Energy-Efficient Provenance Transmission in Large-Scale Wireless Sensor Networks

S. M. Iftekharul Alam
Department of Electrical and Computer Engineering
Purdue University
alams@purdue.edu

Sonia Fahmy
Department of Computer Science
Purdue University
fahmy@cs.purdue.edu

***Abstract*—Large-scale sensor-based decision support systems are being widely deployed. Assessing the trustworthiness of sensor data and the owners of this data is critical for quality assurance of decision making in these systems. Trust evaluation frameworks use data provenance along with the sensed data values to compute the trustworthiness of each data item. However, in a sizeable multi-hop sensor network, provenance information requires a large and variable number of bits in each packet, which, in turn, results in high energy dissipation with extended period of radio communication, making trust systems unusable. We propose an energy-efficient provenance transmission and construction scheme, which we refer to as Probabilistic Provenance Flow (PPF). To the best of our knowledge, ours is the first approach to make the Probabilistic Packet Marking (PPM) approach of IP traceback feasible for sensor networks. We propose two bit-efficient complementary provenance encoding and construction methods, and combine them to handle topological changes in the network. Our TOSSIM simulations demonstrate that PPF requires at least 33% fewer packets and consumes 30% less energy than PPM-based approaches to construct provenance, yet still provides high accuracy in trust score calculation.** [1]

***Keywords*-provenance; trust framework; probabilistic packet marking; energy-efficiency; sensor networks**

I. Introduction

With the recent advances in developing small and smart sensors, wireless sensor networks (WSNs) are being deployed on a larger scale to gather real-time data from the physical world [1], [2]. Investigating permafrost in the Swiss Alps [3], Berkeley's habitat monitoring in Great Duck Island [4], and studying volcanic activity in Ecuador [5] are example applications that exploit WSNs to audit changes in the environment or climate. Sensor-based decision support systems have been implemented for the management of critical infrastructure systems [6] and power grid networks [7]. Global sensor networks [8], sensor networks for large-scale urban environments [9], and physical infrastructure systems [10] indicate potential deployments of hundreds of sensor nodes.

In many sensor applications, the network operates in a multi-hop fashion where battery-powered sensor nodes collect application-specific information and relay through intermediate nodes to a base station. Information collected at the base station is processed and made available to decision makers for further analysis. As quality of decision making is critically dependent on the quality of transmitted information [9], trustworthiness of information and information publishing nodes is important. The identity of the publishers of sensor data is useful for both historical and real-time values [11]. In a multi-hop network, provenance provides knowledge about the publisher and processing path of data since its generation. While some provenance-based trust evaluation frameworks [12], [13] have been proposed, they do not investigate *energy dissipation* due to transmission of provenance throughout the network.

Provenance of an information item can be represented as a tree which is embedded as meta-data with the information item, and updated along the path used to forward the item to the base station. Hence, every intermediate node carries provenance of length proportional to the hop count between that node and the source of the associated information item. In a network with a large diameter (hop count), this increased meta-data length results in an extended period of radio communication and energy dissipation at every intermediate node. We consider a real deployment of a 46-hop network [14] in our simulations, and observe that aggregated energy dissipation of the network increases by 27% when a traditional trust framework is employed. Although large networks can be hierarchically organized [15], they still require a significant number of hops [16], with non-negligible energy usage for aggregated provenance. If we simply incorporate identities of all relay nodes as provenance, practical usability of trust frameworks becomes questionable.

Provenance encoding and transmission has a similar nature to the well-known *IP traceback* problem [17], [18]. IP traceback aims to determine the forwarding paths of spoofed packets in traditional wired networks. Among the many proposed solutions to this problem, Probabilistic Packet Marking (PPM) can be most easily adapted to WSNs [19]. However, PPM assumes trustworthy routers (intermediate nodes) and static routes. Moreover, as we will show via simulations, PPM requires a large number of packets to construct the forwarding path, which makes direct application of PPM to WSNs infeasible.

In this paper, we devise an energy-efficient provenance transmission and construction scheme for large and slowly-varying WSNs. Like PPM, the intuition behind our approach

[1] This work has been sponsored in part by NSF grant CNS-0964294.

978-1-4577-0351-5/11/$26.00 ©2011 IEEE

is to reduce the expected length of provenance information through probabilistic incorporation of node identity, instead of embedding the identity of every node along the information forwarding path. But unlike PPM, our method incorporates a connected subgraph of the forwarding path into a packet and is able to trace the evolution of provenance as topology changes. This reduces the number of packets (and hence convergence time) required to construct provenance. Our simulation results show that the proposed methods consume approximately 30% less energy than the traditional approach, which significantly increases the network lifetime.

The remainder of this paper is organized as follows. We formulate the problem of energy-efficient provenance transmission and define our network and trust models in section II. Section III discusses related work. Section IV explains our approaches to embed and construct provenance. In section V, we discuss practical implementation issues. Section VI presents simulation results. Finally, section VII includes a few concluding remarks.

II. Problem Formulation

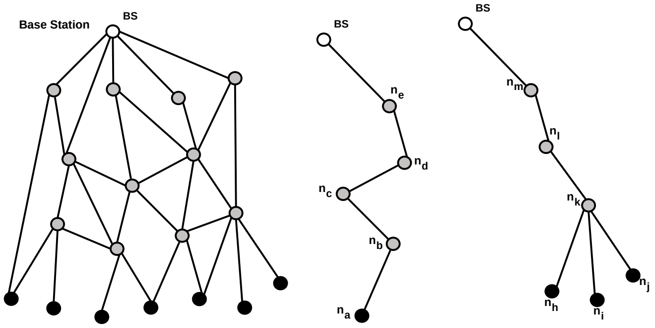
A. Network Model

We consider a multi-hop wireless sensor network where changes in topology due to failure or mobility can occur, but are not frequent. We make the following assumptions regarding the network and traffic:

- A Base Station (BS) acts as a central command authority and the root of a routing tree. It has no resource constraints and cannot be compromised by an attacker.
- The network may or may not be clustered. A clustered network can be constructed by protocols like [20], [21]. A typical cluster consists of a single cluster head and a variable number of cluster members. Some data aggregation functions (e.g., min, max, average) are implemented at the cluster head, which aggregates data from member nodes and forwards the resulting data towards the base station.
- Sensor nodes monitor their surroundings and periodically report to the base station or their designated cluster head (if any).
- Multiple sensors are used for monitoring an event. Thus, within a particular time window, independent observations obtained at cluster heads or the base station from different sensors are concerned with the same event.
- The underlying MAC protocol can be a variant of B-MAC [22] or X-MAC [23], which are compatible with the TinyOS stack. Sleep-wake scheduling is performed in low power listening mode [24].
- A provenance based trust management method such as [12], [13] is used in the application layer to establish and manage trust in an adaptive manner. Provenance information is embedded into sensor data packets as meta-data.

B. Provenance and Trust Model

In a provenance-based trust framework [12], a *trust score* is associated with each data item, and a *reputation value* is attributed to the provider of information. Trust scores and reputation values gradually evolve in an adaptive manner. Specifically, upon reception of an item, the receiver estimates the trustworthiness of the item based on the *value similarity* and *provenance similarity* of information received via multiple paths. The receiver then adjusts the reputation of the information owner based on the newly calculated trust score of the item. This process of trust calculation is typically performed at the base station. However, in a clustered network, nodes that are responsible for aggregation can also compute reputation values of their descendant nodes, and assign a new trust score to the aggregated item based on these reputation values. Every node that manipulates or forwards an information item can update provenance information by embedding its own identity with that item. At the base station, the complete provenance of the item is received in the form of a directed acyclic graph (DAG) of manipulator or forwarder (relay) nodes.



(a) Network (b) Simple provenance (c) Aggregated provenance

Figure 1. An example sensor network and sample provenance graphs.

Consider figure 1(a) where a number of sensor nodes periodically send packets towards the base station through multiple paths. As shown in figures 1(b) and 1(c), provenance forms subgraphs of the network graph – more specifically, trees rooted at the base station.

C. Challenges

Our focus is on how provenance is transmitted and constructed without modifying the basic trust framework. Ideally, each packet should contain identities of all nodes that forward or manipulate that packet, so that the packet contains the entire provenance when it arrives at the base station. However, this requires a large and variable number of bits in the meta-data of the packet, and thus consumes significant energy in the long run. To mitigate this problem, we consider probabilistic incorporation of provenance information – every node embeds its identity into the packet with some probability, and after collecting sufficient packets at the base station, the entire provenance tree or path can be constructed. The higher the percentage of nodes along the forwarding tree that embed their identities into a single packet, the less time it takes to construct the full provenance.

The invariably energy-constrained nature of sensor networks and topological changes caused by failure or mobility of nodes impose the following three challenges to this probabilistic provenance approach: (1) The number of bits required per packet to transmit provenance information should be fixed and small; (2) The number of packets required to construct full provenance should be small. Fast convergence of provenance construction is critical; and (3) Any topological changes should be rapidly reflected in provenance, so that trust score calculation can be performed at the base station with up-to-date provenance information.

D. *Problem Statement*

We consider a network of N nodes, where the maximum length (depth) of any forwarding path (tree) is L. Assume that the maximum number of bits that can be used to embed provenance information in a single packet is B. Based upon this bit budget, there is an integer $m, 1 < m \leq L$ such that at most m consecutive node identities (that is, $m-1$ consecutive edges) can be embedded into a single packet. We must perform the following three operations:

(1) **Provenance Embedding**: In a forwarding tree $G = (V, E)$ rooted at the base station, each node $n_i \in V$ makes an independent decision whether to embed its identity into the packet, starting a connected sub-graph, with probability p_i. We need to design a provenance embedding method to carry a partial path $P =< n_{i_1}, n_{i_2}, \cdots n_{i_m} >$ into a single packet where $n_{i_j} \in V, 1 \leq j \leq m$ and $(n_{i_k}, n_{i_{k+1}}) \in E, 1 \leq k \leq m-1$. This problem is a simple extension of the edge sampling approach in IP traceback [17].

(2) **Provenance Construction**: On the base station side, we must construct the entire provenance tree $G = (V, E)$ by exploiting partial path information collected from a number of received packets, with an upper bound on the number of packets required to construct the provenance.

(3) **Evolution of Provenance**: After topological changes, e.g., due to failures or mobility, we must bound the time that it takes to reflect the changes in the constructed provenance.

III. Related Work

A few provenance-based trust frameworks have been proposed to date [12], [13]. These frameworks do not focus on energy-efficiency in wireless sensor networks. We can relate the problem of provenance transmission to the IP traceback problem that determines the forwarding path of spoofed packets [25]. IP traceback methods include hop-by-hop tracing [26], [27], out-of-band ICMP traceback [28], and in-band probabilistic packet marking [17], [18]. Hop-by-hop tracing is not well-suited to wireless sensor networks due to its large storage requirement. Hot-spot based traceback methods designed for mobile ad-hoc networks [29], [30] store packet information at the nodes, and traceback is performed hop-by-hop to determine the hot-spot where the attacker is located. In our case, provenance information is continuously required at the base station to compute trust scores of descendant nodes. Hot-spot based methods would incur unnecessary delay in trust score calculation. Out-of-band ICMP traceback requires out-of-band communication and increased bandwidth which limit its usability in resource-constrained wireless sensor networks.

In this work, we adapt Probabilistic Packet Marking (PPM) since it does not require additional storage or out-of-band communication. PPM assumes trustworthy routers and static routes which may not hold in our case. Additionally, PPM requires a significant number of packets to construct the forwarding path. Network coding variants of PPM [31], [32] require fewer packets to construct the forwarding path. Network coding approaches, however, have a high computational complexity and increase the length of the packet as marking coefficients are transmitted with the packet. Cheng et al. [33] determine the optimal marking probability for each node to reduce the number of packets required to construct the forwarding path.

IV. Probabilistic Provenance Flow

In this section, we discuss our probabilistic provenance transmission and construction method named Probabilistic Provenance Flow (PPF). We first discuss the assignment of a unique number to every node as a node identifier (ID) before deployment. Then, we propose two complementary provenance embedding methods that differ in how they encode node identifiers. We present provenance construction mechanisms for both encoding methods, and show how the two methods can be combined to handle topological changes.

A. *Node ID Assignment*

For a network of N nodes, we pick a set $Q_P = \{q_1, q_2, \cdots q_z\}$ with the smallest z such that $z \geq N$. An in-place randomized algorithm is used to produce a random permutation of Q_P, $\sigma(Q_P) = \{q_{a_1}, q_{a_2}, \cdots q_{a_z}\}$ and members of $\sigma(Q_P)$ are assigned to all N nodes sequentially. For example, in an 8-node network, we can pick IDs for the nodes from a random permutation of $Q_{11} = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$.

Definition 1. *Let P_n be the largest prime number that is less than or equal to the positive integer n. We define the set of usable IDs, Q_P where P is a prime number:*

$$Q_P = \{n \in \mathbb{N} \mid 2 \leq n \leq P \ and \ 0 \leq n - P_n \leq 7\}$$

Definition 2. *The rank of any node $n \in Q_P$, denoted as $rank(n)$, is the position of n in the same random permutation of Q_P, $\sigma(Q_P)$ that was used to generate IDs. Particularly, in an N-node network, $1 \leq rank(n) \leq N \leq |Q_P|$.*

Definition 3. *For any positive integer $n \in Q_P$, for some P, we define two functions:*

- $prime(n)$ = *The largest prime number that is less than or equal to n = P_n.*
- offset(n) = *The difference between n and P_n = $n - P_n$.*

B. Embedding Provenance with Juxtaposition of Ranks

In the *rank method*, instead of embedding the node ID directly into a packet, $rank(\mathrm{ID})$ (defined in Definition 2) of the node is embedded, since every node ID is uniquely identifiable using its rank. Here, we use the terms *rank* and *identity* interchangeably. Assume that the packet meta-data has space to hold identities of up to m nodes. We use a counter of $\log_2 m$ bits to track the number of already embedded identities in the packet. Initially, the buffer and counter contain zeros. Every node n_i decides to start a connected sub-graph with its identity probability p_i. Once it decides to do so, it overwrites the previous information by doing the following: it zeros out the entire provenance field and then incorporates its identity at the beginning of the buffer and sets the counter to one. If a node decides not to overwrite, it checks for empty buffer space using the counter field. If there is space, it adds its identity into the first available slot in the buffer and increments the counter. Figure 2 shows an example of this method where the buffer space can hold at most four node identities in a single packet.

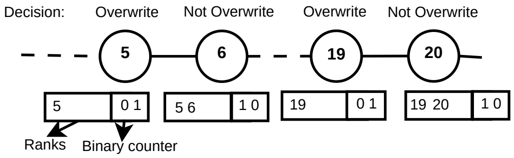


Figure 2. Probabilistic incorporation of provenance using juxtaposition of ranks (numbers inscribed in the circles indicate *rank* of nodes).

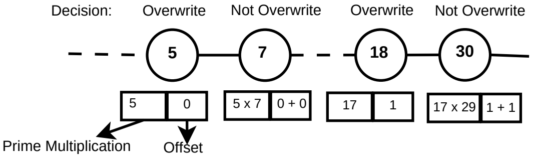


Figure 3. Probabilistic incorporation of provenance using prime multiplication (numbers inscribed in the circles indicate ID of nodes).

C. Embedding Provenance with Prime Multiplication

The rank method is easy to decode, but requires a relatively large number of bits for provenance data. For example, in a 1000-node network, we need a total of 42 bits to transmit provenance information of four nodes in a single packet. We can reduce these bit requirements by using the following *prime method* which still has a simple decoding process.

To store provenance information, we divide the provenance buffer into two parts: *product* and *offset*. Every node n_i has an ID, say ID_i, that is a member of Q_P for some P. As before, once a node n_i decides to start a connected subgraph, it overwrites previously stored information by clearing the entire provenance buffer. It then inserts $prime(ID_i)$ into the product part and *offset*(ID_i) into the offset part of the buffer (as defined in Definition 3). If a node n_j decides not to overwrite, it retrieves the value stored in the product and offset parts. Then, it multiplies the value of the product part with $prime(ID_j)$, adds *offset*(ID_j) to the offset part, and stores the newly calculated values into the respective parts. We will later show that, for a given bit budget, we can determine an upper bound and a lower bound for the number of identities of nodes (m) that can be stored in a single packet using this approach. Figure 3 shows an example with $m = 2$.

We no longer need a counter field to track the number of node identities encoded in the provenance buffer because there is always a unique prime factorization of the *product* part which gives the number of participating nodes.

D. Decoding Partial Provenance

When a packet is received at the base station, the provenance buffer is examined to retrieve the embedded partial provenance (or path) information. With the rank embedding approach, we can easily extract the embedded identities from the provenance buffer using the length field as each node ID uses a fixed number of bits. With the prime embedding method, we assume that information about ordering among nodes is known beforehand (as discussed in the next section). We apply a standard prime factorization algorithm over the product part of the provenance buffer to retrieve the nearest prime numbers and map to node identities.

Retrieving information from the offset part is a version of the subset sum problem [34]. If the number of identities of participating nodes is m, we use a dynamic programming approach to determine possible non-empty subsets of $\{0, 1, 2, 3, 4, 5, 6, 7\}$ with cardinality m that sum to the offset value. For every possible subset, we form partial paths of length $m - 1$ by combining the retrieved nearest prime numbers (say, $X_1, X_2, \cdots X_m$) and the members of the considered subset (say, $o_1, o_2, \cdots o_m$). From prior knowledge, we also identify partial paths of length $m - 1$ such that the nearest prime numbers of node IDs on the paths are $(X_1, X_2, \cdots X_m)$. Then, we consider all possible pairs of newly formed partial paths and the partial paths considered from the past, and calculate a difference score between them using the following formula:

$$\delta = \sum_{i=1}^{m} (o_i - (ID_{a_i} - X_i))^2$$

where $< ID_{a_1}, ID_{a_2}, \cdots ID_{a_m} >$ indicates the partial path based on previous knowledge. Finally, we determine δ_{min}, the lowest difference score over all the possible combinations, and record the corresponding partial path information. When δ_{min} becomes zero, the recorded information is used. Otherwise, topological changes may have occurred in the network and the previously stored provenance information may not be up-to-date. Further processing is necessary to determine the provenance, such as checking other combinations of partial paths by considering nodes that are 1 or 2-hop away from the nodes on the recorded path, or triggering the rank ID embedding approach to recover the order. These extensions will be the subject of our future work.

E. Construction and Evolution of Provenance

With our identity embedding methods, provenance construction is straightforward once we have decoded partial path information from the received packet. After collecting sufficient packets with embedded provenance (i.e., when we have at least one ID from each node), we combine the partial paths to produce the complete provenance graph. However, decoding using the prime approach needs previous knowledge about the order of nodes. This can be obtained by applying the rank method first. After a configurable period of time (generally greater than provenance convergence time) during which the provenance is constructed using the rank method, the prime embedding method can be employed.

In order to keep node order information up-to-date, nodes utilize the rank approach every $t_{embedding}$ seconds. Thus, any topological changes are reflected in the provenance. Based on the frequency of mobility or failures in the network, $t_{embedding}$ can be adjusted. However, a small value of $t_{embedding}$ will reduce the benefits of applying the bit-efficient prime embedding method. We are currently considering a reactive approach to trigger the rank approach only when necessary.

V. Practical Considerations

A. Bit Requirements

Consider a 1000-node network where the maximum number of bits allocated for provenance per packet (B) is 32. Using the rank approach for embedding IDs, at most m=3 ranks of nodes can be encoded into 30 bits and the remaining 2 bits can be used to track the number of encoded nodes. In case of the prime embedding approach, 5 bits can store the sum of the offset values of at most $m = 4$ nodes since offset values vary between 0 and 7. The remaining 27 bits can hold prime multiplication of node IDs. In a static network, careful node ID assignment ensures encoding at least 4 node IDs in a single packet. Our simulations show that randomly assigning node IDs allows encoding 3 or more node IDs in most cases.

B. Convergence Time and Topological Change

Let the average number of IDs received from a packet be m_{avg} (regardless of the embedding method). Assume the probability p of embedding a sub-graph into the provenance buffer is fixed ($\forall_i p_i = p$). With traditional PPM [32], the time for convergence is constrained by the time until the ID of the farthest node is received at the base station which is $\frac{1}{p(1-p)^{L-1}}$. In our method, we are receiving m_{avg} unique IDs instead of one. Thus, the expected number of packets required to construct provenance,

$$E(S) < \frac{1}{m_{avg}}.\frac{L}{p(1-p)^{L-1}}$$

A value of p that is less than or equal to $\frac{1}{L}$ produces a near-optimal result (i.e., reduces the number of required packets).

Convergence time $t_{convergence}$ is nothing but the time required to receive $E(S)$ packets at the base station. If the rank approach of embedding IDs is used every $t_{embedding}$ seconds, then any topological change will be reflected within $t_{evolution} \leq t_{embedding} + t_{convergence}$ seconds.

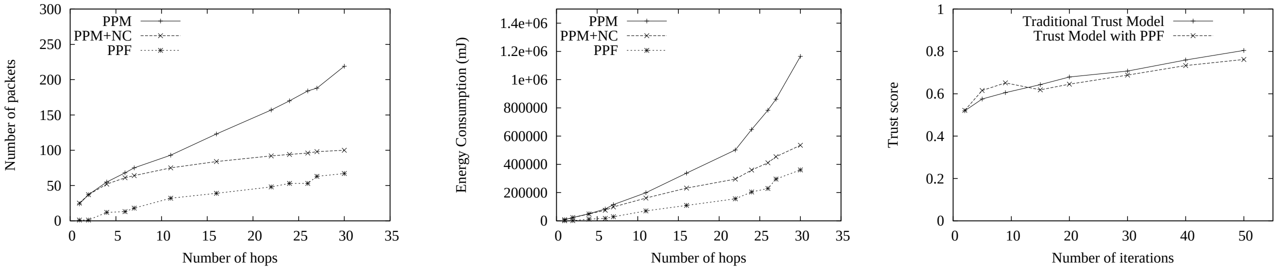
VI. Performance Analysis

We compare our PPF method with two variants of probabilistic packet marking (PPM [17], [18] and PPM with Network Coding [31], [32]) as they are the closest to our approach (though they were designed for wired IP networks). We conduct simulations using TOSSIM [35] for networks with hop counts ranging from 2 to 30, and number of nodes ranging from 3 to 50. For energy analysis, we use POWERTOSSIMZ [36] which uses the *micaz* energy model. All experiments are performed using the transmission rate of 250 kbps, the default transmission rate of the *micaz* mote, where every data-generating sensor sends data towards the base station every 2000 ms (2 s). All results are averaged over 1000 runs.

To make the comparison fair, we place the same constraint on usable bits (32 bits) for provenance embedding in a packet for all three approaches. Though the size of the network can grow arbitrarily, we assume that 1 byte is enough to represent the maximum hop count. Hence, PPM with Network Coding (PPM+NC) requires 14 bits to accommodate three coefficients and the distance field. The remaining 18 bits can be used to store the linear combinations of node IDs. However, embedding 4 node IDs into a packet in PPM+NC requires more than 32 bits. Thus, the maximum number of node IDs carried in a single packet is 1, 3, and 4 for PPM, PPM+NC, and PPF respectively. The probability for embedding a node ID is $p = \frac{1}{25}$.

Figures 4(a) and 4(b) show the number of packets and energy consumption required to construct provenance using the three schemes for increasing numbers of hops. With 1000 runs, the 95% confidence intervals of these experiments have deviations in the range of 0 to 5 from the experimental averages, which statistically assures the correctness of our experiments. The experimental results reveal that PPF requires at least 33% fewer packets and consumes 30% less energy than both PPM-based schemes.

We also integrate PPF with a provenance-based trust model to iteratively compute trust scores. Figure 4(c) shows that the trust score calculated using PPF evolves correctly as soon as the entire provenance is constructed at the base station. PPF accuracy in trust score calculation is similar to



(a) Number of packets

(b) Aggregate energy consumption (in mJ)

(c) Change of trust scores in a 10-hop network

Figure 4. Provenance construction for different hop counts and change of trust scores for different iterations.

the traditional approach that includes every node ID on the forwarding path in the provenance.

VII. Conclusions

We have presented an energy-efficient provenance transmission and construction approach for large-scale multi-hop wireless sensor networks, based on the idea of probabilistic incorporation of node identities. We adapt the probabilistic packet marking (PPM) approach for IP traceback, and propose two complementary provenance encoding methods with a space constraint on the size of provenance data in the packet. Further, we present efficient provenance construction schemes for the two encoding methods, and combine them to deal with topological changes in the network. In contrast to PPM, our proposed approach requires fewer packets to construct network-wide provenance, and significantly reduces the aggregate energy consumption of the network. Most importantly, integration of our scheme with a provenance-based trust model on the TinyOS emulator TOSSIM reveals no degradation in accuracy of trust score calculation. As future work, we plan to design a reactive approach that will accurately reflect topological changes. We will also study how well a complete trust framework can detect and react to various attack and failure scenarios.

References

[1] D. Butler, "2020 computing: Everything, everywhere," *Nature*, vol. 440, pp. 402–405, 2006.

[2] M. Zuniga and B. Krishnamachari, "Integrating future large-scale wireless sensor networks with the Internet," USC Computer Science, Tech. Rep., 2003, cS 03-792.

[3] I. Talzi, A. Hasler, S. Gruber, and C. Tschudin, "Permasense: Investigating permafrost with a WSN in the Swiss Alps," in *Proc. of SenSys*, 2007.

[4] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proc. of SenSys*, 2004, pp. 214–226.

[5] G. Werner-allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," in *Proc. of the Second European Workshop on Wireless Sensor Networks (EWSN)*, 2005.

[6] N. Xu and S. Rangwala, et al., "A wireless sensor network for structural monitoring," in *Proc. of ACM Sensys*, 2004, pp. 13–24.

[7] Y. Yang, D. Divan, R. G. Harley, and T. G. Habetler, "Power line sensornet - A new concept for power grid monitoring," in *IEEE PES Gen. Meet.*, 2006.

[8] K. Fehrenbacher, "A global sensor network launches to fight climate change," January 2011, http://www.reuters.com/article/idUS359029730820110112.

[9] A. Doboli and et al., "Cities of the future: Employing wireless sensor networks for efficient decision making in complex environments," State University of New York, Tech. Rep., April 2008, cEAS Technical Report Nr 831.

[10] "Sensor Andrew at Pennsylvania Smart Infrastructure Incubator," http://www.ices.cmu.edu/psii/sensor-andrew.html.

[11] J. Ledlie, C. Ng, D. A. Holland, K. kumar Muniswamy-reddy, U. Braun, and M. Seltzer, "Provenance-aware sensor data storage," in *Proc. of Workshop on Networking Meets Databases (NetDB)*, 2005.

[12] X Wang et al., "Provenance based information trustworthiness evaluation in multi-hop networks," in *Proc. of IEEE GLOBECOM*, 2010.

[13] H.-S. Lim, Y.-S. Moon, and E. Bertino, "Provenance-based trustworthiness assessment in sensor networks," in *Proc. of the Seventh International Workshop on Data Management for Sensor Networks*, 2010, pp. 2–7.

[14] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proc. of Information Processing in Sensor Networks*, 2007, pp. 254–263.

[15] A. Arora et al., "ExScal: Elements of an extreme scale wireless sensor network," in *Proc. of 11th IEEE International Conference on Real-Time Computing Systems and Applications*, 2005.

[16] K. Iwanicki and M. van Steen, "On hierarchical routing in wireless sensor networks," in *Proc. of the 2009 International Conference on Information Processing in Sensor Networks*, 2009, pp. 133–144.

[17] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for IP traceback," in *Proc. of SIGCOMM*, 2000, pp. 295–306.

[18] D. X. Song and A. Perrig, "Advanced and authenticated marking schemes for IP traceback," in *Proc. of IEEE INFOCOM*, vol. 2, 2001, pp. 878 –886 vol.2.

[19] V. Thing and H. Lee, "IP traceback for wireless ad-hoc networks," in *Proc. of the IEEE 60th Vehicular Technology Conference*, vol. 5, Sept 2004, pp. 3286 – 3290.

[20] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach," in *Proc. of IEEE INFOCOM*, 2004.

[21] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. on Wireless Communications*, vol. 1, no. 4, pp. 660–670, October 2002.

[22] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. of SenSys*, 2004, pp. 95–107.

[23] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. of SenSys*, 2006, pp. 307–320.

[24] "Low power listening," http://www.tinyos.net/tinyos-2.x/doc/html/tep105.html.

[25] A. Belenky and N. Ansari, "On IP traceback," *Communications Magazine, IEEE*, vol. 41, no. 7, pp. 142 – 153, July 2003.

[26] H. Burch, "Tracing anonymous packets to their approximate source," in *Proc. of the 14th USENIX conference on system administration*, 2000, pp. 319–328.

[27] A. C. Snoeren, "Hash-based IP traceback," in *Proc. of SIGCOMM*, 2001, pp. 3–14.

[28] "ICMP traceback messages," http://tools.ietf.org/html/draft-ietf-itrace-04.

[29] Y. an Huang and W. Lee, "Hotspot-based traceback for mobile ad hoc networks," in *Proc. of 4th ACM Workshop on Wireless Security*, 2005, pp. 43–54.

[30] H. Hsu, S. Zhu, and A. Hurson, "A hotspot-based protocol for attack traceback in mobile ad hoc networks," in *Proc. of the 5th ACM Symposium on Information, Computer and Communications Security*, 2010, pp. 333–336.

[31] P. Sattari, M. Gjoka, and A. Markopoulou, "A network coding approach to IP traceback," in *Proc. of IEEE International Symposium on Network Coding*, June 2010, pp. 1–6.

[32] D. Dean, M. Franklin, and A. Stubblefield, "An algebraic approach to IP traceback," *ACM Trans. Inf. Syst. Secur.*, vol. 5, pp. 119–137, May 2002.

[33] B.-C. Cheng, H. Chen, Y.-J. Li, and R.-Y. Tseng, "A packet marking with fair probability distribution function for minimizing the convergence time in wireless sensor networks," *Comput. Commun.*, vol. 31, pp. 4352–4359, December 2008.

[34] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 2001.

[35] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," in *Proc. of SenSys*, 2003, pp. 126–137.

[36] E. Perla, A. O. Catháin, R. S. Carbajo, M. Huggard, and C. Mc Goldrick, "PowerTOSSIM z: realistic energy modelling for wireless sensor network environments," in *Proc. of the 3rd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, 2008, pp. 35–42.