numerical analysis of the p -norm flow optimization problem.

While the concept of information flow has been studied extensively, we still need to answer an important question about the validity of our model. As mentioned earlier, the continuous model is valid as long as the network is dense enough. The important part which is still not discussed is the term “dense” and what is exactly meant by that. In other words, what is the lower bound on the total number of sensor nodes to have a dense network? What is the relation between density of nodes and magnitude of information flow? This question was initially investigated by Toumpis and Tassiulas in [3], where they showed that minimizing the quadratic cost function results in optimal deployment of sensor nodes. This result was based on the assumption that a location (x, y) with node density $d(x, y)$ can support an information flow with magnitude less than or equal to which is proportional to $\sqrt{d(x, y)}$. This assumption was not strictly proven in the paper; it was rather shown in a simple example which can not be generalized to more complicated scenarios.

In this paper we show that the above assumption is not realistic even in very simple examples. For this, we define a new notion of connectivity which is equivalent to implementability of information flow paths with sensor nodes. This happens when number of nodes are large enough to approximate the information flow paths with piece-wise linear routes. With this new definition, we prove that a quadratic relation between the density of nodes randomly distributed throughout the network and magnitude of information flow does not guaranty connectivity. Instead, we show that in a rectangular geometry with a line source at the top producing θ bps and a line BS at the bottom of the network, a total number of $N = O(\theta^2 \log \theta)$ nodes with uniform distribution provides connectivity. Similar to [13], a log factor is the price to pay for the random distribution of nodes.

The rest of the paper is organized as follows. After a short overview of the definition of information flow and its basic properties in Section II, we define a new notion of

connectivity equivalent to implementability of information flux lines in Section III. From there on we use connectivity and implementability interchangeably in the paper. In Section IV we show why a quadratic relation between the density of nodes and the magnitude of information flow is not enough for connectivity. We also set up a simple example and show that $N = O(\theta^2 \log \theta)$ nodes is sufficient to implement information flux lines in a rectangular geometry with a total load of θ bps. In Section V we show by simulation that adding the log factor is also necessary for implementation of information flow, and finally conclude in Section VI.

II. BACKGROUND

Let us start with a short overview of the definition of information flow vector field and its basic properties. We then briefly state the p -norm optimization problem. For more accurate definitions and results see [1], [2] and [11].

Consider a network of N sensor nodes in a region A in the plane. Sensors are intended to collect information and send it to a destination called the data sink in a multihop fashion. Define load density $r(x, y) \geq 0$ as the rate of information generated per unit area at point $z = (x, y)$ in the network. Furthermore, assume $\hat{\theta}(x, y)$ represents a unique direction of flow of information at every point (x, y) in the network. The value of $\hat{\theta}(x, y)$ is undefined at the data sink. This assumption implies that there is a unique path between each location of the network and the destination.

With the above assumptions, we define the information flow vector field as the flux density of paths to the destination. Given a point $(x_1, y_1) \in A$, and choosing a small line segment L passing (x_1, y_1) and perpendicular to $\hat{\theta}(x_1, y_1)$, the magnitude of information flow equals the density of information passing through L . This can be mathematically stated as follows.

$$\vec{D}(x_1, y_1) = \lim_{|L| \rightarrow 0} \frac{\hat{\theta}(x_1, y_1)}{|L|} \int_{\alpha(L)} r(x, y) dx dy \quad (1)$$

In this representation, $\alpha(L)$ denotes the upstream area of L , which is part of the network whose traffic passes through L in its way toward the sink. This is shown in Fig. 1.

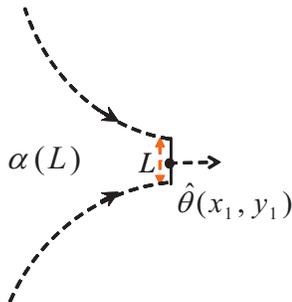


Fig. 1. Upstream area of line segment L passing through point (x_1, y_1) .

The information flow vector field satisfies two basic prop-

erties. First,

$$\begin{aligned} \nabla \cdot \vec{D} &= \rho(x, y), \text{ where} & (2) \\ \rho(x, y) &= r(x, y) - w_0 \delta(x - x_0) \delta(y - y_0), \text{ and} \\ w_0 &= \int_A r(x, y) dx dy. \end{aligned}$$

$\rho(x, y)$ equals the load density function $r(x, y)$ throughout the network except at the data sink, which is assumed to be at (x_0, y_0) in (2). $\rho(x_0, y_0)$ is represented as a delta dirac function of weight w_0 , which is equal to the total load produced in the network. This is to force all the information flow to end at the destination. The second property of information flow is

$$\vec{D}_n(x, y) = 0, \quad \forall (x, y) \in \partial A \quad (3)$$

where \vec{D}_n is the normal component of \vec{D} on the boundary of A , denoted by ∂A . This property states that no information is allowed to pass the geographical boundaries of the network.

Note that definition 1 is a mathematical definition of information flow vector field which depends on the set of paths chosen for the flow of information. Therefore, the value of \vec{D} differs based on the set of paths chosen, but as long as a vector field satisfies properties (2) and (3), it is considered to be an information flow vector field.

From the discussion above, it is obvious that properties (2) and (3) do not specify a unique $\vec{D}(x, y)$ and leave some room for optimization. Therefore, we introduced the p -norm flow optimization problem, which minimizes the p -norm ($p > 1$) of the information flow subject to its basic properties.

$$\begin{aligned} \text{Minimize} \quad & J(\vec{D}) = \int_A |\vec{D}(z)|^p dx dy & (4) \\ \text{s.t.} \quad & \nabla \cdot \vec{D}(z) = \rho(z) \\ & \vec{D}_n(z) = 0, \quad z \in \partial A \end{aligned}$$

It is shown and fully discussed in [11] that the p -norm flow problem has load balancing properties and that the value of p can be chosen based on a trade-off between delay and balancing the traffic over the network.

Tompis and Tassiulas showed in [3] that minimizing the quadratic cost function of \vec{D} , the p -norm flow problem with $p = 2$, results in optimal deployment of nodes in the network.

III. IMPLEMENTING THE FLUX LINES

In this section we define a new notion of connectivity which deals with implementing routes that can approximately follow the information flux lines. In literature [13], the notion of network connectivity is defined as existing at least one path from each node in the network to every other node in the network. Considering information flow vector fields, a different notion of connectivity must be defined. As discussed earlier in [1], we assume there exists a single path between every location in the network to the destination. These paths are mathematically defined as directed curved lines starting at a location (x, y) and ending at the destination. Therefore, paths are not determined by the location; they are rather abstract paths representing the desired paths for transmission of information. Now, routes are defined as sequences of sensor

nodes starting from the source to the destination. The goal is to approximate information paths, also called flux lines, with routes. This approximation is justified if sensor nodes are distributed dense enough in the network. If the number of nodes is large enough to approximate information flux lines, the paths are implementable. A more accurate definition of implementability is given in next section, where we define implementability for a simple rectangular network and generalize it for more sophisticated network topologies in the sequel.

Intuitively, a network needs more sensor nodes to be implementable than to be connected based on its standard definition. This is investigated in Fig. 2. It shows a simple rectangular network with a source located at the left-most and a sink at the right-most of its geometry, and a single flux line between the source and the sink which is the straight line connecting the two nodes. Sensor nodes are uniformly distributed throughout the network. The goal is to approximate the flux line with a piece-wise linear route within the margins specified by dashed lines below and above the flux line. These margins are specified arbitrarily for this specific example; however, as we show later in the sequel, the margins have a linear relation with the transmission range of sensor nodes and are not specified arbitrarily in general.

Starting from the source, each node tries to find a node within its transmission range, specified by the red circles around the nodes, with the most progression toward the sink. In Fig 2(a), although nodes can find a route from the source to the sink and make the network connected, but the route has passed the margins. Hence, the number of nodes is not enough to make the flux line implementable. In Fig. 2(b) nodes are increased such that they can find a route within the margins. This time, the network is connected and the information flow is implementable.

Our goal in this paper is to investigate on the relation between density of nodes and implementability of flux lines. This is done by finding the relation between the magnitude of information flow vector field and the density of nodes. Once this relation is found, it can be used in two different ways. First, assuming that $\vec{D}(x, y)$ is known at every point of the network, this relation can be used to find the order of nodes needed to implement the information flux lines. Second, this relation can be used as the cost function to find the optimal information flow $\vec{D}(x, y)$ that requires minimum number of nodes to carry the traffic from the information source to the destination.

From now on, notions of implementability and connectivity are used interchangeably in the paper.

IV. THE RELATION BETWEEN NODE DENSITY AND INFORMATION FLOW

A. Why a quadratic relation fails

We start with a simple network example and a simple scheduling scheme, and define a new notion of connectivity which is equivalent to ability to implement information flow flux lines with piece-wise linear routes. Then we prove that a quadratic relation between the magnitude of information flow

vector field and the density of nodes does not guaranty implementability. We will generalize this notion of connectivity to other network topologies in the sequel.

Consider a rectangular geometry of width W and length L , as depicted in Fig. 3, with a line source at the top producing information with a total rate of θ bps. The data sink is a line BS located at the bottom of the network. Sensors are distributed uniformly at random throughout the network's geometry and relay information to their downstream. Let C denote the transmission capacity of each node. Now, the information flow can be represented by $n_0 = \frac{\theta}{C}$ flux lines each relaying C bps to the sink. Due to symmetry, the distance between the flux lines is assumed to be uniform dividing the geometry into n_0 sub-regions of equal width. The goal from now on is to implement the flux lines with piece-wise linear routes.

Assume a fixed transmission range of δ for each node. Since no information load is generated inside the geometry, all sensor nodes relay a uniform amount of traffic and the magnitude of information flow vector field is equal to $|\vec{D}| = \frac{\alpha C}{\delta}$, where $\alpha \leq 1$ is a constant specified by the MAC protocol used for delivering information in each sub-region. In order to avoid large footprints and high interference from neighboring sub-regions, the transmission range is assumed to be a linear function of the distance between the flux lines. Therefore, if the distance between the flux lines decreases, the transmission range of each sensor within the rectangular geometry decreases accordingly. The linear relation between the two will make the magnitude of information flow scale linearly with the total load θ . As we show later in the paper, this linear relation is a requirement imposed by the divergence property of \vec{D} .

Fig. 4 shows the network divided into sub-regions of width δ . As mentioned earlier, the width of sub-regions does not affect the order of sensor nodes needed to implement the flux lines as long as it is linearly proportional to the transmission range. Information flux lines are assumed to be at the center of sub-regions directing from the top to the bottom. The parts of the network shown by dashed lines are guard zones added between the sub-regions to prevent interference. The width of each guard zone is chosen to be $\delta\Delta$, where Δ is a nonnegative constant.

In order to prevent interference between sub-regions, the wireless channel can be divided into two sub-channels where every other sub-region shares the same channel. The existence of guard zones guaranty that no two nodes sharing the same channel in different sub-regions will be less than δ apart from each other. Therefore, sub-regions sharing the same channel can not cause interference for each other.

Dividing the network into n_0 sub-regions of width δ and guard zones of width $\delta\Delta$, the total width of the network will be equal to $W = n_0\delta(1 + \Delta)$. Substituting n_0 by $\frac{\theta}{C}$ results in the following relation between the transmission range δ and the total load θ :

$$\delta = \frac{WC}{\theta(1 + \Delta)}. \quad (5)$$

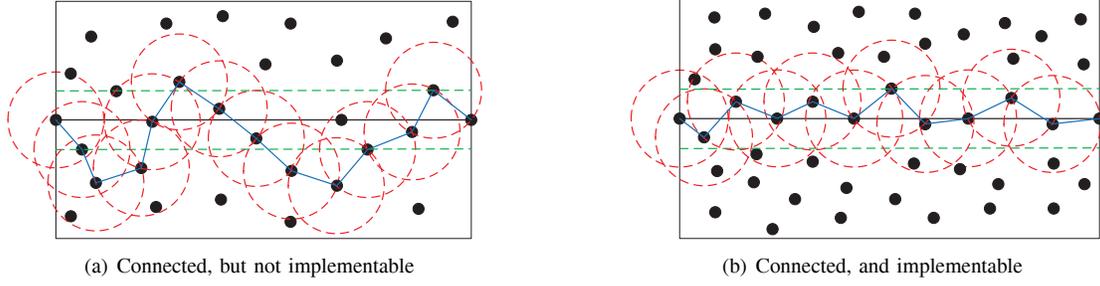


Fig. 2. Comparing the notions of connectivity and implementability for a network.

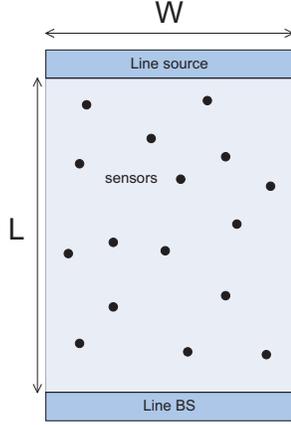


Fig. 3. Rectangular geometry with a line source and a line BS.

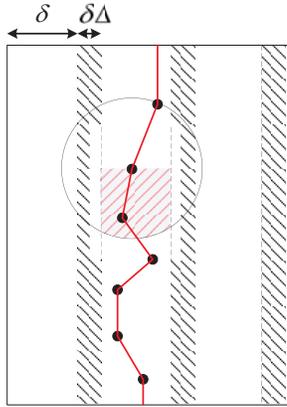


Fig. 4. Sub-regions with guard zones in between.

Increasing the total load θ while C is fixed increases the number of flux lines needed to carry the traffic. As the distance between the flux lines decreases, the transmission range must also decrease to prevent large footprints in the network. This reciprocal relation between θ and δ is clearly shown in relation (5).

With this network setup, the implementability of flux lines is equivalent to having enough nodes distributed uniformly at random throughout the network to be able to implement all information flux lines with piece-wise linear routes within their sub-regions. This definition of implementability of flux lines

can be generalized for other network topologies as well. Consider a general network geometry with an information source and destination. The network geometry can be divided into smaller rectangular geometries, where the size and orientation of each rectangle is chosen such that the information flow inside the rectangle is approximately parallel to its longer sides. Now, the implementation of flux lines in the network can be analyzed locally by analyzing the implementation of flux lines inside each small rectangle similar to the simple example discussed here. The total load carried inside each rectangle is equal to the amount of information flow that enters the rectangle from its side and exits from the opposite side. This is equal to $\theta = |\int_w \vec{D} \cdot d\vec{n}|$, where w denotes either of the shorter sides of the rectangle and $d\vec{n}$ is the differential vector normal to w . Now we can follow a similar approach to the example discussed in this section to find the number of flux lines needed to carry the traffic and form the sub-regions to implement each flux line with piece-wise linear routes.

To implement a flux line, all nodes within its sub-region must have nonempty downstream regions. In other words, all nodes must have at least one node within their transmission range in their downstream toward the BS. Let S_x denote this area for a node in the network. As shown in Fig. 5 this area is a function of x , which is the distance between the node and the left border of its sub-region. Clearly x ranges between $[0, \delta]$, and therefore it can be written as $x = \beta\delta$ for $\beta \in [0, 1]$. Now the area of the downstream region can be computed as follows.

$$S_x = \frac{1}{2}x\sqrt{\delta^2 - x^2} + \frac{1}{2}(\delta - x)\sqrt{\delta^2 - (\delta - x)^2} + \frac{1}{2}\delta^2(\pi - \alpha_1 - \alpha_2) \quad \forall x \in [0, \delta] \quad (6)$$

, where

$$\alpha_1 = \arccos \frac{x}{\delta} \quad \text{and} \quad \alpha_2 = \arccos \frac{\delta - x}{\delta},$$

which can be simplified to

$$S_\beta = \delta^2 \left[\frac{1}{2}\beta\sqrt{1 - \beta^2} + \frac{1}{2}(1 - \beta)\sqrt{1 - (1 - \beta)^2} + \frac{1}{2}(\pi - \arccos(\beta) - \arccos(1 - \beta)) \right] \quad \forall \beta \in [0, 1] \\ = m_\beta \delta^2. \quad (7)$$

This suggests that S_β is a function of δ^2 multiplied by a

coefficient m_β ranging between $m_{min} = 0.7854$ and $m_{max} = 0.9566$ as shown in Fig. 6.

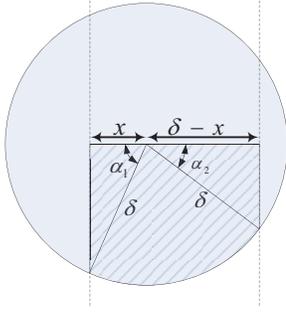


Fig. 5. Downstream region of a node

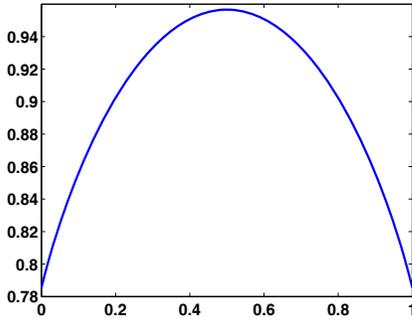


Fig. 6. m_β as a function of $\beta \in [0, 1]$

We will use the following bounds in the proof of the theorems stated in the rest of the paper.

Lemma 1:

$$\exp\left(-\frac{p}{1-p}\right) < 1-p < \exp(-p), \text{ for } |p| < 1 \quad (8)$$

Proof: The right inequality can be easily verified by comparing $1-p$ to the Taylor expansion of $\exp(-p)$ around the origin. For the left inequality, proceed as below:

$$\begin{aligned} 1-p &= \exp(\ln(1-p)), \text{ where} \\ \ln(1-p) &= -\sum_{i=1}^{\infty} \frac{1}{i} p^i \\ &> -\sum_{i=1}^{\infty} p^i = -\frac{p}{1-p}, \text{ for } |p| < 1. \end{aligned}$$

Theorem 1: A total number of $N = O(\theta^2)$ nodes distributed uniformly at random in the rectangular geometry described before does not guaranty connectivity.

Proof: let x_1, \dots, x_N denote the nodes distributed in the network. Furthermore, let S_{β_i} denote the area of the downstream region of node i . Then $Y_i = \sum_{j=1, j \neq i}^N 1(x_j \in S_{\beta_i})$ is a random variable counting the total number of nodes inside the downstream region of node i . Now $p_{\beta_i} = P(x_j \in S_{\beta_i}) =$

$\frac{S_{\beta_i}}{LW}, \forall i, j=1, \dots, N, i \neq j$, which is bounded between $p_{min} = \frac{m_{min}}{LW} \delta^2$ and $p_{max} = \frac{m_{max}}{LW} \delta^2$. Let p_d denote the probability that the network is disconnected. Obviously

$$\begin{aligned} p_d &> P(Y_i = 0) = (1-p_{\beta_i})^{N-1} \\ &\geq (1-p_{max})^{N-1} \geq \exp\left(-\frac{(N-1)p_{max}}{1-p_{max}}\right) \quad (9) \end{aligned}$$

, where we have used Lemma 1 for the last inequality. Now if we choose $N = k\theta^2$ and use relation (5), $Np_{max} = \frac{km_{max}WC^2}{L(1+\Delta)^2}$ which is a positive constant, and

$$\lim_{\theta \rightarrow \infty} p_d \geq \lim_{\theta \rightarrow \infty} \exp(-Np_{max}) > 0. \quad (10)$$

The last line shows that there is a positive probability for the network to be disconnected, and therefore $N = k\theta^2$ nodes is not enough to guaranty connectivity. ■

Note that by adding guard zones between the sub-regions, $\frac{\Delta}{1+\Delta}$ percent of nodes do not participate in routing. These are nodes that are in the guard zones and nodes inside the sub-regions do not communicate with them. However, this does not change our result since we care about the order of nodes rather than specific bounds.

Let $d(x, y)$ denote the density of nodes in the (x, y) coordinate of the network. Due to uniform distribution of nodes, $d(x, y) = \frac{N}{LW}$. On the other hand by using relation (5), $|\vec{D}| = \frac{\alpha C}{\delta} = \frac{\alpha(1+\Delta)}{W} \theta$. Based on Theorem 1, the density of nodes can not have a quadratic relation with the magnitude of the information flow vector field. This contradicts the assumption in [3] based on which it was proved that minimizing the quadratic cost function of $|\vec{D}|$ results in optimal deployment of nodes.

B. A Sufficient Condition For Connectivity

In this section, we will investigate on a sufficient condition for the total number of nodes that guaranty connectivity. This condition is stated in the following theorem.

Theorem 2: Suppose a rectangular geometry with a line source and line BS and a total load of θ bps, as described in the previous section. If $N = k\theta^2 \log \theta$ be the total number of nodes distributed uniformly at random throughout the geometry, the network is connected provided that $k > k_1$, where $k_1 = \frac{2L(1+\Delta)^2}{m_{min}WC^2}$ is a positive constant.

Proof:

$$\begin{aligned} p_d &= P\left(\bigcup_{i=1}^N (Y_i = 0)\right) < NP(Y_1 = 0) \\ &< N(1-p_{min})^{N-1} < N \exp(-(N-1)p_{min}) \end{aligned}$$

Again Lemma 1 is used in writing the last inequality. Now, letting $N = k\theta^2 \log \theta$ and using relation (5), $Np_{min} = \frac{km_{min}WC^2}{L(1+\Delta)^2} \log \theta = \frac{2k}{k_1} \log \theta$ and therefore

$$\lim_{\theta \rightarrow \infty} p_d < \lim_{\theta \rightarrow \infty} \frac{k\theta^2 \log \theta}{\theta^{\frac{2k}{k_1}}}.$$

Hence $\lim_{\theta \rightarrow \infty} p_d = 0$ provided that $\frac{2k}{k_1} > 2$. ■

Theorem 2 states only a sufficient condition for number of nodes that guaranty connectivity. However the simulation result which is shown in next section suggest that $N = O(\theta^2 \log \theta)$ is also a necessary condition for connectivity.

V. SIMULATION

In this section we run a simulation to investigate the relation between θ and total number of nodes that guaranty connectivity. We consider the same rectangular geometry described earlier with $L = W = 10$. It has a line source at the top producing information with a total rate of θ bps, and a line BS at the bottom of the network. Each node is assumed to have a transmission capacity of $C = 1$ bps. The guard zone coefficient $\Delta = 0$ for the purpose of simulation. For each fixed θ , we keep adding nodes one by one uniformly and independent from each other until the network gets connected. For each N , nodes try to route information from the top to the bottom in each sub-region. Each node routes to the node in its downstream region which is closest to the sink. This is shown in Fig. 7 where $\theta = 10$ bps. In part (a), $N = 600$ which is still not enough for all paths to get connected. Unconnected paths are shown by dashed lines. As shown in the figure, three out of ten paths are not connected. The last node in each of these unconnected paths was not able to find a node in its downstream region. In part (b), the number of nodes is increased to $N = 639$, which is the onset for the network to get connected. This occurs when all $n_0 = 10$ paths are implementable. This procedure is repeated 50 times for each θ ranging from 8 bps to 50 bps, taking average from the number of nodes resulted for each fixed θ . Fig. 8 shows the result of this simulation. As can be seen from the figure, the experimental relation between N and θ can be well approximated to be $N = 2.5\theta^2 \log \theta$. Looking back to Section IV-B, $N = k\theta^2 \log \theta$ with $k > k_1$ is a sufficient condition for connectivity. With the parameters of this simulation, $k_1 = 2.5$ which completely matches with our simulation result. This suggests that $N = O(\theta^2 \log \theta)$ is a necessary and sufficient condition for connectivity. However, the necessary part is only shown by simulation.

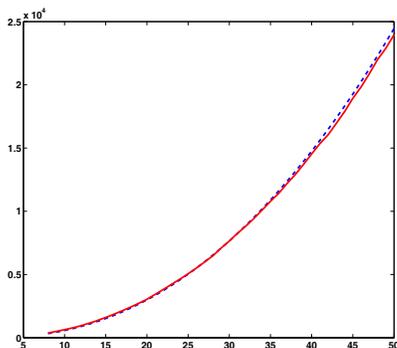


Fig. 8. solid line: simulation result, dashed line: $N = 2.5\theta^2 \log \theta$

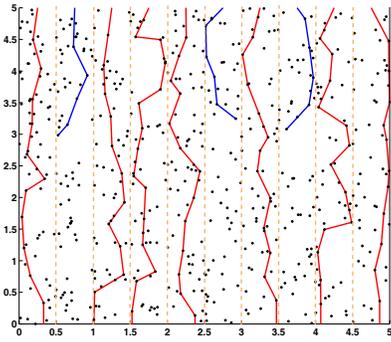
VI. SUMMARY AND CONCLUSIONS

In a dense wireless sensor network, we model flow of information as a continuous function. This model is valid as long as the density of nodes is such that they can approximately follow the information flux lines. In this paper, we provided a more realistic assumption by suggesting a lower bound on the total number of nodes in the network.

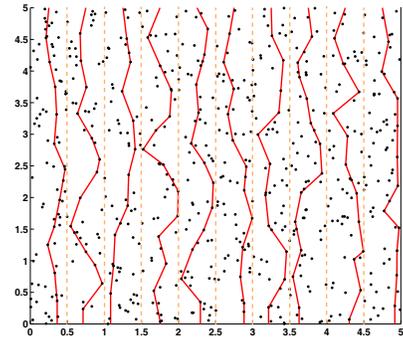
We proved that even in a simple network with a rectangular geometry, a line source and a line BS, a quadratic relation between density of nodes and magnitude of information flow does not guaranty connectivity. This result contradicts the previous belief that minimizing the quadratic cost function results in optimal deployment of nodes by minimizing the number of nodes required to transport information to the sink. We also proved that in a rectangular geometry with a total load of θ bps, a total number of $N = O(\theta^2 \log \theta)$ nodes guaranty connectivity. At the end, the simulation results also suggested the same relation between N and θ , which shows that $N = O(\theta^2 \log \theta)$ is both a sufficient and necessary condition for connectivity. The necessary condition is only shown by simulation, and the mathematical proof is left for a future work.

REFERENCES

- [1] M. Kalantari and M. A. Shayman. Energy Efficient Routing in Sensor Networks. In *Conference on Information Sciences and Systems*, 2004.
- [2] M. Kalantari and M. A. Shayman. Routing in Wireless Ad Hoc Networks by Analogy to Electrostatic Theory. *IEEE International Communications Conference*, 2004.
- [3] S. Toumpis and L. Tassiulas. Packetostatics: Deployment of Massively Dense Sensor Networks as an Electrostatics Problem. *IEEE INFOCOM*, 2005.
- [4] R. K. Ahuja, T. L. Magnanti and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.
- [5] D. Guo and S. Verdu. Replica Analysis of Large-System CDMA. *IEEE ITW*, Paris, France, March-April 2003, pp. 22-25.
- [6] T. Cover and J. Thomas. *Elements of Information Theory*. 1st ed. New York: Wiley Interscience, August 1991.
- [7] G. A. Gupta and S. Toumpis. Optimal Placement of Nodes in Large Sensor Networks Under a General Physical Layer Model. *IEEE Secon*, Santa Clara, CA, September 2005.
- [8] L. Tassiulas and S. Toumpis. Optimal Deployment of Large Wireless Sensor Networks. *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2935-2953, July 2006.
- [9] S. Toumpis. Optimal Design and Operation of Massively Dense Wireless Networks: or How to Solve 21st Century Problems Using 19th Century Mathematics. *Proceedings from the 2006 workshop on interdisciplinary systems approach in performance evaluation and design of computer & communications systems*, New York, NY, 2006, p. 7, ACM Press.
- [10] N. T. Guyen, A.-I. A. Wang, P. Reiher and G. Kuenning. Electric-Field-Based Routing: A Reliable Framework for Routing in MANETs. *ACM Mobile Computing and Communications Review*, vol. 8, no. 1, pp. 35-49, April 2004.
- [11] M. Kalantari, M. Haghpanahi and M. A. Shayman. A p -norm Flow Optimization in Dense Wireless Sensor Networks. *IEEE INFOCOM*, 2008.
- [12] M. Haghpanahi, M. Kalantari and M. A. Shayman. Numerical Flow Optimization in Dense Wireless Sensor Networks. *IEEE GLOBECOM*, 2008.
- [13] P. Gupta, P. R. Kumar. Critical Power for Asymptotic Connectivity. *IEEE Conference on Decision and Control*, December 1998.
- [14] S. Toumpis. Mother Nature Knows Best: A Survey of Recent Results on Wireless Networks Based on Analogies With Physics. *Computer Networks*, vol. 52, pp. 360-383, 2008.



(a) N=600



(b) N=639

Fig. 7. Routing paths- (a): three paths are disconnected. (b): connected network.