I EEE International Conference on Communication, ICC2008,
Proc. Cognitive and Cooperative Wireless Networks Workshop (COCONET), May 19, Beijing, China, pp. 112-117.

A Cost-Quality Tradeoff in Cooperative Sensor Networking

Eyuphan Bulut, 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

***Abstract*—Wireless sensor networks consist of a large number of sensor nodes, each of which senses, computes and communicates with other nodes to collect and process data about the environment. Those networks are emerging as one of the new paradigms in networking with great impact on industry, government and military applications. A sensor network attempts to collect sensing data from the entire domain of its deployment, to process this data to understand phenomena and activities going on in this domain, and finally to communicate the results to the outside world to enable actuators to execute the necessary reactions. However, a sensor node is only capable of sensing events within its limited sensing range, so it has only a localized information about its environment. Hence, to provide the coverage of the entire domain, sensors need to collaborate and share their information with each other. Such sharing increases the knowledge of each sensor about the environment, but it also brings extra communication cost and increases the network operation complexity. In other words, cooperation and data sharing invokes a cost-quality tradeoff in the network.**

In this paper, we study two different sensor network applications: (i) finding an efficient sleep schedule based on sensing coverage redundancy, and (ii) adjusting traffic light periods to optimize traffic flow. In both applications the cost-quality tradeoff arises. In the paper, we study how fast network functionality increases when the level of cooperation raises and how much this increased functionality is offset by the raising cooperation costs. We simulated both applications with different level of cooperation and without it and demonstrated significant improvements in the overall system quality resulting from the properly selected levels of cooperation between the network's nodes.

I. Introduction

Recent advances in wireless communications and electronics have enabled the development of low-power and small size sensor devices. Wireless sensor networks (WSN) consist of a large number of such tiny devices capable of sensing in multiple modes and equipped with a programmable computing chip and wireless communication capabilities. They are utilized in an ever growing range of applications which increasingly require collaboration between sensor nodes to achieve more complex sensing tasks or to improve the quality of information that they provide. In this paper, we consider application specific sensor networks that gather data from their sensors and actuate the network components according to this data. The quality of the network actuation is basically limited by the quality of the data that is available to each individual sensor.

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 can have many advantages over conventional networks with either totally local or fully centralized mode of operation. Some of the significant 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 two different sensor network applications: the sleep scheduling based on nodes' coverage redundancy, and the traffic light adjustment for traffic flow optimization. In the first application, the objective is to find the set of sensors necessary to cover the monitored domain and put the other nodes into sleep mode. At each instance of a network operation, the minimal number of sensor nodes are utilized in sensing while the others are able to conserve their energies. Moreover, the active and sleeping nodes periodically change their roles so that the energy depletion is evenly distributed over all nodes in the network. The challenging part of this problem is to decide which nodes should be active. A well known method applicable here is to check whether each node's sensing area is covered by other nodes in the network. If each node in the network knows the global network topology and the status of each node in it, 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 only know the status of their one-hop neighbors directly. A node with such local knowledge may decide incorrectly that its sensing region is not redundant and therefore will not be able to sleep. 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, the objective is to adjust the traffic lights of an intersection depending on the traffic load so that the waiting times of cars are minimized. There are already some remarkable studies which propose sensor network designs for roads and streets to gather real-time information (i.e. number of cars, their speed, etc.) about the traffic. In most of the previous works, the adjustment of stop light durations in an intersection is determined independently by each node (based on the local traffic around each intersection). However, with the collaboration among the intersection nodes, better decisions can be made in terms of the overall traffic flow. In this paper, we focus on this aspect of the problem and show how much the waiting times of cars decreases with the increasing levels of collaboration among nodes controlling traffic lights at intersections.

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 two example applications discussed above. This section also includes the simulation results for both 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 target and its movements properly and, more importantly, to ensure continuous tracking, currently tracking nodes need to alert the others 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 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.

To the best of our knowledge there are no papers discussing the benefits of cooperation for either the coverage-based sleep scheduling or for the traffic light adjustment selection. Some papers imply that introducing cooperation between nodes changes the performance of a network but they do not compare the network performance with and without cooperation. The cost-quality tradeoff in such applications is also not analyzed in any previous work. In [3], Malik et al. propose a decentralized traffic light control using sensors deployed on the lanes going in and out of the intersection. An intersection agent which gathers data from its lane sensors then adjusts the traffic light durations. The authors mention that with cooperation between intersection agents, the performance of the system could be increased, but such cooperation is neither clearly defined nor discussed.

III. Cooperative Sensor Network Applications

In this section, we first discuss the coverage redundancy based sleep scheduling problem and measure how the collaboration affects the performance of the algorithm. Then, we turn our attention to the traffic light adjustment problem. We discuss the details of the adjustment procedure and present our simulation results which are obtained with and without collaboration between the agent nodes in intersections.

A. 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, if all sensor nodes in such a dense deployment scenario operate at the same time, excessive energy consumption will occur. Moreover, packet collisions will also increase because of the large number of packets being forwarded in the network all the time. 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. This creates a redundancy in data transport and high correlation of activities among the adjacent sensor nodes. To avoid these disadvantages, sleep scheduling algorithms are 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 is not necessary for the monitored domain coverage and 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 backoff 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 for completeness, we also give below a brief summary of a backoff delay solution and explain how it works.

At the beginning of each run of the coverage redundancy algorithm, each node sets its state to active and then chooses a backoff delay by which it postpones the start of its coverage test. When the backoff delay time expires, the node runs the test according to the current states of the nodes in its neighborhood. That means that some of the neighbors, who ran the test earlier, may have changed their state to sleeping, impacting the coverage of the node running the test. The coverage redundancy algorithm is run throughout the network lifetime periodically, so random backoff delay should reflect the changing properties of nodes. A sample backoff delay computation is as follows [8]:

$$t_{backoff-delay} = \left(\alpha \frac{E_r(i)}{E_t(i)} + \beta \frac{N(i)}{N_{max}} + R \right) \times T \quad (1)$$

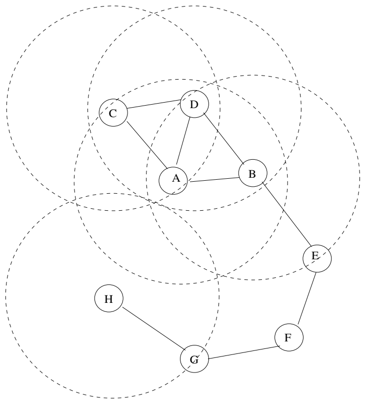

Fig. 1. Node A's sensing area is totally covered by neighbors only if node H is considered in addition to the one-hop neighbors of A.

Here, $E_r(i)$ denotes the remaining battery energy at node *i* and $E_t(i)$ denotes the total battery energy available to the node at the network deployment. A node with small ratio of $E_r(i)$ to $E_t(i)$ should have high priority of moving to sleep mode to promote even energy use throughout the network. $N(i)$ is the neighbor count of node *i* and N_{max} is the maximum neighbor count for any node in a network (this information is established and distributed to all sensor nodes during the initial network set up). A node with high ratio of $N(i)$ to N_{max} should have low priority for turning off its unit because it is likely to effect coverage and connectivity of many other nodes. Moreover, *R* is a uniform random value in the interval $[0, 1-\alpha-\beta]$, *T* is the range from which a random backoff delay is chosen, and α, β are weights of energy and coverage parameters.

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 k-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 and 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 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.

A sensor node knows directly the status of nodes within its transmission range. However, as it is illustrated in the above example, it may be necessary to learn the status of nodes that are more than one hop away to reach a better decision. This can be achieved by cooperation among the nodes. The efficient algorithm to achieve that works as follows. After a node finished its coverage test, if it concluded that it can go to sleep, it broadcasts a sleep message to its neighbors. If k-hop neighbors are considered, this broadcast sleep message is then repeated $k-1$ times by all recipients once, flooding all k or less hop neighbors of the node. Each recipient then removes the original sender from its list of active neighbors. Subsequently, each node knows the status of all nodes that are k or less hops away from it. However, this algorithm increases the messaging cost in the network by a factor of about d for each additional hop included in the neighborhood definition, where d denotes the average connectivity of each node. Since the sleep broadcast is made only by the node that decides to move to sleep mode, the overhead is small in the sparse networks in which the number of redundant nodes is small. On the other hand, the overhead can be very high for dense networks in which high percentage of nodes can sleep at any time.

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, sensing does not cause interference and some passive sensing modes 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 backoff delays computed according to Eq. (1). To simplify the analysis of the results, we only run the algorithm at the beginning of the network lifetime when every node has same energy.

Each node running the coverage test needs to know its active neighbors, so every node in the network needs to keep the status of each neighbor updated. As described above, this is accomplished by a set of broadcasts sent each time the node concludes that it is eligible to sleep. To measure the cost of cooperation for the given hop distance h in the neighbor definition, we approximate the average energy cost of coverage test per sensor node, $E_{h,test}$, as follows:

$$E_{h,test} = \sum_{m=1}^{N} \sum_{i=0}^{h-1} \left(\frac{E_r}{N} \sum_{j \in S_i(m)} n(j) + \frac{E_t}{N} |S_i(m)| \right) \qquad (2)$$

where N denotes the total number of nodes in the network, $S_i(m)$ stands for the set of neighbors of node m i-hops away, $n(j)$ denotes the number of one-hop neighbors of the node j, and $|...|$ returns cardinality of its set argument. Finally, E_r and E_t denote the power needed to receive or transmit one message, respectively. $E_{h,test}$ counts the average number

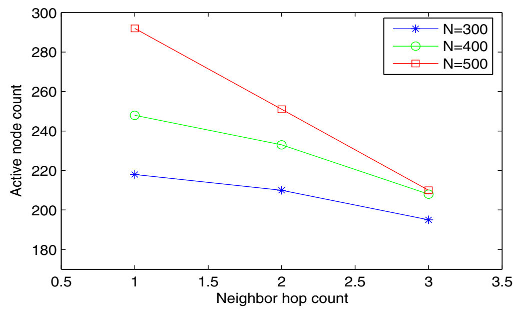

Fig. 2. The numbers of active nodes after running the sleep scheduling algorithm for the different neighbor hop counts and the different numbers of nodes deployed in the region.

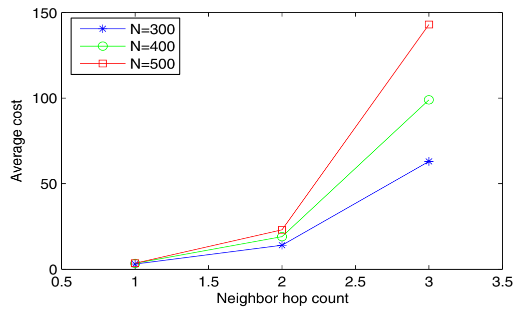

Fig. 3. The average energy costs of running the sleep scheduling algorithm with different neighbor hop counts and different number of nodes deployed.

of recipients of the sleep broadcasts (the first term) and the average number of senders of those broadcasts (the second term) for the neighbors at most h hops away. It can be computed once for the entire network without running the coverage redundancy algorithm. 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. $u_{h,s}E_{h,test}$ 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 deployment over the square region and took the average of the results. Figure 2 shows the average number of nodes that stay active for 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 period of running coverage redundancy algorithm in a network and E_{active} be the energy cost of sensing per node per time unit. As before, $E_{h,test}$ denotes the energy cost of selecting new duties (sleep or active) and $u_{h,s}$ denotes the average percentage of the nodes sleeping during each duty period when h-hop neighbor definition is used in testing. 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:

$$t_{duty} > \frac{u_{h+1,s}E_{h+1,test} - u_{h,s}E_{h,test}}{E_{active}(u_{h+1,s} - u_{h,s})}. \qquad (3)$$

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_{active} can be found and the optimal value of h can be calculated.

B. Traffic Light Adjustment Selection

The traffic light adjustment selection is a well known problem in the intelligent transportation research area. It has been studied by many researchers in the last two decades. The basic challenge is to deal efficiently with the dynamic changes in traffic volume. Many approaches including genetic algorithms, fuzzy logic and reinforcement learning have been proposed for the solution and studied from different perspectives [9], [10]. Furthermore, with the advent of sensor technologies, some sensor network designs have also been proposed for road traffic surveillance and control. In general, maximizing flow of the vehicles, or equivalently reducing the overall waiting time on red lights while maintaining fairness among all vehicle has been the most important goal of this research.

Utilizing wireless sensor networks for providing an intelligent transportation system is still in its infant stage. As every new technology, sensor networks pose challenges to this application. The most important ones are ensuring the accuracy of data generated by sensors, supplying the energy for sensors and designing the proper protocols for data transmission and sharing among sensors.

In this section, we study the problem of collaborative traffic light control with a wireless sensor network. We deploy a sensor network architecture on roads and gather data about the traffic periodically. We also assign a control agent to each intersection to adjust the traffic light durations at that intersection. With this system, we investigate the improvement in the overall waiting time of cars before they reach their destinations accomplished by adjusting the traffic light duration. Additionally, we also analyze the effect of collaboration between intersection agents on the overall waiting time.

We have designed a Java based simulator for the simulation of a simple traffic flow consisting of several interconnected intersections, each equipped with a traffic light. We define a constant t_{period} as the sum of a green and a red light durations and $t_{passage}$ as the time of passage of a car between intersections. A lifetime of a traffic light consists of processing a simple loop. First, its green light is turned on during the assigned green light duration, then its green light turns off and the red light turns on and stay unchanged until the end of red light duration. Then, adjustments to the light durations, if any are needed, are done and the loop executes again.

We assume that cars have constant velocity and they move one unit forward in each simulation time unit. In the simulations, we used a road structure with 3 by 3 (so the total of 9) intersections. We assume that $t_{passage}$ is set to 100 time units. When a car reaches an intersection, it passes if the light is green for its direction, otherwise, it stops and waits until the green light turns on. Moreover, if there are already some cars waiting in the intersection, it stops one unit behind the waiting cars, so we assume that each car occupies one unit of space in the road.

There are various publications that propose sensor network designs for adaptive traffic light control. Our main focus is not on a better design of such a sensor network, but the impact of cooperation. We assume that there is an intersection control agent located at the middle of each intersection and there are several sensor nodes deployed in the road sides which report the number of cars passing in their sensing area. The road side sensors send their data to the intersection agent via multihop packet communication so that intersection agent obtains information about the cars coming from all four directions (east, west, north, and south). However, only cars in the space between the given agent and the nearest neighboring intersection are reported. Figure 4 shows the configuration of a sample road intersection.

For simplicity, we also assume that cars make no turns, and move in either east−west (ew) or north−south (ns) direction. Thus, when the light is green for *ew* cars, it is red for *ns* cars. Likewise, when the light is red for *ew* cars, it is green for *ns* cars.

In the simulations, we use the following car arrival intervals. In all three roads in all directions except east, one car starts its trip in every 10 time units. The car departure intervals from east are 10, 5 and 20 time units. Furthermore, the car departure interval of a middle road from east direction fluctuates between 30 and 5 in every 100 time units. The goal of this setting is to see how fast the traffic lights adjust their durations in response to traffic intensity changes. We evaluate the overall waiting times of cars under four different light switching methods defined as follows.

Constant duration. In this method, red and green light durations are constant and stay the same regardless of the traffic flow. They are set to half of the t_{period} each in the simulations.

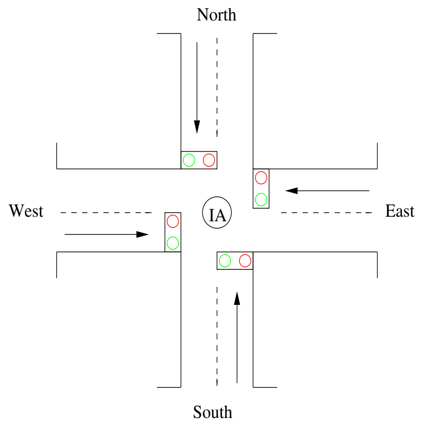

Fig. 4. A road intersection configuration

Counted car adjustment. This technique adjusts the light durations based on the count of cars coming up to the intersection. Let EW denote the number of cars in east-west direction and NS this number for the north-south direction, then the green time of next t_{period} for east-west direction is $t_{period} \times \frac{EW}{EW+NS}$.

Cooperative counted car adjustment. Here, the intersection agents cooperate with neighbor intersection agents deciding the new green and red light durations and use information obtained by their neighbor agents in the formula for the green light duration defined for counted car adjustment method.

Total enumeration. This method finds the optimum value of green light duration by enumerating waiting times of all cars under all possible green light durations and then selecting the duration that minimizes that waiting time. This of course is possible to do only in simulation, so this method is used only for demonstrating the quality of other methods.

Before assigning new values to the light durations, we should also consider the relation between $t_{passage}$ and t_{period}. Information about only those cars that are likely to reach the intersection in the next t_{period} is of interest. There is no need to consider the cars which cannot do that. Since an intersection agent knows only the cars within its range, if the cars out of its range have the ability to reach the intersection, the cooperation with neighbor intersection agents may result in better decisions. Consequently, we need to consider two cases:

If $t_{passage} \geq t_{period}$ then the information that a single intersection agent receives directly is sufficient for deciding the new value of the green light duration. There is no need for cooperation. Otherwise, when $t_{passage} < t_{period}$, information from the neighbor intersection agents can help in a better decision making. Figure 5 shows an illustration of this case. Whether the help is effective depends on the status of the neighboring intersection traffic lights. Consider the leftmost corner in Figure 5. If the neighbor traffic light is red, the cars stop in that neighbor intersection and can not reach the next

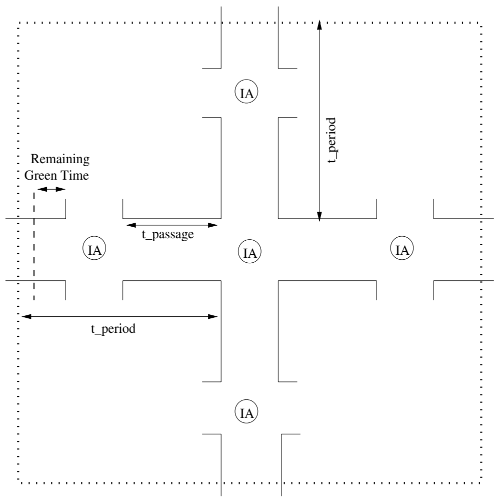

Fig. 5. When t_{period} time is longer than $t_{passage}$, cooperation with other nodes can lead to improved traffic flow. However, only the cars that can pass the intersection within its remaining green time should be considered, as marked in the leftmost corner.

one. On the other hand, if the neighbor traffic light is green, but it changes to red before all cars that have a chance to reach the next intersection can pass, then only information about cars that can pass is of interest.

Accordingly, we ran our traffic simulator for 4000 time units and obtained the following results. Bottom chart in Figure 6 shows the total waiting time of cars when $t_{passage}$ is longer than t_{period}. Since in this case cooperation is irrelevant, we did not show cooperative method in the graph. However, as shown in top chart of this figure, when $t_{passage}$ is shorter than t_{period}, cooperation provides some decrease, compared to the local car counting method, in the total waiting time of cars throughout the simulation.

IV. Conclusions

Cooperative networking is an important method for increasing quality of service in the sensor network applications. In this paper, we analyzed two different sensor network applications: the coverage redundancy based sleep scheduling and the traffic light adjustment selection from the point of view of cooperative networking. We simulated both applications with and without cooperation and observed that the proper level of cooperation 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.

Acknowledgement: 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 Defence, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

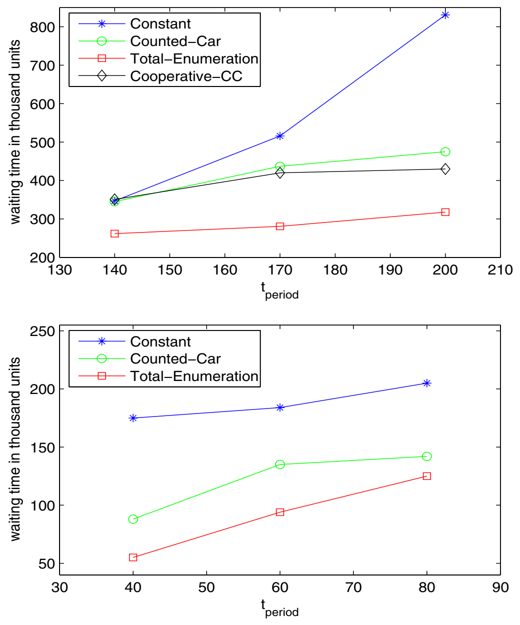

Fig. 6. If $t_{passage}$ is shorter than t_{period}, then cooperation decreases the total waiting time (top chart), otherwise there are no benefits (bottom chart)

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*, Proc. 2nd Workshop on Information Processing in Sensor Networks, 2003.

[3] M. Tubaishat, Y. Shang, H. Shi,*Adaptive Traffic Light Control with Wireless Sensor Networks*, Proc. 4th IEEE Consumer Communications and Networking Conference, 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: An International Journal, 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] M. Wiering, J. Vreeken, J. van Veenen, A. Koopman, *Simulation and optimization of traffic in a city*, Intelligent Vehicles Symposium, IEEE, 2004.

[10] V. Hirankitti and J. Krohkaew, *An Agent Approach for Intelligent Traffic-Light Control*, Proc. 1st Asia International Conference on Modelling and Simulation, (AMS), 2007