Virtual Patrol: A New Power Conservation Design for Surveillance Using Sensor Networks

Chao Gui and Prasant Mohapatra
Computer Science Department
University of California, Davis
{guic,prasant}@cs.ucdavis.edu

Abstract— **Surveillance has been a typical application of wireless sensor networks. To conduct surveillance of a given area in real life, one can use stationary watch towers, or can also use patrolling sentinels. Comparing them to solutions in sensor network surveillance, all current coverage based methods fall into the first category. In this paper, we propose and study patrol-based surveillance operations in sensor networks. Two patrol models are presented: the *coverage-oriented patrol* and the *on-demand patrol*. They achieve one of the following goals, respectively, i)to achieve surveillance of the entire field with low power drain but still bounded delay of detection; ii)to use an on-demand manner to achieve user initiated surveillance only to interested places.**

We propose the "SENSTROL" protocol to fulfill the patrol setup procedure for both models. With the implementation in the GloMoSim simulator, it is shown that patrol on arbitrary path can be set up in a network where each node follows a 98%-time-sleep-2%-time-wake power schedule.

Keywords: Sensor network surveillance, Power conservation, Virtual patrol, Patrol-based surveillance, Trajectory-based forwarding

I. Introduction

Surveillance is a typical application among a wide range of sensor network applications. A surveillance sensor network detects any event of interest in the monitored field. Examples of events include toxic gas leak, structural defect, intruding personnels or vehicles, etc. For any one or multiple occurrences of events, the network is required to generate correct, and more importantly, timely reports about them. In target tracking applications, as a special case, where an event of interest is a moving object, the sensor nodes not only detect them, but also closely track the object's moving path and speed.

A surveillance sensor network is desired to operate unattended for a long time, usually much longer than the battery life-time of a single node. Thus, power conservation is critical and over-deployment of sensor nodes is necessary. In this paper, we consider sleep scheduling in over-deployed sensor networks. Each node can swap between working and sleeping modes and the network only maintains a subset of working nodes. Moreover, sleep scheduling plays an important role in sensor placement planning. The number and the spatial distribution of working nodes at each time are controlled by the sleep schedule.

The interested events often occur sporadically with long and random intervals. Thus, it is very important for the network to let each node have longer sleep time, however, still maintain certain level of ability in detection. The relationship on the amount of sleeping and the network's detectability is only studied very recently in [7], [9], [12]. In [7], several *Quality of Surveillance (QoSv)* metrics are defined for surveillance of moving objects was proposed. In [9], [12], probability based metrics are proposed as well. We can adapt the QoSv metrics in [7] to account for static events by the temporal delay of detection. Let X be the monitored field, and $e(x)$ be the occurrence of an interesting event at location x within the filed. We use $\mathcal{T}$ to denote the time delay of the object being first detected by any sensor. We assume x is uniform random location within X, and let $\mathcal{T}^*$ be the expected value of $\mathcal{T}$. Then, QoSv(X) is defined as: $QoSv(X) \equiv \frac{1}{\mathcal{T}^*}$.

In this paper, we propose a "virtual patrol" model for surveillance operations in sensor networks. The goal is to put the sensor nodes into deeper sleep mode and still maintain the QoSv level. Further, the model can be used for scenarios when the user want to temporarily conduct surveillance within arbitrary sub-region of the monitored field. The organization of this paper is as follows. In Section II, the "virtual patrol" model is introduced. In Section III, we present the SENSTROL protocol, which is used to implement the setup procedure for a patrol. We have implemented the protocol in the UCLA GloMoSim simulator [14]. In Section IV we demonstrate the protocol operations. The performance evaluation results are also shown. In Section V, the related works are summarized. Finally, in Section VI, we provide the conclusions and prospects of future work.

This research was supported in part by the National Science Foundation under the grants CCR-0296070 and ANI-0296034.

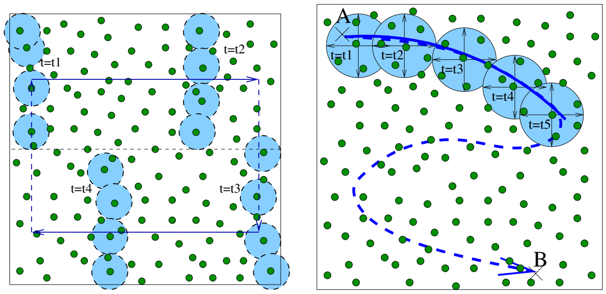
II. The "Virtual Patrol" Model

For the "virtual patrol" model, let us first draw an analogy to the real life solutions of field surveillance. Two types of solutions are possible: stationary watchtowers and patrolling sentinels. Comparing them to sensor field surveillance, current coverage based solutions fall in the first type. Full coverage of the whole field is maintained in all times. Inspired by this analogy, we thus propose a surveillance method that resembles the patrolling sentinel method. In the network field, at each point of time, only a very small subset of the nodes are active in detecting, forming an *active zone*. As the time progresses, the active zone moves along a pre-defined path. We can imagine an active zone as the current location of the patrolling sentinel, then, this sweeping operation can be termed as "virtual patrol." In this section, we will introduce two patrol operation models, namely, the coverage-oriented patrol and the on-demand patrol.

A. Coverage-oriented Patrol

Figure 1(a) shows an example design of coverage-oriented patrol. We will term the imaginary patrolling agent as the *virtual patroller (VP)*. The sensors serving one VP at each time form a vertical straight bar in the field. Thus, the sensors jointly cover a vertical stripe of length being half the width of the network field. The patrol path is as simple as two straight lines in opposite directions. As the vertical stripe moves back and forth in the field, each position is combed by the *patroller*. This procedure of sweeping coverage can be repeated with a given period. The sweep coverage was introduced by Gage [6] in 1992, for surveillance using multi-robot systems.

Under this virtual patrol model, the network's power consumption rate is much lower than the conventional surveillance operations. At each instance of time, only a very small number of sensors are active. However, this method can provide ensured QoSv to the entire field. In each iteration of traversing the coverage-oriented path, each location in the field is swept by the virtual patroller. Any event can be detected with a delay of at most the sweeping period, given that the events



(a) Coverage-oriented patrol. (b) On-demand patrol

Fig. 1. The two virtual patrol models.

persist longer than the sweeping period. By tuning the parameters in the scheme, the network's QoSv can be tuned according to the user's requirement. Besides, this goal is achieved with low-power operation in the network, which is not easily achieved in the conventional operation model.

B. On-demand Patrol

In many scenarios, the user want to conduct a temporary survey to a specific sub-region in the network field. An example scenario is shown in Figure 1(b), where a soldier is assigned to monitor a critical area. The area is the rectangle shown in the figure, in which he deploys a sensor network for help. Position A in the figure is a shield place where the soldier can find hiding and protection against enemy fire. Based on previous information gathered from the network or from other sources, the soldier can plan a patrol path as if he (or she) will traverse in person. An example path is shown in Figure 1(b) as the curve between position A and B. Instead of patrolling in person, the soldier can inject the code that describes the path to the sensor nearest to the start position, point A. The network should ensure that a virtual patrol is conducted by the sensor nodes in the network. The sensors along the path are sequentially woken up and turned off. This wake-up and sleep wave emulates the effect of a "virtual patrol" moving along the path. Events close to the path will be detected and collected along the movement of the "virtual patroller." Compared to personal patrolling, virtual patrolling provides safety from enemy fire and stealth from enemy awareness. Compared to convention sensor network operations, virtual patrolling provides more meaningful and interactive operations. Once a path is submitted to the network, it can be patrolled by the network repetitively for certain amount of time, which is also set by the soldier. This patrol model provides on-demand surveillance operations to the interested places only. Thus, we term this model as the *On-demand patrol*. Power consumption of nodes elsewhere, irrelevant to the patrol path, will be very low.

C. Discussion on Both Models

Although the two models serve different purposes, they can be integrated into a joint operations model. In this case, the coverage oriented patrol can monitor the entire field continuously, and provide a general survey. The information will help the user identify particular places of interest, and then plan and set up an on-demand patrol around those places.

The goal of virtual patrol is to make it an alternative to human based patrolling. However, in some situations, a virtual patrol may not be enough. In these cases, the virtual patrol can provide heuristic information in planning the human based patrol.

III. Implementation of "Virtual Patrol"

To implement the virtual patrol model, we present the SENSTROL (SENSor network paTROL operations) protocol. The procedures are needed for both coverage-based patrol and on-demand patrol. We assume that a new patrol is initiated by a user at the starting point of the patrol path.

A. Problem Description

The following problems needs to be addressed for the implementation of virtual patrol.

1) Given an arbitrary path, how can it be injected into the network and disseminated to all the involved nodes along the path?
2) How to set up the sleep schedules among the involved nodes so that they jointly perform the virtual patrolling?

We assume that the physical deployment of sensor nodes is dense enough and that the desired patrol path is given. At the starting point of the path, the trajectory of the path is coded and injected to the sensor node that is close-by. One possible method to parameterize arbitrary path trajectory is to use Bezier curves. They are used to describe a rich set of arbitrary curves. Certainly, better approximation of the trajectory could be developed and used. Bezier curve is just one example. A cubic Bezier curve, defined by the start and end points and with exactly 2 control points, is represented as:

$$X = Q(u) = Au^3 + Bu^2 + Cu + X_0. \tag{1}$$

Given the coordinates of the start (x0, y0), end (x3,y3), and two control points (x1,y1) and (x2,y2), one can calculate the constants A, B, and C. Thus, the shape of the whole curve is defined.

Given a trajectory $Q(u)$ and a node N_i as shown in Figure 2, we call the point on the trajectory closest to the node as its *correspondent*, denoted as $Q(u_i)$. The corresponding parametric value, u_i, is called the *residual* of N_i on the trajectory. The distance between N_i and its correspondent is called the *residual distance*, denoted as d_i.[1]

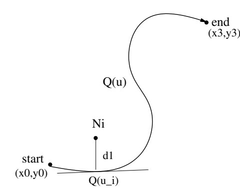
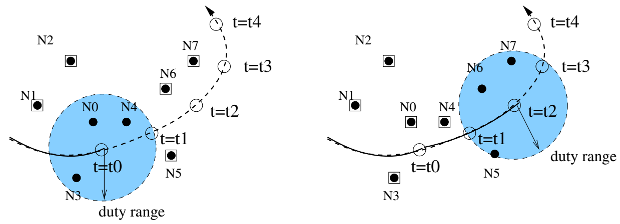


Fig. 2. Trajectory related definitions.

B. Key Concept

The key idea of the protocol is to assume that *there is an imaginary "patroller" moving along the patrol path,* with constant speed v_p. The sensor nodes near or along the patrol path should be active when and only when the "patroller" is within the node's *duty range*. Figure 3 illustrates this operation with an example. In Figure 3(a), the small circles along the trajectory are the marks of the locations of the virtual patroller at several time stamps. At t_0, the patroller is located at the end of the solid part of the trajectory. We draw a shaded circle, centered at the virtual patroller, with radius being the "duty range". At this time stamp, only the nodes that are within this circle should be active, which are nodes N0, N3 and N4. The other nodes should be in sleep. In the figure, each sleeping node is surrounded by a small bracket. Similarly, at t_2, the "patroller" has moved to the new location, and only the nodes N5, N6, N7 need to be awake. Once this rule is carried out to all the involved nodes, the spatial pattern of wake up sequence will be following the desired patrol trajectory with high fidelity. Duty range value would be arbitrated by the user. If we adjust the duty range, we can tune the number of sensor nodes that become on duty at each time. Thus, the intensity of patrolling can be adjusted. We normally set it as the sensing range.

[1]These definitions are given in [20].



(a)At time stamp t0. (b) At time stamp t2.

Fig. 3. The positions of the imaginary patroller at different times. Also shown are the sensors that are awake at time stamps t_0 and t_2.

We require that the physical deployment density satisfy the minimal full coverage condition when all sensors are working. Thus, at any time, the "patroller" is covered by at least one sensor. When multiple sensors are simultaneously working for a patroller, SENSTROL protocol uses an election procedure to select one working node as the leader, which is termed as the current Patroller Host (PH).

The first node of a new patrol has the patrol setup information that will be disseminated into the network. It is composed of the following fields: (i)path trajectory, which contains the parameters for defining the specific Bezier curve; (ii)patrol speed (v_p), which is the constant speed that the imaginary patroller uses; (iii)iteration period (T_i), which is the time the patroller takes to traverse the path from start point to end point; and (iv)patrol duration (T_{PD}), which is the time that the user desires the path to be patrolled.

C. Patrol Setup Phase

The SENSTROL protocol is composed of two phases: the patrol setup phase and the patrolling phase. In the setup phase, the patrol setup information is disseminated to the sensor nodes that are close to the patrol path. Thereafter, each involved sensor node can calculate its own sleep schedule based on received patrol setup information. Then, the sleep schedules are carried out repetitively, which is the main operation for the patrolling phase.

1) Patrol Setup Dissemination: A new patrol begins with the dissemination of patrol setup information. The starting node, which is also the first PH, transmits the PATRO_INFO packets repetitively, carrying patrol setup information. Each packet contains the fields specified in Section III-B. The interval between the broadcasts is denoted as T_{diss}, which translates to the broadcast rate of $1/T_{diss}$ packets per second.

As the time advances, the virtual patroller moves further away from the first PH. When the distance of the virtual patroller and the first PH reaches certain threshold, a new PH will be elected. It will further carry out the broadcasting of setup information. This procedure of PH election and dissemination continues, until the virtual patroller traverses the patrol path.

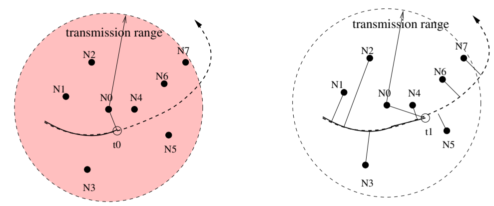
If a normal sensor node completely turns off and sleeps freely, it is no different than a failed node. Thus, all the sensor nodes apply a default sleep schedule when they are not working for any patrols. At each node, time is divided into slots of length T_{slot}. To randomize the slot boundaries of the nodes, each node needs to start an initial sleep time of uniformly random length in the range $[0, T_{slot}]$. At the beginning of each time slot, a node wakes up for a short time period denoted as T_{up}, and then turns off to sleep for the remaining time of the slot. This periodic awake time is intended for the sensor nodes to be able to receive the PATRO_INFO packets. The typical value for T_{slot} is close to $10Sec$ and T_{up} is close to $0.1Sec$. Thus, the typical value of the ratio T_{up}/T_{slot} to be on the order of $1/100$.

For any given node close to the PH, we define T_{delay} as the time delay from when the PH starts transmitting PATRO_INFO packets, to when the node wakes up and receives a packet. Given the broadcast rate and the default sleep schedule setup, we are interested in the property of T_{delay}, and whether all the neighboring nodes of a PH can receive its broadcast within a certain time period.

After a PH starts broadcasting, a node nearby may be sleeping for as long as $T_{slot} - T_{up}$ before first wakeup. To ensure that a node will receive a packet during the first working period, we require T_{up} be equal to T_{diss}. That is, during a node's awake time window, the PH broadcasts at least once. Thus, the property of T_{delay} can be described as the following:

$$T_{delay} = \mathcal{R}_n \cdot T_{diss}, \tag{2}$$

where $\mathcal{R}_n$ is an integer random variable taking any value in $[1, 2, ., n]$ with equal probability, and n equals to $\lfloor \frac{T_{slot}}{T_{diss}} \rfloor$. An example of this procedure is shown in Figure 4.



(a) At time t0, the patroller host, N0, starts to broadcast PATRO_INFO packets periodically. (b) At time t1, neighboring nodes around N0 all knows the patrol setup. (t0 – t1 >= Tslot)

Fig. 4. An example of the dissemination of patrol setup information.

2) Sleep Schedule Setup: Immediately after receiving the PATRO_INFO packet from the PH, a sensor node can calculate its sleep schedule. We use node N6 in Figure 4 as an example to illustrate the calculation steps, which is the same for all nodes.

First, N6 should set up the mapping from "patroller time" and its local time. SENSTROL applies an imaginary reference time, termed as *patroller time*. At the beginning of a patrol, the first node, which is also the first PH, records zero on the *patroller time*, and maps it to its local clock. As the *patroller time* advances, it is maintained at the first PH. When the PH transmits the PATRO_INFO packets, time-stamps of *patroller time* are attached to each packet. A receiver N_i can set up its own mapping using the packet. We use f_i to denote the mapping of *patroller time* to the local time at node Ni. It is of the following form:

$$f_i(t) = t + T_{offset}(i), \tag{3}$$

where t is a time-stamp in patroller time and $f_i(t)$ is the same time-stamp on N_i local time. Since N_i receives the packet P at its local time-stamp $t_{local}(P)$, and P carries a patroller time-stamp $t(P)$, then N_i can get its clock offset by the following equation.

$$T_{offset}(i) = t_{local}(P) - t(P) + \delta, \tag{4}$$

where δ is the minimum error of synchronizing a pair of nodes [2]. Using the *patroller time* as reference, the clocks on the nodes need not to be synchronized. For any event scheduled to happen at a time-stamp on patroller time, all the involved nodes can agree on that absolute time-stamp with negligible error, although they have

[2]The value of δ includes the packet propagation delay, the difference in time-stamping and packet transmission, etc. The delay of going through the protocol stacks should be avoided. The Berkeley MICA-II motes can time stamp a packet as it is transmitted, after all MAC delays have occurred. One implementation synchronizes a pair of nodes to within 2 μsec[8]. Thus, with a careful implementation, δ can be minimized.

different local time value. The issue of difference in clock rates is discussed in Section III-F.

Second, N6 should calculate the time period during which the virtual patroller(VP) is within its duty range. That is, N6 needs to calculate the time-stamp at which the VP moves into (and out of) its duty range, which is denoted as t_{w6} (and t_{s6}). These time-stamps are on the patroller time, and they are $f_6(t_{w6})$ and $f_6(t_{s6})$ on N6's local time, shown in Figure 5.

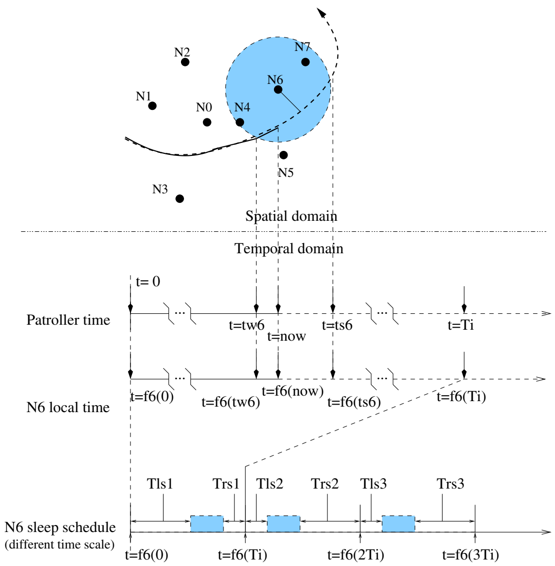


Fig. 5. An example of the sleep schedule calculation conducted at a sensor node, which is node N6 in Figure 4.

Third and the last, N6 sets its future sleep schedule, which is based on the calculated time-stamps in the previous step, and the iteration period T_i received from PATRO_INFO packet. Similar to the default sleep schedule, the time is also divided into slots, but of much larger period. The length of each slot is T_i, which is the time for the VP to traverse the path from start to end. Each slot contains two sleep periods which are separated by one wake period of length T_{wk}. Shown in Figure 5, for the j-th slot, the length of first and second sleep period is denoted as T_{lsj} and T_{rsj}, respectively. For the first slot at N6, the node can sleep till time-stamp t_{w6}, then it should wake up and on sensing duty till time-stamp t_{ts6}, then it can turn off and sleep till time-stamp T_i. For the first slot at N6, T_{ls1} equals to t_{w6}, since the node can sleep till time-stamp t_{w6}. Similarly, T_{wk} equals to $|t_{s6} - t_{w6}|$, and T_{rs1} equals to $|T_i - t_{s6}|$. However, the second slot does not duplicate the first slot. Instead, its temporal structure is exactly the inverted image of the first slot, i.e., T_{ls2} equals to T_{rs1} and T_{rs2} equals to T_{ls1}. Thereafter, the second slot is duplicated into the third and later slots. This is also shown in Figure 5.

3) Patroller Host Handover: When the VP moves close to the boundary of current PH's transmission range, the PH needs to select the next PH and handover the host's duty to its descendant. Under the SENSTROL protocol design, this procedure is described as the following. First, current PH identifies the time when its distance to the VP reaches a threshold of $(1 - \alpha)R_t$, where R_t denotes the transmission range. At this time, and as always, there is a small set of sensor nodes, which have received the PATRO_INFO broadcast and have been awake for its sensing duty. The current PH just needs to select its descendant among the on-duty nodes at this time. An example is shown in Figure 6, in which node N0 is the current PH. At time t_2, the VP has reached the threshold distance, and nodes N5, N6 and N7 are currently on duty, whereas all other nodes are not.

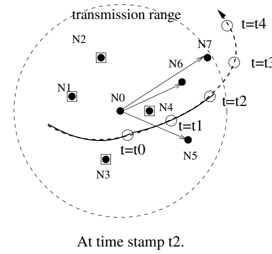


Fig. 6. Illustration of patroller host handover.

The actual handover procedure takes three rounds of messages between the PH and the on-duty nodes. First, the PH starts the procedure by broadcasting a PH_ELECT message, which can either be a separate packet or piggybacked with an immediate PATRO_INFO broadcast. Then, the PH can wait for duration T_{elec} for any returning messages. During this period, it will stop broadcasting PATROL_INFO packets. Upon receiving the PH_ELECT message, each on-duty node chooses a random delay in the range of $[0, T_{elec}]$ and reply to the PH's call for election. In this manner, the replies would not collide at the PH. In the reply message, the sensor node includes its own information such as location and energy level. Then, at the end of T_{elec} time of waiting, the PH can choose its decedent from the on-duty nodes. Possible criteria include the following: to choose the one with lowest residual distance, the one advancing furthest along the path, or the one with highest remaining power. A function combining these various factors can be used as well.

The PH then sends back a confirm message to the selected descendant. The next PH, upon receiving the confirm message can start its role of PH by broadcasting the PATRO_INFO packets. The new PH now also needs to adjust its sleep schedule. The previous schedule makes it awake while the VP is within its duty range. The schedule should be changed so that it will be awake during a longer time when VP is within its transmission range. Thus, the PH handover procedure is finished.

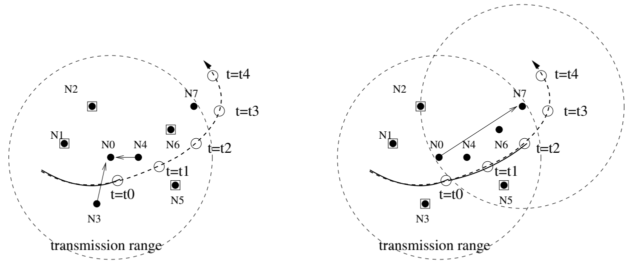
D. Patrolling Phase

When the VP reaches the end point of patrol path, the patrol setup phase is finished. Each sensor nodes along the path has set up its own sleep schedule, and just begin to enter the second time slot in the schedule. As all nodes continue to carry out their schedules, the patrol starts to enter the maintenance phase. In this phase, which should be the major part of a patrol, the PHs do not need to send out PATRO_INFO packets.

As stated in Section III-C.2, the sleep pattern of the second iteration is exactly the inverted image of the first one. This actually means that the patrolling is conducted in the opposite moving direction for the second patrol iteration. For later iterations, the path is all patrolled from the end point to the start point. While the patrol direction does not change the network's quality of surveillance, this design is intended for the possible need of data collection which is presented in Section III-E.

E. Data Collection

Data collection is a special issue for a network under the virtual patrols. The patrol based sleeping has totally changed the network connectivity, namely, the active nodes no longer form a connected topology that needs to span the entire field. However, in an hierarchical sensor network with a backbone for data delivery, data collection will be decoupled with the patrol operations. In such networks, the low end motes only conduct the sensing. There is a group of high



(a) At time stamp t0. (b) At time stamp t2.

Fig. 7. Illustration of data collection along with the patrolling.

end nodes connected by broadband wireless links to form a reliable data delivery service that covers the entire field [19].

In the absence of data delivery backbone, we can make the on-demand patrol to be conducted in the reverse direction, from the end point to the starting point of the patrol path. Once a round of patrol is finished, the information is pushed to the user, who is assumed to be at the starting point. In the patrolling phase, we still imagine that there is a "virtual patroller" moving along the path, from end point to start point. A PH's sleep schedule makes it on duty while the VP is within its transmission range, and the role of a PH has now changed to data collection and forwarding. Whenever a node detects an event of interest, it can broadcast a data packet with the record of its sensor readings. The PH can collect data packets during its awake time. For a surveillance network, the data to be collected are often indication bits, and they are of small volume. This data collection procedure is illustrated in Figure 7(a). At current time t_0, sensor nodes N0, N3 and N4 are on duty, and N0 is the PH. When the next PH wakes up, current PH begin to forward its received data to it. The duty time of consecutive PHs overlap when the VP is within the transmission range of both PHs. Thus, there is plenty of time for several transmission/ACK dialogs to ensure the data is reliable transferred. This procedure is shown in Figure 7(b), where N7 is the next PH. Finally, as the VP arrives the start point, the sensor data gathered during the patrol can be forwarded here to the user. The fundamental limitation of this data collection model is that data is collected and forwarded at the speed of the patroller. This may not satisfy the requirement on QoSv. Whereas with a data delivery service, alert information can travel sever transmission hops within one second.

Under coverage-oriented patrol, to deliver message between two arbitrary locations (rather than nodes), we can use the following method. Let T_P be the time for the patroller to traverse the whole field. The message originator, which is denoted as n_1, will wait at most T_P amount of time before the virtual patroller(VP) reaches its vicinity. Taking this opportunity, n_1 should transmit the message to the patroller host (PH). When the current PH ends its active period, and the next PH down the path starts its active period, the active period of the two should overlap so that current PH can pass the messages it carries to the next PH. The messages consists of the ones passed to it from the previous PH, and the ones it collects during its active period. In this manner, any message is forwarded along the path together with the VP. When the VP reaches the position which the destination of a message is within transmission range, the message will not be passed on, but stay at the PH to be delivered. If the network is dedicated to a surveillance application, all messages will be alarm messages. They will be delivered in the manner to any sink node specified by the user.

Unit			Current Draw	Voltage
MICA -DOT	Processor	Full	8mA	3.3V
		Sleep	$\leq 15\mu A$	
	Radio	Tx	27mA	
		Rv	10mA	
		Sleep	$\leq 1\mu A$	
Data Acq. Unit		Full	1mA	3.3V
		Sleep	0	

TABLE I

DATA SHEET OF MICA-DOT MOTES.

F. Failure Procedure

If one normal sensor node along the path fail, the patrol operation is affected very slightly. The redundancy in sensor deployment ensures that there are multiple on-duty nodes at each time. However, there are two failures that could hamper the patrol operations. We need to discuss the failures and their solutions.

First failure is the clock drifting. With longer advance of time, the effect of the negligible differences among the clocks of the nodes could slowly build up. This can corrupt the operations of patrolling. In this case, we can require the patrolling phase to be bounded to no more than K patrol iterations. Afterwords, all nodes should resume the default sleep schedule, and the patrol setup phase can be repeated.

Second failure is that of the patroller hosts, during the patrol setup phase and the patrolling phase. In the setup phase, if a PH dies before it can conduct the handover procedure to the next PH, the setup phase will be terminated immediately. In the patrolling phase, if the data is collected along with the patroller, a dead PH will terminate and loose the collected data. To detect these cases, we can introduce a keep-alive token, which is forwarded from the end point to the start point, for each round of patrol. Thus, if a patrol is successfully set up, the user at the start point can receive a token after a delay of $2 \cdot Ti$, where Ti is the period of each patrol iteration. The user can initiate another setup procedure when it cannot receive any tokens. If a PH fails during the patrolling phase, the previous PH can detect the absence of its descendant, thus it can initiate a re-election procedure among the on-duty nodes. This is the same procedure described in Section III-C.3.

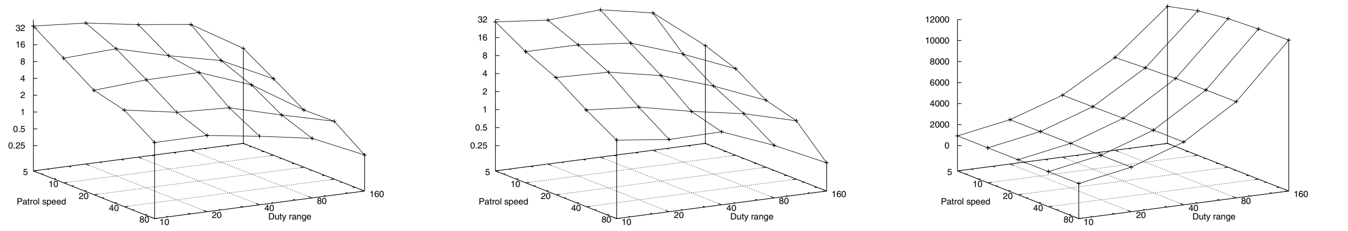
IV. PERFORMANCE EVALUATION

We have implemented the SENSTROL protocol using the GloMoSim[14]. At the physical layer, GloMoSim uses a comprehensive radio model that accounts for noise power, signal propagation and reception. In all the following simulations, we use *Two-Ray* pathloss model for radio propagation. The transmitting power is 10 dBm (10 mW). The receiving sensitivity, which is the measure of the lowest signal power that may be reliably received by the receiver, is set as -65 dBm (0.3 μW). With these settings, the typical transmission range in the GloMoSim simulator is 55.9 meters. In all cases, the sensing range is set to be 20 meters, following a boolean sensing model.

The energy consumption model includes three sources of power draining: the processing unit, radio interface and the data acquisition unit. We use P_{Tx}, P_{Rv},P_{Idle} and P_{Slp} to denote the power consumption of a node at the transmission, receiving, non-sleeping but idle, and sleeping mode, respectively. The actual values are assigned following the data sheet of MICA-DOT motes ,shown in Table I. Thus, we have the value of P_{Tx}, P_{Rv},P_{Idle} and P_{Slp} to be 109mW, 62.8mW, 62.8mW, and 0, respectively.

A. Coverage-oriented Patrols

The simulation setup is the following. The size of the network field is 400m×200m. The number of nodes can be 400, 200 and lower.



(a) Delay of detection (200 nodes). (b)Delay of detection (400 nodes). (c) Energy consumption (200 nodes).

Fig. 8. Coverage oriented patrol.

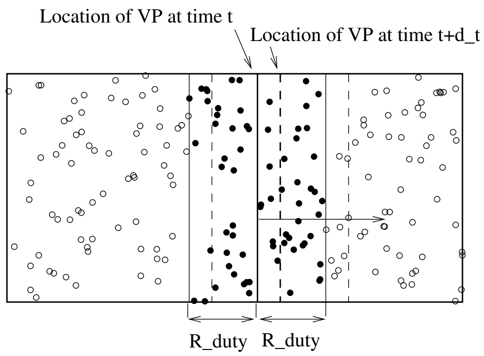


Fig. 9. Scene illustration of coverage-oriented patrol.

Node placement is uniform and random.

The patrol protocol is slightly different than the SENSTROL protocol presented in Section III, and it is illustrated in Figure 9. The virtual patroller (VP) is of a line shape, vertically traversing the field. At each time, if a node's distance to the vertical line is less than R_{duty}, the node should be active. Otherwise, it should be sleeping. As the VP moves horizontally, the zone of active nodes moves along with the VP. This is shown in Figure 9. The VP moves from left border to the right border, and repeats.

Simulation Results on QoSv

To study the QoSv performance, we let an event of interest occur at uniform random position within the field, and at uniformly random time during simulation. It then persists till the end of simulation. The delay from event occurrence to its first detection is recorded. Each experiment is repeated 100 times, and the recorded delay values are averaged. The standard deviation is generally around 3-7% of the mean. The results are shown in Figures 8(a) and (b), in which the node numbers are 200 and 400, respectively.

In both figures, the varying parameters are *patrol moving speed* and the *duty range*. To better understand their influence on QoSv performance, we vary the parameters in a large value range, with [5m/s,80m/s] for patrol moving speed and [10m,160m] for duty range. The axes are plotted in log scale. As shown in both figures, with fixed duty range, the value of detection delay reduces proportionally with the increase of patrol speed. The reason is the following. The period length of patrol iterations is shortened with higher patrol speed, and the detection delay is worst-case bounded by twice the period length. On the other hand, with fixed patrol speed and increasing duty range, the detection delay does not change much, until duty range reaches a high value. To have significant decrease of detection delay, the duty range have to be higher than 40 meters, which is twice the sensing range. This is true in both figures. Finally, our conclusion is that higher patrol speed will effectively reduce detection delay.

Simulation Results on Energy Consumption

Figure 8(c) shows the average energy consumption per node in mJ during the 300-second simulation runs. For our experiment, we only simulate patrolling part of network operations. The procedure repeats itself for each patrol iteration. We found 300s spans over many periods. The number of nodes is 200. The results for the case of 400 nodes are very similar and thus omitted. As shown in the figure, the energy consumption raises dramatically with the increased duty range. With higher duty range, the total awake time at each node is longer. However, with fixed duty range, the energy consumption almost does not change with the different patrol speeds. Even though the period of each patrol iteration has changed, the proportion of active time during each iteration has not changed.

The sensing range is chosen to be less than half of the transmission range, estimated to be 55.9m. If we increase the sensing range, the active zone will cover a larger area, thus, the delay of detection will be reduced. This is the similar effect of increasing duty range. However, the energy consumption will not change much with increased sensing range.

Thus, combining the results on QoSv and energy consumption, we can draw the following conclusion. If we set patrol speed at higher values and duty range at lower values, we can achieve the both goals of lower detection delay and more power saving at each node. The total energy consumption is decided by the duty range, and the default sleep schedule.

B. On-demand Patrols

In this section, we demonstrate the validity of the SENSTROL protocol. The size of the network field is 400m×400m, with 800 nodes following random placement. In Figure 10, the left quarter of the network field is not shown in the snapshots. The patrol path is a half circle, starting at (200,50) and ending at (200,350). The empty dots represent the sleeping nodes and the dark dots represent the active nodes. For the default sleep schedule, T_{slot} and T_{up} are set as 10 seconds and 0.2 second, respectively. Thus, each node by default sleeps for 98% of the time. The patrolling begins at 10.0 second, with node_0 starting to broadcast the PATRO_INFO packets. [3] T_{diss} is set as 0.2 second. Duty range is set a 20 meters.

Shown in Figure 10, at 17.0 second, node_0 has been transmitting patrolling setup packets. If a node receives the packets and determined that it is on the patrol path, the node is double circled in the snapshot. Among the five nodes being double circled, one node has passed its duty time, and yet another one waiting to enter its duty time.

At 22.0 second, the patroller host has been passed away from node_0. Actually, the election of second host is during the interval

[3]The location of node_0 is forced to be at the start point of patrolling path. It serves as the first Patroller Host.

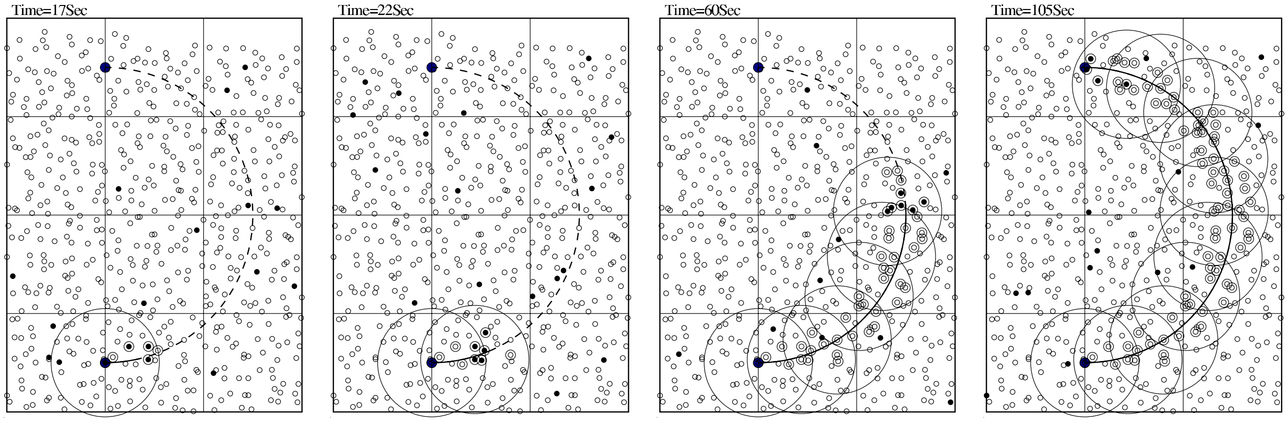


Fig. 10. Snapshots of the setup procedure of an on-demand patrol.

of [19.81, 20.81] second. This can be confirmed by two double-circled nodes, which are out of node_0's transmission range. The later snapshots at 60.0 and 105.0 second are shown as well. The virtual patroller arrives at the end point at 104.2 second.

Energy Consumption Analysis

Figure 11 shows the results of energy overhead per node during the patrol setup. In Figure 11(a), we vary the duty range from 10m to 50m. We also change the default sleep schedule by varying T_{slot}. The three plot lines show nodal energy consumption under different default sleep schedule. They all show that the increase of duty range causes the increase of nodal power drain. With larger duty range, each on-path node should stay up longer while the virtual patroller "passes" beside it. Also notable is that, with T_{slot} more than doubled from 4.0 sec. to 10.0 sec., the power drain does not change in the same ratio. This indicates that, when T_{slot} is 10.0 sec., nearly all nodes along the path has received the patrol information and involved in the setup. Longer T_{slot} will put all nodes in the network into more power-saving status. Thus, a 10.0 sec. T_{slot} is preferable than 4.0 sec.

In Figure 11(b), we vary the moving speed of the virtual patroller from 2 m/s to 8 m/s. At higher patrolling speed, the virtual patroller stays in a nearby node's duty range with less time. It takes less time for virtual patroller to traverse the given path, thus the time for patrolling setup is reduced. These factors contribute reduced energy consumption. Thus, higher patrolling speed is more preferable. However, it is also bounded by the given node density, default sleep schedule and transmission range.

In Figure 11(c), we vary the node density. In the same field, we vary the number of deployed nodes from 500 to 1000. The figure shows that the nodal power drain remains roughly unchanged when the node density is doubled. For each plot-line, the number marking each point is the number of nodes participating in the patrol setup. Along each plot-line, we can observe that with the increase in node density, the number of participated nodes increases, roughly in the same ratio. If we compare the marked numbers in each column, they do not change much with the different T_{slot} values. This trend further confirms the observation in Figure 11(a) that T_{slot} does not affect the patrolling setup very much.

V. RELATED WORK AND COMPARISONS

Power conservation via sleeping has been intensely studied, for nodes in an ad hoc and sensor network. Studies in [2], [4], [5] focus on maintaining optimal topology for data delivery. For coverage aware sleep scheduling, PEAS [18] and GAF [16] can be used for coordinating the sleep pattern. GAF scheme divides the field into grids and only one active node is elected for each grid. Similarly, a MESH scheme is proposed in [7], where horizontal and vertical virtual bars are conceptualized in the field, and only nodes that are located on the bars are active. PEAS uses a probing scheme to avoid the clustering of active nodes. Tuning the probing range can control the density of active nodes. For *quality of surveillance*(QoSv), [7] shows comparisons of these three schemes, together with the trivial random independent sleeping(RIS). Since the *coverage-oriented patrol* is achieving the same purpose as these schemes, we can draw comparisons of them, both in QoSv and in network life-time, using some of the results from [7].

While all other schemes spread the active nodes in the field, the Patrol scheme concentrates them in a small zone. The coverage is only achieved by sweeping the active zone around. For QoSv in monitoring static events, both MESH and Patrol schemes can provide deterministic bound in the delay of detection. [4] For PEAS and GAF, if the active set is static, i.e., active nodes keep working till depletion of energy, a full coverage of sensing ranges is necessary. This requirement induces an active set much larger than MESH and Patrol, however, the delay of detection will be minimal. Both schemes can choose to use a smaller but dynamic active set. The delay of detection, in this case, can only be probabilistically estimated. For RIS, as shown in [7], the active nodes can naturally be clustered, which means a larger delay of detection or larger active set. On the other hand, for QoSv in monitoring moving objects, full coverage is not necessary[7]. It was shown that, by activating roughly the same number of nodes, PEAS and GAF provide better QoSv than RIS. If the Patrol scheme is used for moving objects, the situation could be interesting, and it needs much further study for a conclusion. Since the patrol path is designed for sweep coverage, the chances should be high for a patroller to "catch" the object. However, if the object follows certain path, it can significantly delay or reduce the chance of detection. Moreover, if an intruder knows the patrol path, it can possibly traverse the field while avoiding the patroller.

The network life-time is affected by the size and distribution of active set. If PEAS and GAF use static active set, the network life-time will not be as good as dynamic active set. [5] However, the control overhead for coordinating the sleeping pattern among nearby nodes will add on the communication burden, and reduce life-time. RIS will not have this overhead. For MESH and Patrol, if the sleep schemes are hard-coded to the nodes before deployment, no control overhead is needed. Otherwise, there is control overhead during the setup phase. After that, each node follows their own schemes independently.

[4] In MESH, the virtual bars are sweeping as well, for balancing energy consumption for the entire set of nodes.

[5] In [7], an adaption of PEAS is proposed for this purpose.

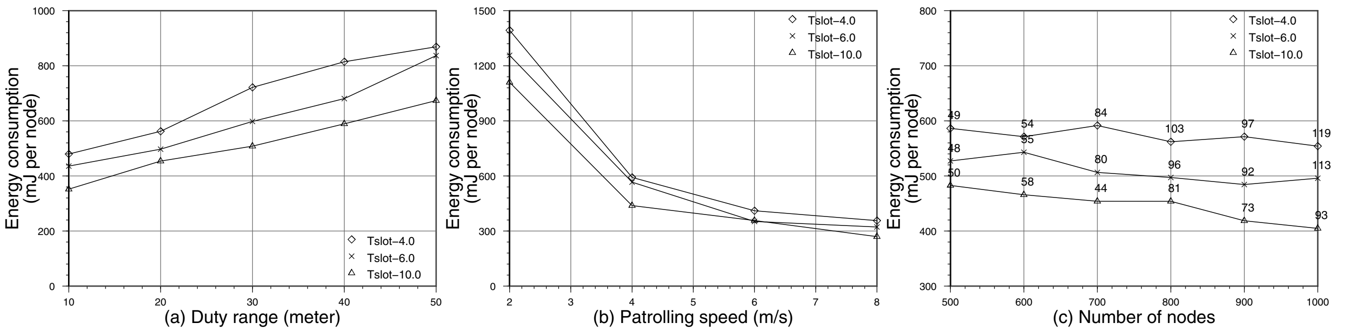


Fig. 11. Energy consumption analysis.

Generally, the active set of Patrol will be much smaller than that of other schemes. Thus, it is expected that the network life-time will be much better in the Patrol scheme. Moreover, the energy consumption will be evenly balanced among the entire nodes.

Our work is also influenced by a number of other related problems. The problem of sensor network surveillance has been studied from various aspects. The network coverage problem are studied in [13], [3], [10], [15], and surveyed by Wu [1]. The network partial coverage is studied in [21], which showed that if only require 95% of the network field to be covered at all times, the achievable life-time increases by 20% compared to full coverage. It is assumed in [9] that the long term average sleep ratio of a node is p. Then the following coverage properties are studied: the probability a point in field being not covered, and the length of the period that the point remains uncovered. A differentiated surveillance model is proposed in [17], in which different parts of the field can be under different intensity of surveillance. In this sense, the SENSTROL can be also categorized into a differentiated surveillance model.

Finally, the concept of trajectory-based forwarding (TBF) in a dense ad hoc network is proposed in [11]. If source node encodes trajectory to traverse and embeds it into each packet, the network assures that the geometric traveled path of a packet follows the desired trajectory with certain fidelity. An implementation design and simulation results are presented in [20]. The SENSTROL protocol extends the TBF by conducting not only forwarding, but also virtual patrolling along an arbitrary path.

VI. Conclusion and Future Work

In this paper, a new operations model for sensor network surveillance is proposed and studied, based on the concept of "virtual patrol". We also identify two possible patrol operations, namely, the coverage-oriented patrol and the on-demand patrol. With the two patrol models, we show that it can achieve not only low power conservation through more sleeping, but also ensured quality of surveillance in the user-specified places or the entire field.

In future, we will adopt a more versatile initiation procedure, in which the user need not be at the starting location. The patrol setup information is forwarded to the node closest to the starting point, which then serves as the first node of the new patrol.

There are several ways to extend the patrol operations model. One is the joint operations model that integrates both patrol models, which will better address user's need in surveillance applications. We also need to investigate the intresting issues of patrolling against moving objects. At last, the current work has not considered multiple patrollers, we will extend the SENSTROL protocol to support these occurrences, with operations including patroller split, patroller merge or rendezvous, etc.

References

[1] M. Cardei and J. Wu. *Handbook of Sensor Networks*, chapter Coverage in Wireless Sensor Networks. CRC Press, 2004.

[2] A. Cerpa and D. Estrin. Ascent: Adaptive self-configuring sensor networks topologies. In *IEEE Infocom*, 2002.

[3] K. Chakrabarty, S.S. Iyengar, H. Qi, and E. Cho. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Transaction on Computers*, 51(12), 2002.

[4] B. Chen, K. Jamieson, and H. Balakrishnan. Span: An energy efficient coordination algorithm for topology maintenance in ad hoc wireless network. In *ACM Mobicom*, 2001.

[5] O. Dousse, P. Mannersalo, and P. Thiran. Latency of wireless sensor networks with uncoordinated power saving mechanisms. In *ACM MOBIHOC*, 2004.

[6] D. W. Gage. Command control for many-robot systems. *Unmanned Systems Magazine*, 10(4), 1992.

[7] C. Gui and P. Mohapatra. Power conservation and quality of surveillance in target tracking sensor networks. In *ACM MOBICOM*, 2004.

[8] J. Hill and D. Culler. Mica: A wireless platform for deeply embedded networks. *IEEE Micro*, 22(6), 2002.

[9] C.-F. Hsin and M. Liu. Network coverage using low duty-cycled sensors: Random & coordinated sleep algorithms. In *International Workshop on Information Processing in Sensor Networks (IPSN)*, 2004.

[10] C.F. Huang and Y.C. Tseng. The coverage problem in a wireless sensor network. In *ACM Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2003.

[11] D. Niculescu and B. Nath. Trajectory based forwarding and its applications. In *ACM International Conference on Mobile Computing and Networking(Mobicom)*, 2003.

[12] S. Ren, Q. Li, H. Wang, X. Chen, and X. Zhang. Probabilistic coverage for object tracking in sensor networks. *ACM Mobile Computing and Communication Review*, 2005.

[13] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *IEEE International Conference on Communication*, 2001.

[14] UCLA. Glomosim. http://pcl.cs.ucla.edu/projects/glomosim.

[15] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *ACM International Conference on Embedded Networked Sensor Systems(SenSys)*, 2003.

[16] Y. Xu, J. Heidemann, and D. Estrin. Geography informed energy conservation for ad hoc routing. In *ACM Mobicom*, 2001.

[17] T. Yan, T. He, and J.A. Stankovic. Differentiated surveillance of sensor networks. In *ACM International Conference on Embedded Networked Sensor Systems(SenSys)*, 2003.

[18] F. Ye, G. Zhong, J. Cheng, S.W. Lu, and L.X. Zhang. Peas:a robust energy conserving protocol for long-lived sensor networks. In *IEEE International Conference on Distributed Computing Systems(ICDCS)*, 2003.

[19] L. Yuan, C. Gui, C. Chuah, and P. Mohapatra. Applications and design for heterogeneous and/or broadband sensor networks. In *First IEEE Workshop on Broadband Advanced Sensor Networks (Basenets)*, 2004.

[20] M. Yuksel, R. Pradhan, and S. Kalyanaraman. An implementation framework for trajectory-based routing in ad-hoc networks. In *IEEE International Conference on Communications (ICC)*, 2004.

[21] H. Zhang and J. Hou. On deriving the upper bound of α-lifetime for large sensor networks. In *ACM MOBIHOC*, 2004.