Adaptive Network Management for Countering Selective Capture in Wireless Sensor Networks

Hamid Al-Hamadi and Ing-Ray Chen
Virginia Tech
{hhamadi, irchen}@vt.edu

***Abstract*—We propose and analyze adaptive network management for countering selective capture which aims to compromise critical sensor nodes close to the base station in a wireless sensor network (WSN) to block data delivery. We consider 3 countermeasures in the protocol design: (1) dynamic radio range adjustment; (2) multisource multipath routing for intrusion tolerance; and (3) voting-based intrusion detection. We identify the best protocol settings in terms of the best redundancy level used for multisource multipath routing, and the best number of voters and the intrusion invocation interval used for intrusion detection under which the lifetime of a WSN is maximized in the presence of selective capture which turns nodes into malicious nodes capable of performing packet dropping attacks and bad-mouthing attacks.**

***Keywords* — Wireless sensor networks, selective capture, multipath routing, intrusion detection, lifetime maximization.**

I. INTRODUCTION

Many wireless sensor networks (WSNs) are deployed in an unattended environment in which energy replenishment is difficult if not impossible. Due to limited resources, a WSN must minimize energy consumption to prolong the system useful lifetime, while satisfying the application specific QoS requirements such as reliability, timeliness and security. This is especially a critical issue in military or mission-critical WSN applications.

It is well known that sensor nodes (SNs) close to the base station (BS) are more critical in gathering and routing sensing data. In the literature, various schemes [1, 3, 12] have been designed for preserving critical SNs from energy exhaustion so as to prolong the system lifetime; however, how to counter selective capture, i.e., critical SNs are targets of select capture attacks, is still an open issue [16].

In this paper, we propose and analyze an adaptive network management algorithm with 3 countermeasures to counter selective capture: (1) dynamic radio range adjustment; (2) multisource multipath routing for intrusion tolerance; and (3) voting-based intrusion detection. We develop a probability model to reveal the tradeoff between energy consumption vs. reliability and security gain with the goal to maximize the lifetime of a query-based WSN. More specifically, we analyze the optimal amount of redundancy for multipath routing and the best intrusion detection settings for detection strength under which the lifetime of a query-based WSN is maximized in the presence of selective capture.

II. SYSTEM MODEL

We consider a query-based WSN with low-power SNs distributed in a geographic area. There is a base station assigned to the WSN that interconnects the WSN to the outside world and that fields queries from the outside world for sensing results. The initial energy of each SN is E_o^{SN}. The deployment area of the WSN is assumed circular with radius r^{BS}.

All SNs are subject to capture attacks. With selective capture, the adversaries (humans or robots) strategically capture SNs and turn them into inside attackers. We represent the capture rate of a SN at a distance x away from the BS at time t by $\lambda_c^{SN}(x,t)$. A possible form is a linear function, i.e., the capture rate drops linearly as the SN is further away from the base station:

$$\lambda_c^{SN}(x,t) = \lambda_c^{max} - \frac{x}{r^{BS}}(\lambda_c^{max} - \lambda_c^{min}) \quad (1)$$

where λ_c^{max} is the maximum capture rate the adversary can possibly have; and λ_c^{min} is the minimum capture rate. With random capture, the adversary randomly performs capture attacks, i.e., $\lambda_c^{random} = (\lambda_c^{max} + \lambda_c^{min})/2$. After a node is compromised it becomes an inside attacker performing packet dropping [7], and bad-mouthing attacks to disrupt the operation of the network. A compromised node performs bad-mouthing attacks by recommending a good node as a bad node, and a bad node as a good node when participating in the voting-based distributed IDS as a voter (described below). As a result, bad-mouthing attacks can cause good nodes being misdiagnosed and evicted from the system, and bad nodes being missed and stayed in the system. This effectively creates an area with a high concentration of bad nodes, especially for critical SN areas with a high capture rate under selective capture.

We consider random deployment where SNs are deployed randomly and distributed according to homogeneous spatial Poisson processes with density λ_o^{SN}. The initial total number of SNs in the system thus is $N_o^{SN} = \lambda_o^{SN}\pi(r^{BS})^2$. Our first countermeasure against selective capture is dynamic radio range adjustment. With random deployment, the initial radio range is denoted by r_o^{SN} such that the average number of neighbor SNs is n_0, which is a system parameter for maintaining connectivity. A SN adjusts its radio range dynamically throughout its lifetime to maintain connectivity such that the average number of 1-hop neighbor SNs remains at n_0. Thus, SNs closer to the BS may have to increase radio range more than SNs away from the BS to counter selective capture. Any communication between two nodes with a distance greater

than single hop radio range between them would require a multi-hop.

Our second countermeasure against selective capture is multipath routing for intrusion tolerance. This is achieved through two forms of redundancy: (a) source redundancy by which m_s SNs sensing a physical phenomenon in the same feature zone are used to forward sensing data to the BS; (b) path redundancy by which m_p paths are used to relay packets from a source SN to the BS. We assume geographic forwarding is being used to packet routing; thus, no path information is maintained.

While data delivery could fail due to hardware failure and transmission failure because of noise and interference, we only consider failure caused by compromised nodes performing packet drop attacks and data modification attacks. We assume that SNs operate in power saving mode (e.g. [3, 12]). Thus, a SN is either active (transmitting or receiving) or in sleep mode. For the transmission and reception energy consumption of sensors, we adopt the energy model in [15] for SNs. We assume that the BS will have pairwise keys with the SNs. A SN also has a pairwise key with each of its neighbors, up to a few hops away for future expandability. Thus, a SN can encrypt data for confidentiality and authentication purposes.

Our last countermeasure against selective capture is voting-based intrusion detection system (IDS) mechanisms to detect and evict compromised nodes. Every SN runs a simple *host IDS* to assess its neighbors. The host IDS is light-weight to conserve energy. It is also generic and does not rely on the feedback mechanism tied in with a specific routing protocol (e.g., MDMP for WSNS [8] or AODV for MANETs [14]). It is based on local monitoring. That is, each node monitors its neighbor nodes only. Each node uses a set of anomaly detection rules such as a high discrepancy in the sensor reading or recommendation has been experienced, a packet is not forwarded as requested, as well as interval, retransmission, repetition, and delay rules as in [2, 4, 13]. If the count exceeds a system-defined threshold, a neighbor node that is being monitored is considered compromised. The imperfection of monitoring due to environment noise or channel error is modeled by a "host" false positive probability (H_{pfp}) and a "host" false negative probability (H_{pfn}) which are assumed known at deployment time.

A voting-based distributed IDS is applied periodically in every T_{IDS} time interval. A SN is being assessed by its neighbor SNs. In each interval, m neighbor SNs around a target SN will be chosen randomly as voters and cast their votes based on their host IDS results to collectively decide if the target SN is still a good node. The m voters share their votes through secure transmission using their pairwise keys. When the majority of voters come to the conclusion that a target node is bad, then the target node is evicted. There is a system-level false positive probability P_{fp} that the voters can incorrectly identify a good node as a bad node. There is also a system-level false negative probability P_{fn} that the voters can incorrectly misidentify a bad node as a good node. In the paper, we will derive the two system-level IDS probabilities based on bad-mouthing attacks performed by inside attackers.

To provide a unifying metric that considers the above two design tradeoffs, we define the total number of queries the system can answer correctly until it fails as the *lifetime* or the *mean time to failure* (the MTTF) of the system, which can be translated into the actual system lifetime span given the query arrival rate. A failure occurs when no response toward a query is received. The cause could be due to energy exhaustion, or packet dropping by malicious nodes. Our aim is to find both the optimal redundancy levels and IDS settings under which the MTTF is maximized, when given a set of parameters characterizing the operational and environment conditions.

III. PROBABILITY MODEL

In this section we develop a probability model to estimate the MTTF of a query-based WSN built with the three countermeasure mechanisms in the protocol design. A parameter can be an *input*, *derived*, *design* or *output* parameter. Specifically, m_p (path redundancy), m_s (source redundancy), m (the number of voters for intrusion detection) and T_{IDS} (the intrusion detection interval) are design parameters whose values are to be identified to maximize the system MTTF, when given a set of input parameter values charactering the operational and environmental conditions. Derived parameters are those deriving from input parameters. There is only one output parameter, namely, the MTTF. Note that most derived parameters are dynamic, i.e., as a function of time. For example, SN density, denoted by $\lambda^{SN}(x,t)$, decreases over time because of node failure/eviction as time progresses. On the other hand, radio range, denoted by $r^{SN}(x,t)$, increases over time to maintain connectivity.

The basic idea of our MTTF formulation is that we first deduce the maximum number of queries, N_q, the system can possible handle before running into energy exhaustion for the best case in which all queries are processed successfully. Because the system evolves dynamically, the amount of energy spent per query also varies dynamically. Given the query arrival rate λ_q as input, the average interval between query arrivals is $1/\lambda_q$. So we can reasonably estimate the amount of energy spent due to query processing and intrusion detection for query j based on the query arrival time $t_{Q,j}$. We then derive the corresponding query success probability $R_q(t_{Q,j})$, that is, the probability that the response to query j arriving at time $t_{Q,j}$ is delivered successfully to the BS. Finally, we compute the MTTF as the probability-weighted average of the number of queries the system can handle without experiencing any failure. More specifically, the MTTF is computed by:

$$MTTF = \sum_{i=1}^{N_q-1} i\left(\prod_{j=1}^{i} R_q(t_{Q,j})\right)\left(1 - R_q(t_{Q,i+1})\right) + N_q \prod_{j=1}^{N_q} R_q(t_{Q,j}) \qquad (2)$$

Here $\left(\prod_{j=1}^{i} R_q(t_{Q,j})\right)\left(1 - R_q(t_{Q,i+1})\right)$ accounts for the probability of the system being able to successfully execute i consecutive queries but failing the $i+1^{\text{th}}$ query. The second term is for the best case in which all queries are processed successfully without experiencing any failure for which the system will have the longest lifetime span.

A. Network Dynamics

Let $\lambda^{SN}(x,t)$ represent the density of SNs at distance x from the BS at time t. Initially at deployment time all SNs are good nodes, so $\lambda^{SN}(x,0) = \lambda_0^{SN}$ for all x's.

As time progresses some SNs are captured and turned into compromised nodes and some SNs fail. Moreover some SNs

may be diagnosed as bad nodes and get evicted from the system. Let T be the capture time of a SN following a distribution function $F_c(t)$. Then, the probability that a SN at location x away from the BS is compromised at time t, given that it was a good node at time t-T_{IDS}, denoted by $P_c^{SN}(x,t)$, is given by:

$$P_c^{SN}(x,t) = 1 - P\{T > t \mid T > t - T_{IDS}\} = 1 - \frac{P\{T > t, T > t - T_{IDS}\}}{P\{T > t - T_{IDS}\}} = 1 - \frac{1 - F_c(t)}{1 - F_c(t - T_{IDS})} \quad (3)$$

In the special case in which the capture time is exponential distributed, $P_c^{SN}(x,t) = 1 - e^{-\lambda_c^{SN}(x,t) \times T_{IDS}}$ for a SN at distance x from its BS. Recall that the voting-based distributed IDS executes periodically with T_{IDS} being the interval. At the i^{th} IDS execution time (denoted by $t_{I,i}$), a good SN at distance x from its BS may have been compromised with probability $P_c^{CH}(x, t_{I,i})$ since the previous IDS execution time $(t_{I,i-1})$. Let $\lambda_{good}^{SN}(x,t)$ and $\lambda_{bad}^{SN}(x,t)$ denote the densities of good, and bad SNs at distance x from the BS at time t, respectively. Then, the densities of good and bad SNs at time $t_{I,i}$ just prior to IDS execution can be recursively estimated from the densities of good and bad SNs at time $t_{I,i-1}$ by:

$$\begin{aligned} \lambda_{good}^{SN}(x, t_{I,i}) &= \lambda_{good}^{SN}(x, t_{I,i-1}) - P_c^{SN}(x, t_{I,i}) \lambda_{good}^{SN}(x, t_{I,i-1}) \\ \lambda_{bad}^{SN}(x, t_{I,i}) &= \lambda_{bad}^{SN}(x, t_{I,i-1}) + P_c^{SN}(x, t_{I,i}) \lambda_{good}^{SN}(x, t_{I,i-1}) \end{aligned} \quad (4)$$

The boundary conditions are $\lambda_{good}^{SN}(x,0) = \lambda_0^{SN}$ and $\lambda_{bad}^{SN}(x,0) = 0$ for all x's.

Every SN dynamically adjusts its radio range for maintaining connectivity with its peers such that on average the number of 1-hop neighbor nodes is n_o to support its intended functions including routing and participating in majority voting IDS as a verifier. In particular, critical SNs must increase radio range more due to more node eviction as a result of more intensive capture and bad-mouthing attacks toward critical SNs.

Let $r^{SN}(x,t)$ denote the radio range of a SN at distance x from its BS at time t so it can find n_o SNs within radio range. Since the SN density is a function of the distance (x) away from the BS, we have to solve $r^{SN}(x,t)$ by integration of the SN population from x-$r^{SN}(x,t)$ to x+$r^{SN}(x,t)$. Let X and Y be two variables denoting the X and Y coordinates in the X-Y coordinate system. Since $Y^2 = r^{SN}(x,t)^2 - X^2$ and $\int_0^{r^{SN}(x,t)} Y dX$ gives the area of the upper semicircle, the expected number of SNs covered by radio range, denoted by $r^{SN}(x,t)$, can be obtained by solving the following equation:

$$2 \int_{-r^{SN}(x,t)}^{r^{SN}(x,t)} \lambda^{SN}(x + X, t) \sqrt{r^{SN}(x,t)^2 - X^2} dX = n_o \quad (5)$$

where the integral gives the expected number of SNs (accounting for density variation along X) located in the upper or lower half circle.

Next we estimate the system-level false positive probability (P_{fp}) and false negative probability (P_{fn}) at time t as a resulting of executing voting-based IDS. For notational convenience, let $n^{SN}(x,t)$ be the number of neighbor SNs of a SN located at distance x from the BS at time t, $\check{n}^{SN}(x,t)$ be the number of forwarding neighbors (with f=1/4 for geographical routing), $n_{good}^{SN}(x,t)$ be the number of good neighbors, and $n_{bad}^{SN}(x,t)$ be the number of bad neighbors at time t. Since we know the densities of good and bad nodes at time $t_{I,i}$ just prior to IDS execution, we have:

$$n^{SN}(x,t) = 2 \int_{-r^{SN}(x,t)}^{r^{SN}(x,t)} \lambda^{SN}(t)(x + X, t) \sqrt{r^{SN}(x,t)^2 - X^2} dX \quad (6)$$

$$\check{n}^{SN}(x,t) = \int_{-r^{SN}(x,t)}^{0} \lambda^{SN}(t)(x + X, t) \sqrt{r^{SN}(x,t)^2 - X^2} dX$$

$$n_{good}^{SN}(x,t) = 2 \int_{-r^{SN}(x,t)}^{r^{SN}(x,t)} \lambda_{good}^{SN}(x + X, t) \sqrt{r^{SN}(x,t)^2 - X^2} dX$$

$$n_{bad}^{SN}(x,t) = 2 \int_{-r^{SN}(x,t)}^{r^{SN}(x,t)} \lambda_{bad}^{SN}(x + X, t) \sqrt{r^{SN}(x,t)^2 - X^2} dX$$

The above information allows us to derive the system-level false positive probability (P_{fp}) and false negative probability (P_{fn}) at time t, as follows (note: we omit distance x and time t in the mathematical expression below for brevity):

$$P_{fp} \text{ or } P_{fn} = \sum_{i=0}^{m-m_{maj}} \left[\frac{C\left(n_{bad}, m_{maj} + i\right) \times C\left(n_{good}, m - (m_{maj} + i)\right)}{C\left(n_{bad} + n_{good}, m\right)} \right] + \sum_{i=0}^{m-m_{maj}} \left[\frac{C\left(n_{bad}, i\right) \times \sum_{j=m_{maj}-i}^{m-i} \left[C\left(n_{good}, j\right) \times \omega^j \times C\left(n_{good-j}, m-i-j\right) \times (1-\omega)^{m-i-j} \right]}{C\left(n_{bad} + n_{good}, m\right)} \right] \quad (7)$$

where m_{maj} is the minimum majority of m, e.g., 3 is the minimum majority of 5, and ω is H_{pfp} for calculating P_{fp} and H_{pfn} for calculating P_{fn}. We explain Equation 7 for the false positive probability at time t below. The explanation to the false negative probability is similar. A false positive results when the majority of the voters vote against the target node (which is a good node) as compromised. The first term in Equation 7 accounts for the case in which more than 1/2 of the voters selected from the target node's neighbors are bad sensors who, as a result of performing bad-mouthing attacks, will always vote a good node as a bad node. Since more than 1/2 of the m voters vote no, the target node (which is a good node) is diagnosed as a bad node in this case, resulting in a false positive. Here the denominator is the total number of combinations to select m voters out of all neighbor nodes, and the numerator is the total number of combinations to select at least m_{maj} bad voters out of n_{bad} nodes and the remaining good voters out of n_{good} nodes. The second term accounts for the case in which more than 1/2 of the voters selected from the neighbors are good nodes but unfortunately some of these good nodes mistakenly misidentify the target nodes as a bad node with host false positive probability $H_{pfp,}$ resulting in more than 1/2 of the voters (although some of those are good nodes) voting no against the target node. Since more than 1/2 of the m voters vote no, the target node (which is a good node) is also diagnosed as a bad node in this case, again resulting in a false positive. Here the denominator is again the total number of combinations to select m voters out of all neighbor nodes, and the numerator is the total number of combinations to select i bad voters not exceeding the majority m_{maj}, j good voters who diagnose incorrectly with $i + j \geq m_{maj}$, and the remaining $m - i$

$-j$ good voters who diagnose correctly. Here we note that more voters do not necessarily provide better detection accuracy since it depends on the percentage of bad node population. That is, if more bad nodes exist than good nodes in the neighborhood, or good nodes have high host false positive probability (H_{pfp}) and host false negative probability (H_{pfn}), then more voters will provide less detection accuracy.

After the voting-based IDS is executed, a good node may be misidentified as a bad node with probability P_{fp} and will be mistakenly removed from the WSN. Consequently, we need to adjust the population of good nodes after IDS execution. Let $\overline{\lambda_{good}^{SN}(x,t_{I,i})}$ and $\overline{\lambda_{bad}^{SN}(x,t_{I,i})}$ denote the densities of good and bad SN nodes located at distance x from the BS, respectively, after IDS execution at time t. Then:

$$\begin{aligned}\overline{\lambda_{good}^{SN}(x,t_{I,i})} &= \lambda_{good}^{SN}(x,t_{I,i}) - \lambda_{good}^{SN}(x,t_{I,i}) \times P_{fp}^{SN} \\ \overline{\lambda_{bad}^{SN}(x,t_{I,i})} &= \lambda_{bad}^{SN}(x,t_{I,i}) - \lambda_{bad}^{SN}(x,t_{I,i}) \times (1 - P_{fn}^{SN})\end{aligned} \quad (8)$$

Therefore for a SN at distance x from its BS, the probability it is a bad SN at time $t_{I,i}$, denoted by $Q_{c,j}^{SN}(x,t_{I,i})$ where j denotes the node id of the SN, is given by:

$$Q_{c,j}^{SN}(x,t_{I,i}) = \frac{\overline{\lambda_{bad}^{SN}(x,t_{I,i})}}{\lambda_{bad}^{SN}(x,t_{I,i}) + \overline{\lambda_{good}^{SN}(x,t_{I,i})}} \quad (9)$$

$Q_{c,j}$ derived above provides critical information because a bad node can perform packet dropping attacks causing a path to be broken if it is on a path from source SNs to the BS.

Here we note that the good/bad node density will remain the same until the next IDS execution (after T_{IDS} seconds) because the IDS only detects and evicts nodes periodically (given that typically node hardware/software failure happens less frequently than security failure). The remaining nodes are good nodes that pass the IDS evaluation and bad nodes that are undetected by the IDS. Thus, $\overline{\lambda_{good}^{SN}(x,t_{I,i-1})}$ and $\overline{\lambda_{bad}^{SN}(x,t_{I,i-1})}$ obtained at time $t_{I,i-1}$ essentially become $\lambda_{good}^{SN}(x,t_{I,i}-1)$ and $\lambda_{bad}^{SN}(x,t_{I,i-1})$, respectively, for the next round of IDS execution at time $t_{I,i}$.

We can also estimate the number of SNs in the WSN at time t as:

$$N^{SN}(t) = \int_0^{r^{BS}} \lambda^{SN}(x,t)\ 2\pi x\, dx \quad (10)$$

B. Query Success Probability

We will use the notation SN_j to refer to SN j responsible to relay the packet for the *jth* hop from the source SN to the BS. Also we will use the notation $x(j)$ to refer to the distance from SN_j to its BS.

Let $D_{SN\text{-}BS}$ be the distance between a SN (selected to report sensor readings) and its BS, which on average is $r^{BS}/2$. Then the average numbers of hops to forward data from a source SN to the BS, denoted by N_{hop}^{SN}, can be estimated as follows:

$$\sum_{j=1}^{N_{hop}^{SN}} r^{SN}(x(j),t) = D_{SN-BS} \quad (11)$$

The equation above equates the sum of hop distances with the source-destination distance.

The success probability for SN_j to transmit a packet to at least p next-hop SN neighbors (with indices k=1, 2, … p) along the direction of the destination node based on geographical routing is given by:

$$\theta_j^{SN}(p) = \sum_{i=p}^{\check{n}^{SN}(x(j),t)} \left[\binom{\check{n}^{SN}(x(j),t)}{i} \prod_{k=1}^{i}(1 - Q_{k,c}^{SN}(x(k),t)) \prod_{k=i+1}^{\check{n}^{SN}(x(j),t)} Q_{k,c}^{SN}(x(k),t)\right] \quad (12)$$

where $Q_{k,c}^{SN}(x(k),t)$ is the probability that SN_k is compromised as derived in Equation 9, and $\check{n}^{SN}(x(j),t)$ is the number of forwarding neighbor SNs for SN_j as derived from Equation 6.

A path starting at SN_j to the BS is successful if in each hop there is at least one healthy next-hop SN neighbor found. Thus, the success probability of a path starting from SN_j (a source node has index j=1) to the BS is given by:

$$\varphi_j^{SN} = \prod_{l=j}^{N_{hop}^{SN}-1} \theta_l^{SN}(1) \quad (13)$$

For the 2nd countermeasure, we create m_p paths between a source SN and the BS for *path redundancy*. The m_p paths are formed by choosing m_p SNs in the first hop and then choosing only one SN in each of the subsequent hops. The source SN will fail to deliver data to the SN if one of the following happens: (a) none of the SNs in the first hop receives the message; (b) in the first hop, i ($1 \le i < m_p$) SNs receive the message, and each of them attempts to form a path for data delivery; however, all i paths fail to deliver the message because the subsequent hops fail to receive the broadcast message; or (c) in the first hop, at least m_p SNs receive the message from the source SN from which m_p SNs are randomly selected to forward data, but all m_p paths fail to deliver the message because the subsequent hops fail to receive the message. Summarizing above, the probability of a source SN (with index j=1) failing to deliver data to the BS through multipath routing is given by:

$$Q_1^{SN}(p) = 1 - \theta_1^{SN}(1) + \sum_{p=1}^{m_p} \theta_1^{SN}(p)\,[1 - \varphi_2^{SN}]^p \quad (14)$$

Consequently, the failure probability of data delivery to the BS from m_s source SNs, each utilizing m_p paths to relay data, is given by:

$$Q_f = [1 - (1 - Q_1^{SN})(1 - Q_{c,1}^{SN}(x(1),t))]^{m_s} \quad (15)$$

Therefore, the query success probability is given by:

$$R_q = 1 - Q_f \quad (16)$$

Note that in the above derivation we omit time for brevity. More precisely, R_q derived above should be $R_q(t_{Q,i})$ since the query success probability is a function of time, depending on the node count and population density at the i^{th} query's execution time (i.e., at time $t_{Q,i}$).

C. Energy Consumption

Now we estimate the amounts of energy spent by a SN located at distance x away from the BS during a query interval $[t_{Q,i}, t_{Q,i+1}]$ and an IDS interval $[t_{I,i}, t_{I,i+1}]$ so as to estimate N_q, the maximum number of queries this SN can possible handle before running into energy exhaustion. When a SN at distance x consumes all its energy, a 'black ring' at distance x away from the BS is formed. Nodes at distance greater than x will have to increase radio range in order to maintain

connectivity with the BS but eventually the system ceases to function. When selective capture is in effect, one can see that a black ring can more easily develop for nodes close to the BS.

To normalize energy consumption over N_q queries, let α be the ratio of the IDS execution rate to the query arrival rate so that αN_q is the numbers of IDS cycles before SN energy exhaustion. Then, we can estimate N_q by the fact that the SN energy consumed due to intrusion detection, and query processing is equal to the initial SN energy as follows:

$$E_o^{SN} = \sum_{i=1}^{\alpha N_q} E_q^{SN}(x, t_{Q,i}) + \sum_{i=1}^{N_q} E_{IDS}^{SN}(x, t_{I,i}) \quad (17)$$

Below we outline how to calculate $E_q^{SN}(x, t_{Q,i})$ and $E_{IDS}^{SN}(x, t_{I,i})$. We first estimate energy consumed by transmission and reception over wireless link. The energy spent by a SN to transmit an encrypted data packet of length n_b bits over a distance r is estimated as [15]:

$$E_T^{SN}(r) = n_b(E_{elec} + E_{amp} r^z) \quad (18)$$

Here E_{elec} is the energy dissipated to run the transmitter and receiver circuitry, E_{amp} is the energy used by the transmit amplifier, and r is the transmission radio range. We use the current SN radio range to derive E_T^{SN}. We set E_{amp} = 10 pJ/bit/m^2 and $z = 2$ when the radio range is less than a threshold distance d_0 (75m) and E_{amp}= 0.0013 pJ/bit/m^4 and $z = 4$ otherwise[15]. The energy spent by a node to receive an encrypted message of length n_b bits is given by:

$$E_R^{SN} = n_b E_{elec} \quad (19)$$

The energy consumed by a SN located at x for processing the i^{th} query, $E_q^{SN}(x, t_{Q,i})$, conditioning on it is being a data delivery path with probability $P_q^{SN}(x, t_{Q,i})$ is the energy consumed for reception (except when it is a source SN) and transmission, i.e.,

$$E_q^{SN}(x, t_{Q,i}) = P_q^{SN}(x, t_{Q,i}) \times [E_R^{SN} + E_T^{SN}(r^{SN}(x, t_{Q,i}))] \quad (20)$$

Since source SNs are randomly picked to answer a query, the probability that a SN at distance x away from the BS is on the data path $P_q^{SN}(x, t_{Q,i})$ is estimated as the probability of a SN at x is needed for data delivery, $(r^{BS} - x)/r^{BS}$, multiplied with the probability that this particular sensor is needed, $m_p m_s / N^{SN}(x, t_{Q,i})$. $N^{SN}(x, t_{Q,i}) = \int_{x - r^{SN}(x,t_{Q,i})}^{x + r^{SN}(x,t_{Q,i})} \lambda^{SN}(X, t)\ 2\pi X\, dX$ is the total number of SNs within the radio range of SNs at distance x.

For intrusion detection every node is evaluated by m voters in an IDS cycle, and each voter sends its vote to the other m - 1 voters. Hence, the energy spent by a SN located at x in the i^{th} IDS cycle, $E_{IDS}^{SN}(x, t_{I,i})$, conditioning on it serving as a voter with probability $P_{IDS}^{SN}(x, t_{I,i})$ for each of its $n^{SN}(x, t_{I,i})$ neighbors is the energy consumed for reception of m-1 votes and transmission of its vote to other m-1 voters, i.e.,

$$E_{IDS}^{SN}(x, t_{I,i}) = P_{IDS}^{SN}(x, t_{I,i}) \times n^{SN}(x, t_{I,i}) \times (m-1)[E_R + E_T^{SN}(r^{SN}(x, t_{Q,i}))] \quad (21)$$

Here the probability that a SN at distance x serves as a voter for a neighbor SN, $P_q^{SN}(x, t_{Q,i})$, is estimated as $m/n^{SN}(x, t_{I,i})$.

The system fails when a SN at distance $x = r_{max}^{SN}$ (SN maximum radio range) depletes its energy since there is no way to maintain connectivity even by dynamic range adjustment. That is, we set $x = r_{max}^{SN}$ to obtain $E_q(t_{Q,i})$, and $E_{IDS}(t_{I,i})$ from Equations 20, and 21, respectively, and then we calculate N_q from Equation 17. The knowledge of N_q along with $R_q(t_{Q,i})$ in Equation 16 allows us to calculate the system MTTF given by Equation 2.

IV. Adaptive Network Management for Countering Selective Capture

The objective of our adaptive network management algorithm is to dynamically adjust the best radio range to maintain network connectivity, to apply the best redundancy level in terms of path redundancy (m_p) and source redundancy (m_s), and to apply the best intrusion detection settings in terms of the number of voters (m) and the intrusion invocation interval (T_{IDS}) to counter selective capture so as to maximize the MTTF, in response to environment changes including SN node density and SN capture rate.

All nodes in the system act periodically to a "T_D timer" event to adjust the optimal parameter setting in response to changing environments. The optimal design settings in terms of optimal $T_{IDS}, m, m_s,$ and m_p are determined at static design time and pre-stored in a table over perceivable ranges of input parameter values. The BS performs a table lookup operation with extrapolation techniques applied to determine the optimal design parameter settings. The action performed by a BS upon a T_D timer event includes (a) determining $T_{IDS}, m, m_s,$ and m_p based on runtime knowledge of node density and attacker strength; and (b) notifying SNs of the new T_{IDS} and m settings. The action performed by a SN upon this T_D timer event is to adjust its radio range to maintain SN connectivity. The action taken upon receiving the control packet from its BS is to update the new T_{IDS} and m settings for intrusion detection. When a T_{IDS} timer event happens, each node in the system uses it current T_{IDS} and m settings to perform intrusion detection. When a data packet arrival event occurs, each SN simply follows the prescribed multipath routing protocol to route the packet. The complexity is O(1) for each SN because of the table lookup technique employed.

When the BS receives a query from a user, it triggers multipath routing for intrusion tolerance using the current optimal m_s and m_p settings to prolong the system useful lifetime. The complexity is also O(1) for the BS.

V. Performance Evaluation

In this section, we present numerical results. Our reference WSN consists of N_o^{SN}= 1500 SN nodes initially deployed with density λ_o^{SN} and the BS sitting at the center of a circular area with radius r^{SN} =300m. The selective SN capture time is assumed to be exponentially distributed following the linear model described by Eq. 1, with λ_c^{min} being once per 4 weeks and λ_c^{max} (=1/T_{comp}) varying in the range of once per half day to once per 3 days. The radio range r_{SN} is dynamically adjusted to maintain network connectivity of n_0= 7 to support basic multipath routing and voting-based IDS functions. The

initial energy level of a SN is $E_0^{SN} = 2$ Joule. The energy parameters used by the radio module are adopted from [6, 15]. The energy dissipation E_{elec} to run the transmitter and receiver circuitry is 50 nJ/bit. The energy used by the transmit amplifier to achieve an acceptable signal to noise ratio (E_{amp}) is 10 pJ/bit/m^2 for transmitted distances less than the threshold distance d_0 (75m) and 0.0013 pJ/bit/m^4 otherwise. The query arrival rate λ_q is a variable ranging from 10^{-2} to 1 query/sec to reveal points of interest. The host IDS false positive probability and false negative probability (H_{pfp} and H_{pfn}) vary between 1% and 5% to reflect the host intrusion detection strength as in [4].

Our objective is to identify the best protocol setting of our countermeasures against selective capture. This includes the radio range to be adjusted dynamically by individual SNs, the best redundancy level used for multipath routing, as well as the best redundancy level in terms of the number of voters and the best intrusion invocation interval used for intrusion detection to maximize the WSN lifetime in the presence of selective capture which turns critical nodes into malicious nodes capable of performing packet dropping attacks and bad-mouthing attacks.

We determine the optimal (m_p, m_s) setting (over the range of 1 to 5) under which the system MTTF is maximized through Equation 2. Tables I and II summarize the optimal (m_p, m_s) values to maximize the lifetime of the reference WSN under selective capture and random capture attacks, respectively, at m=3 (i.e., the number of voters is 3), with T_{comp} ($= 1/\lambda_c^{max}$) and T_{IDS} varying over a wide range of values. We first observe that there exists an optimal (m_p, m_s) setting under which the MTTF is maximized for either case. Furthermore, a higher (m_p, m_s) is needed when the attacker strength λ_c^{max} increases. Also under selective capture attacks, the system must use a higher redundancy level to maximize the MTTF. For example when T_{IDS} = 4 hrs and T_{comp}= 0.5 days, the optimal (m_p, m_s) setting is (2, 5) under selective capture (in Table 1) but is only (1, 5) under random capture attacks (in Table 2). This is because selective capture requires the system to apply more redundancy to cope with more critical nodes being compromised. The system is better off in this case to use higher redundancy to ensure secure routing at the expense of more energy consumption to maximize the system MTTF.

TABLE I: Optimal ($\boldsymbol{m_p}$, $\boldsymbol{m_s}$) to counter Selective Capture with m=3 and varying $\boldsymbol{T_{comp}}$ ($\boldsymbol{1/\lambda_c^{max}}$) and $\boldsymbol{T_{IDS}}$

T_{comp}	T_{IDS}=1hr	2hrs	4hrs	6hrs	8hrs
0.5 days	(1,2)	(1,4)	(2,5)	(2,5)	(2,5)
0.75 days	(1,2)	(1,2)	(1,4)	(2,5)	(2,5)
1 day	(1,2)	(1,2)	(1,2)	(1,4)	(2,5)
2 days	(1,2)	(1,2)	(1,2)	(1,2)	(1,2)

TABLE II: Optimal ($\boldsymbol{m_p}$, $\boldsymbol{m_s}$) to counter Random Capture with m=3 and varying $\boldsymbol{T_{comp}}$ ($\boldsymbol{1/\lambda_c^{max}}$) and $\boldsymbol{T_{IDS}}$

T_{comp}	T_{IDS}=1hr	2hrs	4hrs	6hrs	8hrs
0.5 days	(1,2)	(1,2)	(1,5)	(1,5)	(2,5)
0.75 days	(1,2)	(1,2)	(1,2)	(1,5)	(1,5)
1 day	(1,2)	(1,2)	(1,2)	(1,2)	(1,5)
2 days	(1,2)	(1,2)	(1,2)	(1,2)	(1,2)

We next analyze the effect of the intrusion detection interval T_{IDS} (representing the intrusion detection strength) on the system MTTF. Whether to use a small or large T_{IDS} value depends on the attacker strength λ_c^{max}. When the attacker strength is high (i.e., when $T_{comp} = 1/\lambda_c^{max}$ is small), as evidenced by the frequency at which bad nodes are detected by the IDS and evicted, we must counter it with high detection strength (a small T_{IDS}). Conversely, when the attacker strength is low, a large T_{IDS} could be used to save energy to maximize the MTTF. Figures 1 and 2 show the MTTF vs. (m_p, m_s) under small (T_{IDS}=1 hrs) and large (T_{IDS}=3 hrs) detection intervals, respectively, for the case when the attacker strength is high (T_{comp} = 0.5 days). We again set m=3 to isolate its effect. We observe that at the optimal (m_p, m_s) setting, the MTTF under T_{IDS}=1 hrs (Figure 1) is much higher than the MTTF under T_{IDS}=3 hrs (Figure 2). This is because when the system is subject to a high capture rate, the system is better off to apply high detection strength (a small T_{IDS} at 1 hrs) at the expense of more energy consumption to quickly detect and evict compromised nodes, instead of applying low detection strength (a large T_{IDS} at 3 hrs), so as to increase the MTTF. This trend applies to both selective and random capture attacks. We also see that the MTTF under selective capture is much lower than that under random capture because with selective capture critical nodes are more easily compromised and back holes more easily formed near the BS to cause system failure.

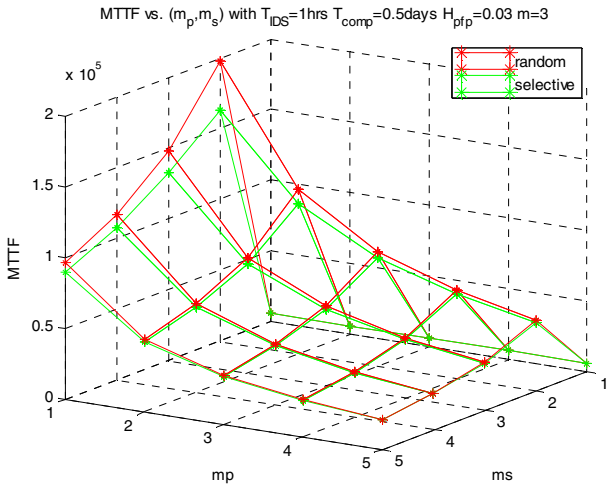

Figure 1: MTTF vs. (m_p, m_s) with High Detection Strength in the presence of High Attacker Strength.

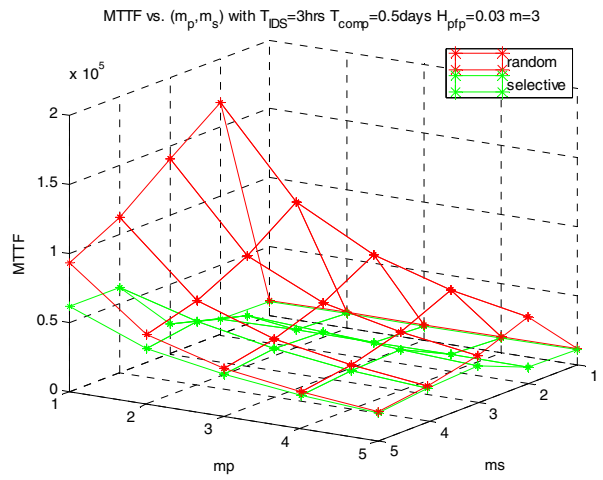

Figure 2: MTTF vs. (m_p, m_s) with Low Detection Strength in the presence of High Attacker Strength.

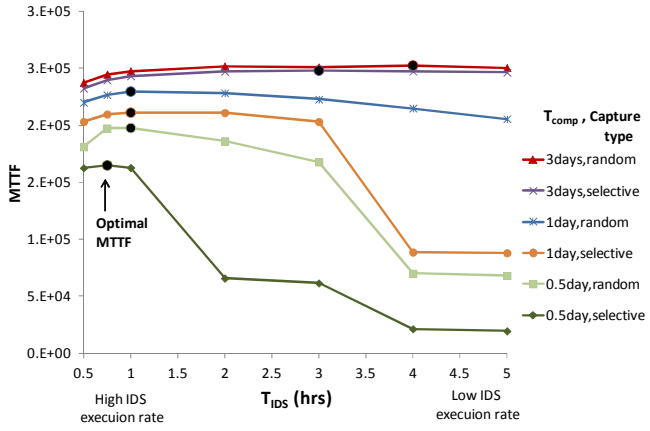

Figure 3: Effect of Countering Attacker Strength with $\boldsymbol{T_{IDS}}$ on the System MTTF under Random Capture vs. under Selective Capture.

Figure 3 compares the effect of T_{IDS} on the MTTF under random capture vs. selective capture at the optimal (m_p, m_s) setting under random capture vs. selective capture. We again observe that there exists an optimal T_{IDS} value (marked by a black dot) at which the MTTF is maximized. Furthermore, the optimal T_{IDS} value under selective capture in general is smaller than that under random capture because the system has to increase detection strength to cope with selective capture which creates more compromised critical nodes.

In Figure 4 we summarize the damaging effect of selective capture attacks compared with random capture attacks. It shows that selective capture has a devastating effect on the MTTF compared with random capture. The effect is especially pronounced when the attacker strength λ_c^{max} is high (left end of the graph where $T_{comp} = 1/\lambda_c^{max}$ is small). The MTTF at the optimal (m_p, m_s) setting under selective capture is relatively low compared with that under random capture because the success probability for a node to transmit a packet to at least p next-hop SN neighbors (Equation 11) is low as the node is close to the BS, as many critical nodes are compromised due to selective capture. The optimal (m_p, m_s) setting identified best balances the probability of query success probability (Equation 16) vs. energy consumption (Equation 17) to maximize the system MTTF.

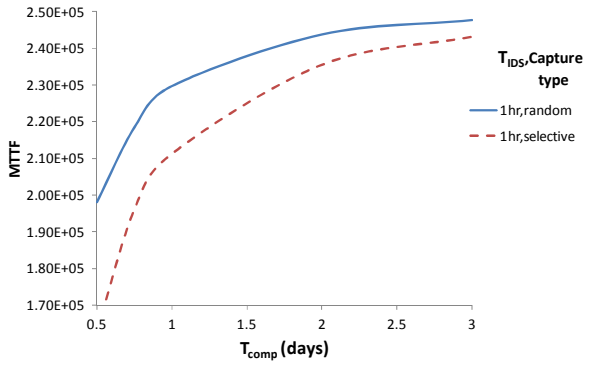

Figure 4: WSN lifetime under Random Capture vs. under Selective Capture with varying $\boldsymbol{T_{comp}}$

Figure 5 vividly displays how the "good SN density" $\lambda_{good}^{SN}(x,t)$ evolves over time under selective capture vs. under random capture. It confirms that $\lambda_{good}^{SN}(x,t)$ decreases over time because of capture, and the rate at which $\lambda_{good}^{SN}(x,t)$ declines for SNs with $x < 1/2$ under selective capture is higher than that under random capture. The effect of selective capture on good node population is especially pronounced for critical nodes near the BS (i.e., when x=1/16 or 1/8).

Figure 6 displays how a SN at distance x dynamically adjusts its radio range to counter selective capture so as to maintain sufficient network connectivity and improve packet delivery reliability. It confirms that with the "dynamic radio range adjustment" countermeasure, a SN increases its radio range over time to maintain network connectivity. Further, under selective capture because critical nodes (i.e., when x is small) are more likely compromised, and subsequently detected and evicted from the system, a critical node must increase its radio range more rapidly to maintain network connectivity and improve packet delivery reliability to effectively counter selective capture. Figure 6 demonstrates that critical SNs (e.g., when x=1/16 or 1/8) are able to more rapidly adjust radio range to maximize the system MTTF.

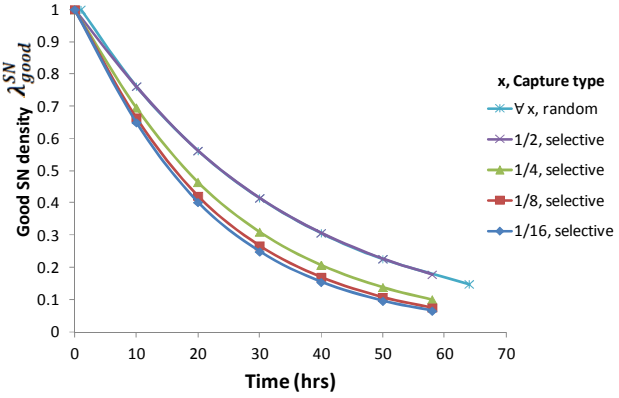

Figure 5: Density of Good SNs at Distance x vs. Time.

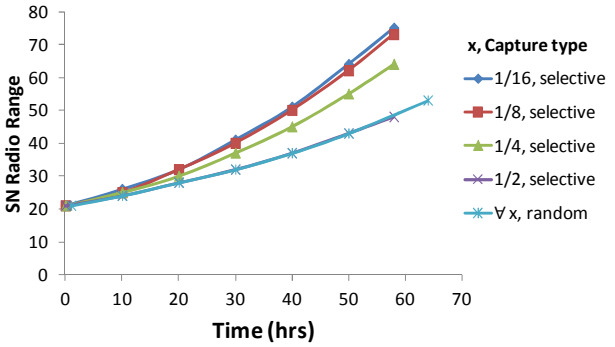

Figure 6: Adjusting Radio Range at Distance *x* vs. Time.

VI. CONCLUSION

In this paper we proposed and analyzed adaptive network management with three countermeasures for coping with selective capture aiming to create holes near the base station in a wireless sensor network to block data delivery. Through numerical analysis, we demonstrated that our countermeasures are effective against selective capture. There exist best protocol settings in terms of the best radio adjustment, the best redundancy level for multipath routing, the best number of voters, and the best intrusion invocation interval used for intrusion detection to maximize the system lifetime. Leveraging the analysis techniques proposed in this paper, one can obtain optimal protocol settings at static time, store them in a table, and apply a simple table lookup operation at runtime to determine optimal settings for adaptive network management to maximize the system lifetime without runtime complexity.

This paper considers three countermeasures against selective capture attacks. For future work, we plan to consider selective deployment, i.e., populating more critical nodes than edge nodes to effectively counter selective capture. We also plan to consider more sophisticated inside attacker behaviors including opportunistic, random and insidious behaviors [10] and investigate countermeasures against these attacker types. Finally, we also plan to investigate the use of trust/reputation management [5, 9, 11] augmented with fuzzy failure criteria [17, 18] to strengthen intrusion detection through "weighted voting," leveraging knowledge of trust/reputation of neighbor nodes, as well as to tackle the "what paths to use" problem in multipath routing for intrusion tolerance in WSNs. This may involve the use of trust-based admission control strategies [19-21] and location services [22-25] to increase the probability of path success probability for data delivery.

ACKNOWLEDGMENT

This work is supported in part by the U. S. Army Research Office under contract number W911NF-12-1-0445.

REFERENCES

[1] I. R. Chen, A. P. Speer, and M. Eltoweissy, "Adaptive Fault-Tolerant QoS Control Algorithms for Maximizing System Lifetime of Query-Based Wireless Sensor Networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 2, pp. 161-176, 2011.

[2] V. Bhuse and A. Gupta, "Anomaly intrusion detection in wireless sensor networks," *J. High Speed Netw.,* vol. 15, no. 1, pp. 33-51, 2006.

[3] G. Bravos and A. G. Kanatas, "Energy consumption and trade-offs on wireless sensor networks," *16th IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications*, pp. 1279-1283, 2005.

[4] A. P. R. da Silva, M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," *1st ACM Workshop on Quality of Service & Security in Wireless and Mobile Networks*, Montreal, Quebec, Canada, 2005.

[5] I.R. Chen, F. Bao, M. Chang, and J.H. Cho, "Trust management for encounter-based routing in delay tolerant networks" *IEEE Globecom 2010,* Miami, FL, Dec. 2010.

[6] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.,* vol. 1, no. 4, pp. 660-670, 2002.

[7] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *IEEE Int. Workshop on Sensor Network Protocols and Applications*, pp. 113-127, 2003.

[8] Y. Lan, L. Lei, and G. Fuxiang, "A multipath secure routing protocol based on malicious node detection," *Control and Decision Conference*, pp. 4323-4328, 2009.

[9] F. Bao, I.R. Chen, M. Chang, and J.H. Cho, "Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 161-183, 2012.

[10] R. Mitchell and I. R. Chen, "Effect of Intrusion Detection and Response on Reliability of Cyber Physical Systems," *IEEE Transactions on Reliability,* vol. 62, no. 1, pp. 199-210, 2013.

[11] F. Bao, I. R. Chen, M. Chang, and J.H. Cho, "Trust-based intrusion detection in wireless sensor networks," *IEEE International Conference on Communications*, Kyoto, Japan, June 2011.

[12] S. Qun, "Power Management in Networked Sensor Radios A Network Energy Model," *IEEE Sensors Applications Symp.*, pp. 1-5, 2007.

[13] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly detection in wireless sensor networks," *IEEE Wireless Communications,* vol. 15, no. 4, pp. 34-40, 2008.

[14] D. Somasundaram and R. Marimuthu, "A Multipath Reliable Routing for detection and isolation of malicious nodes in MANET," *17th IEEE Conf. on Computing, Communication and Networking*, pp. 1-8, 2008.

[15] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Trans. Mobile Comput.,* vol. 3, no. 4, pp. 366-379, 2004.

[16] Y. Zhou, Y. Fang, and Y. Zhang, "Securing wireless sensor networks: a survey," *IEEE Communications Surveys & Tutorials,* vol. 10, no. 3, pp. 6-28, 2008.

[17] I.R. Chen, and F.B. Bastani, "Effect of artificial-intelligence planning-procedures on system reliability," *IEEE Transactions on Reliability*, vol. 40, no. 3, pp. 364-369, 1991.

[18] I.R. Chen, F.B. Bastani, and T.W. Tsao, "On the reliability of AI planning software in real-time applications," *IEEE Transactions on Knowledge and Data Engineering*,vol. 7, no. 1, pp. 4-13, 1995.

[19] I.R. Chen and T.H. Hsi, "Performance analysis of admission control algorithms based on reward optimization for real-time multimedia servers," *Performance Evaluation*, vol. 33, no. 2, pp. 89-112, 1998.

[20] S.T. Cheng, C.M. Chen, and I.R. Chen, "Dynamic quota-based admission control with sub-rating in multimedia servers," *Multimedia systems,* vol. 8, no. 2, pp. 83-91, 2000.

[21] O. Yilmaz, and I.R. Chen, "Utilizing Call Admission Control for Pricing Optimization of Multiple Service Classes in Wireless Cellular Networks," *Computer Communications*, vol. 32, no. 2, pp. 317-323, 2009.

[22] B. Gu and I.R. Chen, "Performance analysis of location-aware mobile service proxies for reducing network cost in personal communication systems," *Mobile Networks and Applications*, vol. 10, no. 4, pp. 453-463, 2005.

[23] I.R. Chen, T.M. Chen, and C. Lee, "Performance evaluation of forwarding strategies for location management in mobile networks," *The Computer Journal*, vol. 41, no. 4, pp. 243-253, 1998.

[24] I.R. Chen, T.M. Chen, and C. Lee, "Agent-based forwarding strategies for reducing location management cost in mobile networks," *Mobile Networks and Applications*, vol. 6, no. 2, pp. 105-115, 2001.

[25] Y. Li and I.R. Chen, "Design and performance analysis of mobility management schemes based on pointer forwarding for wireless mesh networks," *IEEE Trans. on Mobile Computing*, vol. 10, no. 3, pp. 349-361, 2011.