This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

Dynamic Optimization of Secure Mobile Sensor Networks: A Genetic Algorithm

Rahul Khanna*, Huaping Liu†, and Hsiao-Hwa Chen‡

*Intel Corporation, 2111 NE 25th Ave., Hillsboro, OR 97124, USA. E-mail: rahul.khanna@intel.com

†School of EECS, Oregon State University, Corvallis, OR 97331 USA. E-mail: hliu@eecs.oregonstate.edu, tel: +1 541 737 2973

‡Institute of Communications Engineering, National Sun Yat-Sen University, Taiwan. E-mail: hshwchen@ieee.org

Abstract— **We propose a reduced-complexity genetic algorithm for secure and dynamic deployment of resource constrained multi-hop mobile sensor networks. Mobility and security are relatively expensive operations since they involve both communication and computation. Furthermore, these operations have to co-exist with optimal node and route assignments. The goal of this paper is to achieve optimal secure coverage and improved battery life using dynamic re-locatability. The genetic algorithm is used to adaptively configure optimal position and security attributes by dynamically monitoring network traffic, packet integrity, and battery usage. This results in minimization of the power consumption of the sensor system while maximizing the sensor objectives (coverage and exposure).**

I. Introduction

Low-cost integration and small-size micro-sensors [1]–[5] have generated significant interest in the area of disposable sensors. These are motion capable, randomly deployed, infrastructure-less, data-centric sensors equipped with data processing capabilities and sensory circuits that cannot be charged (or rarely charged) or replaced. These sensors are constrained in energy, bandwidth, storage, and processing-capabilities and find their uses in the areas of homeland-security, disaster-recovery, target-identification, reconnaissance, medical applications, defense applications [6], and intrusion-detection, etc. Individual sensors process the sensory data and transmit to the target (sink) in a secure manner. Mobility reduces communication overhead (maximize the battery and sensor's life) by relocating these sensors and helping set up energy-efficient route for non-redundant secure data transmission from source to the sink. Although mobility and secure routing have been widely researched for ad hoc networks, it is still an unexplored area for resource constrained mobile sensor networks.

In this paper we develop an evolutionary algorithm [7] that divides and positions the randomly deployed mobile sensors into an optimal number of independent clusters with cluster-head and optimal route. Once deployed, these sensors further maximize their coverage by moving (or re-orienting) themselves at the expense of battery life. Cluster-head collects data from its member sensors and sends them to the sink in a compressed and secure manner via the most cost-effective router. Sensors may be deployed in a hostile environment and may require enablement of security attributes adaptively based on observed data integrity and battery usage.

Genetic algorithm (GA) is a stochastic search technique that mimics the natural evolution proposed by Charles Darwin in 1858. GA has been successfully applied to a wide range of combination problems. They are particularly useful in applications involving design and optimization, where there are large numbers of variables and where procedural algorithms are either non-existent or extremely complicated. Simple GA converges to a single solution.

This paper extends our previous work on self-organization of static sensor networks [14] by adding mobility and security to improve data integrity, coverage, and network life. The goal is to develop a long-lasting secure sensor network containing mobile nodes with non-renewal and limited energy resource. To achieve this goal we discover clustered topology, optimal locality with optimal routes to the sink. These clusters have the ability to fuse the collected data at the cluster head, which are then routed to the sink using one or more hops.

II. Related work and motivation

Mobile sensor networks consist of randomly deployed disposable sensors where configurable objectives cooperate with one another to maximize coverage and battery life. At deployment, sensors could diffuse into the environment via random-walk. In this paper we use four competing objectives that create an energy-efficient sensor network: (a) *Cluster membership* that keeps on changing because of dead or depleted nodes, (b) *Routes to sink* that keeps on changing to avoid high-cost paths (like multiple clusters using the same inter-cluster router to route data to the sink), (c) *Sensor position* that dynamically adapts based on predicted optimal coverage, node traffic, and overhead traffic. The optimal position is predicted for cluster heads, routers, and sensor nodes based on factors that constitute the fitness function, and (d) *Sensor security* that dynamically enables the security attributes based on security threat and battery usage. Overall, the sensor network relies on continuous random motion to bring nodes into optimal contact for various reasons such as security, the shortest path for clusters-heads, and load migration, etc.

Previous work related to mobile sensor networks include dynamic approaches where sensor nodes are deployed one at a time, with each node making use of data gathered from previously deployed nodes to determine its optimal deployment location [8], potential fields to reduce deployment time [9], self-organization strategies and algorithms for responsive adaptation of sensor nodes to coverage of a field with multiple dynamically changing contexts [10], optimal deployment of sensors toward critical region to ensure quality of the readings of the value of interest [11]. Work related to sensor security include a solution [12], [13] using simple symmetric cryptographic algorithms. Asymmetric cryptographic algorithms are not suitable for providing security on wireless sensor networks due to limited computation, power, and storage resources available on sensor nodes. Although most of the schemes described above are promising, they do not deal with the sensor networks holistically that require optimization of the competing objectives (clustering, positioning, routing and security) for a high energy efficiency.

This paper is an extension of earlier work by Khanna *et al.* [14] that introduced a multi-objective genetic algorithms [15]

This work was supported in part by the National Science Foundation under grant DBI 0529223 and in part by Science and Technology Plan of Zhejiang Province, China under grant No. 2006E10010.

H. Liu is also with the Provincial Key Laboratory of Information Network Technology, Zhejiang University, Hangzhou, 310027, China.

1-4244-0353-7/07/$25.00 ©2007 IEEE

(MOGA) approach for achieving the first two objectives, i.e., *cluster membership* and *routes-to-sink*, for static sensors. The bulk of the work done focused on maximizing the coverage while minimizing the battery usage in stationary sensor networks. For problems where there are several, often conflicting objectives, an MOGA is used which evolves a set of solutions (the population) towards the Pareto-optimal front where trade-off analysis can be performed to select a suitable solution. This paper introduces an approach to deal with a more complex problem where secure coverage per unit of power of motion-capable sensors is maximized by analyzing two additional objectives: *sensor position* and *sensor security*, using secure protocols and locomotive abilities of these sensors. These objectives help generate optimal parameters related to (a) resolving routing imbalances, (b) optimal sensor allocation for various functions, (c) resolving load imbalances, (d) reducing overhead traffic, (e) optimal positioning of sensors to avoid shadowing effect, redundant usage, sub-optimal clustering, (f) load migration, and (g) security overhead for secure communication. Data security provides a unified and efficient scheme for maximum reliability and privacy. Mobility on the other hand provides an ability to re-position (or self-repair) the sensor (nodes, routers, cluster-heads) strategically so as to maximize the overall objective (cluster membership, security overhead, and routes) with an extra degree of freedom. However motion costs battery life and therefore sensors cannot be moved very frequently. Therefore, an efficient dynamic re-positioning and security uses long-range prediction based on historical trends or generational improvement over a period of time with the primary goal of maximizing the coverage in a resource constrained environment.

A. Representation of Static Sensors

As a part of our previous work, each individual sensor node is allocated a functional assignment using using genetic algorithm. These functions are represented as (a) inactive node (powered off), (b) cluster-head (CH), (c) inter-cluster router (ICR), and (d) sensor node (NS). Each cluster is represented by a cluster-head, and cluster-members are represented by inactive/active node sensors and ICRs. Cluster-head is responsible for data-fusion from various node-sensors and inter-cluster router is responsible for routing cluster data (from cluster-head) to the sink. In later sections we will introduce the mobility and security aspect to the GA fitness function along with its chromosome representation and co-existence with existing fitness parameters of importance. Algorithmic details regarding clustering, naming, routing using GA can be found in [14]. The fitness parameters defined in the earlier paper for optimal clustering of static sensor networks are:

1) **Coverage Fitness (CF)** optimizes the blanket coverage with an objective to maximize the total detection area.
2) **Cluster-Head Fitness (CHF)** defines the fitness based on the uniformity of the sensor nodes and cluster-heads.
3) **Node Communication Fitness (NCF)** defines the power required to communicate with the cluster-head that can be computed using the path loss.
4) **Battery Status Fitness (BF)** defines thresholds used to optimize node assignments w.r.t. battery status/usage.
5) **Router Load Fitness (RLF)** penalizes routers (ICR) if they cater to more than the average number of cluster-heads and ICR to avoid overloading.
6) **Sensor Effector Fitness (SEF)** interprets the power consumed by the sensory action of clusters. The net effect of SEF is to re-arrange the sensor nodes such that the sensor data transmission is uniformly optimized by fusion, elimination or compression methods.
7) **Total Node Fitness (TNF)** is evaluated in the GA algorithm for the appropriate node assignment as

$$\mathrm{TNF} = \alpha_1\mathrm{CHF} + \alpha_2\mathrm{NCF} + \alpha_3\mathrm{BF} + \alpha_4\mathrm{RFL} + \alpha_5\mathrm{SEF} + \alpha_6\mathrm{CF} \tag{1}$$

where $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6 = 1$ and α_i depends upon the relative significance of the component. These values can be made adaptive using an external heuristics.
8) **Route Selection Fitness Function (RSFF)** generates balanced routes based on node allocation using GA based on node fitness function. During setup operation, both CH and ICR start sending data on the most cost effective routers.

III. Mobility Extensions

In this section we will discuss locomotive details of the GA with a goal to achieve energy efficient deployment. In the GA modeling, TNF and RSFF cost functions are expanded with the parameters of the locomotion architecture and encryption algorithms explained later in Section IV that affect the transmission process such as available bandwidth, network bandwidth, packet size, CPU power consumption, RF distance, optimal routing, and protocol overhead. The final optimality equation is derived for the optimization and encryption decision process which is implemented by GA.

Mobility allows the nodes to seek out power optimization, request data fusion from other nodes to perform cooperative sensing, seek out repair, and locate data portals from which to report. But mobility comes with a price as locomotion is costly in terms of node size and power consumption. An optimal locomotion strategy is achieved by sensor node's ability to monitor its own power as well as its interaction with environmental dynamics. Locomotion is employed to maximize the fitness by rearranging its position to achieve the optimality of the parameters defined in the following sections.

A. Coverage Uniformity Fitness (CUF)

CUF expresses the coverage improvement by filling the coverage holes and maximizing the detection area using sensor movement. This is done by re-balancing the communication distances between member nodes of the cluster. Closely placed nodes are rewarded for moving away while farther nodes are rewarded for moving toward each other to attain coverage equilibrium. When the distances between nodes become optimal, the distance to the farthest neighboring node and the required transmission power are minimized which helps maximize the NCF. CUF is expressed as

$$\mathrm{CUF} = 1 - \frac{1}{2M}\sum_j \min\left(1,\ |d_{j\_\mathrm{min}} - d_{j\_\mathrm{mean}}|/d_{j\_\mathrm{mean}}\right) + \min\left(1,\ |e_{j\_\mathrm{min}} - e_{j\_\mathrm{mean}}|/e_{j\_\mathrm{mean}}\right) \tag{2}$$

where M is the number of clusters, d_{j_min}, d_{j_mean} are, respectively, the minimum and mean communication distances between nodes in cluster j, and e_{j_min} and e_{j_mean} are, respectively, the minimum and mean communication distances between nodes and cluster-head in cluster j.

Uniformly distributed sensor nodes spend energy more evenly than nodes with an irregular topology. GA exploits the sensor motion ability by positioning the sensors in a manner to increase the coverage, reduce the inter-node interference, and minimize the power required to communicate.

B. Cluster-Node Migration Fitness (CNMF)

CNMF aids in improving the uniformity of sensor nodes and cluster-heads by rewarding the migration of sensor nodes between cluster-heads with low CHF. Migration helps to achieve higher CHF if sensor migration is from high-density clusters to those with lower density. Cluster-node migration fitness can be expressed as

$$\text{CNMF} = \frac{1}{2N}\sum_{n}^{N}(\chi_{ns} + \chi_{nt}) \tag{3a}$$

$$\chi_{ns} = \min(1, \max(-1, (\rho_{ns} - \rho)/\rho)) \tag{3b}$$

$$\chi_{nt} = \max(0, \min(1, (\rho - \rho_{nt})/\rho)) \tag{3c}$$

where n is the n-th migration pair (source-target cluster), N is the total number of migration pairs, χ_{ns} is the source cluster's measure of excessive number of sensor nodes, χ_{nt} is the target cluster's measure of depleted number of sensor nodes, ρ_n is the number of nodes attached to this cluster-head, and ρ is the average number of nodes per cluster in a system calculated as

$$\rho = \text{Total Sensor Nodes}/\text{Total Cluster Heads}. \tag{4}$$

The fitness expression rewards the migration of sensor nodes if they reside in the low CHF clusters with high diffusion gradient between source and target clusters.

C. Cluster-Head Migration Fitness (CHMF)

CHMF rewards movement of the cluster-head and inter-cluster routers with lower router load fitness. Movement of the CH and ICR can help attain higher RLF due to the following:

1) ICR or CH movement can change the membership of the ICR based on various factors defined in [14]. This can result in optimizing the number of CH/ICR attached to the ICR that was moved.
2) ICR can also move to exchange roles with another functional node (cluster-head, sensor node). This can help maintain the existing topology by exchanging the nodes with higher battery capacity for router purposes (and exchanging the functional objectives).

Cluster-head migration fitness is expressed as

$$\text{CHMF} = \frac{1}{N}\sum_{n}^{N}\frac{1}{1+\eta_n}((1 - \text{RLF}_n) + \eta_n(1 - \text{BF}_{ns} + \text{BF}_{nt})) \tag{5}$$

where N is the total number of nodes-in-motion, RLF_n is the router load fitness (Section II-A) of n-th node, BF_{nt} is the battery fitness (Section II-A) of non-ICR node that is exchanged with n-th ICR node with BF_{ns}, η_n is the boolean that represents the presence of exchange pair for the n-th ICR.

It is evident from (5) that sensor movement is rewarded on ICRs and CHs with lower battery and router load fitness. Node movement influences the router's load by re-balancing and the battery capacity by exchanging functional objectives.

D. Node Motion Fitness (NMF)

The average distance traveled by a node is related to its movement at the expense of battery life. So, the expected distance is an important estimate of energy required for nodes with limited energy supply. Hence it is desired to stabilize the motion characteristics while achieving the overall system objectives (coverage and longer network life). These characteristics are related to motion frequency and oscillations.

1) *Motion Frequency* measures an average movement of the sensor in a given amount of time bounded by a threshold which is a function of the battery life defined by battery fitness. Larger movements of sensors with limited battery life is penalized which makes it highly prohibitive to achieve locomotion as the system ages.
2) *Location Stability* measures an inability of nodes to attain stable position due to competitive objectives. Nodes are penalized for having excessive movement or un-sustained oscillations.

Node motion fitness can be expressed as

$$\text{NMF} = ((1 - F_i(Q, \text{distance})) + (1 - \phi_i(n)))/2 \tag{6}$$

where $\phi_i(n)$ is the i-th sensor node's penalty measure for visiting the same location for n times ($0 \leq \phi_i(n) \leq 1$), $F_i(\cdot)$ is the i-th sensor node's penalty with $0 \leq F_i(Q, \text{Node Type}) \leq 1$, Q is the battery status represented in quantized steps, *distance* is the estimated distance traveled by the node which is estimated indirectly using energy-based localization based on multiple energy reading at different known sensor locations.

The signal energy measured on the i-th sensor over a time interval t, denoted by $y_i(t)$, can be expressed as

$$y_i(t) = \frac{G_i.S(t)}{|\boldsymbol{r}(t) - \boldsymbol{r}_i|^{\alpha}} + \epsilon_i(t) \tag{7}$$

where G_i is the gain factor of the i-th sensor, α (≈ 2) is an energy-decay factor, and $\epsilon_i(t)$ is the cumulative effects of the modeling error of the parameters, $S(t)$ denotes the energy emitted by the target at time t, $\boldsymbol{r}(t)$ is a $D \times 1$ vector denoting the coordinates of the target at time t, $\boldsymbol{r}_i$ is a $D \times 1$ vector denoting the cartesian coordinates of the i-th stationary sensor.

E. Sensor Data Fitness (SDF)

SDF measures sensor data efficiency with the net effect to re-position the sensor node such that its data transmission is uniformly optimized by fusion, elimination, or compression methods. This is further improved by optimizing the quality of sensing for a given SNR. Optimal sensing in a resource constrained (communication, battery, etc.) can be represented by $\theta(B, F)$, where B is the QoS requirements related to sensing operation and F is the timer policy. While QoS property is implemented to take advantage of variable data compression and fusion rules, a timer is implemented to vary the bit-rate depending upon conditions (density of sensors, etc.) of the sensor. Sensor movement is rewarded by reducing the average energy requirements in a cluster by:

1) Reduced variance in the timer activity due to load sharing by the recently moved sensor.
2) Reduction in the number of bits because of new fusion rule triggered due to recently moved sensors, and because of elimination of redundant sensing due to movement of redundant sensors.

Net result of the reward process is the optimal sensor density and bits per second for a given SNR. SDF is expressed as

$$\text{SDF} = \frac{1}{N}\sum_{n}^{N}(\lambda_1\psi(F, n) + \lambda_2\psi(B, n)) \tag{8a}$$

$$\psi(X, n) = \min\left(1, \max\left(0, \frac{X_{\mu}^{n}(s-1) - X_{\mu}^{n}(s)}{X_{\mu}^{n}(s-1)}\right)\right) + \min\left(1, \max\left(0, \frac{X_{\sigma}^{n}(s-1) - X_{\sigma}^{n}(s)}{X_{\sigma}^{n}(s)}\right)\right) \tag{8b}$$

where $\lambda_1 + \lambda_2 = 1$, $F = \{F_1, F_2....F_N\}$, and $B = \{B_1, B_2....B_N\}$ represents the average frequency and bit rate of each sensor node of the cluster n in which sensor node movement has been detected, $\psi(X, n)$ represents the improvement gain by a sensor parameter X represented by change in its mean(X_{μ}^{n}) and variance(X_{σ}^{n}) between consecutive sampling instances (s) in cluster n, λ_1 and λ_2 can be adjusted based on the sensor implementation.

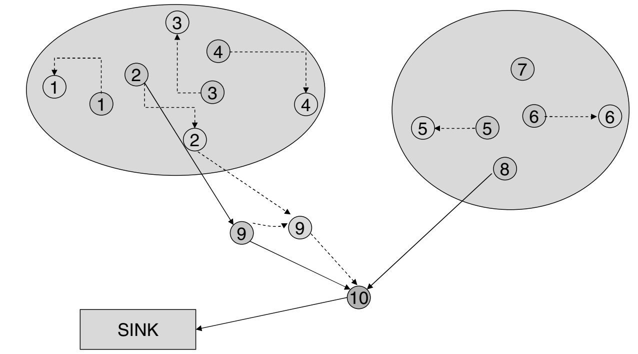
The total fitness associated with node movement is given by total node motion fitness

$$\text{TNMF} = \alpha_1\text{CUF} + \alpha_2\text{CNMF} + \alpha_3\text{NMF} + \alpha_4\text{CHMF} + \alpha_5\text{SDF} \tag{9}$$

where $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5$ = 1 and individual weight is dependent upon implementation.

F. Node Placement Genetic Algorithm

With the TNMS, we can design the algorithm for optimal node deployment using the GA operators. The GA executes in the sink or a similar centralized entity, where it repeats upon multiple triggers. These triggers are related to battery alert, deteriorating route fitness alert, and periodic action. Once the optimal fitness is achieved, the deployment corresponding to that fitness is committed and the sensors are instructed to assume the new positions by relinquishing the old positions.



```
(0100001 1100000 1011000 0010111) (1110001 0011100 1000000 1010101) (0010001) (0000000)
(1110001 0110001 0001101 1010011) (1010001 0010100 1000000 1010101) (0010001) (0000000)
(0110000 0110000 1011000 0010111) (1110001 0011100 1000000 1010101) (0010001) (0000000)
(    1        2        3        4    )(    5        6        7        8    )(    9    )(   10   )
```

Fig. 1. Node re-positioning as a result of genetic algorithm. In this example, nodes 1, 2 undergo 3 replacements; nodes 3, 4 undergo 2 replacements; nodes 5, 6, 9 are replaced only once. Other nodes do not move.

(a) **Chromosome Representation**: The chromosome of the GA is the building block to a solution of the problem at hand in a form that is suitable for the genetic operators and the fitness function. Chromosome string is formed using each individual sensor node's motion vector represented by a 7-bit binary number called 'gene' as shown in Fig. 1. The chromosome string hierarchy can be defined as
$((\widehat{\theta}x\theta x\widehat{S}xS)_1(\widehat{\theta}x\theta x\widehat{S}xS)_2(\widehat{\theta}x\theta x\widehat{S}xS)_3......)_1.......$
$((\widehat{\theta}x\theta x\widehat{S}xS)_1(\widehat{\theta}x\theta x\widehat{S}xS)_2(\widehat{\theta}x\theta x\widehat{S}xS)_3......)_n$
where $(\widehat{\theta}x\theta x\widehat{S}xS)_i$ represents the motion vector with the following properties:

  a) $(\widehat{\theta}\theta)$ represents 0^o(00), 90^o(01), 180^o(10), 270^o(11) angular movement
  b) $(\widehat{S}S)$ represents number of finite steps the sensor travels in the direction given by angular movement
  c) Sensor is moved only if one of the x values is 1.

(b) **Initial population**: Initial chromosome strings are seeded partially randomly using a random number generator (RNG) and partially using the population of previous samples. Population uses the gene structure as defined in Section II-A. This population is coded with gene structure as defined in Section II-A.

(c) **Evaluation**: Each chromosome string is evaluated for the fitness using the TNMF function (for node placement) as defined by Eq. (9).

(d) **Reproduction**: Reproduction allows individuals (strings) with larger fitness to have a higher probability of contributing an offspring in the next generation. Since the TNMF defines the fitness value, the chromosome with the highest TNMF value has a better chance to take part in reproduction. The algorithm uses the standard weighted roulette wheel method to select n individuals to the mating pool that produces N chromosomes using a crossover probability. During reproduction, we choose multiple cross-over points whose locations are calculated using an RNG.

(e) **Mutation**: Reproduced N chromosomes are transferred to the mutation pool where the mutation operator mutates them according to adaptive mutation probability which is inversely proportional to the average fitness. We will choose a maximum mutation probability p_m.

$$p_g = p_m(1 - (N * \text{TNMF}_{\text{avg}})/\text{NMF}_{\text{total}}). \qquad (10)$$

Mutation uses function flip (toss of a coin) to decide whether to invert the bit or not.

(f) **Selection**: Finally n chromosomes are chosen out of $N + n$ (n parent and N children) according to their fitness values and are carried over to the next generation.

IV. Security Extensions

So far we have defined three sensor objectives that execute in parallel using genetic algorithms. The first two objectives i.e., TNF and RSFF are defined in [14] and the third objective, NMF, is defined in Section III. The reliable functioning of these three objectives depend on the secure communications between various functional elements (nodes and sink). This requires identifying compromised or falsely added nodes, secure re-deployment/addition of nodes, and preventing passive listening by a malicious intruder using elements of authentication, integrity, privacy (or confidentiality), and anti-playback. All communications need to be secure to avoid data intercept, analysis and alteration by an intruder who can device methods to reduce the effectiveness of the sensor network. This has to be done in a manner such that time required to circumvent the security measures using brute-force methods takes longer than the life of the network.

In our security model, the sink is considered a trusted component that establishes a necessary trust relationship for secure forwarding of data between various node types. Nodes closest to the sink form the most trusted relationships. Farther nodes build the hierarchy of trust starting from the sink which is apparent from the pre-determined routing decisions that are created during setup and later during re-configuration [14]. Ingredients of security architecture create a trust relationship between various node-types for the reasons related command/message execution, data forwarding, etc. Any authentication is mediated through sink and components of the trusted routing hierarchy. To achieve a power efficient authentication we employ certain elements of secure network encryption protocol (SNEP) [13]. Encryption portion of the protocol is executed between first-initiator (FI) and the sink with other components in the hierarchy acting as an authenticated (or un-authenticated) pass-through. First initiator is a cluster-head or an ICR that either initiates a command or participates in data fusion for delivery to sink. Elements of security are:

1) ***Master Key (MK)*** derives keys for symmetric encryption (K_{encr}), message authentication (K_{auth}), and generates pseudo-random numbers (K_{rand}) [13]. The derived keys can be changed randomly upon request by the sink. The master key is shared between a node and the sink *a priori* and used for exclusive node-sink messaging. A pseudo-random number is generated using a derived key K_{rand} and a counter C. This number is inserted in the message before encryption to avoid plain-text attacks.

$$K_{\text{rand}}^{n+1} = \text{MAC}(K_{\text{rand}}, C^n) \qquad (11)$$

2) ***Inter-Node Communication Key (INCK)*** is the sink-mediated shared key between two nodes that authenticates (INCK_{mac}) the messages between them. Since the sink is aware of the routing hierarchy, it encapsulates an INCK={($\text{INCK}^0_{\text{mac}}$), ($\text{INCK}^1_{\text{mac}}$)} for each ICR (or CH) that takes part in authentication. Each node decrypts the encapsulated packet using its K_{encr} (derived from the master key) and extracts its INCK. $\text{INCK}^0_{\text{mac}}$ and $\text{INCK}^1_{\text{mac}}$ are the MAC keys used at ports 0 and 1, respectively.
3) ***Encryption and Authentication*** Similar to the SPIN protocol [13], we use counter-mode block cypher for encryption/decryption and CBC-MAC [16] for authentication. The counter-mode block cypher requires a shared counter between a node and the sink which is incremented on each block. Since it is a stream-cipher, the message length is the same as the plain text and hence a lower communication overhead. While some routers can be used as pass-through, other routers enforce admission control using MAC based authentication. Sink can change the authentication requirements of ICR and CH depending on energy requirements and perceived security threat as measured by the battery quantization levels and the number of bad packets.

Battery limitations and computational overhead prevent us from maintaining the same threat levels by employing encryption and authentication mechanisms on all nodes. Ideal enabling reduces the computational overhead while maintaining the adequate security levels by identifying the strategic nodes. Strategic nodes are optimally enabled for security by evaluating the battery status, network traffic, malformed or retries on a specific route and number of nodes in a single route handling authentication. For the purpose of GA, we evaluate a fitness function that competes for the optimal enablement of the security ingredients on the sensor nodes.

A. Secure Node Fitness (SNF)

SNF rewards the security enabling on nodes based on the perceived threat involving data integrity for secure communication. Sink keeps track of all the ill-formed packets received on a particular route. While routes (CH→sink) are penalized for carrying malformed and retried packets, they are rewarded for enabling authentication on routers (ICR) and encryption on cluster-heads. Additional penalty is awarded if the authentication is enabled disproportional to threat level quantized to M levels. While system can react proportionally to the perceived threat, it may not be enabled in an energy efficient manner. SNF rewards energy efficient enablement of security attributes that are measured against battery quantization levels and rate of battery usage. Sink uses hysteresis to compute the battery usage that is indicative of data communication, average number of connecting nodes, and locomotion, etc.

$$\text{SNF} = 1 - \frac{1}{2R}\sum_{i=1}^{R}\left(\left|\frac{\theta_i}{M} - \frac{\lambda_1 K_i}{N} - \lambda_2\right| + \sum_{n=1}^{N}\frac{I_i^n F_i^n(Q,\psi)}{N}\right) \tag{12}$$

where $\lambda_1 + \lambda_2 = 1$ with λ_2 being the reward contribution due to encryption of the first initiator (FI), R is the total number of routes, θ_i is the threat level of route i as calculated by sink, K_i is the number of nodes (ICR(s) and CH) that are enabled for authentication and encryption in route i, N is the total number of nodes (ICR(s) and CH) in route i, I_i^n=1 (else 0) if node n in route i is enabled for authentication, and $F_i^n(.)$ is penalty for enabling admission control on node i on route j that has battery level at Q and rate of battery usage at ψ.

B. Security Enablement Genetic Algorithm

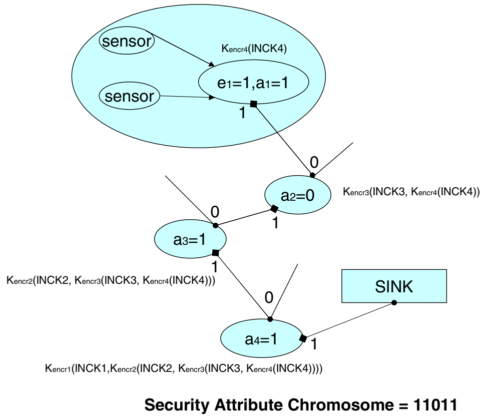

sensor
sensor
K_{encr4}(INCK4)
e_1=1,a_1=1
1
0
0
a_2=0
K_{encr3}(INCK3, K_{encr4}(INCK4))
1
a_3=1
1
K_{encr2}(INCK2, K_{encr3}(INCK3, K_{encr4}(INCK4)))
0
SINK
a_4=1
1
K_{encr1}(INCK1,K_{encr2}(INCK2, K_{encr3}(INCK3, K_{encr4}(INCK4))))
Security Attribute Chromosome = 11011

Fig. 2. Sink mediates the INCK keys at ICR and CH ports. These keys enable the authentication based on security attribute chromosome generated by genetic algorithm (GA). For example, node 2 does not require message authentication at port 0, but requires at port 1. $\text{INCK}^1_{\text{mac2}} = \text{INCK}^0_{\text{mac3}}$.

In this section we design the genetic algorithm for enabling security attributes (authentication & encryption) on nodes. These nodes are represented with a chromosome string that is formed using each individual sensor node's security policy represented by a 2-bit binary number and defined as

$$(e_1 a_1 a_2..a_N)_1 (e_1 a_1 a_2..a_N)_2..(e_1 a_1 a_2..a_N)_R$$

where $(e_1 a_1 a_2..a_N)_i$ represents the security attributes (e_n & a_n) on node n of route i, and e_n and a_i represent the encryption bit and authentication bit, respectively. It should be noted that the first node is always a CH where data encryption can optionally take place by setting $e_n = 1$. Genetic Algorithm (GA) steps are similar to those defined in section III-F.

Security settings competes with node-selection or locomotion. Nodes are assigned functions or locations based on the corresponding fitness factors which may be suboptimal for securing the packets due to battery conditions. This triggers re-configuration until all objectives reach an acceptable convergence. Like node/route selection, and mobility estimation, this is a dynamic process that repeats over system's life-time.

V. Results and Discussion

Experimental setup consists of 100 nodes at random positions in a 30×30 space. Individual node picks up a random coordinate between $(0, 0)$ and $(30, 30)$ and assigns itself an UUID and a random battery capacity between 0 and 15. For simplicity, each node is given a coverage area of 3×3. Once all the nodes have placed themselves in the listen mode, GA is run with the cross-over rate of 60% and an initial mutation of 6%. Experiment assumes line-of-sight propagation between sensor nodes. The software simulates the sink operation and runs in conjunction with NS-2 software that simulates the network traffic. It executes the GA that generates the motion path as well as security attributes. It also calculates the fitness parameters (Sections III and IV) based on network traffic, battery usage, and integrity of received packets. A separate process in the sink simulator runs a predictive algorithm that estimates the traffic and data integrity into the future using past hysteresis. This data is used to estimate the fitness parameters during the GA run. While each GA objective tends to compete with others to converge at the system equilibrium, the end result is to maximize the network life for optimal coverage.

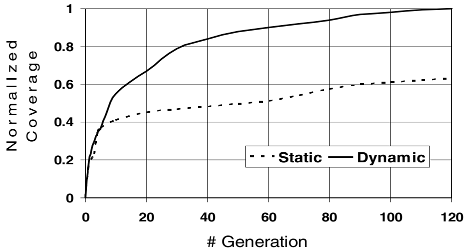


Fig. 3. Coverage as a function of n-th generation for static and dynamic deployment.

Fig. 3 shows the coverage as a function of the number of generations for static and dynamic deployment. It is observed that coverage is increased to about 30% as a result of dynamic deployment due to locomotion. While coverage is improved, energy cost may be increased due to sensor motion which affects the sensor-network life. Locomotion is accompanied with the communication overhead due to (a) encryption and authentication of motion commands and (b) temporary packet loss and data corruption due to node motion that triggers enhanced authentication attributes on the communication routes (GA Run). While battery cost of locomotion is compensated by communication cost reduction due to node-redeployment, it still reduces the overall benefit.

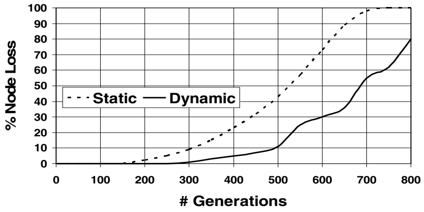


Fig. 4. Percentage of nodes lost (due to battery) as a function of n-th generation for static and dynamic deployment (50% threat level, θ_i = 0.5).

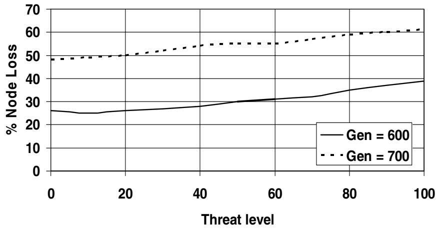


Fig. 5. Percentage of nodes lost (due to battery) as a function of threat level θ_i at the end of 600-th and 700-th generation.

Fig. 4 shows the node loss versus of the number of generations. It is found that a dynamic deployment significantly outperforms the static one with a 15-20% reduction in the number of lost nodes. While nodes are lost exponentially in static deployment case, they die gradually in clusters for the dynamic deployment case due to better distribution of the total energy. Coverage loss due to death of statically deployed nodes results in increased transmission energy and longer routes. Genetic algorithm enhances the coverage, life and integrity of the sensor network using the security and mobility extensions in addition to optimal node assignments. Furthermore, security extensions promotes added improvement over existing methods by dynamically switching authentication and encryption based on threat levels (Fig. 5) as perceived by the sink. Energy savings are realized due to reduced computation and header-data overhead on safe nodes (CH and ICR).

VI. Conclusion

This paper presents secure, dynamic, and energy-efficient deployment of mobile sensors using a multiple-objective genetic algorithm. This approach maximizes coverage and network life by exploiting mobility which optimally relocates sensor node that further optimizes node assignments, route and security attributes. We observe incremental improvement over static deployment that involved optimal functional and route assignments using GA [14]. An interesting outcome is the regional uniformity of the communication distances proportional to the sensor activity in that region. We observe a better distribution of energy among various functional nodes attributed to an extra degree of freedom that relocates the node strategically to achieve better battery utilization (fitness). This also reduces frequent re-clustering because now the roles are exchanged by just exchanging the positions while maintaining fitness parameters in equilibrium. Additionally, we develop a novel approach of adapting the security attributes proportional to the perceived threat and in a manner that promotes efficient battery usage and minimizes the effects of aberrant nodes. We will investigate the ability to exclude aberrant nodes from the network. Furthermore, we will also investigate effects of activity migration between hot and cold regions as well as better characterization of energy distribution over network life.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, Mar. 2002.

[2] K. Akkaya, M. Younis, "A Survey on routing protocols for wirelsss sensor networks," *Ad Hoc Networks*, vol. 3, pp. 325–349, May. 2005.

[3] R. Min, M. Bhardwaj, S.-H. Cho, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan, "Low power wireless sensor networks," in *Proc. Int. Conf. VLSI Design*, Bangalore, India, Jan. 2001.

[4] J. M. Rabaey, M. J. Ammer, J. L. da Silva, Jr., D. Patel, and S. Roundy, "PicoRadio supports ad hoc ultra low power wireless networking," *IEEE Computer*, vol. 33, pp. 42–48, July 2000.

[5] R. H. Katz, J. M. Kahn, and K. S. J. Pister, "Mobile networking for smart dust," in *Proc. 5th ACM/IEEE MobiCom*, Seattle, WA, Aug. 1999.

[6] H. O. Marcy, *et al.* "Wireless sensor networks for area monitoring and integrated vehicle health management applications," in *Proc. AIAA Conf. Guidance, Navigation, and Control*, Portland, OR, USA, Aug. 1999.

[7] D. Goldberg, *Genetic Algorithm in Search, Optimization and machine learning*, Addison-Wesley Publishing Company, Inc., 1989.

[8] A. Howard, M. J. Mataric, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor network," *Autonomous Robots*, vol. 13, pp. 113–126, Sep. 2002.

[9] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem.," in *DARS'02*, 2002.

[10] S. Y. Wu, C. H. Liu, and C. K. Tzeng, "Self-organization for dynamic context coverage in capability-constrained mobile sensor networks.," in *Proc. IEEE SUTC'06*, 2006.

[11] G. Trajcevski, P. Scheuerman, and H. Bronnimann, "Mission-critical management of mobile sensors (or, How to Guide a Flock of Sensors)," in *Proc. 1st Workshop DMSN*, Toronto, Canada, Aug. 2004.

[12] H. Cam, S. Ozdemir, D. Muthuavinashiappan and P. Nair, "Energy efficient security protocol for wireless sensor networks," in *Proc. IEEE*, pp. 2981-2984, 2003.

[13] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. D. Tygar, "SPINS: Security protocols for wireless sensor networks," in *Proc. ACM Mobile Comp. and Network*, 2001.

[14] R. Khanna, H. Liu and H. H. Chen,"Self-organization of sensor networks using genetic algorithms," in *Proc. IEEE ICC*, Istanbul, June 2006.

[15] J. Horn, J, N. Nafpliotis, and D. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization: Evolutionary Computation," in *Proc. 1st IEEE Conf. Computational Intelligence*, pp. 82–87, 1994.

[16] U. S. National Institute of Standards and Technology (NIST), "DES model of operation," in *Federal Information Processing Standards Publication*, 81 (FIPS PUB 81).