Hierarchical Trust Management for Wireless Sensor Networks and Its Application to Trust-Based Routing

Fenye Bao, Ing-Ray Chen, Moonjeong Chang
Department of Computer Science
Virginia Tech
{baofenye, irchen, mjchang}@vt.edu

Jin-Hee Cho
Computational and Information Sciences Directorate
U.S. Army Research Laboratory
jinhee.cho@us.army.mil

ABSTRACT

In this work, we propose a highly scalable cluster-based hierarchical trust management protocol for wireless sensor networks to effectively deal with selfish or malicious nodes. Unlike prior work, we consider multidimensional trust attributes derived from communication and social networks to evaluate the overall trust of a sensor node. Our peer-to-peer trust evaluation method leverages the cluster-based hierarchical structure for efficient communications. We develop a probability model using stochastic Petri net techniques to analyze the performance of the proposed trust management protocol. We validate the protocol design by comparing *subjective trust* generated as a result of protocol execution against *objective trust* obtained from actual node status. We apply our hierarchical trust management protocol to trust-based geographical routing as an application. Our results demonstrate that trust-based geographic routing under identified design settings can approach the ideal performance level achievable by flooding-based routing in message delivery ratio and message delay without incurring substantial message overhead. Furthermore, it can significantly outperform traditional geographic routing protocols that do not use trust concept in selecting forwarding nodes in message delivery ratio over a wide range of design parameter settings.

Keywords

Trust management; geographic routing; wireless sensor networks; trust-based routing.

1. INTRODUCTION

A wireless sensor network (WSN) is usually composed of a large number of spatially distributed autonomous sensor nodes (SNs) to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants. A SN deployed in the WSN has the capability to read the sensed information and transmit or forward information to base stations or a sink node through multi-hop routing. While SNs have popularly used for various monitoring purposes such as wild animals, weather, or environments for battlefield surveillance, they also have severely restricted resources such as energy, memory, and computational power. Further, wireless environments give more design challenges due to inherently unreliable communications. A more serious issue is that nodes may be compromised and perform malicious attacks such as packet dropping or packet modifications to disrupt normal operations of a WSN wherein SNs usually perform unattended operations. A large number of SNs deployed in the WSN also require a scalable algorithm for highly reconfigurable communication operations. In this work, we consider a scalable hierarchical structure to deal with a large number of SNs with trust management mechanisms to identify selfish or malicious nodes for trust-based routing in WSNs.

We propose a hierarchical trust management protocol for cluster-based WSNs for efficient communications. Unlike prior work, we consider multidimensional trust attributes derived from communication and social networks to evaluate the overall trust of a sensor node (SN) for WSN applications wherein both *social trust* and *QoS trust* are important for mission execution. We apply our hierarchical trust management protocol to *trust-based geographical routing* as an application. Traditional geographic routing [5, 6] uses geographic location information to select the next forwarding node closest to the destination node, so that a message if delivered successfully may be delivered with the shortest delay. However, in the presence of selfish and malicious nodes, geographical routing may result in low message delivery ratio because the next forwarding node selected may be compromised or selfish, resulting in message losses. Unlike traditional geographical routing, trust-based geographical routing uses both trust and distance as criteria to select the most trustworthy neighbor nodes among those closest to the destination node for message forwarding so that a message may be delivered successfully with a high probability. The key design issues considered include trust formation (i.e., how a peer-to-peer trust value is formed), trust aggregation (i.e., how information is aggregated in parallel), and trust composition (i.e., what trust components are considered and their optimal weights) of the hierarchical trust management protocol and its application to trust-based geographical routing.

In the literature, trust has been used in WSNs for assessing the availability, reliability, or security property of a node (e.g., whether a node is malicious or not) based on past interaction experiences [1, 4, 7, 8, 10, 12]. Ganeriwal et al. [4] proposed a reputation-based framework for data integrity in WSNs. The proposed reputation system takes information collected by each node using a *Watchdog* mechanism (for direct monitoring and observations) to detect invalid data and uncooperative nodes. Yao et al. [12] proposed a parameterized and localized trust management scheme for WSN security, particularly for secure routing, where each node only maintains highly abstracted

SAC'11, March 21-25, 2011, TaiChung, Taiwan.

parameters to evaluate its neighbors. Aivaloglou and Gritzalis [1] proposed a hybrid trust and reputation management protocol for WSNs by combining *certificate-based* and *behavior-based* trust evaluations. However, [1, 4, 12] cited above only considered a node's QoS property in trust evaluation based on a flat architecture. Shaikh et al. [10] proposed a group-based trust management scheme for clustered WSNs in which each SN performs peer evaluation based on direct observations or recommendations, and each cluster head (CH) evaluates other CHs as well as SNs under its own cluster. This work is similar with ours in that a hierarchical structure is employed for scalability. However, they only considered QoS metrics (i.e. the message delivery ratio in a time window) based on direct observations. Liu et al. [7] and Moraru et al. [8] also proposed trust management protocols and applied them to geographic routing in WSNs. However, no hierarchical trust management was considered for managing clustered WSNs. Also, their work evaluated trust based on QoS aspects of a SN only such as packet dropping and the degree of cooperativeness while our work considers both QoS and social trust for trust evaluation of a SN.

2. SYSTEM MODEL

We consider a cluster-based WSN consisting of multiple clusters, each with a cluster head (CH) and a number of SNs in the corresponding geographical area. The CH in each cluster may be selected based on an election protocol such as HEED [13]. A SN forwards its sensor reading to its CH through SNs in the same cluster and the CH then forwards the data to the base-station or the destination node (or sink node) through other CHs. Leveraging this two-level of hierarchy in the WSN, our trust management protocol is conducted using periodic peer-to-peer trust evaluation between two SNs and between two CHs. At the SN level, each SN is responsible to report its peer-to-peer trust evaluation results towards other SNs in the same cluster to its CH which applies statistical analysis and performs CH-to-SN trust evaluation towards all SNs in its cluster.

Unlike prior work, we compose our trust metric by considering both *social trust* and *QoS trust* to take into account the effect of both aspects of trust on trustworthiness. Social trust may include friendship, honesty, privacy, similarity, betweenness centrality, and social ties (strengths) [3]. QoS trust may include competence, cooperation, reliability, task completion capability, etc. In this work, we adopt *intimacy* (for measuring social ties) and *honesty* (i.e., whether a node is compromised or not) to measure social trust derived from social networks. We choose *energy* (for measuring competence) and *unselfishness* (for measuring cooperativeness) to measure QoS trust derived from communication networks. The *intimacy* trust component reflects the relative degree of interaction experiences between two nodes. The *honesty* trust component indicates whether a node is compromised (being an inside attacker) or not based on intrusion detection capability in the system such as software-based code attestation [2]. Energy is one most important metric in WSNs since SNs are constrained in energy and we use energy as a QoS trust metric to measure if a SN is capable of performing its intended functionality. The unselfishness trust component reflects if a SN can cooperatively execute the intended protocol.

Our trust management protocol can apply to any WSN consisting of heterogeneous SNs with vastly different initial energy levels and different degrees of maliciousness or selfishness. We consider a clustered WSN in which a SN may adjust its behavior dynamically according to its own operational state and environmental conditions. A SN is more likely to become selfish when it has low energy or when it has many unselfish neighbor nodes around. Further, a SN is more likely to become compromised when it has low energy (a node with high energy may perform better energy-consuming defenses against attackers) or when it has more compromised neighbors around. A compromised SN can perform various attacks such as message dropping, good-mouthing attacks (recommending a bad node as a good node), and bad-mouthing attacks (recommending a good node as a bad node). A CH consumes more energy than SNs. After a SN or CH is compromised, it may consume more energy to perform attacks. On the other hand, a selfish node consumes less energy than unselfish nodes as its selfish behavior is reflected by stopping sensing functions and arbitrarily dropping messages.

3. HIERARCHICAL TRUST PROTOCOL

Our hierarchical trust management protocol maintains two levels of trust: *SN-level* trust and *CH-level* trust. Each SN evaluates other SNs in the same cluster while each CH evaluates other CHs and SNs in its cluster. The peer-to-peer trust evaluation is periodically updated based on either direct observations or indirect observations. When two nodes are neighbors within radio range, they evaluate each other based on direct observations via snooping or overhearing. Each SN sends its trust evaluation results toward other SNs in the same cluster to its CH. Each CH performs trust evaluation toward all SNs within its cluster. Similarly, each CH sends its trust evaluation results toward other CHs in the WSN to a "CH commander" which may reside on the base station if one is available, or on a CH elected if a base station is not available. The CH commander performs trust evaluation toward all CHs in the system. The election protocol is outside of the scope of the paper.

These two levels of peer-to-peer trust evaluation process consider four different trust components as described earlier: intimacy, honesty, energy, and unselfishness. The trust value that node i evaluates towards node j at time t, $T_{ij}(t)$, is represented as a real number in the range of [0, 1] where 1 indicates complete trust, 0.5 ignorance, and 0 distrust. $T_{ij}(t)$ is computed by:

$$T_{ij}(t) = w_1 T_{ij}^{intimacy}(t) + w_2 T_{ij}^{honesty}(t) + w_3 T_{ij}^{energy}(t) + w_4 T_{ij}^{unselfishness}(t) \quad (1)$$

where w_1, w_2, w_3, and w_4 are weights associated with these four trust components with $w_1 + w_2 + w_3 + w_4 = 1$.

3.1 Peer-to-Peer Trust Evaluation

Here we describe how peer-to-peer trust evaluation is conducted, particularly between two SNs or two CHs. When a trustor (node i) evaluates a trustee (node j) at time t, it updates $T_{ij}^X(t)$ where X indicates a trust component as follows:

$$T_{ij}^X(t) = \begin{cases} (1-\alpha) T_{ij}^X(t-\Delta t) + \alpha T_{ij}^{X,direct}(t), \\ \quad if\ i \text{ and } j \text{ are neighbors}; \\ \underset{k \in N_i}{\text{avg}} \{\gamma T_{ij}^X(t-\Delta t) + (1-\gamma) T_{kj}^{X,recom}(t)\}, \\ \quad otherwise. \end{cases} \quad (2)$$

In Equation 2, if node i is a 1-hop neighbor of node j, node i will use its direct observations ($T_{ij}^{X,direct}(t)$) and past experiences ($T_{ij}^X(t-\Delta t)$ where Δt is a trust update interval) toward node j to update $T_{ij}^X(t)$. A parameter α ($0 \le \alpha \le 1$) is used here to weight

these two contributions and to consider trust decay over time. A larger α means that trust evaluation will rely more on direct observations. Here $T_{ij}^{X,direct}(t)$ indicates node *i*'s trust value toward node *j* based on direct observations accumulated over the time period $[0,t]$ possibly with a higher priority given to recent interaction experiences over the time period $[t-\Delta t,t]$. Below we describe how each trust component value $T_{ij}^{X,direct}(t)$ can be obtained based on direct observations:

• $T_{ij}^{intimacy,direct}(t)$:This measures the level of interaction experiences. It is computed by the number of interactions between nodes *i* and *j* over the maximum number of interactions between node *i* and any neighbor node over the time period $[0,t]$.

• $T_{ij}^{honesty,direct}(t)$: This refers to the belief of node *i* that node *j* is not compromised based on node *i*'s direct observations toward node *j*. It can be a binary quantity, 0 or 1, based on the result of IDS deployed on node *i* about whether or not node *j* is compromised at time *t*.

• $T_{ij}^{energy,direct}(t)$: This indicates the percentage of node *j*'s remaining energy that node *i* directly observes at time *t*. Node *i* can overhear or even monitor node *j*'s packet transmission activities over the time period $[0,t]$ to estimate $T_{ij}^{energy,direct}(t)$.

• $T_{ij}^{unselfishness,direct}(t)$:This provides the degree of unselfishness of node *j* as evaluated by node *i* based on direct observations over$[0,t]$.Node *i* can apply overhearing and snooping techniques to detect selfishness behaviors and may give recent interaction experiences a higher priority over old experiences in estimating $T_{ij}^{unselfishness,direct}(t)$.

On the other hand, if node *i* is not a 1-hop neighbor of node *j*, node *i* will use its past experiences ($T_{ij}^{X}(t-\Delta t)$) and recommendations ($T_{kj}^{X,recom}(t)$ where *k* is a recommender) to update $T_{ij}^{X}(t)$. A parameter γ is used here to weight these two contributions and to consider trust decay over time as follows:

$$\gamma = \frac{1}{1+\beta T_{ik}^{honesty}(t)} \quad (3)$$

Here we introduce another parameter $\beta \geq 0$ to specify the impact of "indirect recommendations" on $T_{ij}^{X}(t)$ such that the weight assigned to indirect recommendations is normalized to $\beta T_{ik}^{honesty}(t)$ relative to 1 assigned to past experiences. Essentially, the contribution of recommended trust increases proportionally as either $T_{ik}^{honesty}(t)$ or βincreases. Instead of having a fixed weight ratio $T_{ik}^{honesty}(t)$ to 1 for the special case in which $\beta = 1$, we allow the weight ratio to be adjusted by adjusting the value of β and test its effect on protocol resiliency against malicious recommendation attacks such as good-mouthing and bad-mouthing attacks. Here, $T_{ik}^{honesty}(t)$ is node *i*'s *honesty* trust value toward node *k* as a recommender (for node *i* to judge if node *k* provides correct information). We note that node *i* can choose all its 1-hop neighbors(N_i) as recommenders. The new trust value $T_{ij}^{X}(t)$ in this case would be the average of the combined trust values of past trust information and recommendations collected at time *t*.

3.2 CH-to-SN Trust Evaluation

Each SN reports its trust evaluation toward other SNs in the same cluster to its CH. The CH then applies statistical analysis principles to $T_{ij}(t)$ values received to perform CH-to-SN trust evaluation towards node *j*. Further, the CH can also leverage $T_{ij}(t)$ values received to detect if there is any outlier as an evidence of good-mouthing or bad-mouthing attacks. Based on the resulting CH-to-SN trust evaluation result toward node *j*, the CH determines whether node *j* needs to be excluded from sensor reading and routing duties.

4. PERFORMANCE MODEL

We develop a probability model based on stochastic Petri nets (SPN) [9] techniques to describe the behavior of each SN or CH in a WSN. It provides a basis for obtaining global status of nodes in the system, thereby allowing us to derive *objective trust* against which subjective trust obtained as a result of executing our hierarchical trust management protocol can be checked and validated. We use SPN as our analytical tool due to its capability to represent a large number of states for complex systems where an underlying model is a semi-Markov or Markov model. Further, we develop novel iterative hierarchical modeling techniques to avoid state explosion problems and to yield efficient solutions. Figure 1 shows the SPN model that describes the behavior of a SN (or a CH). We consider a heterogeneous WSN consisting of *N* SNs uniformly distributed in an *M* by *M* square-shaped operational area. Each SN is attached to a CH based on its location and so the system will have N_{CH} clusters with N_{CH} CHs. CHs and SNs have radio range of *R* and *r* respectively. The trust update interval is Δt. Nodes are stationary after the initial deployment.

Below we explain how we construct the SPN model for describing the behaviors of a single node and how we compose a performance model for the entire WSN using a number of such SPN models (one for each node in the system).

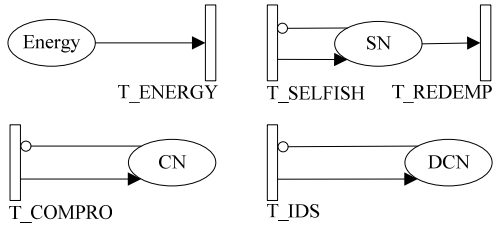

Figure 1: SPN Model for a Sensor Node or a Cluster Head.

• **Energy**: Place *Energy* indicates the remaining energy level of the node. The initial number of tokens in place *Energy* is set to E_{init}. A token will be released from place *Energy* when transition *T_ENERGY* is triggered. The rate of transition *T_ENERGY* indicates the energy consumption rate. A CH consumes more energy than a SN. The energy consumption rate is also affected by a node's state. It is lower when a node becomes selfish. It is higher when a node is compromised because it takes energy to perform attacks. We denote $\Delta_{E-sensor}$, Δ_{E-CH} and $\Delta_{E-compromised}$ as the energy consumption rates per Δt time for a normal SN, a normal CH, and a compromised node, respectively, which can be obtained by analyzing historical data with $\Delta_{E-sensor} < \Delta_{E-CH} < \Delta_{E-compromised}$. Thus, the energy consumption rates for a selfish SN and a selfish CH are $\rho\Delta_{E-sensor}$ and $\rho\Delta_{E-CH}$ per Δt time unit, respectively, where ρ is the rate of the energy consumption of a selfish node.

• **Selfishness**: We model the selfish behavior of a node as follows: A node may become selfish to save energy. A selfish node may

stop reading data and drop packets it receives. An unselfish node may decide whether it will be selfish or not upon every time interval T_s according to its remaining energy and the number of unselfish neighbors. A selfish node can be redeemed as unselfish based on trust evaluation performed in every trust update interval (Δt). We model these behaviors by putting a token into place *SN* when transition *T_SELFISH* is triggered and removing the token from place *SN* when transition *T_REDEMP* is triggered. A token in place *SN* thus indicates that the node is selfish. A node's selfish probability is modeled by:

$$P_{selfish} = \frac{1}{2}\left(\frac{E_{consumed}}{E_{init}} + \frac{N_{neighbor}^{unselfish}}{N_{neighbor}}\right) \quad (4)$$

where $E_{consumed}$ is energy consumed and E_{init} is the node's initial energy level. Thus $E_{consumed}/E_{init}$ represents the percentage of energy consumed. $N_{neighbor}^{unselfish}/N_{neighbor}$ is the percentage of unselfish neighbors where $N_{neighbor}^{unselfish}$ is the number of unselfish neighbors and $N_{neighbor}$ is the total number of neighbors. A node's selfish probability tends to be lower when a node has more energy and higher when the node has more unselfish neighbors as there are sufficient unselfish neighbors around to take care of sensor tasks. Thus, the rates of transitions *T_SELFISH* and *T_REDEMP* are given by $P_{selfish}/T_s$ and $(1 - P_{selfish})/\Delta t$ respectively. We assume all nodes are unselfish initially with no token in place *SN*.

• **Honesty**: A node becomes compromised when transition *T_COMPRO* fires and a token is put in place *CN*. The rate to *T_COMPRO* is modeled by:

$$\lambda_c = \lambda_{c-init}\left(\frac{E_{init}}{E_{remain}} + \frac{N_{neighbor}^{compromised}}{N_{neighbor}^{healthy}}\right) \quad (5)$$

where λ_{c-init} is the initial node compromising rate which can be obtained by first-order approximation based on historical data about the targeted network environments. E_{init} and E_{remain} indicate a node's initial energy and remaining energy, respectively. $N_{neighbor}^{compromised}$ and $N_{neighbor}^{healthy}$ are the numbers of compromised and healthy (not compromised) nodes in the neighborhood. $N_{neighbor}^{compromised}/N_{neighbor}^{healthy}$ refers to the ratio of the number of compromised 1-hop neighbors to the number of healthy 1-hop neighbors. Equation 5 models that a node is more likely to be compromised when it has low energy to perform energy-consuming defense mechanisms, and when there are more 1-hop neighboring compromised nodes around it due to their collusive attacks. We model the IDS behavior through transition *T_IDS*. A compromised node can be caught by IDS with the rate $(1 - P_{fn})/T_{IDS}$ for transition *T_IDS* where P_{fn} is the IDS false negative probability and T_{IDS} is the IDS detection interval. When a compromised node is caught by IDS, a token will move to place *DCN*. In addition, we model false positives generated by the IDS (i.e., diagnosing a good node as a bad node) by associating a rate of P_{fp}/T_{IDS} with transition *T_IDS* which is enabled only when the node is not compromised, that is, when there is no token in place *CN*. Note that all nodes are healthy, i.e., not compromised, initially.

The overall performance model for describing the behaviors of a WSN consists of *N* SPN subnet models one for each SN, and N_{CH} SPN subnet models one for each CH, with vastly different energy consumption, selfish/redemption and compromise rates. Below we describe how one could leverage SPN outputs to obtain subjective trust and objective trust values for performance evaluation of our hierarchical trust management protocol.

4.1 Subjective Trust Evaluation

Table 1: Instantaneous Subjective Trust $T_{ij}^{X,direct}(t)$ for Component X based on Direct Observations.

Item	Value	Condition (of node *j*)
$T_{ij}^{intimacy,direct}(t)$	*a* / *c*	*If mark*(*SN*) = 1 AND *mark*(*CN*) = 0
	b / *c*	*If mark*(*CN*) = 1
	1	Otherwise
$T_{ij}^{honesty,direct}(t)$	1	*If mark*(*DCN*) = 0
	0	Otherwise
$T_{ij}^{energy,direct}(t)$	*mark*(*Energy*)/E_{init}	none
$T_{ij}^{unselfishness,direct}(t)$	1	*If mark*(*SN*) = 0
	0	Otherwise

Recall that under our proposed trust management protocol, node *i* will subjectively assess its trust toward node *j*, $T_{ij}(t)$, based on its direct observations and indirect recommendations obtained toward node *j* according to Equations 1 and 2. In particular, node *i* will apply monitoring, snooping and overhearing techniques to watch node *j* (a 1-hop neighbor to node *i*) closely to compute $T_{ij}^{X,direct}(t)$ based on direct observations over the time period $[0, t]$. As a result, $T_{ij}^{X,direct}(t)$ computed by node *i* will fairly accurately reflect actual status of node *j* at time *t*. Leveraging the SPN model developed which provides actual status of each node dynamically, we can easily obtain this *instantaneous* subjective trust $T_{ij}^{X,direct}(t)$ of node *i* toward node *j* in component *X* at time *t* as listed in Table 1. In particular, $T_{ij}^{honesty,direct}(t)$, $T_{ij}^{energy,direct}(t)$, and $T_{ij}^{unselfishness,direct}(t)$, can be easily computed by simply checking the status of node *j* at time *t* in node *j*'s SPN model; $T_{ij}^{intimacy,direct}(t)$ is computed based on interaction experiences for packet forwarding events. We consider four types of interactions, given that node *i* is the initiating node: (1) *Requesting*: Node *i* broadcasts a packet delivery request to its 1-hop neighbors; (2) *Reply*: Nodes that are closer to the destination node than node *i* will reply to node *i*; (3) *Selection*: Node *i* selects up to *L* nodes with the highest trust values to forward the packet; (4) *Overhearing*: Node *i* overhears if the packet has been forwarded.

In practice, node *i* will keep track of its interaction experiences with node *j* to compute $T_{ij}^{intimacy,direct}(t)$. Let the average numbers of interactions of node *i* with a selfish node, a compromised node and a normal node be *a*, *b* and *c*, respectively. Then the instantaneous subjective trust $T_{ij}^{intimacy,direct}(t)$ of node *i* toward node *j* based on direct observations will be *a*/*c*, *b*/*c*, or *c*/*c*, respectively, depending on if node *j* is a selfish node, a compromised node, or a normal node. The values of *a, b, c* are computed dynamically. Below we predict their values from node *i*'s perspective for the case in which a selfish node drops 50% of packets and a compromised node drops 100% of packets. On the one hand, if node *i* requests a neighbor to forward a packet then (1) the expected number of interactions between node *i* and a selfish node *j* is 25%×50%×3 because there will be three interactions (reply, selection, and overhearing) only if the selfish node is in a quadrant closer to the destination node (with 25% probability) and does not drop the packet (with 50% probability);

(2) the expected number of interactions between node i and a compromised node j is 0 because a compromised node discards all requests from node i; and (3) the expected number of interactions between node i and a normal node j is 25%×3 because there will be three interactions only if that node is in a quadrant closer to the destination node (with 25% probability). On the other hand, if node i receives a request from node j to forward a packet, the expected number of interactions will be 25%×2 because from node i's perspective there will be two interactions (reply and selection) only if node i is in the quadrant closer to the target node. Summarizing above, we can predict:

$$a = 25\% \times 50\% \times 3 + 25\% \times 2;$$
$$b = 0 + 25\% \times 2; \quad (6)$$
$$c = 25\% \times 3 + 25\% \times 2.$$

Once $T_{ij}^{X,direct}(t)$is computed, note i will compute $T_{ij}^{X}(t)$ based on Equation 2 and subsequently compute $T_{ij}(t)$ based on Equation 1.

4.2 Objective Trust Evaluation

To validate subjective trust evaluation, we compute objective trust based on actual status as provided by the SPN model output. The objective trust value of node j, $T_{j,obj}(t)$, is also a weighted linear combination of four trust component values:

$$T_{j,obj}(t) = w_1 T_{j,obj}^{intimacy}(t) + w_2 T_{j,obj}^{honesty}(t) + w_3 T_{j,obj}^{energy}(t) + w_4 T_{j,obj}^{unselfishness}(t) \quad (7)$$

where $T_{j,obj}^{intimacy}(t)$, $T_{j,obj}^{honesty}(t)$, $T_{j,obj}^{energy}(t)$, and $T_{j,obj}^{unselfishness}(t)$ can be obtained directly from the SPN model output reflecting node j's actual status at time t.

5. TRUST EVALUATION RESULTS

Table 2: Default Parameter Values Used.

Param	Value	Param	Value	Param	Value
M	900m	R	150m	r	50m
N	900	N_{CH}	81	Δt	10$min.$
α	0.5	β	0.5	$1/\lambda_{c\text{-}init}$	[4,18]hrs
$\Delta_{E\text{-}sensor}$	10$min.$	$\Delta_{E\text{-}CH}$	20$min.$	$\Delta_{E\text{-}compromised}$	30$min.$
ρ	1/3	T_{IDS}	10$min.$	P_{fp},P_{fn}	0.5%
T_s	[10,60]$min.$			w_1, w_2,w_3, w_4	0.25
E_{init}	[18,24]hrs for SNs, [36,48]hrs for CHs.				

In this section, we show numerical results obtained through model-based evaluation as described in Section 4. Table 2 lists default parameters used. We consider a WSN with 900 SNs (and 81 CHs) evenly spread out in a 900m×900m operational area based on uniform distribution. We set radio range R=150m and r=50m. The initial energy lifetime of a SN varies from 18hrs to 24hrs while the CHs have much higher initial energy lifetime ranging from 36hrs to 48hrs. The WSN is assumed to be deployed in a hostile environment with the node's average compromising interval in the range of 4hrs to 18hrs. We consider the worst case of good-mouthing (providing the highest trust value of 1 for a malicious node) and bad-mouthing attacks (providing the lowest trust value of 0 against a good node). Further, we use P_{fp} = P_{fn} = 0.5% which deems acceptable [2]. Below we present CH-to-SN trust evaluation results for a SN arbitrarily chosen based on peer-to-peer trust evaluation results reported by other SNs in the same cluster, and compare them against objective trust evaluated based on the SN's actual status. We vary parameter values to reflect changes to the environmental and operational condition and test their effects on subjective vs. objective trust values obtained. The node is a good node at time t=0 and then becomes a bad node based on its compromise rate.

Figure 2 compares subjective trust (using equal weight with w_1:w_2:w_3:w_4=0.25:0.25:0.25:0.25) vs. objective trust obtained, with α varying over a wide range (using a larger α indicates that subjective trust evaluation relies more on direct observations compared with past experiences). We fix β to 0.5 to isolate out its effect. We observe that subjective trust initially approaches objective trust as more recent direct observations are used. However, we also observe a crossover time point (for $\alpha \geq 0.5$) after which subjective trust is lower than objective trust. This implies that when sufficiently large amount of direct information is used for trust evaluation, subjective trust tends to be underestimated but does not cause any risk by over-trusting a trustee.

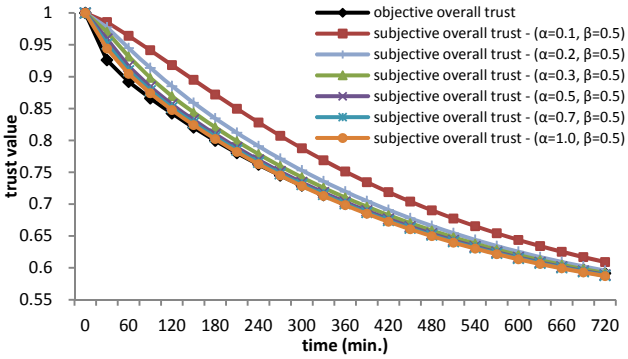

Figure 2: Effect of α on Trust Evaluation.

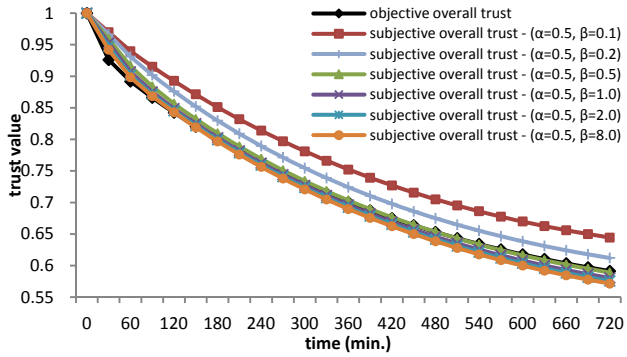

Figure 3: Effect of β on Trust Evaluation.

Figure 3 shows the effect of β on subjective trust. A higher β value indicates that subjective trust evaluation relies more on indirect recommendations provided by the recommenders compared with past experiences. We vary β from 0.1 to 8.0 to cover a wide range of possible values with α fixed at 0.5 to isolate out its effect. One can see that the subjective trust value approaches the objective trust value as β increases, but underestimates the trust value once a cross-over time point is reached particularly for $\beta > 1$.

Figures 2 and 3 show that α = 0.5 and β = 0.5 yield subjective trust values very close to objective trust values with the mean square error percentage less than 1%. The choice of the best α and β values depends on the given set of parameter values as those listed in Table 2 characterizing the environmental and operational conditions. The model-based analysis methodology developed in

the paper allows the best combination of α and β values to be determined. Overall, we observe a close correlation between subjective trust evaluation and objective trust evaluation, thus supporting our claim that subjective trust obtained as a result of executing our proposed hierarchical trust management protocol approaches true objective trust.

6. APPLICATION: TRUST-BASED GEOGRAPHIC ROUTING

We apply the proposed hierarchical trust management protocol to *trust-based geographic routing* as an application. In *geographic routing*, a node disseminates a message to a maximum of L neighbors closest to the destination node (or the sink node). In *trust-based geographical routing*, node i forwards a message to a maximum of L neighbors not only closest to the destination node but also with the highest trust values $T_{ij}(t)$. We conduct a performance analysis to compare our trust-based geographical routing protocol with baseline routing protocols, namely, flooding-based [11] and traditional geographic routing. In *flooding-based routing*, a node floods a message to all its neighbors until a copy of the packet reaches the destination node. It yields the highest message delivery ratio and the lowest message delay at the expense of the highest message overhead.

Recall that for all routing protocols, the source SN first forwards a message to its CH (through multiple hops if necessary). Then, the CH forwards the message to the sink node through other CHs. Without loss of generality, we normalize the average delay for forwarding a message between two neighboring SNs to τ. The average delay between two neighboring CHs is normalized to 2τ. We collect data for delivering 1000 messages, each with a source sensor and a sink node randomly selected. We consider two cases: $L=1$ and $L=2$ for both *trust-based geographic routing* and *geographic routing*. We use the optimal set of (α, β)=(0.5, 0.5) identified in Section 5 to ensure subjective trust is close to objective trust. We also use parameter values as listed in Table 2 for characterizing environmental and operational conditions. In the comparative analysis, we vary the degree of selfish or compromised nodes from 0% to 90%. Note that 30% of compromised or selfish nodes means that 30% of nodes are compromised or selfish in the system without a fixed ratio being used for these two types of nodes.

Figure 4 shows the message delivery ratio under various routing protocols. Our trust-based geographic routing protocol (L=1 or L=2) outperforms traditional geographic routing (L=1 or L=2) and approaches flooding-based routing, especially as the percentage of compromised or selfish nodes increases. The delivery ratio for all three routing protocols drops below 0.1 when the percentage of compromised or selfish nodes is higher than 80%. We observe that even the message delivery ratio of our trust-based geographic routing without redundancy ($L=1$) is higher than that of the geographic routing with redundancy ($L=2$) when the percentage of compromised or selfish nodes is higher than 40%. We attribute this to the ability of trust-based geographic routing being able to successfully avoid forwarding messages to untrustworthy nodes based on $T_{ij}(t)$ values obtained from our hierarchical trust management protocol.

Figure 5 shows the average delay for those messages that are successfully delivered under various routing protocols. Flooding-based routing has the best performance since it can always find the shortest path to reach the destination sink node through flooding. Geographic routing ($L=1$ or $L=2$) has almost the same performance with flooding-based routing due to its greedy nature for selecting nodes closest to the destination sink node for message forwarding. Trust-based geographic routing with L=1 has the highest delay but with L=2 approaches the performance of flooding-based routing and geographic routing. The average delay of all routing protocols drops as the percentage of compromised or selfish nodes increases. Further, the average delay of all routing protocols is below 3τ when the percentage of compromised or selfish nodes is higher than 80% since the message can be successfully delivered only if the source sensor and the sink node are close to each other.

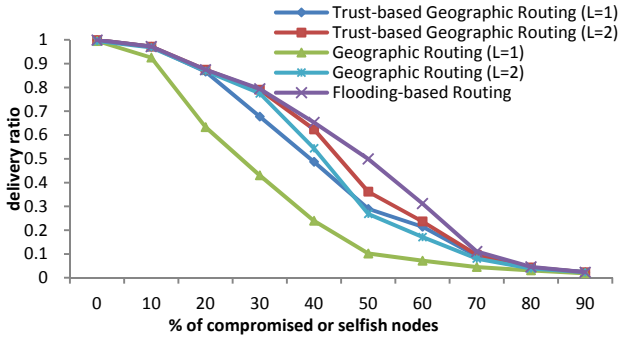

Figure 4: Message Delivery Ratio.

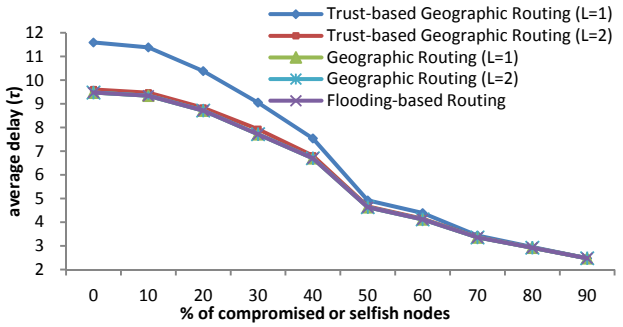

Figure 5: Message Delay.

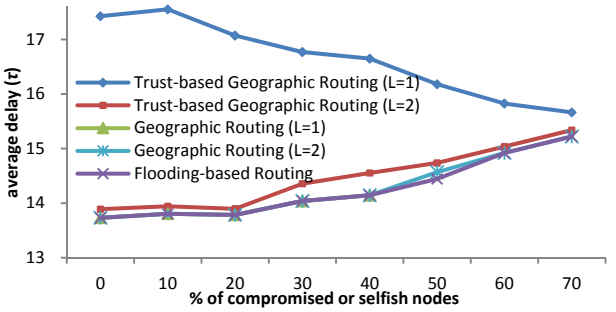

Figure 6: Message Delay with Source Sensor and Sink Node at a Distance Away.

Figure 6 shows the average delay for those messages that are successfully delivered for a special case in which the source SN and the sink node are at least a distance ($700m$) away. We create this case to ensure there are sufficient intermediate nodes on any path to reach the sink node. Compared with Figure 5, we observe (a) trust-based geographic routing with L=2 again approaches flooding-based routing, especially as the percentage of compromised or selfish nodes increases; (b) traditional geographic

routing with L=1 fails to deliver any message when the percentage of compromised or selfish nodes is higher than 50% because there is no short route to reach the destination node over a long distance, while trust-based geographical routing with L=1 can still deliver messages; (c) the message delivery delay increases as the percentage of compromised or selfish nodes increases due to more messages being dropped by selfish or malicious nodes resided on shorter routes.

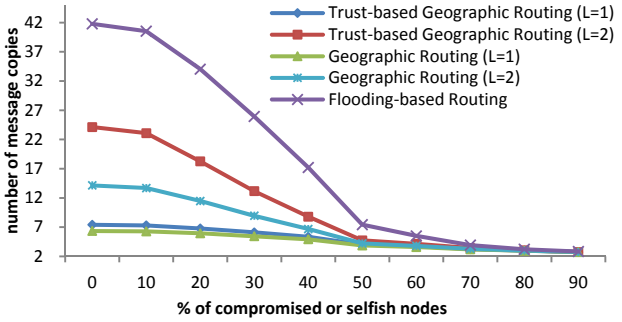

Figure 7: Message Overhead.

Figures 4-6 above suggest that trust-based geographical routing with L=2 can achieve ideal performance in message delivery ratio and message delay. Below we study the message overhead issues. Figure 7 shows the message overhead in terms of the number of message copies propagated before the destination sink node receives one copy. Both geographic routing and trust-based geographic routing perform significantly better than flooding-based routing. Trust-based geographical routing incurs more message overhead than traditional geographical routing because the path selected by trust-based geographical routing is often the most trustworthy path, not necessarily the shortest path. Nevertheless, we observe that the overhead increase of trust-based geographical routing over traditional geographical routing is small compared with that of flooding-based routing over traditional geographical routing. The system thus can effectively trade off message overhead for message delivery ratio and message delay. Finally, we observe that the number of message copies propagated for all three routing protocols is close to 3 when the percentage of compromised or selfish nodes is higher than 80%. The reason is that the message can be successfully delivered only when the source node and the sink node are close to each other. Otherwise, there is a high probability that compromised and selfish nodes reside on a long route will drop the message copies received.

Overall Figures 4-7 demonstrate that our trust-based geographic routing protocol with L=2 can significantly improve the delivery ratio and message delay (close to those of flooding-based routing) in the presence of compromised or selfish nodes, without sacrificing too much message overhead. Here we note that the system can effectively trade off message overhead (energy consumption) for high delivery ratio and low message delay by adjusting the level of redundancy (L). As L increases the performance of our trust-based geographical routing protocol in delivery ratio and message delay will approach that of flooding-based routing.

7. CONCLUSION

In this paper, we proposed a hierarchical trust management protocol for cluster-based wireless sensor networks, considering two aspects of trustworthiness, namely, social trust and QoS trust. We developed a probability model utilizing stochastic Petri nets techniques to quantitatively analyze the protocol performance, and validated subjective trust against objective trust obtained based on actual node status. We applied our hierarchical trust management protocol to trust-based geographic routing and demonstrated that our trust-based geographic routing performs close to the ideal performance of flooding-based routing in delivery ratio and message delay without sacrificing much in message overhead compared with traditional geographic routing protocols which does not use trust.

8. REFERENCES

[1] E. Aivaloglou and S. Gritzalis, "Hybrid Trust and Reputation Management for Sensor Networks," *Wireless Networks*, vol. 16, no. 5, Jul. 2010, pp. 1493-1510.

[2] I.R. Chen, Y. Wang and D.C. Wang, "Reliability of Wireless Sensors with Code Attestation for Intrusion Detection," *Information Processing Letters*, 2010.

[3] E.M. Daly and M. Haahr, "Social Network Analysis for Information Flow in Disconnected Delay-Tolerant MANETs," *IEEE Transactions on Mobile Computing*, vol. 8, no. 5, May 2009, pp. 606-621.

[4] S. Ganeriwal, L.K. Balzano, and M.B. Srivastava, "Reputation-Based Framework for High Integrity Sensor Networks," *ACM Transactions on Sensor Network*, vol. 4, no. 3, May 2008.

[5] B. Karp and H. T. Kung. "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *6th ACM Annual International Conference on Mobile Computing and Networking*, Boston, USA, 2000, pp. 243–254.

[6] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. "Geographic Routing Made Practical," *2nd USENIX/ACM Symposium on Networked System Design and Implementation,* May 2005, pp. 217–230.

[7] K. Liu, N. Abu-Ghazaleh, and K.-D. Kang, "Location Verification and Trust Management for Resilient Geographic Routing," *Journal of Parallel and Distributed Computing*, vol. 67, no. 2, 2007, pp. 215-228.

[8] L. Moraru, et al., "Near Optimal Geographic Routing with Obstacle Avoidance in Wireless Sensor Networks by Fast-Converging Trust-Based Algorithms," *3rd ACM Workshop on QoS and Security for Wireless and Mobile Networks*, Chania, Greece, Oct. 2007, pp. 31-38.

[9] R.A. Sahner, K.S. Trivedi and A. Puliafito, *Performance and Reliability Analysis of Computer Systems*, Kluwer Academic Publishers, 1996.

[10] R.A. Shaikh, et al., "Group-Based Trust Management Scheme for Clustered Wireless Sensor Networks," *IEEE Transacions on Parallel and Distributed Systems*, vol. 20, no. 11, Nov. 2009, pp. 1698-1712.

[11] T. Spyropoulos, K. Psounis, C.S. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: the Multiple-Copy Case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, Feb. 2008, pp. 77-90.

[12] Z. Yao, D. Kim, and Y. Doh, "PLUS: Parameterized and Localized Trust Management Scheme for Sensor Networks Security," *3rd IEEE Int'l Conf. Mobile Ad-Hoc and Sensor Systems,* Oct. 2006, pp. 437-446.

[13] O. Younis and S. Fahmy, "HEED: A Hybrid Energy Efficient, Distributed Clustering Approach for Ad Hoc Sensor Network", *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, Oct.-Dec. 2004, pp. 366-379.