

BALANCING COST-QUALITY TRADEOFF IN COOPERATIVE AD HOC SENSOR NETWORKS

Eyuphan Bulut, Jin Zheng, Zijian Wang and Boleslaw K. Szymanski

Department of Computer Science and Center for Pervasive Computing and Networking

Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, USA

{bulute, wangz, szymansk}@cs.rpi.edu and zhengj3@rpi.edu

Abstract—In wireless ad hoc and sensor networks, each node is capable of functioning using only its local information about the environment. However, such a node can reach only locally optimal decisions that may prevent the network from ever reaching the global optimum performance for the given application. To avoid this problem, each node needs to cooperate with others to gain knowledge about the overall network and environment properties so that its decision contributes to the network's global objectives.

This paper models the cooperation between nodes by measuring the level of information sharing between the neighbors. If h -cooperation is applied, each node shares its information with all nodes which are at most h hop away from it. As the cooperation level raises, knowledge of each individual node about its environment increases, thus, it can make better decisions in meeting the main objective of the network application. On the other hand, it also brings extra communication cost and increases the network operation complexity. Therefore, these two contradicting aspects of cooperation cause a cost-quality tradeoff.

In this paper, we investigate the effects of this tradeoff in three different types of sensor network applications: (i) finding an efficient sleep schedule based on sensing coverage redundancy (ii) routing in a network with failure-prone nodes (iii) routing in a network with a mobile sink node. In all of these applications, we simulated different levels of cooperation and showed significant improvements in the overall system quality when the optimal level of cooperation between the network's nodes is chosen.

I. INTRODUCTION

Recent advances in wireless communications and electronics have enabled the development of low-power and

This research was sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

small size sensor devices. Wireless sensor networks (WSN) consist of a large number of such tiny devices capable of sensing in multiple modes. An emerging trend in wireless sensor networks is to use cooperative communication and networking to achieve higher quality of service. A sensor network employing such cooperation has many advantages over conventional networks with either totally local or fully centralized mode of operation. Some of those advantages include: (i) better decision making thanks to sharing resources and information via distributed transmission and processing, (ii) increased reliability of sensed data resulting from coordinated sensing, and (iii) improved efficiency of operation that is achieved via careful coordination of activities. However, such cooperation creates a complex network structure with increased energy cost and messaging overhead. Consequently, a cost-quality tradeoff arises during the design of a sensor network and its applications when deciding the level of cooperative networking.

In this paper, we study cooperation of sensor nodes in three different sensor network applications: (1) the sleep scheduling based on nodes' coverage redundancy, (2) routing in a network with failure-prone nodes, and (3) routing in a network with a mobile sink node.

In the first application, the goal is to find the set of sensors necessary to cover the monitored domain and put the other nodes into sleep mode. A popular approach to decide which nodes should be active is to check whether each node's sensing area is covered by other nodes or not. If each node in the network knows the global network topology with the status of all nodes, then the problem is easy and the only remaining challenge is to prevent message collisions while each node is performing the test. However, in sensor networks, the nodes know directly only the status of their one-hop neighbors. Hence, to minimize the number of active nodes, each node needs to collaborate with others to assess the current coverage of its sensing region. In this paper, we focus on this aspect of the sleep schedule algorithm and measure how the quality of sleeping decisions improves with the level of collaboration invoked.

In the second application, we study the routing of packets in a network with failure-prone nodes. More precisely, when a time interval Δt tends to zero, the probability of a node failing in such time interval tends to a failure rate that is characteristic parameter of the network. We assume that each node knows the whole topology at the beginning. When a node needs to send data, it calculates the shortest route to the sink using its current knowledge of other nodes statuses. Then, it forwards the data to next hop on this path. However, if any node on the path to sink fails, the packet is dropped. For a node to keep the status of other nodes in its local data updated, it needs to collaborate with others. Furthermore, when the cooperation level increases, each node learns the status of more nodes in the network so that the shortest path to the sink node is calculated more accurately. In this paper, we study how the routing performance in such a network is affected by different cooperation levels.

In the last application, we again study routing but this time we assume that nodes are stable but the sink is mobile. We assume that each node knows the next hop towards which it will forward the packets on their way to the destination. However, in a network with a mobile sink node, this information needs to be kept updated, so that packets are forwarded quickly towards the current position of the mobile sink node. To know the current location of sink node, the node needs to collaborate with other nodes that might have the updated location of sink node. When the cooperation level increases, the node can learn the updated knowledge of sink node more quickly. But as in other applications, increased level of communication brings extra communication cost with it. In this paper, we study the performance of routing in such a network with different levels of cooperation.

The rest of the paper is organized as follows. In section II we present some of the related work about cooperative sensor network applications. Then in Section III, we elaborate on the three example applications discussed above. This section also includes the simulation results for all applications. Finally, we close the paper with conclusions and discussions given in Section IV.

II. RELATED WORK

There are many applications where sensors cooperate with each other to increase the network functionality. Yet, only a few papers discuss the benefits of cooperation for sensor network applications or analyze the advantages and disadvantages of cooperation.

A certain level of cooperation is commonly used in target tracking applications. The objective is to track the location of an object within the range of some sensors. Sensors need to cooperate to detect the object and its movements properly and, more importantly, to ensure continuous tracking. Nodes currently tracking the object need also to alert others, especially those into range of which the object is moving. The number of sensors tracking an object affects the tracking errors. In [1], the authors study the sleeping schedule in a target tracking application and discuss the tradeoff between the energy savings and the tracking errors that result from keeping asleep some of the sensors in whose sensing range the object is present. They propose efficient sleeping policies that optimize this tradeoff. [2] is another paper where energy-quality tradeoff for target tracking in wireless sensor networks is discussed.

Cooperation of nodes is also important for routing in delay tolerant networks where most of the time a specific source-to-destination path does not exist. In [3], authors explore the effects of node cooperation for three well-known routing algorithms proposed for delay tolerant networks with respect to the message delivery delay and the transmission overhead. They refer to the cooperation as a node's willingness in receiving or forwarding a message.

To the best of our knowledge, there are no papers discussing the benefits of cooperation for the applications presented in this paper. Some papers imply that introducing cooperation between nodes changes the performance of a network. Yet, they do not compare the network performance with and without cooperation. The cost-quality tradeoff in such applications seems also not have been analyzed in any previous work.

III. COOPERATIVE SENSOR NETWORKING

In this section, we first present our cooperation model then we discuss three different cooperative applications in detail. For each application, we also provide the simulation results that show the effects of cooperation on the performance of the application.

A. *The Cooperation Model*

Many ad hoc definitions of cooperation among network nodes have been created. Yet, all of them share the common underlying idea of a help that a node receives from other nodes. This help sometimes amounts to information sharing, sometimes means a support for another node's activity and sometimes results in not preventing other nodes from doing their duties. In this

paper, we model the cooperation as a function of the range of a node's information sharing. In other words, if h -cooperation is applied, each node broadcasts its information to all other nodes that are at most h hops away. In this definition the information shared between nodes is application specific and can contain locations of nodes, their current status or their knowledge about the position of other nodes (e.g., sink).

In an ad hoc sensor network, cooperation between nodes can improve the network's functionality, but it also requires extra communication that imposes an energy cost on the network. Therefore a careful design of the cooperation is needed to obtain overall optimal performance of the network. In our cooperation model, the cost of h -cooperation to a single node can be approximated as follows:

$$\delta(h) = \sum_{m=1}^N \sum_{i=0}^{h-1} \left(\frac{E_r}{N} \sum_{j \in S_i(m)} n(j) + \frac{E_t}{N} |S_i(m)| \right) \quad (1)$$

where N denotes the total number of nodes in the network, $S_i(m)$ stands for the set of neighbors of node m that is i -hops away, $n(j)$ denotes the number of one-hop neighbors of the node j , and $|\dots|$ returns cardinality of its set argument. Finally, E_r and E_t denote the power needed to receive or transmit one message, respectively. In the formula, the first term counts the average number of recipients of a node's broadcasts and the second term counts the average number of senders of those broadcasts for the neighbors at most h hops away. Note that, if the density of the network is known in advance, average number of neighbors of a node can be easily estimated so that $\delta(h)$ can be computed once for the entire network. Then, this cost result can be used later throughout the lifetime of the network to decide the proper cooperation level that maximizes the benefit.

The improvement that cooperation brings to the network's performance is application specific. Therefore, it is not possible to find a general formula that shows the benefit of cooperation. Moreover, it does not depend only on the cooperation level but also on some other parameters. For instance in first, second and third application, these parameters are coverage range of sensors, percentage of failure-prone nodes and the speed range of the sink node, respectively.

B. Coverage Redundancy Based Sleep Scheduling

Sensor networks are usually deployed with high densities (up to 20 nodes/ m^3 [4]) to extend network reliability and lifetime. However, simultaneous running of all nodes

can cause excessive energy consumption and more frequent packet collisions. In a dense deployment, sensing areas of the sensor nodes may overlap and the same data may be sent to sink from different sensor nodes. To avoid this redundancy, sleep scheduling algorithms are commonly applied in sensor networks. As a result, only a necessary set of sensors stays active and the remaining sensors are put into sleep mode to conserve energy. Some examples of sleep scheduling algorithms are presented in [5], [6], [7].

In coverage redundancy based sleep scheduling, a node is put into sleep mode if it is redundant in terms of its sensing coverage. In other words, if the sensing area of the node is also sensed by other nodes in the network, the node's sensing input is not necessary for the monitored domain coverage and this node can go to sleep mode to conserve its energy. There are two important points here. The first one is the order in which the nodes perform the coverage test and the second one is the hop distance at which a node is considered a neighbor for the purpose of this test. The first issue is usually solved by use of back-off delay to impose a unique order of testing. The second issue creates a cost-quality tradeoff for the test. In this paper, we focus on this second issue but interested reader can look into our previous work [9] for more information about back-off delay solution and how it works.

Neighborhood information used in the test depends of course on the neighbor's definition. In sensor network algorithms, a neighbor is defined most often as a node that is just one hop away from the given node. Also in the coverage test, the sensing ranges of only one-hop neighbors are typically considered. However, some nodes whose sensing areas overlap the sensing area of the node in questions may require more than one-hop to be reached from that node. Excluding such nodes from consideration in the coverage test may change its outcome. Hence, it may be beneficial to generalize the neighbor definition to include h -hop neighbors.

Consider the example illustrated in Figure 1. The active nodes B , C , and D are one-hop neighbors of the node A , and the active nodes E , F , G , H are 2, 3, 4 and 5 hop neighbors of node A , yet they have common sensing areas with it. If node A only considers its one-hop active neighbors while deciding whether it is eligible for sleep or not, it must decide to be non-eligible, since the sensing area of node A is not totally covered by its active one-hop neighbors. However, if other nodes close to A are also considered, the sensing area of node A is totally covered by active nodes. Hence, the coverage

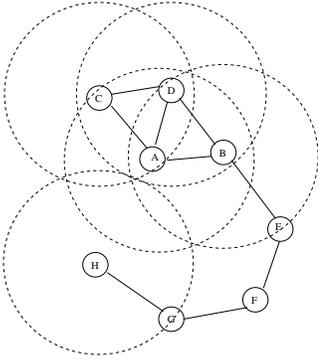


Fig. 1. Node A's sensing area is totally covered by neighbors only if node H is considered in addition to the one-hop neighbors of A.

test will benefit from considering all the nodes which are closer to the sensor node than its sensing area diameter, regardless of their hop distance to that node.

We have simulated such a sleep scheduling algorithm in the following configuration. We deployed identical sensor nodes with 100m sensing range into a square region of the size 500m by 500m. We assigned a small node transmission range (30m) to emphasize the cooperation effect. Such simulation setting is justified for two reasons. First, communication requires a lot of power and creates interference in densely deployed sensor networks. Second, passive sensing does not cause interference, so passive sensing modes often have large sensing ranges. Then, for different numbers of randomly placed nodes, we find a set of active nodes necessary to sense the whole region by applying coverage test to each node in a random order dictated by the back-off delays. To simplify the analysis of the results, we only run the algorithm at the beginning of the network lifetime when every node has the same energy.

Let $u_{h,s}$ denote the average percentage of the nodes sleeping under the sleep schedule with h -hop neighbor definition. By definition, it is a non-decreasing function of h . Using Eq. 1, $\delta(h)u_{h,s}$ is easy to compute and gives a pretty realistic approximation of the energy cost per node of using the different numbers of hops, h , in the neighbor definitions. In the simulations, we use h as a parameter defining the level of cooperation, varying it from 1 to 3. For instance, if $h = 2$, then the nodes that are reachable in one or two hops from the given node are considered in its coverage test.

We ran each simulation scenario 10 different times with different random node deployments over the square region and took the average of the results. Figure 2 shows the average number of nodes that stay active for

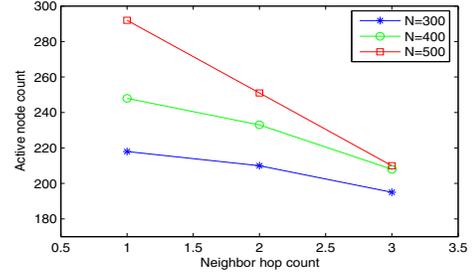


Fig. 2. The numbers of active nodes after running the sleep scheduling algorithm for the different neighbor hop counts and the different numbers of nodes deployed in the region.

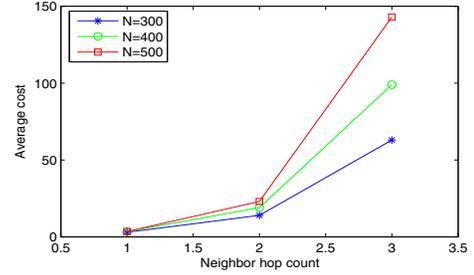


Fig. 3. The average energy costs of running the sleep scheduling algorithm with different neighbor hop counts and different number of nodes deployed. Here the cost unit is the average cost of communication of two nodes.

different node counts deployed in the region and different neighbor hop counts. In each case, this number decreases when the neighbor hop count increases. Since including the neighbors beyond one-hop in the test requires cooperation with other nodes, this indicates the benefit of cooperation in this application. However, as it is shown in Figure 3, the average cost of running the coverage test increases with the level of cooperation. Hence, there is a tradeoff between the cost of the cooperation and the cost of making suboptimal sleep scheduling selections. With the proper level of cooperation more energy can be saved than lost by it.

Let t_{duty} be the interval between two subsequent executions of coverage redundancy algorithm in a network and E_{active} be the energy cost of sensing per node per time unit. Note that, $\delta(h)$, the energy cost when h -coordination is applied, is equal to $E_{h,test}$ which is the energy cost of selecting new duties (sleep or active) of nodes in this specific application. Furthermore, as it is indicated above $u_{h,s}$ denotes the average percentage of the nodes sleeping under the sleep schedule with h -hop neighbor definition. Then, the total energy used per node

per time unit, $E_{h,total}$, can be computed as:

$$E_{h,total} = (1 - u_{h,s})E_{active} + u_{h,s} \frac{E_{h,test}}{t_{duty}}$$

It is clear that if t_{duty} grows to infinity, then $E_{h,total} = (1 - u_{h,s})E_{active}$. Therefore, the performance for very large t_{duty} is a non-decreasing function of h . If $u_{h+1,s} > u_{h,s}$, in other words, if increasing the hop distance of neighbors increases the number of sleeping nodes, then we can compute how large t_{duty} should be for $h + 1$ solution to be more efficient than h solution from the inequality:

$$t_{duty} > \frac{E_{h+1,test}}{E_{active}} + \frac{u_{h,s}(E_{h+1,test} - E_{h,test})}{E_{active}(u_{h+1,s} - u_{h,s})} > \frac{E_{h+1,test}}{E_{active}}$$

For any values of duty period satisfying this inequality, the algorithm with $(h+1)$ -hop neighbor definition uses less energy than the algorithm with h -hop neighbor definition. Consequently, when sensing, communication costs and duty periods for a network are given, $E_{h+1,test}$ exceeding $t_{duty}E_{active}$ can easily be found and the optimal value of h can be calculated.

C. Routing in a network with failure-prone nodes

In an ad hoc sensor network, one of the significant challenges is routing of packets. This topic has been studied deeply and many routing algorithms have been proposed. In this part of the paper, we will focus on a conventional routing algorithm and show how its performance can be improved by network node cooperation.

Assume that we have a sensor network in which each node can fail or go to sleep at any time instance with a probability given by the constant failure rate f . That is, when the time interval Δt tends to zero the probability that any given node breaks in this interval tends to f . In a real life sensor network, such failures are very common because there may be obstacles on the path between adjacent nodes or some nodes may be permanently unreachable (which has the same effect as a failed node for the rest of the network) for example by periodically going to sleep. We assume that each node is given the whole network topology in the network set up phase and it uses this information for routing packets via the shortest path to the destination. All packets sent by a node are assumed to follow the same shortest path to the destination. Since nodes are failure-prone, to route correctly, the node needs to know the current status of other nodes in the network. If any node on the path to destination node is not active, the packet is dropped. Therefore, as a node increases the number of other nodes

of which it can keep the updated status, the paths it chooses for routing becomes more reliable, so that higher delivery rate is achieved.

By h -cooperation, each node can learn the status of all nodes that are h hops away with extra communication cost defined by $\delta(h)$. This introduces a tradeoff between the delivery rate depending on the reliability of path to destination node and the energy cost of cooperation. Using the same idea presented in the previous application, the proper level of cooperation can be found, so that maximum benefit from the network can be obtained.

To simulate such a network, we randomly and uniformly deployed 60 identical nodes with communication range 100m in a 500m by 500m area. In each simulation round, each node creates a packet to a random destination and sends it over the calculated shortest path. To be able to find the shortest path, each node assumes by default that all other nodes are active. Depending on the cooperation level, the node updates their current status. Let F denote a set which contains the nodes that decide their status at the beginning of each simulation round and select to be either active or not, randomly, as a result of failure or going to sleep. We set the failure rate f . It is obvious that when the size of this set increases, the network changes status of the nodes more frequently. We simulated different sizes of set F and different cooperation levels. Furthermore, for each pair of these parameters, we ran 10 different simulations and took the average of results. Figure 4 shows the average delivery ratio obtained in such a network environment. As it is seen clearly from the graph, when size of set F increases, delivery ratio decreases and when cooperation level increases, delivery ratio increases. This indicates the benefit of cooperation in this application.

Other than simulations, we have also find the expected delivery rate, D_r , of the network analytically. Let N_l denote the number of all paths with length l in the network. Since each node selects a random destination and sends a packet over the shortest path to that destination, the expected delivery rate can be formulated by finding the expected path length and the probability that all nodes on this path are active. Given the number of nodes in the network (N), the size of set F and the cooperation level (h), the expected delivery rate when h -cooperation is in use can be denoted as:

$$E[D_r] = \left(\sum_{l=1}^{l=h} N_l + \sum_{l=h+1}^{l_{max}} N_l (P_a)^{(l-h)} \right) / \binom{N}{2}$$

Here, P_a is the probability that a node is active and l_{max} is the longest shortest path between any two differ-

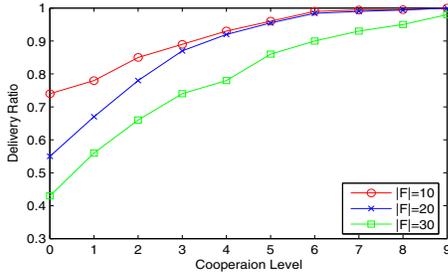


Fig. 4. The average delivery ratio obtained with different number of failure-prone nodes in different levels of cooperation.

ent nodes in the network (measured in inter-node hops). Since the set F is selected randomly among all nodes in the network and for the nodes in set F , it is possible to be active or sleeping with same probability, expected value of P_a is calculated as $E[P_a] = (1 - F/2N)$. To calculate the value of $E[D_r]$, we only need to find the values of N_l for all possible l values in the network. It is easy to find the expected value of N_1 with the parameters given. However when $h \geq 2$, it is a difficult problem. In some studies, this problem has been analyzed from different perspectives and attempts to solve it were based on various ideas. For instance, in [11], authors use Effective Radius (ER) idea and give a general formula to find an approximation of N_l . Using that formula, we can obtain the values of N_l for all possible values of L and calculate $E[D_r]$. For an easy validation of the correctness of this analysis, we obtained instead the values of N_l from the simulation and used it in our formula for $E[D_r]$ calculation. We have observed that the results are very close to those obtained from the presented simulations.

D. Routing in a network with mobile sink node

In this part, we again study a routing algorithm and effects of cooperation on its performance metrics. However, in this case we define the network consisting of stable nodes and a mobile sink node. There are many examples of such network environments from real life in which a robot, a tank or an unmanned aerial vehicle (uav) is such a mobile sink node.

We assume that each node in the network knows the next hop to forward a packet towards the sink node. However, since the sink is mobile, its location information in a node's record may not be updated, so that packet might be forwarded to a wrong neighbor. To know more current location of the sink node, a node needs to collaborate with other nodes and ask their latest information about the sink's location. The more nodes it

asks for this information, the more updated the location of sink node becomes in its records. Accordingly, the delivery rate of all packets can be increased and the average delay of successfully delivered packets can be reduced thanks to increased cooperation.

Our cooperation model enables each node to learn the updated information of sink node from other nodes. As the cooperation level increases, the nodes learn the updated knowledge of the sink node more quickly. But this brings extra communication cost to the network. Therefore, as in the two previous applications, there exists a tradeoff between the delivery rate and the energy consumption of the network. Similarly, we again can find out the proper level of cooperation which provides maximum benefit using the idea presented in the first application.

To simulate this application, we randomly and uniformly deployed one mobile sink node and 70 stable sensor nodes with 60m communication range on a 500m by 500m torus. The sink node moves according to the random direction mobility model defined in [10]. We did two different simulations with different speed ranges of sink node ([2-10]m/s and [10-30]m/s). In each simulation unit, the sink moves to its new location and sends a message to inform the nodes within its range about its existence. All nodes which register the sink within their range update their knowledge. That is, they change next hop to *sink*, hop count to destination to 1 and time of discovery to the current time. We assume that all nodes in the network create packets with an average inter-creation time. At the beginning of the network lifetime, each node finds out how many hops away they are from the sink and the next hop (neighbor) to reach the sink. We assume the shortest route, so next hop is basically a neighbor of the node which is the next node on this shortest path to destination. Each node can learn this easily by flooding. During the simulation, when a node receives a packet, it forwards it to this next hop. However as it is seen in Figure 5, if the next-hop is sink node, according to the current knowledge, but the sink is not there, the packet is dropped.

In each simulation unit, if h -coordination is used, each node sends its information (known hop count to the destination and the time of discovery) to all other nodes which are at most h hops away. When a node receives such messages from neighbors, it selects the most updated one and updates its knowledge according to it. Hence, when cooperation level increases, nodes learn the new location of the sink quickly and route more packets using the correct path.

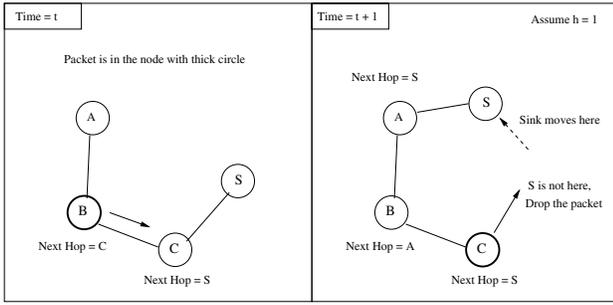


Fig. 5. Packets are dropped if the sink node is not present where it is supposed to be according to the available knowledge.

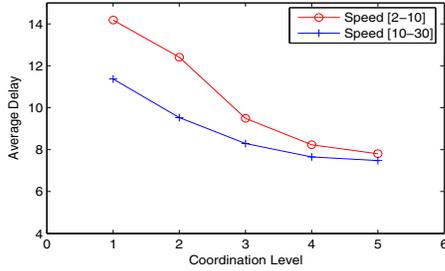


Fig. 6. The average delivery delay obtained with different speed ranges of the sink with the different levels of cooperation.

Figure 6 shows the average delay obtained in the network with different cooperation levels and speed ranges for the mobile sink node. Note that only the successful deliveries contribute to the delay. When the cooperation level increases, the average delay decreases, as expected. Moreover, as it is seen in Figure 7, when the cooperation level increases, delivery rate increases. Additionally, we observe that the speed range of the sink node also effects both the delivery rate and the average delay. At first glance, it seems that when velocity range is high and if this enables the sink node enter other nodes' ranges before the packet delivery time, other nodes can learn the sink's location more quickly. Thus, average delay decreases and delivery rate increases. However, as a future work, we will study this effect and also the effects of other parameters on the performance of applications in detail.

IV. CONCLUSIONS

Cooperative networking is an important method for increasing quality of service in the sensor network applications. In this paper, we analyzed three different ad hoc and sensor network applications, each of which includes one significant aspect of wireless communication (i.e. coverage, failure, and mobile sink). We simulated all

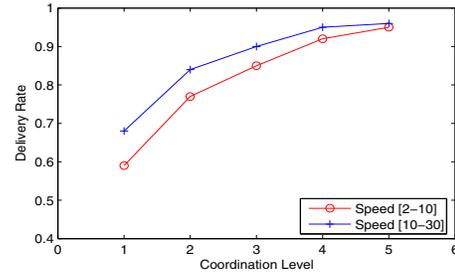


Fig. 7. The average delivery rate obtained with different speed ranges of the sink with the different levels of cooperation.

applications with different levels of cooperation and observed that selecting the proper one leads to a significant increase in the application's performance. On the other hand, we also noticed that the cost of network operation increases with the increase in the level of cooperation. Therefore, we conclude that the cooperation among sensor nodes in a sensor network should be carefully designed considering the cost-quality tradeoff.

REFERENCES

- [1] J. Fuemmeler and V. V. Veeravalli, *Smart Sleeping Policies for Energy Efficient Tracking in Sensor Networks*, IEEE Transactions on Signal Processing, 2007.
- [2] S. Pattem, S. Poduri, and B. Krishnamachari, *Energy-Quality Tradeoffs for Target Tracking in Wireless Sensor Networks*, Workshop on Information Processing in Sensor Networks, 2003.
- [3] A. Panagakis, A. Vaios, and I. Stavrakakis, *On the Effects of Cooperation in DTNs*, 2nd International Conference on Communication Systems Software and Middleware, 2007.
- [4] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, *Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks*, Proc. 7th Annual ACM Conference on Mobile Computing and Networking, Rome, Italy, July 2001.
- [5] H. Zhang and J. C. Hou, *Maintaining sensing coverage and connectivity in large sensor networks*, Wireless Ad Hoc and Sensor Networks, Vol. 1, No. 1-2, pp. 89-123, January 2005.
- [6] E. Bulut and I. Korpeoglu, *DSSP: A Dynamic Sleep Scheduling Protocol for Prolonging the Lifetime of Wireless Sensor Networks*, Proc. IEEE AINA, Canada, 2007.
- [7] J. Branch, G. Chen and B. Szymanski, *ESCORT: Energy-efficient Sensor Network Communal Routing Topology Using Signal Quality Metrics*, Proc. ICN, LNCS, vol. 3420, Springer-Verlag, Reunion Island, France, April, 2007.
- [8] E. Bulut, *Connectivity and Coverage Preserving Sleep Scheduling Mechanism with Predictive Coverage and Multiple Mode Selections in Wireless Sensor Networks*, M.S. Project, Bilkent University, Turkey, 2007.
- [9] E. Bulut, Z. Wang, B. K. Szymanski, *A Cost-Quality Tradeoff in Cooperative Sensor Networking*, in Proc. of ICC, 2008.
- [10] T. Spyropoulos, K. Psounis, C. S. Raghavendra, *Performance Analysis of Mobility-assisted Routing*, MobiHoc, 2006.
- [11] L. Ma, X. Cheng, F. Liu, F. An, J. Rivera *iPAK: An In-Situ Pairwise Key Bootstrapping Scheme for Wireless Sensor Networks*, IEEE Trans. Parallel Distrib. Syst. 18(8), 2007.