SCUB: A Distributed, Utility-based Architecture for Closed Loop Control with Wireless Sensor Networks[*]

Kangyuan Zhu and Stephen D. Patek
Systems and Information Engineering
University of Virginia
Charlottesville, VA

ABSTRACT

This paper develops an architecture for task assignment in wireless sensor networks in support of closed-loop tactical operations. Our approach is based on a mathematical framework, which, based on multi-attribute utility theory, allows either end-users to easily express the value (utility) that they would attach to sensor information based on features associated with the data, such as the geographic scope and relevance, observation type, age, and others aspects of the data that relate to system objectives. We have applied the utility-based framework in the design of a distributed task assignment algorithm for a simple pursuit-evasion (patrol) scenario, which involves (i) a set of soldiers patrolling subregions within a larger Area of Interest (AOI), (ii) intruders that randomly appear on the boundary of and traverse across the AOI, and (iii) a network of finite-energy, wireless sensors reporting tripwire data to the soldiers, queuing them in their mission to pursue and capture the intruders. While utility framework is quite general, the algorithm we have developed is “sensor centric” in the sense that individual sensors must decide whether the detections that they have made warrant dissemination and if so to which end-users. Simulation studies based on this scenario show that in terms of mission-level objectives including both intruder capture rate and system lifetime, utility-based framework for task assignment can significantly improve upon baseline task-assignment framework, in which each sensor indiscriminately reports all data to all available end-users.

Keywords

Sensor networks, Network architecture, Task assignment, Utility

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.4 [**Distributed Systems**]: Distributed Applications

General Terms

Algorithms, Architecture, Design

1. INTRODUCTION

The last decade has seen a tremendous amount of research and development directed toward sensing technologies, sensor development, and the application of wireless sensor networks (WSNs). These efforts have targeted applications in many domains, such as military applications, environmental applications, health applications, home applications, and other commercial applications [2, 46, 21]. On the other hand, since sensor networks significantly differ from other kinds of networks [36], various challenging problems are being addressed [5, 21, 2, 1, 34, 47, 43, 36] within the research community, such as architectures, network communications, and data processing.

System architecture is one of the crucial issues that significantly affect system performance. Due to principle differences in the underlying communication technologies, appropriate WSN architectures can vary drastically with respect to individual nodes and to the network as a whole [21], and can be categorized as two classes: (i) Node architecture, which involves the embedded microcontroller, radio transceivers, batteries and the operating system in the individual sensor node [21, 43, 36, 3, 15, 14, 4, 30], and (ii) Network architecture, which takes into account various aspects regarding the system performances of the networks as a whole, such as communication cost, quality of service, network scalability [21, 36, 45, 17, 33, 16, 39].

For individual nodes, many researchers aim to optimize hardware organization, operating system, and power characteristics in a single sensor node. Asada et al. [3] addressed some important and extensive design issues, such as hardware organization, power characteristics, and tiny micro-threading OS issues. Hill et al. [15, 14] introduced a methodology for system design for single-node systems and presented an operating system and three generations of a hardware platform designed as a systematic architecture. Barr et al. [4] designed and implemented a distributed, power-aware, adaptive operating system called MagnetOS specifically targeting ad hoc and sensor networks. It showed that with MagnetOS the system could reduce energy consumption, avoid hotspots

[*]This work was supported in part by the Wireless Internet Center for Advanced Technology (WICAT), an NSF-sponsored IUCRC.

and increase system longevity. Min et al. [30] proposed a power-aware design methodology that emphasized the scalability of energy consumption with factors such as available resources, event frequency, and desired output quality, at all levels of the system hierarchy.

At the network level, researchers have focused on network design and protocol structures that efficiently meet the characteristics of sensor nodes and the requirements of practical applications. A lot of previous work on network architectures has been done for traditional wired networks. Wesson et al. [45] proposed two network structures: the anarchic committee (AC) structure and the dynamic hierarchical cone (DHC) structure. AC structure is a completely interconnected network without any hierarchy, and each node in the network can communicate with any other node. On the other hand, DHC is a tree structure in which nodes are only allowed to communicate directly with adjacent layers. In order to overcome the drawbacks of (i) expensive communications costs in the AC structure and (ii) vulnerability (single points of failure) in the DHC structure, a hybrid structure called "flat tree" structure was proposed, which involves organizing the network nodes as many complete binary trees and completely connecting all the roots [17, 33]. Different from the flat tree structure, Iyengar et al [16] proposed to use deBruijn graph (DG) [39] to connect nodes at each layer to reduce integration errors, which are resulted from the integration of multiple sensor outputs in the presence of noises or faults.

Much of the work mentioned above on network architectures has focused on (or at least) has application to traditional wired networks. More recently, network structures have been proposed that explicitly take into consideration the intended applications and characteristics of wireless sensor networks [34, 2, 1, 5]. Lim [23] proposed an architecture for information dissemination in self-organizing sensor networks, which involve application systems, configurable systems, sensor networking, and physical device layers. Estrin et al. [11] proposed a localized algorithm of directed diffusion to establish flexible, efficient data delivery paths in sensor networks. Heidemann et al. [13] designed a diffusion filter architecture, a software structure for a distributed event system that allows user-supplied software to interact with event routing. Subramanian and Katz [42] proposed a generic architecture for self-configurable systems where a large number of sensors coordinated amongst themselves to achieve a large sensing task.

Despite the diversity of available node-level and network architectures, so far WSNs are mainly used to collect data for specific applications. Hence, data processing and/or other high-level application functions supports have to be integrated with the sensor networks [34, 21]. Recent research tends to integrate data processing requirements with node-level and network-level considerations. Qi et al. [35] presented an architecture for mobile-agent-based distributed sensor networks (MADSN), which compared to various other distributed sensor networks offers some important benefits, such as reduced bandwidth requirements, better network scalability, better extensibility and stability. Ganesan et al. [12] presented a data handling architecture, DIMENSIONS, a system that provided a unified view of data handling in sensor networks, which had the ability to incorporate the resource constraints and spatio-temporal interpretation of the physical world. Madden and Franklin [24] developed the Fjords architecture for managing multiple queries over many sensors and showed that the architecture could limit the sensor resources used while maintaining high query throughput through the test deployed on Interstate 80 near Berkeley. Shen et al. [40] introduced a sensor information networking architecture named SINA, which could facilitate querying, monitoring, and tasking of sensor networks. Users could access information within a sensor network deployed with SINA using declarative queries, or perform tasks using programming scripts.

Although various data processing methods are integrated with node and network architectures, most current research on architectures still focuses on the collection of data, with the flow of information going mainly from sensors to the users. Thus, while there might be a large set of diverse applications supported by sensor data, the responsibility of the WSNs is mainly to collect and transmit data. Future sensor networks are likely to be deployed in a manner that data collection is only part of the overall responsibility of the WSNs. For example, in the military applications of WSNs tactical operations are apparently different from other applications of WSNs in that (a) distinct end-users are working amidst sensors in the field and may directly receive message from individual sensors in a decentralized fashion, (b) end-users are mobile and have dynamic features, (c) sensors make decisions about data dissemination based on both the information they collect and end-users' dynamic features, (d) end-users are intelligent enough to efficiently process the information they received, and (e) standard metrics of network performance, such as data delay and data throughput, may not reflect the overall system objectives. Apparently, in this application of WSNs, the overall system objectives are in mission levels, such as intruder capture rate and system lifetime, which can not be reflected only with standard metrics of network performances. Moreover, it is readily seen that the data collection from individual sensors to the end-users is only part of the system operations. These unique features for tactical operations thrown into question the applicability of existing WSN's architectures and may leave significant opportunities for improvements.

In this paper we attempt to establish an architecture that can reflect overall system objectives for specific applications of WSNs. In particular, we develop a sensor-centric, utility-based (SCUB) architecture for task assignment in wireless sensor networks in which sensors individually make judicious decisions about what information should be forwarded to a collection of mobile receivers based on the "value" of the information encoded within a utility function.

Outline

In the remainder of this section, we review related work in the management of wireless sensor networks. We present the SCUB architecture as well as a baseline architecture called "Send-All-to-All" in Section 2, where we also formulate the optimal of SCUB task assignment problem and further provide a heuristic sub-optimal SCUB solution. In Section 3 we design a testbed called *patrol scenario*, which involves multiple soldiers capturing randomly arriving intruders in a fixed area of interest, to illustrate the application of SCUB architecture with WSNs. In Section 4, we provide the detail parameters in *patrol scenario* as well as the preliminary numerical results to support our assertion that SCUB architecture can have strong impact on task assignment for tactical operations comparing with the baseline architecture. Finally, we wrap up the paper in Section 5 and offer directions for future research.

Literature Review
Both Sensor-centric and utility-based approaches have been widely applied in various WSN research efforts. The benefits of sensor-centric and decentralized approaches are summarized in several papers [27, 37]. Sheth et al. [41] presented a design of a decentralized fault diagnosis system for WSNs, which is capable of distinguishing between multiple root causes of degraded performance and providing efficient feedback into the network to troubleshoot the fault. Wen and Sethares [44] proposed a decentralized algorithm for WSN clustering, with which each sensor adopts a random waiting timer and local criteria to determine whether to form a new cluster or join a current cluster without a centralized controller. Makerenko et al. [27] developed an approach adhering to scalable decentralized algorithms for both data fusion and the decision-making layers of the system for an indoor Active Sensor Network (ASN). Makarenko and Durrant-Whyte [26] presented an algorithm for Bayesian decentralized data fusion (BDDF) and its extension to information theoretic control. Chamberland and Veeravalli [8] investigated a binary decentralized detection problem in which a network of wireless sensors provided relevant information about the state of nature to a fusion center. Sadagopan et al. [38] advocated a systematic decentralized approach to designing networks based on utility functions to achieve the global optimal load for data gathering tree. Mainland et al. [25] presented a self-organizing resource allocation for achieving efficient resource allocation in sensor networks based on the decentralized, utility-defined action selections of individual sensors. Kannan et al. [20, 19] presented a sensor-centric approach with a game theoretic algorithm to achieve an energy-constrained, reliable, data-centric information routing algorithm for WSNs under different constraints. Ridley et al. [37] described the theoretical and practical development of a decentralized air and ground sensing network for target tracking and identification based on the information-filter formulation of the Kalman filter algorithm and information-theoretic methods derived from Bayes theorem.

Utility theories also have been used in various fields of WSNs to represent the value of information. Chen and Sha [9] formulated data transport problem in WSNs as an optimization problem to achieve the maximal amount of utility collected at sinks subject to flow, energy, and channel bandwidth constraints. Byers and Nasser [7] presented a model for numbers of sensors participating sensing to define appropriate global objectives based on utility functions and specify the cost for energy consumption. Byers and Nasser also developed distributed algorithms to maximize the utility derived from the sensor networks over its lifetime. Zhao et al. [48] introduced and developed the definition of information utility and several approximate measures of the information utility, with which the paper was intended to determine the participants in a sensor collaboration by dynamically optimizing the information utility of data for a given cost of communication and computation. Nama et al. [32, 31] characterized the trade-off of system performance and lifetime by considering a cross-layer design problem in a WSN with orthogonal link transmissions, which jointly maximized the network utility and lifetime. Kang and Li [18] used the information utility measurement for decision making in sensor selection of clustering considering three key factors: sensing quality, communication cost and power level. Bian et al. [6] proposed a framework to select a sequence sensor sets which had the maximal utility while not exceeding the available energy for sensor selection techniques.

2. ARCHITECTURAL ISSUES

There are two main questions to resolve in designing an architecture for task assignment: (1) Where should the authority for making decisions about consumption of network resources lie? (2) If control authority is decentralized, what information needs to be exchanged (or fused) in order to ensure effective and efficient operation of the network?

If the sensor network generally has the authority to decide what observations to transmit to what users, there is a wide spectrum of possible architectures. On one extreme, the decision-making could be centralized within a single *resource manager* separate from the sensing functionality of the network, that schedules the transmission of specific observations to specific users based on a global understanding of what observations are available to transmit to which users. Assuming that this global understanding comes for free, centralized decision-making should allow for the best possible trade-off between application performance and system longevity subject to energy constraints. However, unless there are side channels for coordination and control, the network resources that would be needed to keep the centralized decision-maker informed may be comparable to the resources needed to simply report all observations to end-users. It may be that a more decentralized approach is at least as effective.

One possibility would be to have a group of resource managers whose collective control authority is partitioned according to geographic considerations or according to role, so that individual resource managers only resource a local understanding of what observations are available to transmit. Another possibility would be to eliminate the notion of separate resource managers and to concentrate decision-making within the sensors themselves. In this case, individual sensors could be tasked with coordination amongst themselves to ensure effective operation of the system. Even here there is a trade-off between the efficiencies achieved through explicit coordination and the consumption of network resources in sensors keeping each other informed.

In this paper we have focused on understanding the performance that can be achieved with no explicit coordination between sensors by developing the Sensor-Centric, Utility-Based (SCUB) architecture, which is completely decentralized, i.e. no notion of separate resource managers and no explicit coordination between individual sensors. For comparison purpose, in Section 2.1 we briefly outline a baseline solution for data dissemination derived from existing WSN's network architectures. In Section 2.2 we develop the Sensor-Centric, Utility-Based (SCUB) architecture, which is completely decentralized and no explicit information exchanges between sensors.

2.1 Baseline Solution: Send-All-to-All

Given that there are multiple mobile end-users operating within the sensor network, a straightforward approach to data dissemination is to require all sensors to send all detections to all users, up to constraints on network resources including power, energy, bandwidth, and link availability. This "Send-All-to-All" architecture for task assignment is simple, and its main benefit is that sensors do not have to waste resources by communicating with one another (or with a base station) to determine whether it is appropriate to send sensor data to an individual user. On the other hand, all sensor information within this architecture is treated as equally important, when in reality information from neighboring sensors may be highly correlated and/or irrelevant to accomplish mission objectives. In addition, the "Send-All-to-All" architecture assumes that all information is equally relevant to all end-users, where in reality some users may find the data more actionable than others. Implementation of this approach requires a routing protocol for sensor to end-user communications, and, in our evaluation of this scheme, we assume (as described below) that sensors are periodically informed about the positions of end-users.

2.2 SCUB Architecture

All decision making authority in SCUB architecture is concentrated within individual sensors and no explicit communication occurs between sensors regarding the consumption of common resources. Instead of indiscriminately reporting all information to all end-users, in SCUB architecture sensors independently and periodically make decisions regarding which observations to send to which end-user and when to transmit observations. Decisions are based on a common model for the utility that end-users would attach to data given that they receive it. The utility model serves to quantify the value that users attach to data based on a vector of features associated with the data (e.g. type of observation, location, age, etc.) and on a vector of user-adjustable parameter values that express the relative importance of features associated with the data. Users must provide the sensors with the parameter values needed to evaluate the utility associated with observation data. By introducing a utility model into the framework, we hope to achieve a common value system for all sensors in the system, so that, even though they are operating independently, they still have a means of discriminating between transmission opportunities and hopefully consuming network resources by transmitting high-value observations.

As a more formal description of the utility framework, we assume that there is a set S of sensors that constitutes the sensor network. Each sensor $s \in S$ keeps in memory a history O_s of recent observations, and for each observation $o \in O_s$ there is a set of messages M_{so} that could be sent based on those observations. (It may be possible to send a number of different messages based on the same observation, e.g. imagery with different levels of compression. In the "Send-All-to-All" architecture, the strategy for sending imagery would have to be predefined.) The set $M_s = \cup_{o \in O_s} M_{so}$ represents the collection of all messages that are currently available for sensor s to send based on its recent history of observations. Let G_s represent the set of users that are potential recipients of messages $m \in M_s$ where each "user" may actually correspond to a group of end-users, applications, and/or software agents that could benefit by receiving the message m. In choosing which, if any, of the messages $m \in M_s$ to transmit to user $g \in G_s$, the sensor must compute the utility $u_g(m)$ that g would place on each message m if it were to be received. Mathematically, the function u_g takes the form $u_g(m) = U_g(f(m); \theta_g)$, where $f(m) = (f_1(m), f_2(m), \dots f_F(m))$ is a vector of features associated with the message, and $\theta_g = (w_1, w_2, \dots, w_W)$ is a vector of user-adjustable parameters that help to quantify the utility of the message m for user g. Features associated with messages could include age of the corresponding observation, location of the observation, location of the observation relative to current position of the user, etc. In other words, features are quantitative characteristics of the data associated with the message, not the message itself. The user adjustable parameters in θ_g include various scaling coefficients and weights, which ensure that the overall assessment of utility $u_g(m)$ is normalized to lie between zero (corresponding to no value) and one (corresponding to maximal value).

Having the ability to compute utility values for each message $m \in M_s$ for each soldier $g \in G_s$, the sensor s must use this information to determine which available message to transmit to which end user next, if indeed any are worth transmitting. While the SCUB architecture dictates that no explicit coordination between sensors is allowed, it remains as a task assignment policy issue to determine precisely how message utilities feed into the decision process.

2.2.1 Optimal SCUB Task Assignment

Since the SCUB architecture dictates that sensors act independently in forwarding observation data to users, what we mean by "optimal" task assignment is best characterized in game-theoretic terms. Our goal here is mainly to suggest a mathematical framework for understanding the general issues in SCUB task assignment.

As introduced in Section 2.2, we envision a sensor network comprised of a set S of sensors that constitutes the sensor network. Each sensor $s \in S$ keeps in memory a history O_s of recent observations, and for each observation $o \in O_s$ there is a set of messages M_{so} that could be sent based on that observation (perhaps corresponding to data sent at varying levels of compression). As before we assume that each sensor s has an associated set of users to which it can send messages, and $u_g(m) = U_g(f(m); \theta_g)$ denotes the utility associated with a message $m \in M_s$ for user $g \in G_s$, where $f(m)$ is a vector feature values for the observation associated with m and θ_g is a vector of utility parameters set by the user.

To characterize the state of the task assignment process, we assume that each sensor s also maintains a history H_s of messages that have already been transmitted to users in G_s. Thus, in considering whether to send a given message $m \in$

M_{so} to a particular user $g \in G_s$ the sensor can determine from H_s whether it has already sent a related message to that particular user, thereby having the ability to avoid self-generated redundancy in its transmissions.

Another factor to consider in determining what messages to transmit is current consumption of resources in the network, such as bandwidth. Abstractly, let R denote the set of resources associated with the sensor network. Let C_r denote the capacity of resource $r \in R$, and let x_r denote the current utilization of resource r. For example, if resource r corresponds to a communications link in the network, then C_R would correspond to the bandwidth associated with the link and x_r would correspond to the percentage of that bandwidth that is currently being utilized (measured on an appropriate timescale). In this case, x_r naturally characterizes the state of that communications channel, and the task assignment policy embedded within the sensor simply will not consider sending any message that would cause x_r to exceed the capacity C_r of the channel. (The message may be transmitted later when the channel is less congested.) For resources like bandwidth, the utilization level x_r may fluctuate up and down according to the need for sensors to transmit observation data. For other resources, like total battery energy, the consumption of the resource is monotonic.

Optimal SCUB resource management involves each sensor sending messages to users, subject to resource constraints, so as to maximize the expected aggregate accumulated utility associated with the information that is transmitted. To account for the independent operation of each sensor, we assume that each sensor $s \in \mathcal{S}$ implements a (possibly randomized) Strategy

$$\mu_s(M_s, G_s, H_s, (x_r)_{r\in\mathcal{R}})$$

that prescribes the message $m \in M_s$ to be transmitted to user $g \in G_s$ next (if any) depending on the current consumption of resources $(x_r)_{r\in\mathcal{R}}$ and on the history H_s of messages sent so far. Thus, we seek to compute a profile of strategies $(\mu_{s_1}, \mu_{s_2}, \ldots, \mu_{s_S})$ that maximizes:

$$F(\mu_{s_1}, \ldots, \mu_{s_{|S|}}) = \mathbb{E}\left\{\int_0^T g_{\mu_{s_1}, \mu_{s_2}, \ldots, \mu_{s_{|S|}}}(t)dt\right\}, \quad (1)$$

where T is the random time horizon of the problem and $g_{\mu_{s_1}, \mu_{s_2}, \ldots, \mu_{s_{|S|}}}(t)$ is the random impulse train associated with discrete chunks of utility associated the transmit-decisions that are made by sensors in the network under the strategies $\mu_{s_1}, \mu_{s_2}, \ldots, \mu_{s_{|S|}}$. Note that, in addition to accounting for the possibly randomized behavior of the sensors in the network, the expectation Equation (1) involves uncertainty over the random nature of observations made by individual sensors, which in turn depends on the randomness associated with the environment. $F(\mu_{s_1}, \ldots, \mu_{s_{|S|}})$ can be thought of as a mission-level utility function, representing the collective goals and objectives of all end-users for the duration of the systems lifetime. Optimal SCUB task assignment, can thus be thought of as an identical-interests game, in which each sensor $s \in S$ is a player implementing a strategy μ_s to achieve a high-value global equilibrium.

2.2.2 *Sub-Optimal SCUB Task Assignment*

Realistically, a complete mathematical specification of probabilistic model of Equation (1) can only be made in the context of specific applications built around prohibitively specific scenarios. Moreover, even if the were available, the computation of an optimal profile of task assignment strategies most likely be intractable. Consequently, in the remainder of the paper (as a first cut analysis), we only consider a heuristic strategy designed to improve the chances of achieving system-level objective, while falling short of constituting an optimal solution. We refer to the strategy as "Myopic Utility Maximization," in which, at every transmission opportunity, each sensor will send the message m^* that offers the highest utility to a corresponding user g^*. In other words, the message/user pair (m^*, g^*) for the next transmission is a maximizing solution to the optimization problem $\max_{g\in G_s, m\in M_s} u_g(m)$.

We emphasize that myopic utility maximization is not the only task assignment strategy admitted by the SCUB architecture. For example, in on-going work, we are evaluating other heuristics that incorporate a notion of cost or disutility associated with the consumption of WSN resources that offsets the raw utility that end-users perceive in receiving sensor data. The trick to such an approach is to adaptively adjust resource costs to reflect long-term objectives.

3. ILLUSTRATION: PATROL SCENARIO

To illustrate the application of the SCUB architecture, we have developed a simple testbed called *patrol scenario* as shown in Figure 1, which involves (1) soldiers patrolling subregions within a larger Area of Interest (AOI), (2) a network of sensors reporting detections of intruders to soldiers, and (3) the pursuit and capture of intruders that traverse the AOI. The scenario is coded in JAVA as a discrete simulation event with a fixed discrete simulation interval.In this section we show how a sensor-centric, utility-based task assignment framework can be adapted to suit the overall mission, which is to maximize the rate at which intruders are captured while efficiently managing the use of battery power associated with the sensor network.

We assume that all sensors within the AOI are tripwire sensors with 360 degree field of view and given detection radius, evenly distributed within the AOI and powered by two AA batteries. Each sensor has the ability to make detections of intruders, to receive, transmit, and forward messages, and to engage in antenna idling. Such functions consume different amounts of power and energy. The sensors have ability to detect the intruders with a fixed detection probability once the intruders are within their detection ranges. Intruders randomly appear at the boundary of the AOI and then randomly cross the AOI until captured or until from the system by leaving the AOI. Soldiers are assigned different sub-areas (rectangular) within the AOI and randomly patrol within their sub-areas until they are cued as to the presence of intruders by sensors, at which time they will commence to pursue the intruders, moving at a faster speed than the intruders. Soldiers in pursuit of intruders may require more than one detection report before the intruder is within visual range, in which case the soldier will pursue the intruders without needing (or wanting) additional messages. If a soldier receives multiple messages from different sensors, the soldier only moves in the direction of the nearest intruder. Upon capturing an intruder or upon concluding that the targeted intruder has escaped, soldiers become idle and proceed to patrol randomly until they cued by another detection report from one or more sensors. While in pursuit, soldiers are allowed to leave their assigned sub-areas. The details of the implementation are introduced in the following subsections.

3.1 Sensor Features and Network Model

In accordance with the architectures for decentralized task allocation in WSNs, the flow of information across the sensor network for the patrol scenario is controlled by the sensors themselves, each of which independently implements a task assignment policy designed to maximize the utility of the messages they transmit. We assume that all sensors are in continuous operation (the radio is always active) and are set up to detect the presence of intruders. Also, all transmissions from each sensor are isotropic and has a given effective radio range. In addition, we assume that each sensor is aware of the energy remaining within its batteries and that each sensors will cease transmitting once its energy level has dropped to within a given threshold of zero.

When a sensor detects an intruder and attempts to send a message to a designated soldier, the message may have to be routed through multiple sensor hops. We make the following assumptions about the routing protocol.

- Each senor has complete information of the network topology. Once a sensor is out of energy, all the other sensors are informed of the change of the network topology.
- The chosen route is defined as the shortest path (minimal hops) from origin to destination, as could be computed by the Bellman-Ford algorithm.
- Each time when multiple shortest paths are presented, one path is chosen randomly with uniform distribution.
- The decision to send a detection report to a particular soldier involves sending the message to all (functioning) sensors within radio range of the last reported position of the soldier, in the hope that one of these "destination sensors" is still within communications range of the intended recipient. If the destination sensors can communicate with one another, then only one path among the shortest paths from the origin to the closest destination sensor is used, and the message is then forwarded to the other destination sensors as in a very localized multicast protocol. Otherwise, multiple paths are chosen in such a way that all destination sensors can receive the message from the origin. When a destination sensor receives a message, it will attempt to transmit the message to the soldier. If the soldier is still within communications range of the sensor, then the message will be successfully received by the soldier without confirmation.

Subject to the constraints of channel bandwidth, there exists link delay from the original sensors to the destination sensors due to the aggregate of messages being supported by links on the route. For simplicity we assume that all transmissions are perfectly scheduled and the signal collisions are completely avoided [7, 10]. Hence, the link delay only depends on the traffic load in the route and would be computed as follows based on packet's sharing the bandwidth.

$$d_i = \sum_{j=1}^{j=n} \frac{2 \times P_j}{C \times T} \tag{2}$$

Here d_i denotes the link delay for message i; n is the total number of sensors in the route including the sender and receiver; P_j is the number of packets need to be routed in sensor j in one simulation interval; the coefficient 2 denotes the time division of radio usage for receiving and transmitting; C denotes the channel capacity in packets per second; and T is the simulation interval for the discrete-event simulation.

3.2 Intruder Movement Policy

We assume that the AOI is rectangular and has been split up into two kinds of two rectangular sub-areas of equal size: upper-half and lower half; left-half and right-half. Once an intruder is generated randomly on the boundary of the AOI, the destination of the intruder is also randomly identified, with uniform distribution, in the opposite half-area. For example, when a intruder is generated at the lower boundary of AOI (which is also the boundary of lower half AOI), its destination is identified randomly within the upper-half sub-area while the destination is chosen within right-half sub-area if the intruders is generated on the left boundary of AOI. After the intruder arrives at its destination, it then proceeds along the shortest path out of the AOI. Having fixed a target destination, the intruder's movement is modeled as a random process, capturing the effect of irregularities in the terrain and uncertainty about how to proceed. In our discrete-time simulation of the patrol scenario, we insert a random deviation into an intruder's movement, calling the intended direction of the intruder (with no deviation) the "basic direction," while calling the random direction deviation as "variability angle." The maximal variability angle is a parameter of the model, and in each of the process intruders randomly and independently select a variability angle with uniform distribution within the boundaries established by the maximal variability angle.

3.3 Solider Movement Policy

For soldiers, we assume that the AOI is rectangular and has been split into two rectangular patrol areas of equal size: the lower patrol area and the upper patrol area. Each soldier is assigned to patrol one of these two areas. Soldiers will randomly patrol within his/her patrol area at walking speed until cued as to the existence of an intruder by a sensor. Soldiers in pursuit of an intruder move at running speed. In addition, we assume that soldiers have two additional characteristics: capture range and visual range. The soldier movement in the scenario conforms to the following doctrinal policies:

- At each time interval, soldiers will carry out a look-around process within their visual range. Once soldiers find intruders within their visual range, they will pursue the closest intruder with running speed without further information from sensors.

- Once a soldier captures an intruder, if the other soldiers pursuing the same intruder can observe this happening, they will abort pursuit and become idle.
- When a soldier is idle and is outside of his patrol area, he will move toward the center of his assigned patrol area with walking speed. Upon reaching this destination, if he is still idle, he will commence patrolling randomly at walking speed.
- Any soldier can switch to pursue the closer intruder when receiving multiple messages.
- When a soldier reaches the destination, if he can not find an intruder within his visual range, he will abort to continue to pursue the intruder and become idle.
- When a soldier is idle within his sub patrol area, he moves with random direction and walking speed.

Even when the soldier is within visual range of the intruder, his movement is still modeled as random, using the same random process as for the intruders.

To successfully route messages to the soldiers, the sensors need to be informed periodically of each soldier's position. On the other hand, in order to save energy and network bandwidth, we assume that when a intruder is within a soldier's visual range the soldier is ineligible to receive messages from sensors and can independently pursue the intruder. With these considerations in mind, we assume that soldiers engage in the following position update strategies:

- Regular Update - When a soldier finishes a task (say, upon capturing an intruder), the soldier will report to the sensors that he is eligible to receive message and report to all sensors his position. When a soldier sees an intruder within his visual range, he will report to the sensors that he is ineligible to receive messages and will simultaneously update his current position.
- Extra Update - When a soldier is not in pursuit for a given period of time (the "update interval"), the soldier will report his position to all sensors. The update interval is a parameters of the model/architecture, which can be adjusted if the sensors need more precise information about soldier positions.

3.4 Features and Marginal Utility Model

We model the utility of a message m for soldier g as a multiplicative function of predefined marginal utilities that are associated with the various features of the observation. We assume that there are three features of interest as follows.

- *Feature 1: Age of Associated Observation* - Depending on the interests of a given soldier g, the age of the observation associated with a message can be an important factor in determining whether to transmit the message. Mathematically, this feature is computed as

$$f_1(m) = t - t_{obs}(m),$$

where t is the current time, and $t_{obs}(m)$ is the time at which the observation was made. We assume that the marginal utility that soldier g attaches to this feature is linear and saturates at zero, i.e.

$$u_g^1(f_1(m)) = [1 - w_{1,1} f_1(m)]^+,$$

where $w_{1,1}$ is a g-specified coefficient that describes the maximum age a message can have and still be of any value.

- *Feature 2: Distance to Patrol Area* - The distance from the detection to the patrol area associated with soldier g can also be an important factor in determining whether to transmit a message. Mathematically, this feature is computed as

$$f_2(m) = dist((x_{obs}(m), y_{obs}(m)), (w_{2,1}, w_{2,2}, w_{2,3}, w_{2,4})) \times I,$$

where $dist$ refers to the shortest Euclidean distance between (i) the position of the observation $(x_{obs}(m), y_{obs}(m))$, which in the scenario we take to be the position of the sensor, and (ii) the patrol area for soldier g, with $(w_{2,1}, w_{2,2})$ and $(w_{2,3}, w_{2,4})$ describing the upper left and lower right coordinates of the patrol area (rectangular). Parameter I is an indicator whether the observation is made within the sub-patrol area of the soldier. When the observation is made within the sub-patrol area of the soldier, $I = 0$. Otherwise, $I = 1$. As before, we assume that the marginal utility associated with the distance to the patrol area of soldier g is linear, saturating at zero, i.e.

$$u_g^2(f_2(m)) = [1 - w_{2,5} f_2(m)]^+,$$

where $w_(2,5)$ describes the largest distance to soldier g's patrol area that a detection can have and still have value. The parameters $(w_{2,1}, w_{2,2}, w_{2,3}, w_{2,4}, w_{2,5})$ are specific to (and must be specified by) the soldier g.

- *Feature 3: Distance to Soldier Location* - The third and final feature associated with sensor detections is the relative distance between observation and the current position of a given soldier g. Mathematically, this feature is computed as

$$f_3(m) = dist((x_{obs}(m), y_{obs}(m)), (w_{3,1}, w_{3,2})),$$

where $dist$ again refers to the shortest Euclidean distance between (i) the position of the observation $(x_{obs}(m)$, $y_{obs}(m))$, which we again take to be the position of the sensor itself, and (ii) the most recently updated position $(w_{3,1}, w_{3,2})$ of soldier g. We assume that the marginal utility for this feature is also linear, saturating at zero, i.e.

$$u_g^3(f_3(m)) = [1 - w_{3,3} f_3(m)]^+,$$

where $w_{3,3}$ describes the largest distance to soldier g that a detection can have and still have value. Again, the parameters are specific to (and must be specified by) g. Thus, if the soldier is capable of rapid movement, then it will have to update its parameter vector frequently.

3.5 Message Overall Utility

The marginal utilities described above only reflect preferences for sensor information based on individual features and cannot by themselves fully represent the preferences of the soldiers, which are possibly a complex function of age, distance to patrol area, and relative distance. Drawing insights from decision theory (see, for example, [22]), we model soldier-g's overall utility for message m as a multiplicative function of his marginal utilities, i.e.

$$u_g(m) = \frac{1}{k_0}\left[-1 + \prod_{i=1}^{3}\left(1 + k_0 k_i u_g^i(m)\right)\right],$$

where (i) the number 3 denotes the number of features for sensors, (ii) the parameters k_1, k_2, k_3 are weighting parameters set by g to reflect the importance of each feature, and (iii) the parameter k_0 is a normalizing constant set to be a solution to the equation

$$1 + k_0 = \prod_{i=1}^{3}(1 + k_0 k_i).$$

Each weighting parameter k_i $(i = 1, 2, 3)$ should be chosen to reflect the overall utility of a message where the i-th feature has its best possible value and all the other features have their worst possible values. From decision theory, utility functions of this form are characteristic of an individual whose preferences exhibit a quality known as *mutual utility independence,* where, holding any fixed subset of attributes (features), the individuals preferences between alternatives are independent of the values of the fixed attributes. While it may not be the case that any given soldier's preferences exhibit mutual utility independence, the multiplicative functional form for $u_g(m)$ is still a useful model in that, by virtue of its being a nonlinear function of the marginal utilities, it can represent preferences where features can either compliment or substitute for each other. In order for a sensor to be able to evaluate the overall utility of a message m for soldier g, the soldier must provide values for the weighting parameters k_1, k_2, k_3, and to the extent that these values are changing over time, the soldier is also responsible for updating the values of these parameters for the duration of the patrol mission.

4. NUMERICAL EVALUATION

Section 3 presents the design principles and doctrines of patrol scenario. Here we provide the detail parameter values and simulation flow in Section 4.1 and the corresponding preliminary experimental results in Section 4.2.

4.1 Simulation Test Bed

Here, we outline the main assumptions and parameters of the simulation test bed in Section 4.1.1 below. In Section 4.1.2, we briefly describe the internal operation of the simulation.

4.1.1 Assumptions and Parameters of the Simulation

The simulation testbed is designed primarily to reflect the features of the patrol scenario from Section 3. The simulation accounts for a rectangular AOI, cut into two rectangular patrol areas to which individual soldiers are assigned. Intruders randomly appear at the boundary of the AOI, move across the interior, and either escape or are captured by soldiers who receive cues from sensors. The implementation of the simulation is general enough to allow for different types of experiments to be conducted (say for varying numbers of soldiers, sensors with different characteristics, diverse task assignment policies, etc.). In the following paragraphs, we list the parameters that can be adjusted for experimental purposes, along with the default values used in the experimental results of Section 4.2.

General Scenario Parameters
The simulation evolves in discrete stages with each stage corresponding to a fixed sampling interval, which is set as 1.0 second per update. The dimensions of the AOI are given in "distance units" with 10000 distance units corresponding to a linear mile. The simulation starts with each sensor being fully charged, with energy being consumed at different rates depending on the traffic load. The simulation accounts for one or more shared communications channels with a fixed capacities that affects the link delay of the messages. All sensors are evenly spaced throughout the AOI according to a grid pattern. The intruders are generated at the boundary of AOI and move randomly towards their destinations until captured by the soldiers cued by the messages from sensors or escaping from the AOI. The simulation is completed once all the sensors are out of energy.

Sensor Characteristics
Most of the data for the sensors are from references [29, 28, 25] based on experiments on MICA2 mote. Mainland et al. [25] showed that the energy needed for sensor sampling for MICA2 is $8.41 \times 10^{-5} J$ per sample, and the Datasheet for MICA2 [28, 29] shows the powers needed for radio idling, receiving and transmitting are 30mW, 30mW and 81mW, respectively. The corresponding energies for radio receiving and transmitting are computed based on the powers and the radio time needed, which can be computed with the size of message and the capacity of channel. Here, again, we assume that there is no data collisions in the transmissions. The energy for radio idling in one interval is computed based on the idling power and the idling time in one interval except the radio's receiving and transmitting. Similarly, we assume the channel capacity is 38.4 Kbps, which is corresponding to the MICA2 bit rate. We also assume that the size per message is 200 kilobytes, which is assumed to be one packet. The simulation assumes a constant probability of detection within any sensors detection range. The simulation does not currently account for false alarms. All the Parameters that describe characteristics of the sensors are given in Table 1.

Intruder Characteristics
Parameters that describe characteristics of the intruders are given in Table 2, including the intruder variability, intruder speed and the likelihood with which new intruders arrive at the boundary of the AOI. It is noted that the true movement direction of intruders is the basic direction, from current position to the destination, plus the variability angle, which also applies in the soldier movement when the soldier is in pursuit.

Soldier Characteristics
Parameters that describe characteristics of the soldiers are given in Table 3. The simulation is implemented in such a way that the number of soldiers patrolling the AOI can be determined at runtime. In our experiments we have only

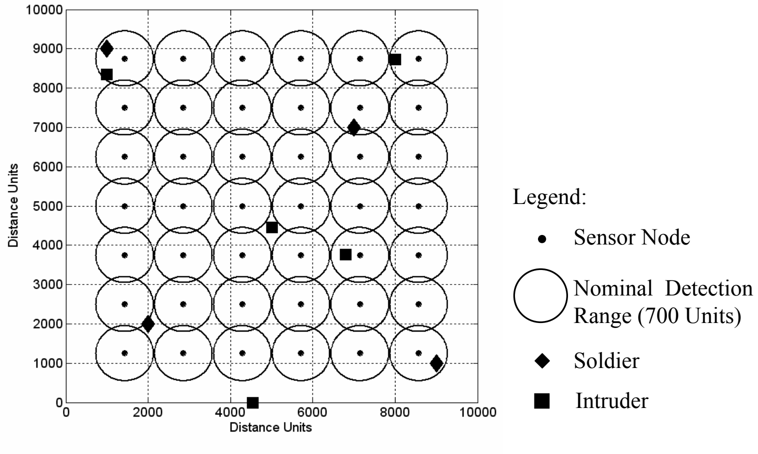

Figure 1: Sensor Network Configuration

Parameter Name	Value
Message size	200 Bytes, one packet
Channel bit rate	38.4 Kbps, 24 packets
Full energy	14040 Joules, 2×AA batteries
Sensor sampling	8.41×10^{-5}Joules
Idle power	30mW
Receive power	30mW
Transmit power	81mW
Number of sensors	42
Detection range	700 distance units radius (.07 mile, 113 meters)
Radio range	1500 distance units radius (0.15 mile, 242 meters)
Detection probability	0.8 per interval if intruder is within detection range

Table 1: Parameters Descriptive of the Sensors

Parameter Name	Value
Move variability boundary	-45-45 degrees, uniformly distributed, independent draws
Intruder speed	10 units per interval (1.63 meters / second)
Intruder arrival probability	0.015 per interval (randomly on boundary of AOI)

Table 2: Parameters Descriptive of the Intruders

considered between two and four soldiers with patrol areas and initial positions as given in the table below. Whenever there are two soldiers in the system, the two soldiers are "Soldier 1" and "Soldier 2." Whenever there are three soldiers, they are "Soldiers 1, 2, and 3." All soldiers move randomly (with random directions) while on patrol, and they move at a fixed "patrol speed." Once a soldier is cued as to the existence of an intruder he/she always moves in the direction of basic direction plus the variability towards the intruder at a fixed "pursuit speed." All soldiers share a common intruder capture and visual ranges, and they update their positions with all sensors at regular and extra updates as shown in Section 3.

Utility Function Parameters

The simulation assumes that all soldiers perceive utility for sensor data in the same way. Except for the fact that soldiers have distinct sub-patrol areas and current positions, we assume that in evaluating the utility of sensor data all sensors use the same utility function parameters for all soldiers. The default parameter values for sensor utility appear in Table 4.

4.1.2 Simulation Flow

The simulation evolves in discrete stages, with each corresponding to a fixed interval of time. Here, we briefly outline the operational flow of the simulation. Each stage begins with the random determination of whether a new intruder appears on the boundary of the AOI. Next, the new positions for the existing intruders are computed randomly according the mobility model discussed in Section 3. Any intruder that either enters into or remains in the detection radius of a sensor runs the risk of being detected, and the simulation will next randomly compute the entire set of detections based on the most recent set of intruder movements. For each such detection, the corresponding sensor will determine, according to the task assignment architecture being implemented, which soldier if any should receive a message about the detection. The first step of this determination is to reassess the set of soldiers who are eligible to receive data, i.e. soldiers who are not already in pursuit of an intruder within their sight.

The next step is to determine the destination sensors for each messages based on the historical positions of chosen soldiers. Then, the route for each message is computed with Bellman-Ford algorithm and randomly chosen if multiple shortest routes are present. The third step is to determine the traffic loads (in packets) in each senor and the link delay for each message based on the traffic load in the whole network. Correspondingly, the energy consumptions for sensor sampling, radio idling, receiving and transmitting on each sensor are determined. The fourth step is to release the message from the destination sensors once the time elapsed is equal to the link delay of the message. If the designated soldier is still in the radio range of any destination sensor, the message is successfully delivered to the soldier. If a soldier is informed of more than one intruder, then he/she will choose the closest one to pursue. The last step of each stage is to compute new locations for each soldier. Each soldier who is not in pursuit of an intruder will move randomly according to the mobility model discussed earlier. Soldiers who are pursing intruders move towards the reported location of the intruder.

4.2 Simulation Experiments

The simulation test bed of Section 4.1 allows us to compare the SCUB task assignment with the "Send-All-to-All" task assignment for the patrol scenario. In this section, we present preliminary experimental results that validate the assertion that SCUB task assignment can have strong effect on mission performance.

Before presenting numerical results, we define our system performance metrics in Section 4.2.1. Next, in Section 4.2.2, we outline the experimental variables, and in Section 4.2.3 we present and discuss our results.

4.2.1 Performance Metrics

In our preliminary experiments so far we have focused on two principle attributes, (i) success rate, i.e. the fraction of intruders captured, and (ii) number of sensors alive. Both success rate and number of sensors alive are computed against the time horizon. And both attributes are important considerations in configuring the sensor network.

For success rate since it is a discrete simulation and there exists stochastic variance, we average the success rate in a fixed time window size. The success rate is computed with the following equation:

$$SR^{t_i,t_j} = \frac{N^{t_i,t_j}_{captured}}{N^{t_i}_{atLarge} + N^{t_i,t_j}_{generated} - N^{t_j}_{atLarge}} \tag{3}$$

Where SR^{t_i,t_j} denotes the average success rate in time period $(t_i, t_j]$; $N^{t_i,t_j}_{captured}$ denotes the total number of intruders captured in time period $(t_i, t_j]$; $N^{t_i}_{atLarge}$ means the number of intruders at large in time t_i; $N^{t_i,t_j}_{generated}$ denotes the total number of intruders generated in time period $(t_i, t_j]$. We computed the averages above for different experimental trials within windows of $28,800$ seconds.

For the other metric, number of sensors alive, we sampled the number of sensors that have sufficient energy remaining to continue making detections throughout the lifetime of the network. We computed average number of sensors alive for different experimental trials over time windows of 200 seconds.

4.2.2 Experimental Variables

We choose five important experimental variables among all of these to test the two task assignment architectures. The names of variables and their range are listed as follows.

- Intruder and soldier's variability boundaries (in deg.) : 0, *-45-45,* -90-90
- Sensor detection range (in distance units) : 400, 600, *700,* 800, 900
- Number of soldiers: 2, *3,* 4
- Intruder arrival probability: 0.001, 0.005, *0.015,* 0.025

Parameter Name	Value
Random patrol direction (idle)	0-360 deg., uniformly distributed, independent
Move variability boundary (in pursuit)	-45-45 deg., uniformly distributed, independent
Patrol speed	10 units / interval (1.63 meters / second)
Pursuit speed	15 units / interval (2.44 meters / second)
Soldier extra position update	60 intervals / update (1 minutes)
Message for position update	200 kilobytes, one packet
Intruder capture range	50 distance units (8 meters)
Soldier visual range	500 distance units (80 meters)
LL corner of patrol area, soldiers 1 & 3	(0, 0) “coordinate” in distance units
UR corner of patrol area, soldiers 1 & 3	(10000, 5000) coordinate
LL corner of patrol area, soldiers 2 & 4	(0, 5000) coordinate
UR corner of patrol area, soldiers 2 & 4	(10000, 10000) coordinate
Soldier 1 initial position	(2000, 2000) coordinate
Soldier 2 initial position	(7000, 7000) coordinate
Soldier 3 initial position	(9000, 1000) coordinate
Soldier 4 initial position	(1000, 9000) coordinate

Table 3: Parameters Descriptive of the Soldiers

Feature	Parameter Value
0	$k_0 = 0.36$ (normalizing constant)
1	$w_{1,1} = 1/5$, $k_1 = 0.3$
2	$w_{2,1}, \ldots, w_{2,4} =$ UL/LR region of interest, $w_{2,5} = 1/5000$, $k_2 = 0.3$
3	$w_{3,1}, w_{3,2} =$ historical position of soldier, $w_{3,3} = 1/14000$, $k_3 = 0.3$

Table 4: Soldier Utility Parameters for Sensor Data

- Solider extra position update cycle length (in seconds): 1, *60,* INF (no update)

Here in each variables the italic number is the nominal value. When we test the task assignment architectures on one variable, all the other variables are set as the nominal values and all the other parameter values are kept fixed at the default values listed in Section 4.1.1. It is also noted that we do not select the values evenly for some variables. The simulation has lots of stochastic sources. It may need large number of runs to obtain statistical significance of the system performances if two levels of the same variable is very close. For current settings of the simulation and 20 runs for each trial, it takes 3 days with parallel simulation in 10 computers.

4.2.3 Preliminary Results

In order to compare the effectiveness of SCUB task assignment over ”Send-All-to-All” task assignment, we have conducted 20 independent trials for each pair of (i) experimental variables and (ii) task assignment architectures. We do not show the variations for all the results, but the variances of the results with the nominal parameter settings.

Soldier and Intruder Movement Variability

The changes of success rates and number of sensors alive along with the time horizon are shown in Figures 2 and 3 for both SCUB and ”Send-All-to-All” task assignments, respectively. For success rate, it is shown that with the more variability the success rate for ”Send-All-to-All” task assignment is going up while for SCUB task assignment it is going down. In ”Send-All-to-All” task assignment, once an intruder is detected by a sensor, all the eligible soldiers may receive the message only if the message can be successfully routed to the soldiers. It results in that multiple soldiers may pursue the same intruder. Hence, the more movement variability, the longer the intruder stay in the AOI and the more success rate of the intruders captured by soldiers. On the contrary, for SCUB task assignment, only the soldier with maximal utility can receive the message, and normally only one soldier pursues an intruder. The more movement variability (e.g. the more complex geography) makes the soldier harder to capture the intruder. However, for any variability boundary in this scenario, the success rates with SCUB task assignment are significantly greater than those with ”Send-All-to-All” task assignment throughout the whole system lifetime. It is also noted that after 3.5×10^5 seconds, the success rates for ”Send-All-to-All” task assignment go down dramatically since the numbers of sensors alive are reduced quickly as shown in Figure 3.

For number of sensors alive, the trends are the same for both SCUB and ”Send-All-to-All” task assignments. The more variability, the less sensors alive. The more variability makes the intruders stay in AOI longer before they escape the AOI or even are captured by soldiers. However, it is noted that the time period with all sensors alive in SCUB task assignment is significantly longer that that in ”Send-All-to-All” task assignment. For any movement variability, the number of sensors alive with SCUB task assignment is significantly more than that with ”Send-All-to-All” task assignment throughout the entire system lifetime.

Sensor Detection Range

The average success rate and number of sensors alive for different sensor’s detection ranges are shown in Figures 4 and 5

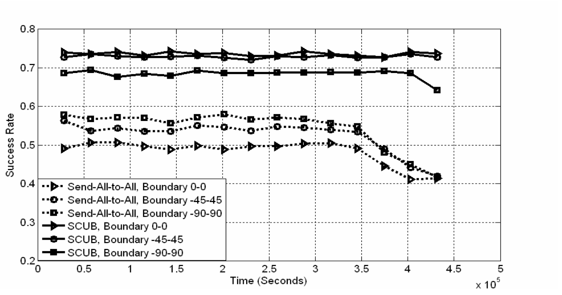

Figure 2: Success rate as a function of movement variabilities

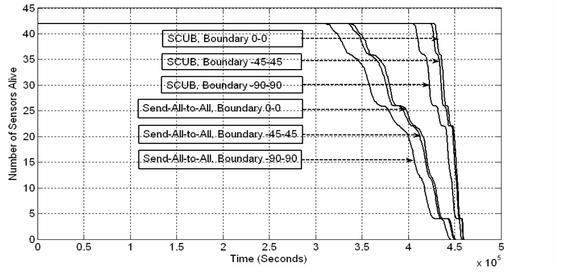

Figure 3: Number of sensors alive as a function of movement variabilities

for both SCUB and "Send-All-to-All" task assignments, respectively. For any sensor detection range, the success rate with SCUB task assignment is significantly greater than that with "Send-All-to-All" task assignment throughout the entire system lifetime. It is also noted that the success rate reaches the maximal with detection range of 700 units for SCUB task assignment while it is 600 units for "Send-All-to-All" task assignment. When the detection range is less or larger than 700 units for SCUB task assignment and 600 units for "Send-All-to-All" task assignment, the success rates go down. It is noted that this trend is related to the sensor and soldier characteristics. In this scenario we assume that the sensor can not precisely locate the intruder's location and only report its own location once an intruder is detected. Also, the soldier's visual range is 500 distance units. When the detection range is small, the sensors can not effectively detect the intruders, which results in the low success rate since soldiers can not obtain enough information from sensors. But it seems that SCUB task assignment suffers more from the less coverage since sensors send messages only to one soldier once detecting an intruder. When there is severe overlap in sensor coverage (800 and 900 units), since the soldier's visual range is only 500 units it is more likely that the soldiers can not find the intruders even they reach the destination (sensor's location). We also note that after 3.5×10^5 seconds in "Send-All-to-All" task assignment, the success rate for detection range of 700 units is below those with detection range of 400 and 600 units. The reason is that after 3.5×10^5 seconds the number of sensors alive for detection range of 700 units is significantly lower that those with detection range of 400 and 600 units as shown in Figure 5. The same reason applies for the phenomenon that in SCUB task assignment success rate with detection range of 600 units is lower that that with 400 units after 4.5×10^5.

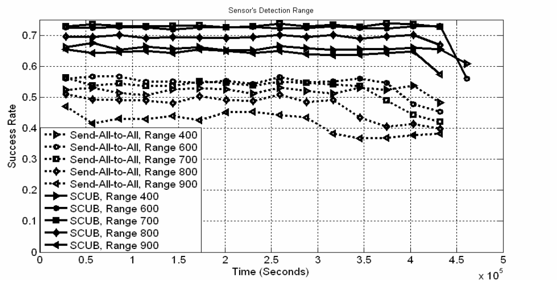

Figure 4: Success rate as a function of sensor's detection ranges

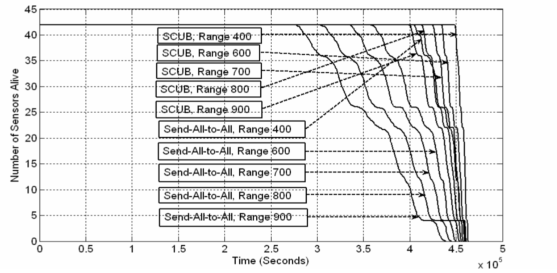

Figure 5: Number of sensors alive as a function of sensor's detection ranges

For number of sensors alive, throughout the entire system lifetime the number of sensors alive in SCUB task assignment is larger than that in "Send-All-to-All" task assignment since sensors with "Send-All-to-All" task assignment transmit more messages and consume more energy. It is also noted that with larger detection range the number of sensors alive becomes less since sensors with larger coverage have more opportunities to detect intruders and transmit messages.

Number of Soldiers

The average success rate and number of sensors alive for 2 to 4 soldiers are shown in Figures 6 and 7 for both SCUB and "Send-All-to-All" task assignments, respectively. For both strategies, the more soldiers, the larger success rate. However, the magnitude of improved success rate by adding one more soldier is larger for SCUB task assignment than "Send-All-to-All" task assignment. For any number of soldiers, the success rate with SCUB task assignment is significantly greater than that with "Send-All-to-All" task assignment throughout the entire system lifetime.

For number of sensors alive, for "Send-All-to-All" task assignment the more soldiers the less number of sensors alive while for SCUB task assignment the more sensors alive. The more soldiers assigned to the AOI, the larger number of eligible soldiers. For "Send-All-to-All" task assignment, since sensors need to send messages to all the eligible soldiers, it consumes more energy. On the contrary, in SCUB task assignment a sensor only need to send one message to the eligible soldier with the maximal utility value. The more eligible soldiers makes the maximal utility higher and the route shorter. Therefore, for SCUB task assignment, the

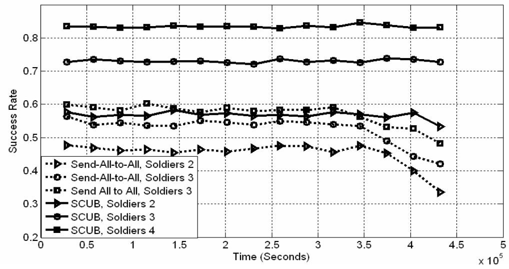

Figure 6: Success rate as a function of number of soldiers

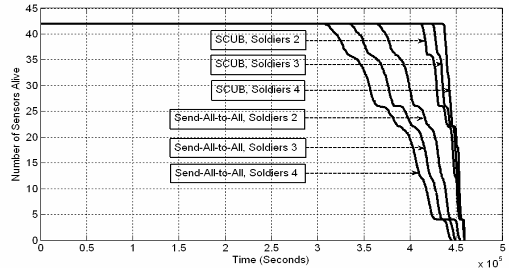

Figure 7: Number of sensors alive as a function of number of soldiers

more soldiers the more number of sensors alive while for "Send-All-to-All" task assignment it is exactly on the contrary.

Intruder Arrival Rate

The average success rate and number of sensors alive for different intruder arrival probabilities are shown in Figures 8 and 9 for both SCUB and "Send-All-to-All" task assignments, respectively. For both architectures, the larger intruder arrival probability, the less success rate. It is noted that with the increase of intruder arrival probability, the success rate for "Send-All-to-All" task assignment is reduced dramatically. That is, with small intruder arrival probability SCUB and "Send-All-to-All" task assignments appear to have the same success rate. However, "Send-All-to-All" task assignment is more sensitive to intruder arrival probability. Moreover, throughout the system lifetime, the success rates of SCUB task assignment are significantly better than those of "Send-All-to-All" task assignment with the range of intruder arrival probabilities from 0.001 to 0.025.

For number of sensors alive, the larger intruder arrival probability, the less number of sensors alive. With larger intruder arrival probability, the sensors have more opportunities to detect intruders and transmit more messages, which consumes more energy.

Position Update Interval

The average success rate and number of sensors alive for different soldier position update cycle lengths are shown in Figures 10 and 11 for both SCUB and "Send-All-to-All" task assignments, respectively. For both architectures, it seems that the cycle length does not has significant impact on the success rate though the success rates are slightly improved with short cycle length. However, it is noted that this observation is related several factors, such as the number of soldiers, the intruder arrival probability and the definition of the regular update as shown in 3. This observation shows that with the soldier number of 3 and the intruder arrival probability of 0.015 the regular soldier position update is sufficient for sensor to compute the utility values to discriminate the soldiers.

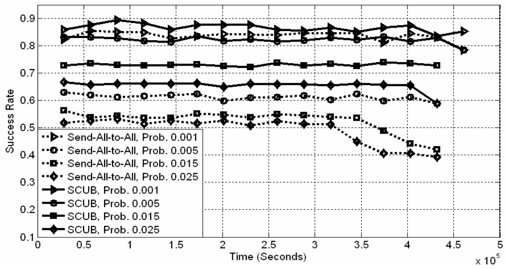

Figure 8: Success rate as a function of intruder arrival probabilities

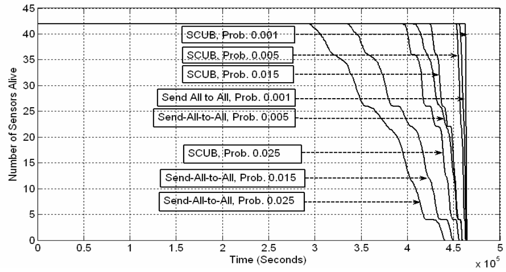

Figure 9: Number of sensors alive as a function of intruder arrival probabilities

For number of sensors alive, the effects of different cycle lengths are also trivial. The number of sensors alive is slightly larger when the cycle length is shorter. We explain it is affected by the success rates with different cycle lengths. With a shorter cycle length, the success rate is improved slightly, which means that slightly more intruders are captured before escaping. That saves the sensor's energies. However, we have to mention that this trivial observation is under two assumptions in the scenario: (i) The required powers for radio idling and receiving are the same and both are 30mW, and (ii) The radio is always active. It can be conjectured that the observation will be different if the two assumptions do not hold.

Uncertainty Analysis

We list the variances of success rate and number of sensor alive for nominal parameter settings as shown in Figures 12 and 13 for both SCUB and "Send-All-to-All" task assignments. For success rate, it is found that SCUB task assignment has less variance and fluctuation than "Send-All-to-All" task assignment. For number of sensors alive, we have the same observation. Those show that nominally the results from SCUB task assignment has less variance than

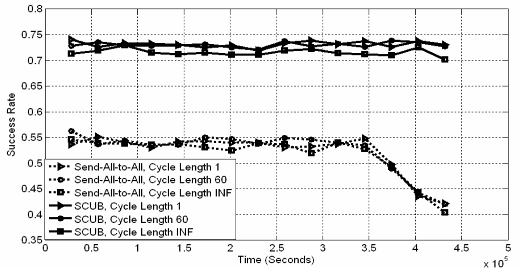

Figure 10: Success rate as a function of soldier extra position updates

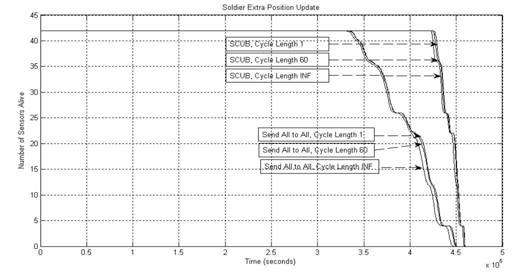

Figure 11: Number of sensors alive as a function of soldier extra position updates

those from "Send-All-to-All" task assignment, and appear to be more stable.

5. CONCLUSIONS

Our work on architectures for task assignment algorithms in distributed sensor networks, particularly on the SCUB architecture for task assignment is at a very preliminary stage. SCUB itself seems to represent an approach to the problem of allocating the scarce resources of sensor networks, being highly decentralized and yet cognizant of the diverse information requirements of multiple users. An important point of departure in our work on SCUB from other related work is our focus on application layer performance metrics, rather than on lower-level performance metrics, such as the throughput and delay of the wireless channel. Our view is that research on sensor networks should not treat the sensor network as yet another type of data network, but rather as a platform for applications.

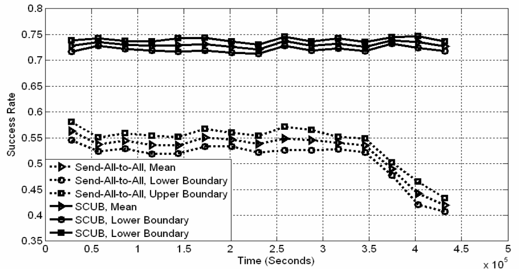

Figure 12: 95% confidence boundaries of success rate for nominal parameter settings

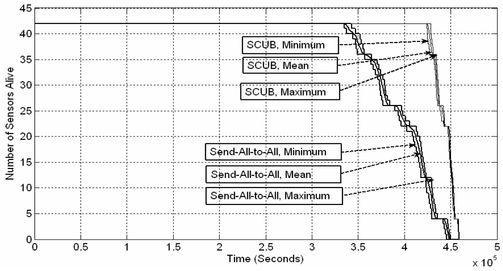

Figure 13: Boundaries of number of sensors alive for nominal parameter settings

In future work we plan to further develop the SCUB architecture, along with competing architectures that are more centralized (involving resource brokers/managers) for avoiding redundant communications over the shared wireless channel. We also plan to refine the patrol scenario as a test bed for research on information management algorithms. The patrol scenario presented in this report is quite simplistic and would be improved for (i) the terrain associated with the AOI (the current model may is a bit simple), (ii) the sensor communication protocol. For example, now we assume that there is no signal interference and packet collisions, and the sensor radio is always active. For future work, some communication protocols, such as carrier sensing medium access (CSMA), will be implemented and in order to save sensor's energy with different duty cycles for the radio will be set up.

Finally, we observe that, strictly speaking, SCUB is not a decision-theoretic approach to task assignment. That is, in requiring sensors to prioritize transmission opportunities based on utility, the system isn't necessarily making the *best* decision from any individual's point of view. We use the notion of a utility function to build a unified quantitative means of comparing transmission opportunities, and we have merely found it convenient to draw upon insights from decision theory in establishing a form of the utility function that accounts for the substitutability and complementarity of features. We hope that insights from group decision theory will similarly apply in assessing utility functions for the value that a group of peers (not individual soldiers) would attach to sensor data, where each member may have different views about the importance of features. Accounting for group preferences in this framework will be a major challenge moving forward.

6. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the assistance of Xinyan Pan in helping to formulate the "patrol scenario" of this paper.

7. REFERENCES

[1] A. Ahmed and M. Eskicioglu. Current researches on sensor networks. Technical report, Telecommunication Research Labs,Winnipeg, Canada, 2004.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks: the International Journal of Distributed Informatique*, 38(4):393–422, March 2002.

[3] G. Asada, M. Dong, T. Lin, F. Newberg, G. Pottie, and W. Kaiser. Wireless integrated network sensors: Low power systems on a chip. In *European Solid State Circuits Conference*, October 1998.

[4] R. Barr, J. Bicket, D. Dantas, B. Du, T. Kim, B. Zhou, and E. Sirero. On the need for system-level support for ad hoc and sensor networks. *ACM Operating Systems Review*, 36(2):1–5, April 2002.

[5] A. Bharathidasan and V. Ponduru. Sensor networks: An overview. Technical report, UC Davis, 2002.

[6] F. Bian, D. Kempe, and R. Govindan. Utility-based sensor selection. In *The Fifth International Symposium on Information Processing in Sensor Networks*, April 2006.

[7] J. Byers and G. Nasser. Utility-based decision-making in wireless sensor networks. *Mobile and Ad Hoc Networking and Computing, IEEE*, August 2000.

[8] J. Chamberland and V. Veeravalli. Decentralized detection in sensor networks. *IEEE Transactions on Signal Processing*, 51(2):407–416, February 2003.

[9] W. Chen and L. Sha. An energy-aware data-centric generic utility based approach in wireless sensor networks. In *Proceedings of the third international symposium on Information processing in sensor networks*, April 2004.

[10] B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *IEEE International Conference on Communications*, Montreal, Canada, June 1997.

[11] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: scalable coordination in sensor networks. In *ACM/IEEE International Conference on Mobile Computing and Networks*, pages 263–270, 1999.

[12] D. Ganesan, D. Estrin, and J. Heidemann. Dimensions: why do we need a new data handling architecture for sensor networks? *Computer Communication Review*, 33:143–148, 2003.

[13] J. Heidemann, F. Silva, Y. Yu, D. Estrin, and P. Haldar. Diffusion filters as a flexible architecture for event notification in wireless sensor networks. Technical report, Information Sciences Institute, USC, April 2002.

[14] J. Hill. System architecture for wireless sensor networks. Technical report, PhD thesis, UC Berkeley, May 2003.

[15] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In *Proceedings of ASPLOS-IX*, November 2000.

[16] S. Iyengar, D. Jayasimha, and D. Nadig. A versatile architecture for the distributed sensor integration problem. *IEEE Trans. Comput.*, 43(2):175–185, 1994.

[17] D. Jayasimha, S. Iyengar, and R. Kashyap. Information integration and synchronization in distributed sensor networks. *IEEE Transactions on Systems, Man and Cybernet*, 21(21):1032–1043, 1991.

[18] H. Kang and X. Li. Power-aware sensor selection in wireless sensor networks. In *The 5th International Conference on Information Processing in Sensor Networks, Work-in-Progress*, 2006.

[19] R. Kannan, S. Sarangi, and S. Iyengar. Sensor-centric energy-constrained reliable query routing for wireless sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):839–852, 2004.

[20] R. Kannan, S. Sarangi, S. Iyengar, and L. Ray. Sensor-centric quality of routing in sensor networks. In *INFOCOM*, March 2003.

[21] H. Karl and A. Willig. A short survey of wireless sensor networks. Technical report, Technical University Berlin, Berlin, Germany, 2003.

[22] R. Keeney and H. Raiffa. *Decisions with multiple objectives: Preferences and value tradeoffs.* John Wiley and Sons, New York, 1976.

[23] A. Lim. Distributed services for information dissemination in self-organizing sensor networks. *Journal of the Franklin Institute*, 338(6):707–727, September 2001.

[24] S. Madden and M. Franklin. Fjording the Stream: An Architecture for Queries Over Streaming Sensor Data. In *ICDE*, pages 555–566, 2002.

[25] G. Mainland, D. Parkes, and M. Welsh. Decentralized, adaptive resource allocation for sensor networks. In *Proceedings of the 2nd USENIX/ACM Symposium on Networked Systems Design and Implementation*, 2005.

[26] A. Makarenko and H. Durrant-Whyte. Decentralized data fusion and control in active sensor networks. In *Proceedings of the Seventh International Conference on Information Fusion*, 2004.

[27] A. Makerenko, E. Nettleton, B. Grocholsky, S. Sukkarieh, and H. Durrant-Whyte. Building a decentralized active sensor networks. In *11th Intl Conf on Advanced Robotics*, Coimbra, Portugal, 2003.

[28] MICA2. Mote Datasheet, 2004. http://www.xbow.com/Products/Product_pdf_files/ Wireless_pdf/6020-0042-01_A_MICA2.pdf.

[29] M. Miller and N. Vaidya. A MAC protocol to reduce sensor network energy consumption using a wakeup radio. *IEEE Transactions on Mobile Computing*, 4(3):228–242, May 2005.

[30] R. Min, M. Bhardwaj, S. Cho, A. Sinha, E. Shih, A. Wang, and A. Chandrakasan. An architecture for a power-aware distributed microsensor node. In *IEEE Workshop on Signal Processing Systems*, October 2000.

[31] H. Nama, M. Chiang, and N. Mandayam. Utility lifetime tradeoff in self regulating wireless sensor networks: A cross-layer design approach. In *Proc. IEEE ICC*, June 2006.

[32] H. Nama and N. Mandayam. Optimal utility-lifetime tradeoff in self-regulating wireless sensor networks: a distributed approach. In *Conference on Information Sciences and Systems*, Princeton, March 2006.

[33] L. Prasad, S. Iyengar, R. Kashyap, and R. Madan. Functional characterization of sensor integration in distributed sensor networks. *IEEE Transactions on Systems, Man and Cybernet*, 21(5):1082–1087, 1991.

[34] H. Qi, S. Iyengar, and K. Chakrabarty. Distributed sensor networks-a review of recent research. *Journal of the Franklin Institute*, 338(6):655–668, September 2001.

[35] H. Qi, S. Iyengar, and K. Chakrabarty. Multiresolution data integration using mobile agents in distributed sensor networks. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 31(3):383–391, August 2001.

[36] P. Rentala, R. Musunuri, S. Gandham, and U. Saxena. Survey on sensor networks. In *Proc. of International Conference on Mobile Computing and Networking*, 2001.

[37] M. Ridley, E. Nettleton, S. Sukkarieh, and H. Durrant-Whyte. Tracking in decentralised air-ground sensing networks. In *Proceedings of the Fifth International Conference on Information Fusion*, pages 616–623, 2002.

[38] N. Sadagopan, M. Singh, and B. Krishnamachari. Decentralized utility-based sensor network design. *Mobile Networks and Applications*, 11(3):341–350, 2006.

[39] M. Samantham and D. Pradhan. The debruijn multiprocessor network: a versatile parallel processing and sorting network for vlsi. *IEEE Trans. Comput.*, 38(4):576–581, 1989.

[40] C. Shen, C. Srisathapornphat, and C. Jaikaeo. Sensor information networking architecture and applications. *IEEE Personal Communications*, 8(4):52–59, August 2001.

[41] A. Sheth, C. Hartung, and R. Han. A decentralized fault diagnosis system forwireless sensor networks. In *Mobile Adhoc and Sensor Systems Conference, IEEE*, 2005.

[42] L. Subramanian and R. H. Katz. An architecture for building self-configurable systems. In *ACM/IEEE Workshop on Mobile Ad Hoc Networking and Computing*, August 2000.

[43] M. Tubaishat and S. Madria. Sensor networks: An overview. *IEEE Potentials*, 22(2), April 2003.

[44] Y. Wen and W. Sethares. Automatic decentralized clustering for wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 5:686–697, 2005.

[45] R. Wesson, F. Hayes-Roth, J. Burge, C. Stasz, and C. Sunshine. Network structures for distributed situation assessment. *IEEE Transactions on Systems, Man and Cybernet*, 11(1):5–23, 1981.

[46] N. Xu. A survey of sensor network applications. *IEEE Communications Magazine*, 40(8):102–114, August 2002.

[47] W. Zhang, Z. Deng, G. Wang, L. Wittenburg, and Z. Xing. Distributed problem solving in sensor networks. In *AAMAS*, pages 988–989, 2002.

[48] F. Zhao, J. Shin, and J. Reich. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal Processing Magazine*, 19(2):61–72, March 2002.