Mobility Control with Local Views of Neighborhood in Mobile Networks *

Zhen Jiang
Dept. of Computer Science
West Chester University
West Chester, PA 19383

Jie Wu
Dept. of Computer Sci. and Eng.
Florida Atlantic University
Boca Raton, FL 33431

Robert Kline
Dept. of Computer Science
West Chester University
West Chester, PA 19383

Abstract

Recent work in mobile ad hoc networks, simply MANETs, has drawn attention to the mobility capability of each node. In [5], it is proved that the optimal positions of the relay nodes along a single active flow must lie entirely on the line between the source and destination with each node spaced evenly along such a line. Based on this, we propose two distributed schemes to control the relay nodes in MANETs to approach their optimal positions in the local relative coordinate system, one using one-hop neighborhood and the other using two-hop neighborhood. Unlike the one presented in [5] using only one-hop neighborhood, our methods have no oscillation problem and will converge more quickly. To reduce the overhead in synchronization, outdated neighborhood (lagging by one round of information exchange and update) is used in our two-hop neighborhood based approach. The simulation results shows the substantial improvement on the speed of achieving the optimal configuration and the total moving distance of nodes.

Keywords*: Mobile ad hoc networks (MANETs), distributed algorithm, mobility control, neighbor information (neighborhood).*

1 Introduction

Recent research work has drawn attention to the control schemes in MANETs in order to achieve optimal configuration in data flows for improving communication performance. In [5], the optimal configuration of a single active flow is established and then a complete distributed iterative scheme is proposed to move each node toward its optimal position. Simply, in a synchronous round based system, at each round, every node is required to compute the average of its two neighbors and then move to that new position. As we will discuss later in this paper, the round-off error [1] may cause the oscillation problem. In [5], instead of reaching the expected target position, the node only moves toward that point. The movement is damped and will not suffer from large oscillation. But the oscillation problem is not solved completely. Indeed, the damping slows down the convergence and causes delay in achieving optimal configuration. This will decrease the effectiveness of such mobility control in some real time applications such as target tracking systems [4, 6, 8], in which the tracking data must be transferred quickly enough to catch up with the moving target. Thus, providing a practical scheme which can form a stable data flow quickly while keeping the relay nodes at or close to their optimal positions is becoming increasingly important.

In this paper, we focus on a way to collect and distribute the location information, which is discovered in the GPS-free positioning algorithm [3], so that the above averaging algorithm using such information can converge quickly to be used in data transmission. The challenge is twofold. First, the information must be accurate enough to represent the exact configuration. The effect of the round-off error, which may cause oscillation in the implementation, must be considered. Second, the collection and distribution process must be practical and energy efficient. In other words, it must be simple, have no complex computation, and must not require any costly information, which demands expensive equipment or a lot of multicasting/broadcasting.

A short summary of our approach follows. First, in the averaging process using 1-hop neighborhood [5], the minimum moving distance per round, $MDPR$, as a certain parameter for each node deployed, is introduced to avoid the oscillation caused by round-off error. Instead of using the damping process, a node will reach its target position immediately. However, it will be moved only if the distance from the current position to the target position is larger than this $MDPR$. Then, by using 2-hop neighborhood, the above averaging process can converge more quickly and each node has less movement. To reduce the overhead in synchronization, the use of outdated 2-hop neighborhood, also called

*The work was supported in part by NSF grants ANI 0083836, CCR 9900646, CNS 0422762, CNS 0434533, and EIA 0130806. Contact E-mail: zjiang@wcupa.edu.

inconsistent neighborhood, is discussed. Our simulation results show the substantial improvement of our control schemes on the number of rounds needed in the converging process in the synchronous round based system (i.e., the speed of achieving optimal configuration) and the corresponding total moving distance of nodes.

The remainder of the paper is organized as follows: Section 2 introduces some necessary notions and preliminaries, including the related control schemes and the problems in their implementation. Our control schemes in the local relative coordinate system are presented in Sections 3. Section 4 shows the simulation results. Section 5 concludes this paper and provides ideas for future research.

2 Preliminary

We assume that all the nodes have the same communication range. The nodes inside the communication range are called neighbors and two neighboring nodes are directly connected. To send the data in an efficient way, the power spent at each node is determined by its physical distance to the target neighbor, which is called transmission range. Thus, a network can be represented by a simple undirected graph $G = (V, E)$, where V is a set of vertices (nodes) and E is a set of undirected edges. An undirected edge (u, v) denotes the connection between two neighboring nodes u and v. The *neighbor set* $N(u)$ of node u is defined as $\{w \mid (w, u) \in E \text{ or } (u, w) \in E\}$. Our mobility control algorithm is orthogonal to the routing discovery protocol. Each node u has the location (x_u, y_u), simply denoted by $L(u)$. $\mid L(u) - L(v) \mid$ is the distance between two nodes u and v. $L'(u)$ denotes the target location of u in its movement. $L_u(v)$ is the location of node v in the relative coordinate system at node u. Specifically, $L_u(u) = (0, 0)$.

We assume that a path from the source s to the destination d has been discovered using a routing protocol, e.g., a greedy routing protocol or one of the ad hoc routing protocols. We label the nodes from the source to the destination 0, 1, ..., n. We call node u_0 the source, node u_n the destination, and nodes u_1, ..., u_{n-1} relay nodes. For each relay node u_i, we have $u_{i-1}, u_{i+1} \in N(u_i)$. Thus, any information of u_i can be shared with u_{i-1} and u_{i+1} by the information exchange among neighboring nodes. To simplify the discussion, we describe the schemes in synchronous round based system. All the schemes presented in this paper can be extended easily to an asynchronous round based system; however, to make our schemes clear, we do not pursue the relaxation.

After each relay node knows its position, the optimal configuration of relay nodes for a single active flow is established in [5] as follows: Assume that the energy cost function is a non-decreasing convex function. Then the optimal positions of the relay nodes must lie entirely on the line between s and d. Furthermore, the relay nodes must be evenly spaced along the line. A uniform distributed algorithm that allows the relay nodes to move to their optimal position is also introduced in [5] (see in Algorithm 1). The key ingredient of this algorithm is the simple averaging step. Note that although a relay node computes the average of its two neighbors, the node only moves toward this point, instead of reaching it in one round. In other words, the movement is damped. Such an algorithm is denoted as MCD.

Algorithm 1 (MCD): Mobility control at each relay node u_i [5].

1. Exchange $L(u_i)$ with u_{i-1} and u_{i+1}.
2. Receive $L(u_{i-1})$ and $L(u_{i+1})$. Set $L'(u) = \frac{L(u_{i-1})+L(u_{i+1})}{2}$.
3. Set damping factor $g \in (0, 1]$, move toward $L(u_i) + g \cdot (L'(u_i) - L(u_i))$.

[5] also claimed that MCD will converge and eventually evenly distribute all the relay nodes on the line between s and d. However, in the final converging stage, it only requires a node to move a very short distance. Because of the round-off error, such a distance cannot be expressed precisely in most computer languages such as C and will cause inappropriate round-in or round-out. For example, a 5-hops path contains nodes $s(92.\cdots34, 3.\cdots32)$, $u_1(86.\cdots14, 9.\cdots64)$, $u_2(80.\cdots93, 16.\cdots95)$, $u_3(74.\cdots75, 22.\cdots28)$, $u_4(69.\cdots55, 28.\cdots60)$, and $d(63.\cdots37, 34.\cdots94)$, where $XX.\cdots YY$ stands for the coordinate value beginning with XX as integer part and ending with YY in its decimal part. In the first round, node u_1 is expected to move to $(86.\cdots135, 9.\cdots635)$. However, the target location of u_1 comes out as $(86.\cdots13, 9.\cdots63)$ because a round-off error occurs. Respectively, u_2, u_3, and u_4 will move to $(80.\cdots94, 16.\cdots96)$, $(74.\cdots74, 22.\cdots27)$, and $(69.\cdots56, 28.\cdots61)$. In the next round, all the nodes will bounce back to the positions in the previous round; that is, an oscillation occurs. According to our simulation results, the oscillation will occur in most cases ($> 70\%$) when g is fixed and set to 1. Our simulation results also show that when g is randomly generated, the percentage of achieving stabilization can be improved but the problem cannot be solved completely.

3 Mobility Control in the Local Relative Coordinate System

In this section, we rewrite the 1-hop neighborhood based control scheme in [5] in the local relative coordinate system

Algorithm 2 ($MC1$): 1-hop neighborhood based mobility control at relay node u_i, subject to $MDPR$.

1. Set $L_{u_i}(u_i) = (0,0)$ and build the local coordinate system.
2. Apply the procedure in [3] to obtain $\{L_{u_i}(v) \mid v \in N(u_i)\}$ in the local coordinate system, which only needs one round information exchange.
3. Get $L_{u_i}(u_{i-1})$ and $L_{u_i}(u_{i+1})$. $L'_{u_i}(u_i) = \frac{L_{u_i}(u_{i-1})+L_{u_i}(u_{i+1})}{2}$.
4. If $\mid L'_{u_i}(u_i) - (0,0) \mid > MDPR$, move to $L'_{u_i}(u)$.

and solve its oscillation problem by introducing the constraint of $MDPR$ on mobility of each node. Then, to move the nodes closer to their optimal positions and more quickly, a control scheme based on 2-hop neighborhood is presented. To reduce the extra overhead in synchronization, the use of inconsistent view of neighborhood is introduced in this control scheme.

3.1 1-hop neighborhood based control

At each relay node u_i, the location of neighbors u_{i-1} and u_{i+1} in the relative coordinate system of u_i, $L_{u_i}(u_{i-1})$ and $L_{u_i}(u_{i+1})$, can be determined in the positioning algorithm in [3] in only one round of information exchange. Such an algorithm does not rely on GPS (Global Positioning System) and only uses the distances between the nodes to build the relative coordinate system in which the nodes' positions are computed. Then, we can apply the averaging algorithm in [5] to estimate the target location of u_i, i.e., $L'_{u_i}(u_i) = \frac{L_{u_i}(u_{i-1})+L_{u_i}(u_{i+1})}{2}$. Without damping, node u_i will reach the new position in our scheme before the next round starts. As we discussed in Section 2, caused by the round-off error, the relay nodes may oscillate between two consecutive positions but within a small range. To prevent the node from falling into unstoppable movement and wasting energy on such oscillation, the oscillation range is set as the threshold called *Minimum Moving Distance per Round*, $MDPR$. A relay node will move only if the distance from the current position to the target position is larger than this $MDPR$. This is sufficient to avoid oscillation while keeping the nodes close to their optimal positions. The oscillation range relies on the difference between the calculated approximation of a number and its exact mathematical value. According to the table in [2], it can be set as 0.0001 ($> 2 * 0.00003$ to cover all cases). The detailed process can be seen in Algorithm 2, which is denoted as $MC1$.

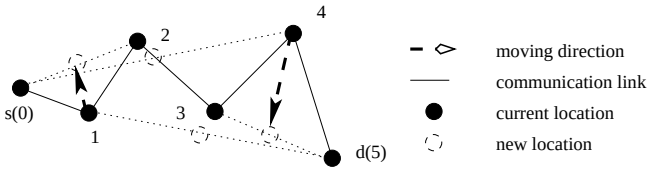

Figure 1. Illustration of the averaging algorithm using 2-hop neighborhood.

After applying $MC1$, the relay nodes may not locate exactly at their optimal positions due to the use of $MDPR$. However, we will show in the simulation results in the next section that such a gap is very small and can be ignored. Actually, the existence of such a gap can be accepted as the tradeoff cost for achieving a stable path quickly. It is noted that a relay node in $MC1$ will move exactly like the one in MCD when $g = 1$. When the $MDPR$ blocks its movement, the other nodes will move like those in MCD in an asynchronous round based system. Due to the use of $MDPR$, the nodes in $MC1$ will stop to move before they reach the final converging stage in MCD; that is, each node will reach its stable status earlier. Thus, the properties of MCD discussed in [5], such as the connection of all the relay nodes, still hold in $MC1$.

3.2 2-hop neighborhood based control

By using 2-hop neighborhood, the above averaging process can converge faster. It is noted that the collection of such information requires the extra overhead of synchronization. In a more efficient way, each node applies the positioning algorithm only once at the beginning of network construction and then uses only one round of information exchange at each round to update the position record reactively when the movement occurs. When two neighboring relay nodes share not only their updated locations but also the recorded locations of all their neighbors, a relay node can collect the location information of all its 2-hop neighbors from the corresponding 1-hop neighbors. In this way, a significant amount of positioning computation and communication for neighborhood collection can be saved while the 2-hop neighborhood based averaging algorithm can still be applied. It is noted that the 2-hop neighborhood collected at that 1-hop neighbor in the previous round may not represent the exact position in the current round. Thus, the 2-hop neighborhood collected in this way is inconsistent. However, as we show in simulation results in the next section, the averaging algorithm using such inconsistent information can still have better performance than $MC1$. The detailed process is shown in Algorithm 3, which is denoted as $MC2$.

In $MC2$, the update of location of a 1-hop neighbor node

Algorithm 3 ($MC2$): Inconsistent 2-hop neighborhood based mobility control at a relay node u_i, subject to $MDPR$.

1. Send $L_{u_i}(u_{i-1})$, $L_{u_i}(u_{i+1})$, and the position change of node u_i in the previous round $L'_{u_i}(u_i)$ to both u_{i-1} and u_{i+1}.
2. Receive information from u_{i-1} and u_{i+1} and update their records at u_i: $L_{u_i}(u_{i-1}) = L_{u_i}(u_{i-1}) + L'_{u_{i-1}}(u_{i-1})$ and $L_{u_i}(u_{i+1}) = L_{u_i}(u_{i+1}) + L'_{u_{i+1}}(u_{i+1})$.
3. Determine the 2-hop neighborhood if any: $L_{u_i}(u_{i-2}) = L_{u_{i-1}}(u_{i-2}) + L_{u_i}(u_{i-1})$ and $L_{u_i}(u_{i+2}) = L_{u_{i+1}}(u_{i+2}) + L_{u_i}(u_{i+1})$.
4. $L'_{u_i}(u_i) = \frac{L_{u_i}(u_{i-2})+L_{u_i}(u_{i+2})}{2}$. Otherwise, only 1-hop neighborhood is available and $L'_{u_i}(u_i) = \frac{L_{u_i}(u_{i-1})+L_{u_i}(u_{i+1})}{2}$.
5. If $\mid L'_{u_i}(u_i) - (0,0) \mid > MDPR$, move to $L'_{u_i}(u_i)$ and save the position change for updates at neighbors in the next round.
6. Rebuild the relative coordinate system of u_i and update each position record according to the new position of origin point.

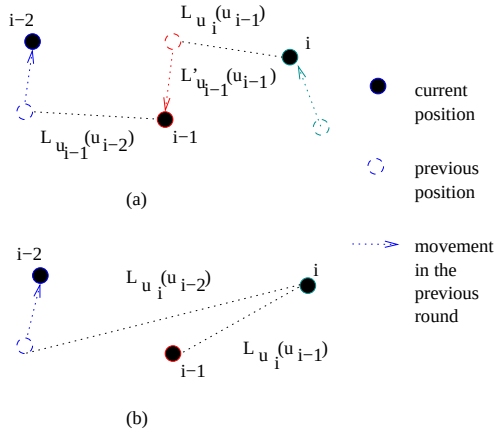

Figure 2. Illustration of information collection in MC2. (a) Information at u_{i-1} before the information exchange. (b) Neighborhood collected at u_i after information exchange.

is triggered by receiving the corresponding position change in the previous round. For a node u and its 1-hop neighbor v, when v moves from $(0,0)$ to $L'_v(v)$ in its local system, the new position in the relative coordinate system of u can be described as $L_u(v) = L_u(v) + L'_v(v)$ (see u_i and u_{i-1} in Figure 2). At node u, after this update for v, the location of another 1-hop relay neighbor of v, i.e., the 2-hop neighbor of u, say node w, can be determined by this $L_u(v)$ and its location at neighbor v, which has been updated after the movement of v in the previous round (see u_{i-2}, u_{i-1}, and u_i in Figure 2). Therefore, we have $L_u(w) = L_v(w) + L_u(v)$. It is noted that the update only occurs at relay nodes because only they can move.

In $MC2$, the location of 1-hop neighbors can be collected precisely. However, the collection of 2-hop neighborhood is lagging by one round in $MC2$. Thus, the averaging process in $MC2$ will converge in an asynchronous way. Figure 3 shows a sample of the convergence in a 6-hop data flow in $MC2$. The relay nodes in $MC2$ were not used in any data flow before, did not move in the previous round, and have the consistent location information at its 2-hop neighbors in the first round (see in Figure 3 (c)). After that, each relay node will move vacillatingly to approach its optimal position (see in Figure 3 (e)).

4 Simulation

In this section, we verify the improvement of our new schemes on speed and cost of converging from a simulator. In a synchronous round based system, the speed of achieving stabilization is measured by the number of rounds needed for convergence. The cost of mobility control schemes primarily comes from the energy consumed in node movement which is determined by the distance a node moves. The simulation is conducted to test the number of rounds needed and the total distance the nodes move in flows with different length from 4-hops to 14-hops in $MC1$ and $MC2$. The transmission range of each relay node eventually achieved in different schemes is also tested to measure the ability of each scheme to achieve optimal configuration. The radius of communication range is set to be 10 meters for each node [7]. All the nodes are deployed randomly. After the deployment, we randomly pick the source node s. From s, for each node selected, we randomly pick its succeeding node (along the path) from its neighbors. In this way, the worst case in each routing protocol can be generated and tested in our simulator. Then, we apply $MC1$

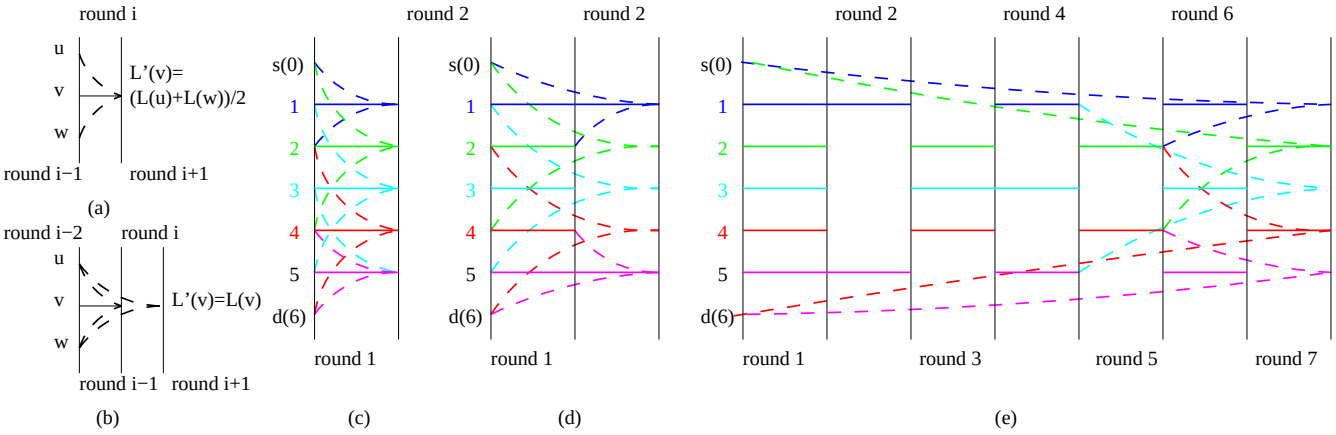

Figure 3. (a) Information collection at node v in round i. (b) Unchanged location in rounds i. (c) Round 1 in $MC2$. (d) Round 2 in $MC2$. (e) Round 7 in $MC2$.

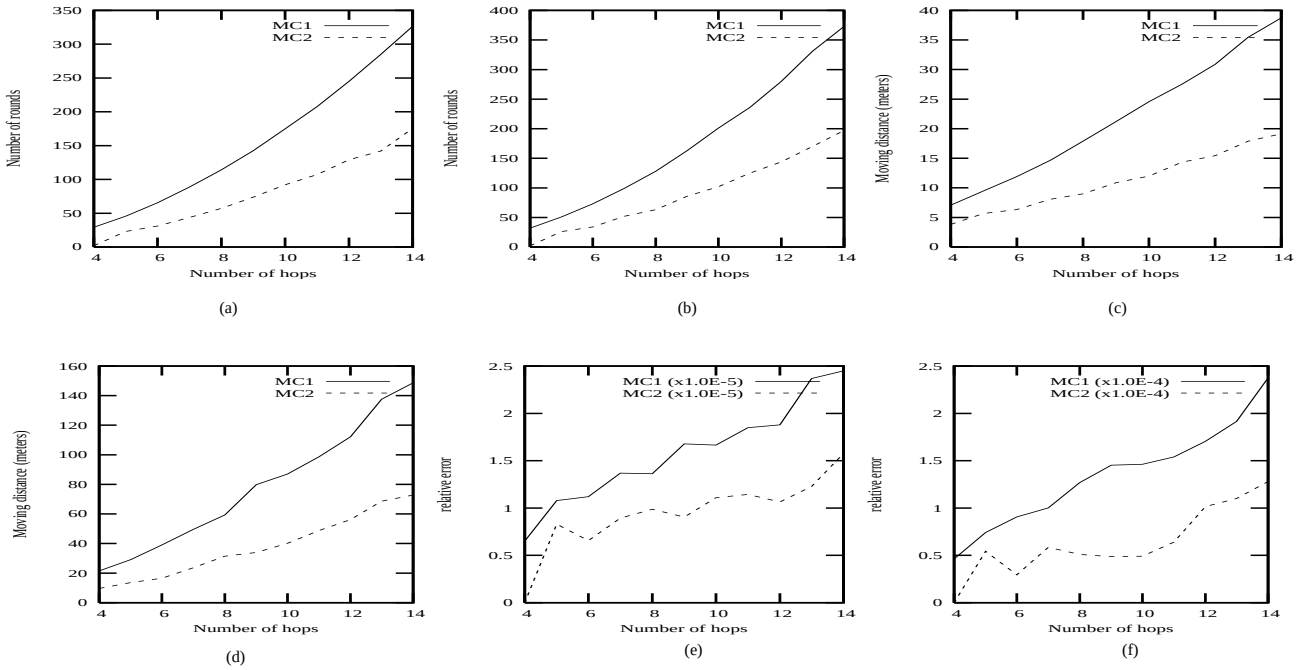

Figure 4. Experimental results when $MDPR = 0.0001$. (a) Average number of rounds. (b) Maximum number of rounds. (c) Average moving distance of a node. (d) Maximum moving distance of a node. (e) Average relative error of the achieved configuration on transmission range to the optimal one. (f) Maximum relative error.

and $MC2$ on those flows and compare the results in the following figures. It is noted that a flow with 4-hops is the least case in which we can apply $MC2$ and we will probably never see any flow longer than 14-hops in an application system with a 10-meter communication range.

We make the following observations from the comparison shown in these figures.

1. With 2-hop neighborhood, even with inconsistent information, each node in $MC2$ needs fewer rounds to become stable (seen in Figure 4 (a) and (b)) and makes more accurate movement (seen in Figure 4 (c) and (d)) than the one does in $MC1$.

2. The simulation results show that $MC1$ and $MC2$ can move the relay nodes close to optimal positions; that is, the relative errors are all less than 0.00025 when $MDPR = 0.0001$, although the $MDPR$ prevents them from reaching their expected target positions. With more accurate information, $MC2$ can move nodes closer to optimal positions than $MC1$ does (see in Figure 4 (e) and (f)).

3. MCD presented in [5] has oscillation in more than 70% of flows. Both $MC1$ and $MC2$ are proved to converge within a certain number of rounds in simulation. They are oscillation-free and can perform better than MCD. $MC1$ is extended directly from MCD by introducing the constraint of $MDPR$. The convergence of $MC1$ and its performance shown in Figure 4 prove the effectiveness of this $MDPR$.

5 Conclusion

In this paper, we introduce the $MDPR$ in the implementation of averaging algorithm so that oscillation caused by round-off error can be avoided. New mobility controls using 2-hop neighborhood, inconsistent with the exact position, are proposed. The process to collect and distribute each kind of location information is also presented. In each control scheme, the connection of relay nodes is guaranteed. A simulation is developed to test the performance and the cost of each control scheme and prove the improvement of our schemes on achieving optimal configuration in a data flow. In our future work, we will apply our results to create a new routing process so that when the nodes move and form dynamic networks, not only can the connected path be constructed but also the data can be transmitted in an energy efficient way. Also, the mobility control will be extended for multiple flows as the connectivity constraints are considered in $MC1$ and $MC2$.

References

[1] http://mathworld.wolfram.com/RoundoffError.html.

[2] http://en.wikipedia.org/wiki/Rounding_error.

[3] S. Capkun, M. Hamdi, and J. Hubaux. GPS-free positioning in mobile ad hoc networks. *Proc. of the* 34^{th} *Annual Hawaii International Conference on System Sciences.* 2001, pp. 3481-3490.

[4] A. Cerpa, J. Elson, M. Hamilton, J. Zhao, D. Estrin, and L. Girod. Habitat monitoring: Application driver for wireless communications technology. *Proc. of ACM SIGCOMM Workshop on Data Communications in Latin America and The Caribbean Costa Rica*. 2001, pp. 3-5.

[5] D. Goldenberg, J. Lin, A. Morse, B. Rosen, and Y. Yang. Towards mobility as a network control primitive. *Proc. of the* 5^{th} *ACM international symposium on Mobile Ad Hoc Networking and Computing (Mobihoc'04)*. May 2004, pp. 163-174.

[6] R. Gupta and S. Das. Tracking moving targets in a smart sensor network. *Proc. of the* 58^{th} *IEEE Vehicular Technology Conference (VTC'03)*. Vol. 5, Oct. 2003, pp. 3035-3039.

[7] D. Tian and N. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. *Proc. of the 1st ACM Workshop on Wireless Sensor Networks and Applications*. 2002.

[8] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor colaboration for tracking applications. *IEEE Signal Processing Mag.* Vol. 19, 2002, pp. 68-77.