PROTRU: LEVERAGING PROVENANCE TO ENHANCE NETWORK TRUST IN A WIRELESS SENSOR NETWORK

by

GULUSTAN DOGAN

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy,
The City University of New York

2013

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

Theodore Brown

Date — Chair of Examining Committee

Theodore Brown

Date — Executive Officer

Ted Brown

Nancy Griffeth

Abdullah Uz Tansel

Jin-Hee Cho

Supervision Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

PROTRU: Leveraging Provenance to Enhance Network Trust in a Wireless Sensor Network

by

Gulustan Dogan

Advisor: Professor Ted Brown

Trust can be an important component of wireless sensor networks for believability of the produced data and historical value is a crucial asset in deciding trust of the data. A node's trust can change over time after its initial deployment due to various reasons such as energy loss, environmental conditions or exhausting sources. Provenance can play a significant role for supporting the calculation of information trust by recording the data flow and snapshots of the network. Furthermore provenance can be used for registering previous trust records and other information such as node type, data type, node location, average of the historical data. We will introduce a node-level trust-enhancing architecture for sensor networks using provenance. Our network will be cognitive in the sense that our system will react automatically upon detecting anomalies. Through simulations we will verify our model and will show that our approach can provide substantial enhancements in information trust as compared to the traditional network approaches.

Acknowledgements

One of the joys of completion is to look over the journey past and remember all the professors, friends and family who have helped and supported me along this long but fulfilling road.

I would like to express my heartfelt gratitude to Professor Ted Brown for being an exceptional mentor. I could not have asked for a better role model, inspirational, supportive, and patient. I would also like to thank my committee, Professor Nancy Griffeth, Professor Abdullah Uz Tansel and Dr. Jin-Hee Cho who provided encouraging and constructive feedback. It is no easy task, reviewing a thesis, and I am grateful for their thoughtful and detailed comments. To the many anonymous reviewers at the various conferences, thank you for helping to shape and guide the direction of the work with your careful and instructive comments.

It has been an exceptional experience to be a member of the graduate program under the direction of Prof. Theodore Brown. My personal academic achievements would not have been possible without the supportive staff at the Graduate Center who administered the grants that funded my graduate studies and travels. I would like to thank Ms. Lina Garcia in Computer Science Department for being very helpful and responsive. I have benefited greatly from the teaching fellowship that funded the majority of my dissertation work.

As such, I am grateful to Prof. Yedidyah Langsam of the Brooklyn College CIS department for providing me with interesting courses that complemented my studies and for the favorable teaching schedules that made this all possible. Over the years, I have been fortunate to learn from and work with professors whose creativity has broadened my thinking and whose guidance has been invaluable.

I would also like to give a special thank you to my brother Eren for being supportive and encouraging throughout the dissertation process. I would not have contemplated this road if not for my parents, Ismail and Asiye, who instilled within me a love of creative pursuits, science and language, all of which finds a place in this thesis. To my parents, thank you.

To my parents Ismail and Asiye who instilled within me a love of knowledge.

Table of Contents

List of Figures

Chapter 1

Introduction

Provenance plays an important role in wireless sensor networks. Wireless sensor network technology is a fast growing concept. Tiny and cheap nodes are employed in large numbers in difficult environments such as forests, lakes, etc. to do data collection for many purposes [6]. Originally wireless sensor networks were motivated by surveillance in battlefields for military however in time they were used in many areas [90]. Some examples of these areas are active volcano monitoring[138, 137], the microclimate monitoring throughout the height of a redwood tree [32], buildings and bridge monitoring [49, 100] and health-care monitoring [48] to name a few.

In most of the sensor networks, there is no centralized control over data collection, as in the traditional databases, hence data is often copied, moved, created, updated and deleted in an uncontrollable way. Provenance makes it possible to have a clear picture of the dataflow in a sensor network by tracking the evolution of the data systematically. Provenance plays an important role in deciding about the qualities of the data such as its trustworthiness, its accuracy and its verifiability. Provenance can be used in order to find out causes of

faulty behavior, to figure out the circumstances that determine the connectivity of the network, to increase the trustworthiness of data after elimination of the causes. Therefore, provenance management should be an interest in sensor networks in order to have an understanding of how the results are obtained for later use such as trust assessment, fault tolerance, troubleshooting, result reproduction and performance optimization. In our work we use provenance for trust assessment in wireless sensor networks (WSNs).

Wireless sensor networks are very vulnerable environments due to computational and energy constraints. In addition wireless sensor networks are very open to physical world effects such as a person walking on a field can step on a sensor and make it dysfunctional. Trust is quite important for self-configurable and autonomous systems such as wireless sensor networks [41]. A wireless sensor network has to depend on accurate data. Therefore keeping the trust of a data item as up-to-date as possible is a clear concern and trust assessment is a crucial task. Network trust may depend on several factors such as the path traveled by the data, the trust of the source, time elapsed after the transmission. As enhancing trust involves understanding causal chains of events, dataflow oriented provenance model becomes a solid reference of the phases data goes through [25]. The dataflow oriented provenance model which we use in our architecture, makes it possible to have a clear picture of the dataflow by keeping the source node and destination node information and their states.

A trust management scheme can make a WSN tolerant to node failures by assisting decision making process. For example, a node can decide to cooperate with a node or not based on the feedback it receives from the trust model. Trust research on WSN is new; few systems have considered it [47, 143]. More research has been done on Trust in Ad-hoc and P2P networks. Although these network types have many similarities to WSN, still a

separate trust management system has to be developed for WSN because of their specific characteristics such as energy and computation constraints. For example, some of the trust models for ad-hoc networks use a central reputation mechanism that needs a ”manager” overseeing the trust of the network [109]. This approach is not very applicable to sensor networks because of energy and scalability issues. Our major contribution of this work is leveraging provenance in order to restructure the network for maintaining the trust, an idea that we believe is new. Not many research has been done combining trust, provenance and wireless sensor networks and to our knowledge, our architecture is the only one which combines these three areas for network restructuring. Using provenance in trust assessment for wireless sensor networks is a new research area with many open questions which we feel privileged for exploring.

This thesis is organized as follows. Chapter 2 establishes terminology on provenance that will be used liberally throughout this thesis. Chapter 3 will review background on Trust and Chapter 4 will review background on Sensors. In Chapter 5, we will present Related Work. Chapters 6 presents the contributions of this thesis work. Chapter 7 presents the experimental results. We conclude in Chapter 8 and present future research in Chapter 9.

Chapter 2

Background on Trust

Trust is quite important for self-configurable and autonomous systems such as WSNs [41]. WSNs are very vulnerable environments due to computational limitations, energy constraints and network attacks. In addition, WSNs are very open to physical world effects such as a person walking on a field can step on a sensor and make it dysfunctional. A trust management scheme can make a WSN tolerant to node failures and misbehavior by assisting decision making process. For example, a node can decide to cooperate with a node or not based on the feedback it will receive from the trust model. Trust research on WSN is very new, few systems have considered it [47, 143]. More research has been done on Trust in Ad-hoc and P2P networks. Although these network types have many similarities to WSN, still a separate trust management system has to be developed for WSN because of their specific characteristics such as energy and computation constraints.

Data collection is very important in the process of designing a trust management system. The system should be history aware, past behaviors should be taken into consideration [41]. Our system is history aware as the trust calculation is considering past behaviors. Moreover, every node keeps their past behavior statistics regarding the data they produce such as

average error of the created data in the past time intervals. One of the biggest constraints is the overhead that can be caused by the trust model. Trust model should be as lightweight as possible [41].

There are many different data that can be used as input of the trust model. For example, a node that is not alive for a long time or a node that appears or disappears randomly may not be trusted. On the communication layer, a node which is misreporting will not be trusted. For instance, a node which is giving a fire alarm when conditions are calm should be given a low trust value [41].

2.1 Definitions: Trust, Trustworthiness, Risk, Reputation

Josang et al. [63] define trust and trustworthiness based on the definitions of Gambetta [46]. Solhaug et al. [118] define trustworthiness as objective probability that the trustee performs a particular action on which interests of the trustor depend. Trust is a subjective probability varying from 1 (complete trust) to 0 (complete distrust) [63].

As trust is the believed probability and trustworthiness is the actual probability, there can be a difference between them. This difference introduces the risk factor [26]. Risk increases if the trust is misplaced.

Reputation is also a concept that is very related to trust. Sometimes reputation and trust is used in the same context however they have different meanings. Reputation is the opinion of an entity about the other entity. However trust is derivation of reputation of an entity.

2.1.1 Information Trust

There are different types of trust such as social trust, cognitive trust, and communication trust. In this work, we are assessing the information trust of data items and sensor nodes. Information trust or data trust refers to the trust placed on data produced by objects or processes. Information trust in a network is important because it can prevent erroneous data to accumulate in the network. In a network, a node can (i) create data (ii) process the data such as fusion (iii) pass the data along. The trust of data depends on the trust of the node that creates the data and the trust of the nodes the data has visited. Information trust in a network can be categorized into three: (i) creator node's subjective view of the trust (ii) objective trust assessment of the data by the neighboring nodes (iii) changes in information trust as the data travels along the network.

2.1.2 Properties of Trust

Cho et al. [26] list characteristics of trust as follows in their survey paper.

- dynamicity : Due to node failures and mobility, sensor network is highly dynamic hence trust should be dynamic too.
- subjectivity : Nodes might decide to put different levels of trust on same nodes due to dynamicity of the network [1].
- transitivity : Trust is not necessarily transitive. For transitivity, we need two types of trust, trust in a trustee and trust in recommendations of the trustee.
- asymmetry : Trust is not necessarily transitive. A node may trust a node however the trustee node may not trust the trustor.

- context-dependency : Trust is context-dependent [9]. For instance, a node may trust the image data coming from a node but may not trust the audio data coming from the same node.

2.2 Trust and Reputation in Different Domains

Below I give information about trust literature in social science, e-commerce, distributed systems and ad-hoc networks based on the survey of Momani and Challa [90].

2.2.1 Trust in Social Science and E-Commerce

Trust is very related to social sciences because it is a part of human life [114]. It has a very big impact on human relationships such as making friends, sharing secrets, selling/buying things, working together. It eases the everyday life by helping in the decision making process, delegation, certification, resource access [111].

One of the motivating domains for trust research is e-commerce. In internet, buyers and sellers have a trust relationship. Buyers will buy from sellers that they trust. The trust will be formed based on the reputation of the seller. Seller gains a reputation based on past behavior. E-commerce systems such as eBay [110], Yahoo [110], Keynote [10, 11] keep a centralized trust authority to maintain the reputation and trust values.

Abdul-Rahman and Hailes designed a trust model based on sociological characteristics of trust [3]. In their model, entities are given trust based on their reputation (indirect) and their direct experiences. They also consider the word-of-mouth mechanism. Agents put different importance (weights) to opinions of different agents.

Josang and Ismail developed a reputation system for electronic markets [64]. Most reputation systems are intuitive and ad-hoc however they have built their reputation systems on beta probability density function in statistics. The beta distribution is mapped to an opinion, which is a belief about the truth of statements.

2.2.2 Trust in Distributed and Peer-to-Peer Systems

In distributed systems, there is no central authority for assessing the trust of entities. Hence entities form their own opinions of trust by exchanging information with their peers. Generally methods from game theory [141], bayesian networks [135] are used for trust calculation distributedly.

Aberer and Despotovis were one of the first researchers to propose a reputation management system for P2P systems [4]. They employ algorithms and data structures that require no knowledge from a central authority. The trust model is based on the past interactions between the nodes. One drawback of their method is that only the negative feedbacks are considered and the system is sensitive to misbehavior of peers. The resurrecting duckling model in [121] and its descendants [7] use out-of-band channels to authenticate key exchange. The established trust between the nodes is binary, either secure or not secure.

There are other trust models for peer-to-peer systems which we do not want to go into details of as we are interested in trust models for sensor networks. Other trust mechanisms surveyed by Momani and Challa[90] are SECURE[18], Distributed Trust Model[2], Bayesian Network Model [135], UniTec[70], BambooTrust[71], B-trust model[107].

2.2.3 Trust in Ad-hoc Networks

In ad-hoc networks, nodes join to networks or move networks very often. There are no trusted nodes to support the network functionality. Trust relationship between the nodes is also dynamic as the network is constantly changing [152].

A majority of the trust mechanisms in ad-hoc networks use game theory and bayesian network approaches. Two examples of these systems are CONFIDANT [16] and CORE [87].

2.3 Trust in WSN

There are some surveys done on trust in WSNs [5, 41]. WSNs face different kinds of attacks such as eavesdropping, fabrication, injection, modification of packets, node capturing and many others [90]. These attacks raise issues such as privacy, accountability, data integrity, data authentication and data freshness. Some research has been done on security of WSN as surveyed by Momani and Challa [134, 101, 152, 131, 97, 156, 104, 121, 153, 106]. Cryptographic mechanisms do not completely solve the problems. System faults, erroneous data, bad routing by malicious nodes can cause network breakdowns. Cryptography is not sufficient to solve the security problems, cryptographic approaches should be integrated with tools from domains such as statistics, e-commerce, social sciences. Some nodes can behave maliciously or selfishlessly. Trust architectures have to discover and isolate these nodes. There are different approaches followed by researchers [152].

- Maintain a trust and reputation table for all nodes in a sub-network
- Use a watchdog mechanism to monitor the behavior of the nodes

- Discover faulty nodes and exclude them from the network
- Reward nodes so that they comply with protocol rules
- Use of low-cost cryptography to protect the integrity of the data

Trust has been researched for a long time [82]. It has been studied by many disciplines such as social sciences [112], economy [37]. Yet we cannot say that there is a formal definition of trust. Trust establishment between the nodes is a security approach that is very effective in WSNs. As nodes work cooperatively, trust relationship between them improve the security of WSN.

Security and trust are very related concepts and sometimes they are used interchangeably [105]. However security is different than trust. It is broader than trust and overhead is higher in security.

Trust is used in restructuring WSNs such as omitting nodes, adding nodes, merging clusters. Trust establishment is a must because WSN depends on cooperative and trusting nature of its nodes. However due to limited resources in WSN, it is not possible to use the traditional cryptographic approaches [40]. Different trust mechanisms are needed for wireless sensor networks. Trust in WSNs is still an open and challenging field.

2.3.1 Best Practices of Trust in WSNs

Lopez et al. describe the best practices of trust management in WSNs in their survey [80] based on the other surveys [41, 62, 61].

2.3.2 Trust and Reputation

Reputation and trust should be maintained separately in a WSN. Reputation builds in time. To make an accurate decision, trust should be calculated based on reputation. Without reputation trust will have a value based on the instant behavior. For instance, a node that has behaved maliciously in the previous 10 time intervals can behave good, however the network will not be deceived by the last action as the reputation will reflect the bad behavior history.

2.3.3 First-Hand Information Gathering

There are many events in a sensor network that can be used as a base for trust computing such as hardware errors, energy issues, node relocations, sensor reading deviations. These are considered first-hand information and they should be taken into consideration. A trust management system that considers multiple sources of information will be more robust.

2.3.4 Second-Hand Information Gathering

As sensor networks consist of nodes that are working collaboratively, second hand information should be considered for trust management. A node can have local intelligence. To some extent it can detect abnormal activities of itself and can report this to its neighbour nodes. It can also report a bad behavior of its neighbor to another neighbor. However when considering second hand information, we should be careful about bad mouthing attack. Bad mouthing attack happens in WSNs when a node gives bad reports about a good node and good reports about a bad node misleading the trust calculation.

2.3.5 Initial Values

The nodes in the network should be given initial trust values at the deployment time. As we assume that a network administrator has configured them and tested them, they should all be equally trusted. However the system should be suspicious about the nodes added to the network after the deployment, as they can be part of a white-washer attack, where a node throws away its bad reputation by creating a new identity.

2.3.6 Granularity

Nodes in a wireless sensor network might have different actions such as sensing, routing. Different trust values should be assigned to different actions of a sensor node.

2.3.7 Updating and Aging

Trust should build overtime. When trust of a node is updated, the past trust values should not be overwritten. The previous trust state of the network should be remembered. If bad behaviors are not remembered, the network will be vulnerable to on-off attacks [99].

Chapter 3

Background on Provenance

Provenance comes from French, from the verb provenir 'come or stem from' and from Latin provenire, from pro- 'forth' + venire 'come'. The concept of "provenance" originally derives from art and is defined broadly as the origin, history, chain of custody, derivation or process of an object [23].

In some sensor networks, there is no centralized control over data as in the traditional databases, hence data is copied, moved, created, updated and deleted in a decentralized way. Provenance plays an important role in deciding about the qualities of the data such as trustworthiness, accuracy, and verifiability. Provenance management should be a concern in sensor networks in order to have an understanding of how the results are obtained for later use such as fault tolerance, troubleshooting, result reproduction and performance optimization.

3.1 Motivations for Provenance

Provenance become more important when the financial industry imploded in 2008. In 2010, US Congress passed the Oversight Act which mandated that every major financial transaction has a verifiable record. This had a motivating effect on provenance community such as Department of Defense's Orange Book in Security Community in the 1970s [25]. After the Oversight Act, USA started to follow stronger rules for transparency for financial data. Data origin of reconciled papers began to be recorded as provenance for later reevaluations.

Many areas utilize provenance as listed below. However the areas which had major impact on provenance tide are the intelligence community and hospital information systems. Provenance of dossiers is important in the world of intelligence because it introduces the way of hiding the authors, keeping the authenticity of the document. In the applications for Homeland Security, this approach has been used [38]. The same requirement is also present in business use cases where job reviewers should be kept anonymous while it is guaranteed that they are valid [54].

One other domain that motivated provenance in computing is Hospital Information Systems [129]. Provenance of data changes is very critical in this domain. The origin of medical results should be traceable, and when a medical decision is made, the base for this decision should be recorded. For example, Health Care Portability and Accountability Act (HIPAA) mandates logging of access and change histories for medical records, which is doable through provenance [35].

3.2 What is Provenance?

Provenance is defined broadly as the origin, history, chain of custody, derivation or process of an object [23]. In disciplines such as art, archaeology, provenance is crucial to value an artifact as being authentic and original. In computational world, as all kinds of information can easily be changed, provenance becomes important way of keeping track of alterations [23], whereas in science it can be crucial for the quality and trustworthiness of the result.
Although provenance has a formal definition, it has various meanings in different domains. User communities are calling data related to the origin of the object in their applications as *provenance*. It is important to look for features and requirements of applications in order to define what provenance is for them.
A couple of different usages in various domains can be listed as follows to give the reader a glimpse of provenance [25] :

- Wireless sensor networks keep sensor location, timestamp of observations, node type as provenance data.
- Operating systems log important system events to aid system administration and intrusion detection.
- File systems record basic metadata such as file creation, modification, ownership and permissions.
- Version control systems record metadata about when changes have been made and by whom.
- Compilers use source line number tagging to aid in pointing to the sources of compile-time and run-time errors.

- Specific database curators maintain detailed high-level records of who has inserted, modified or deleted data and (sometimes) where it has been copied from.
- Web browsers retain history information about which sites have been visited and when.

3.3 Properties of Provenance

In his work, Muniswamy-Reddy list these properties as requirements for provenance to be useful [95].

Data Coupling: Data and provenance should be coupled together through the applications and systems because provenance will not be sound without the actual data. An object and its provenance timeline should match. Otherwise the data can be old but provenance information can be updated or vice versa which will lead to erroneous results. Any transaction made on the data should be reflected to the corresponding provenance data such as an update, deletion, and manipulation. Data coupling can be a problem when various information is kept as provenance. While some provenance information is dynamic, the rest can stay static and the connection between them might get lost. For instance, provenance values such as GPS location of a sensor node, energy left at the node can change while node id remains same. A solution can be made by combining the varying provenance data with a fixed provenance such as connecting the GPS information (variable provenance) with sensor ID (fixed provenance). Coupling is handled differently in various systems. In some systems, data is tightly coupled to provenance, data and metadata are stored in the same storage system (file system or database) with the same keys or even they can be attached to the same data file as done in the headers of NASA Flexible Image Transport System.

Data Independent Persistence: Provenance should stay permanent although the objects are temporary and link between the objects should not be lost in any situation such as deletion of an object, a system crash. For example, if object P is the ancestor of an object L, this information is stored in provenance of the object P. If at some point in the system P gets deleted, its provenance should be kept. Otherwise provenance graph will be disconnected and there will not be a meaningful picture of the provenance of the system. We should still be able to derive that P is the ancestor of L although P does not exist anymore.

Efficiently Queriable: Although it is a fact that provenance is created more often than it is queried, provenance should be able to be efficiently queried. In systems with a few number of objects, users will easily be able track the provenance data. However efficient querying becomes crucial in systems with large number of objects where users do not have the precise knowledge of the objects they want to access. For example, a user might want to selectively lookup the ancestors of an object in a system where ancestry relations form long chains. At this point an efficient querying mechanism is mandatory. Otherwise the stored provenance data is inaccessible and is of reduced value [95].

3.4 Provenance in Different Fields

In this section, we review how provenance is defined in different fields.

3.4.1 Provenance in Art

In the study of fine art, provenance refers to the documented history of some art object [91]. Provenance of a painting is a history of its ownership. Based on the documented history, the object is considered authentic or fake. For instance, if it cannot be verified that Mona

Lisa was created by Leonardo Da Vinci then the painting is considered less valuable.

3.4.2 Provenance in Science

The idea of documenting the provenance of a data item comes from the arts as stated above, but recently the eScience community has taken a great deal of interest in documenting the steps, data sets and processes used in a research result. One of the hallmarks of scientific research is that it can be duplicated by others; doing so allows validation and moreover presents the additional research ideas that a paper creates. eScience research has focused on reproducible research and the use of workflow technologies to illustrate the steps taken in a scientific experiment for reproducibility. This has led to introduction of scientific workflows. Commercial and open source scientific workflow systems have started to be developed to allow scientists automate the steps taken during their research without going into the burdens of scripting [84]. Some popular scientific workflow management systems are myGrid/Taverna [98], Kepler [15], VisTrails [45], and Chimera[43].

The work done on provenance management in eScience community has focused primarily on scientific workflows. Provenance management should be a concern in order to have a better understanding of how the results are obtained. Although the processes are straightforwardly connected, the reproducibility of the papers results or additional examinations would be enhanced by having clear workflow and data provenance. Scientific workflow systems automatically capture provenance information during workflow creation and execution to support reproducibility [124]. Having this motivation, provenance has been studied by several approaches in eScience Community. Provenance research in eScience community has included work on different domains and applications such as sensor data access, analysis [8] and provenance-based fault-tolerance mechanisms [28, 127] focusing

primarily on data and workflow provenance. As this area progresses, scientists can be able to share provenance derived from different systems, analyze and visualize it. Resulting science collaboratories such as Open Provenance Model, will change the way people do science [44].

3.4.3 Provenance in Computing

Computing literature divides provenance into data provenance and workflow provenance [92]. Data provenance gives a detailed record of the derivation of a piece of data that is the result of a transformation step [125] whereas workflow provenance is the information or metadata that characterizes the processing steps of information from input to output [124]. Database community was first to address the issue of provenance. Cui et al. [31] were among the first researchers to formalize provenance of data in the context of relational databases calling it *lineage* of a tuple. Each tuple present in the output of a query is associated with a set of tuples present in the input. The associated tuples are called lineage. Basically the lineage of a tuple is defined as the input data that contributed to the tuple. Later based on this intuition, provenance has been re-defined.

Although provenance was first addressed by database community, later it was used by many research communities such as network[155], internet [20], trust[51], file systems[113].

3.4.4 Provenance in Databases

Trio, Panda, DBNotes and SPIDER are examples of provenance systems with database approaches. Trio keeps track of the provenance of tuples. Essentially it tracks tuples that are used as inputs in generating another tuple. It associates each tuple with a confidence level hence it supports uncertainty data model. It is a very interesting work in the sense that it

extends SQL to support lineage and accuracy [139]. Panda goes beyond Trio by supporting process provenance [60]. In Trio which tuples are ancestors of a tuple is tracked whereas in Panda also processes that produce a tuple's state are tracked. DBNotes is an annotation management system for relational database systems [125]. Attributes are tagged with multiple annotations. When the query is executed, annotations of relevant attribute values are propagated to attribute values. SPIDER computes data provenance over schema mappings that uses non-annotation approach [125]. Schema mappings are the logical assertions of the relationships between an instance of a source schema and an instance of the target schema. They describe how data in the source and target instances are defined.

3.5 Provenance in Sensors Network Domain

Mostly sensor data is treated as data with real time value but it has historical value too if thought in a broader sense. To heal, adjust and manage sensor networks, historical sensor data is required. For sensor data to have historical value, it has to have provenance. Under this fact, research on provenance in sensor networks is crucial. Sensor networks is a domain with its specific requirements and restrictions such as energy limitations, environmental conditions. Although there is research done on provenance in eScience and databases, provenance in sensor networks should be studied separately taking constraints of sensor networks into consideration.

Park and Heidemann list only some of the differences of provenance in sensor networks from eScience and database community [102]. One difference is that in scientific workflows, data is taken from static databases whereas sensor networks have live data feeds.

Secondly, ownership and access are not handled in scientific workflows whereas reporting sensor's identity and user privileges are important in sensor networks. Thirdly, scientific computations can last for many days on supercomputers but sensor data is more lightweight, in turn provenance of sensor data is lightweight too. The other difference is that in database research, provenance is purely related to metadata and data transactions whereas in sensors there might be a need to capture the executable codes and process chains (workflow provenance). In databases, data provenance is computed only for the current state of the tables however in sensor networks old snapshots of the data is valuable too. By maintaining explicit timestamp on sensor data provenance, old results can be traced.
Provenance research in sensor networks is still very new and open to various directions. Many research has been done regarding different aspects of sensor networks. For instance some research has leveraged provenance in fault recovery and success validation [122], whereas in other research, provenance information associated with sensor data has been used in answering domain specific complex queries [103, 102].

3.5.1 Data Provenance in Sensor Networks

Data provenance is crucial in data streaming applications which use sensors. Data quality becomes an important metric in sensor network applications since there are multiple sources and processing is done by multiple nodes. Data provenance refers to valuable information in assessing data quality such as which node created, modified, deleted data, the manipulations data went through [75]. Data provenance in databases is classified as why, how and where provenance. This classification can also be used in sensor networks provenance models where there is a central data storage.

Students

	name	studentid
t1:	Jane	415-1200
t2:	Jill	831-3000

Courses

	studentid	course taken	course dept	grade
t3:	415-1200	corc1312	math	A
t4:	415-1200	cisc2210	cis	A
t5:	831-3000	cisc1210	cis	B
t7:	831-3000	corc3210	math	B

Figure 3.1: Example database: student portal

Query1:

```
SELECT s.name, c.grade
FROM Students s, Courses c
WHERE s.studentid = c.studentid
c.type = 'math'
```

Result of Query1:

name	grade
Jane	A
Jill	B

Figure 3.2: Example query

3.5.1.1 Why Provenance

First, we will examine "why provenance" in databases. "'Why provenance" holds information about the input which cause output to be produced. It is the one which is closest data provenance type to the lineage definition of Cui et al. [31]. In Figure 3.2, as each of the two source tuples t_1 and t_3 justify the existence of the Jane tuple, why provenance of the tuple (Jane, A) of Query1 becomes the source tuple set $\{t_1,t_3\}$. Source tuples t_1 and t_3 are named "proof" or "witness" for the Jane output tuple according to Query1 because they are the two tuples contributing to Jane output tuple.

Why provenance tells about the source tuples that witness the existence of an output tuple in the result of the query but it does not describe how an output tuple was formed according to the query. Green et al. [53] has introduced the concept of polynomials of defining how output tuple was derived. They call these polynomials "provenance semirings" in their paper. A semiring is a system consisting of a set S together with two binary operations, addition and multiplication [14]. Later the idea provenance semirings led to the introduction of how provenance which we define below.

To our knowledge we are the first ones to define "why provenance" for sensor networks. It is used for describing the nodes that contributed to a data result. For instance, why provenance of the data in *Hub 1* in Figure 3.3 will be the set $\{$A,B,C,D$\}$.

3.5.1.2 How Provenance

"How provenance" gives more specific information about the way query runs. In the query in Figure 3.1, for all students the student name and their grades in the courses offered by mathematics department are derived. In the select, two tables Students, Courses are combined. Why provenance of the output tuple (Jane, A) is $\{t_1,t_3\}$. We can only learn that

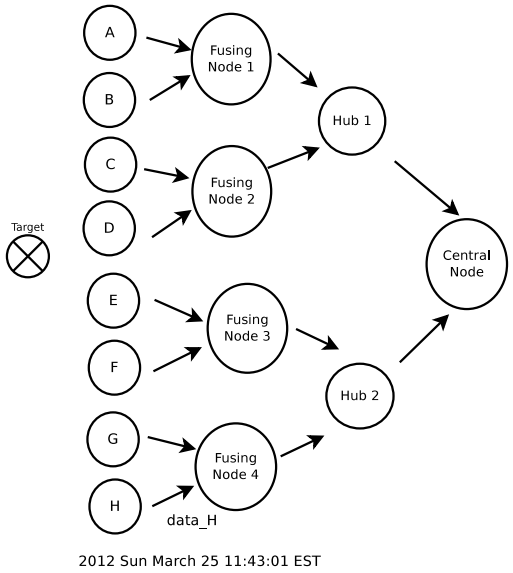

Figure 3.3: Data path used for why provenance

tuples t_1 and t_3 witness the output tuple (Jane, A) but we cannot learn how the tuple was formed. At this point how provenance comes in. How provenance of the output tuple (Jane, A) is represented as a polynomial (t_1 + t_3) giving information about the way the query runs in order to gather the tuple. The polynomial (t_1 + t_3) means t_1 and t_3 contribute to the query by a select. If there were two queries, outer and inner then we would use multiplication representing the join operation between inner and outer queries. Why provenance can be derived from the how provenance but the converse is not always possible.

We define the how provenance in sensor networks as follows. It is a polynomial with variables as sensor ids and operations of addition and multiplication. Two sensor ids are combined with addition. This means that their data is concatenated for the result. Multiplication stands for any kind of data aggregation operation done at fusion nodes such as running aggregation algorithms on the data such as finding dissimilar data, averaging.

For instance, how provenance of the data in Hub 1 in Figure 3.3 will be the formula in Equation 3.5.1.

$$A * B + C * D \tag{3.5.1}$$

3.5.1.3 Where Provenance

"Where provenance" explains precisely from where an attribute value v in the output was copied according to query Q. It describes the relationship between input and output locations [24]. Notation (R,t,a) refers to a location showing field A of a tuple t of a relation R. The where provenance of the value "Jane" in the first tuple in the result of Query1 is the location (Students, t_1, name) in the input database, since "Jane" was copied from the name attribute of the tuple t_1 in the Students relation, according to Query1. Similarly, the where-provenance of the value "Jerry" in the second output tuple is the input location (Students, t_2, name).

We define where provenance in sensor networks as follows. It is a tuple consisting of sensor id and sensing time showing which sensor created the data at what time. For instance, data_H forwarded by sensor node H at time 2012 Sun March 25 11:43:01 EST will have the where provenance of (H, 2012 Sun March 25 11:43:01 EST)

3.5.2 Utilizing Provenance in Sensor Networks

Basically, if there is any input data changing to an output data (data provenance) or any process chain (workflow provenance), provenance can be included. The areas where provenance is utilized in sensor networks can be listed as follows [54]:

- **Security and Safety:** Data need provenance in order to ensure authenticity. In the virtual world, data that a node is creating can be changed and this can be used for

malicious, illegal purposes. If there is provenance tracking on the node in a sensor network, it can be used to verify if the data is authentic or not. For instance, Sultana et al. propose a provenance-based mechanism to identify malicious packet dropping adversaries in sensor networks [123]. They transmit provenance information along with the sensor data to isolate the malicious link.

- **Failure Recovery:** Provenance is frequently used for failure recovery and success validation. Failure recovery and success validation are alike in the sense that there is a verification in both of them. In failure recovery, we want to avoid repeating a specific bad outcome. In success validation, we want to validate, replicate and generalize some good results. They both include understanding causal chain of events and counterfactual possibilities [25]. In our previous work, we used provenance for troubleshooting a sensor network. We analyzed data fusion graphs called provenance graphs to find out sources correlated with anomalous results [68].

- **Development Process:** Provenance answer the questions "how an object is created", "through what changes the object went through" [95]. The changes between two snapshots can be found out by the provenance logs if a sensor network is unstable. In our previous work, we store value dependencies as provenance to capture the sensor network snapshots [39]. Later the snapshots are compared to detect the changes in the network such as node movements, energy changes.

- **Optimization:** With provenance, tracing and monitoring can be done very efficiently. This information can be used in optimization. For example, in distributed system deployments, it is crucial to figure out resource allocation patterns to minimize access times and provenance can capture the patterns. A scenario can be how

to locate the nodes nearer to their frequent users in a sensor network. By analyzing the provenance logs, the most efficient allocation can be found out and deployed.

- **Trust:** Provenance has also been used for assessing and enhancing trust in various environments such as sensor networks[39], multi-hop networks [52], social networks [51] and semantic web [20]. We described provenance work in trust in more detail in Section 3.7.

- **Real-time Diagnostics:** Queries can monitor a network for runtime anomalies. For instance, when number of changes in a routing table in a network exceeds a threshold, an alarm event can be generated. Upon this alarm, the system may generate a distributive recursive query over the network to detect the source of malicious activities [154].

3.5.3 What is kept as Provenance in Sensor Networks

When provenance is concerned, one of the main questions is what to keep as provenance. Can all data regarding origin and chain of custody of an object or data be considered as provenance? There is not a clear-cut answer to this question because provenance data is formed by the requirements of every domain. For instance, in a document versioning system, provenance is required for tracking the changes made to documents. Hence the appropriate provenance data for a document versioning system will be the version control numbers, version change dates, the track of changing authors. Provenance in a scientific workflow system will be the software used in each process, the datasets used, the order processes are called.

For this work, we are interested in what data can be considered as provenance for sensor

networks. There are various types of provenance. One of them is network provenance. Network provenance refers to provenance data kept in network [154]. One of the most commonly used network provenance data is reachability in a network. Zhou et al. use network provenance for enforcing distributed trust management in networked information systems [154].

Sensor specification such as location of the sensor, type of the sensor, output structure of the sensor are frequently kept as provenance data in sensor networks. Data such as angle of a video sensor should be recorded as provenance in order to make sense of the data [73]. IBM T.J. Watson center has a work on a biomedical sensor system for online healthcare analysis. The system is called Century [85, 12]. They store provenance information describing which data and processes contributed to an event in the system. Vijayakumar and Plale have proposed a system capturing provenance data for meteorology forecasting domain [130]. They keep timestamp, type and description of sensed data and events in the system as provenance.

Patni et al. keep location of the sensor, timestamp of observations, attributes of the sensor itself as provenance data. For instance, they keep what kind of sensor the node is. They make use of provenance data in their system. One example usage is as follows. As a motion sensor is not useful in the context of a blizzard, they disregard the readings from motion sensors when looking for events related to a blizzard [103].

3.6 Provenance Management in Sensor Networks

To use sensor data for its historical value it has to be named, stored and indexed properly. Sensor data is location specific but also has to be combined with other sensors' data to make more meaning out of it. For example, immediate traffic data is useful for ticketing

drivers but it is also useful in other ways such as finding the correlation between weather and traffic, time of the day and traffic. It can also be combined with data from other cities to make statistical observations.

There are three key components of a provenance management system: a capture mechanism, a representational model, and an infrastructure for storage, retrieval. There are different kinds of management solutions for these systems with their trade-offs.

3.6.1 Capturing Provenance in Sensor Networks

Capture mechanisms have three main classes: workflow, process and operating system based. Most workflow systems support provenance from their initial design. Process-based systems should be instrumented to capture provenance from the information coming from the processes. OS-based systems should do post-processing to extract provenance because they are not coupled with workflows or processes [44].

3.6.1.1 Granularity

There are two types of granularity, coarse-grained and fine-grained. Coarse-grained provenance tracks dependencies between input and output at a very abstract level (e.g., streams) whereas fine-grained provenance tracks dependencies for individual data items [50]. For most of the sensor network applications fine-grained provenance should be kept because we should be able to track source data from the result. For instance, for some applications sensor readings are sent to a central system. These systems detect critical situations such as machine overheating or low inventory. Human operators need to understand from which inputs these events are derived. For a machine overheating event, it should be understandable which sensors measured high temperature values and this information is provided by

fine-grained provenance. Park et al. keep fine-grained provenance for their Sensornet re-publishing system [102]. Huq et al. also propose a provenance system for sensor data streaming applications combining fine-grained and coarse-grained provenance [58].

3.6.1.2 Collection Methods

There are two collection methods automated monitoring and manual capturing.

Automated Monitoring:

This method is also called observed provenance. Provenance data is automatically gathered without depending on user input. In this type of provenance collection, an administrator can specify some parameters to the monitoring process but cannot directly interfere with the collection phase. Advantage of automated monitoring is that it is resistant to malicious attacks since users cannot alter the provenance data. On the other hand, as data is collected automatically, integrating provenance data with the application should be done by software developers, which can be a hard task.

Manual Capturing:

The provenance data collected manually is called disclosed provenance. In this collection type, manual sensors depend on user input for provenance gathering, hence provenance data can be manipulated easily by malicious users. Advantage of manual capturing is that integration with the application will not be an issue as the collection depends on user input.

3.6.2 Representation of Provenance

There is no standard for lineage representation in provenance community. As every domain has specific needs, it is still a challenge to build a metadata standard. Provenance is represented using different techniques depending on the underlying data processing system.

Two major approaches used are annotations and inversion [116].

In the annotations approach, metadata is collected as annotations and descriptions about data and processes. In this approach, provenance metadata is pre-computed and readily available for querying. When a transformation occurs, the datum is annotated with the provenance information. Each datum can have annotations attached to it. For instance, in a sensornet republishing system, every data can contain a url to its source. A problem with this approach is that not all objects support annotations. In most systems only attribute values can be annotated whereas tuples, relations or functions cannot be annotated. Mostly provenance systems with annotation approach use XML while others use domain ontologies in languages like RDF and OWL to capture semantic information [84], [151], [96]. Ledlie et al. use annotation-based approach for maintaining provenance for sensor data [74]. They mark sensors when they are replaced with newer models having slightly different properties, or when software on the sensor devices was upgraded. These descriptions and annotations are searchable in their system.

In the non-annotation (inversion) based approach, the state of the database is captured before and after a transaction. Provenance information is gained by comparing two states of the database [125]. In the inversion approach, the information about the queries and output data is used in order to derive the input data.

While the inversion method is more compact, the annotation method provides richer information. Inversion method is not suitable for sensor networks as sensor networks are very dynamic and a compact view of the network at different snapshots is not accessible. One other reason of why inversion method is not suitable to sensor networks is that processing of sensor data is arbitrary and hard to invert [102].

3.6.2.1 Open Provenance Model

Researchers who joined *First International Provenance and Annotation Workshop* (IPAW'06) wanted to know about other systems' capabilities and expressiveness such as design principles, representations, retrieval methods, storage choices. Hence they agreed on organizing an event for sharing the provenance approaches of different systems and the idea of *First Provenance Challenge* was born.

In the First Provenance Challenge, participants ran a Functional Magnetic Resonance Imaging workflow and executed a pre-identified set of queries. The strength of the systems were questioned according to their success of executing provenance queries and displaying the asked provenance information.

In the Second Provenance Challenge, systems were asked to exchange provenance information and interoperability of the systems were measured. At the end of the Second Provenance Challenge, researchers decided that there should be a standardization of the way provenance is modeled, stored, queried and changed to make the systems compatible with each other. Following this consensus, authors met in a workshop in 2007, crafted and iterated a data model called Open Provenance Model. In the following paragraphs, Open Provenance Data Model's specifications, properties and design principles will be briefly described.

Entities:

The core requirements of the model are listed below. These are the initially aimed requirements whereas authors decided to allow the addition of more features to the model in the future.

- a compatibility layer where different systems share their provenance information

- implementability such that tools can be built on top of this model
- a precise definition of the model, a digital representation of provenance for any object
- a set of valid inferences that can be made on provenance graphs

Open Provenance Model is based on three primary entities.

Artifact: Immutable piece of state which may have a physical embodiment in a physical object or a digital representation in a computer system. They are represented by circles.

Process: Action or series of actions performed on or caused by artifacts and resulting in new artifacts. They are represented as rectangles.

Agent: Contextual entity acting as a catalyst of a process, enabling, facilitating, controlling, affecting its execution. They are represented as octagons.

Dependencies:

Provenance of objects is modeled as a directed acyclic graph (DAG) representing a past execution but never a prediction of future events. Dependencies are shown with edges, an edge represents a causal dependency between its source, denoting the effect, and its destination, denoting the cause. There are five predefined causal relationships *used, wasGeneratedBy, wasTriggeredBy, wasDerivedFrom, wasControlledBy* and edges are labeled with one of these causal relationships.

3.6.2.2 Current Provenance Representation Models in Sensor Networks

Below is a list of some provenance systems in sensor networks and the way they represent provenance.

A couple of sensor network systems have used OPM in sensor network community. Liu et al. have modeled a provenance-aware virtual sensor system using the open provenance

model [78]. They also have work where they use OPM for provenance tracking, they use concepts such as derivedFrom to note the causal relationships between the raw and/or other virtual sensor and the new virtual sensor [79]. Stephan et al. leverage open provenance model as a multi-tier model for global climate research [122].

Park and Heidemann et al. have work on republishing sensor data. Republishing is the process of transforming sensor data so that it can involve multiple steps and different users. In their system, many sensors are attached to Internet and are publishing data to one of many centralized sensor stores. Data provenance is the information of the source and the transformation applied to the source [102]. Their Open Provenance Model record two basic provenance relationships; the derivation between republished data to source data and transformation of republished data.

Lange et al. has work on provenance aware sensor networks for real-time data analysis [73]. In their system provenance is used for tracing data back to its source. As OPM does not specify standards for querying or recording provenance data, they use a modified version of OPM as their provenance model. They use distributed temperature sensors for measuring temperature over long distances. Hence sensor data come from several sources and it has to be combined with other sensor data to make sense. They create a workflow that is capable of recording provenance so that the sources and processes that are involved in producing a set of results can be easily tracked [86].

Lim et al. has work on using provenance for trust assessment in sensor networks which we described in more detail in Section 3.7. They represent provenance as a tree illustrating the data flow path. However in some applications such as document workflow systems, there can be loops and more complex graphs are created [75].

3.6.3 Provenance Storage

Provenance storage is also an ongoing research issue. Many approaches have been tried but they all have disadvantages and advantages. In a network, provenance can be stored locally or distributedly. In local provenance, provenance data such as whole reachability tree is stored at a node. In distributed provenance, pointers to previous nodes are stored to reconstruct provenance on demand. For instance Zhou et al. keep provenance (reachability) locally for their work [154].

One other classification in network provenance is online provenance or offline provenance. Online provenance is only kept for the network state that is currently valid. Offline provenance is kept even when the derivations are expired. Online provenance is helpful for runtime reaction or adjustments such as detecting network anomalies, restructuring network [39]. For instance when a node is detected to be malicious, all the routing updates associated with this node can be deleted. Offline provenance is useful for working on data that has expired for historical results such as finding the contact rate between two nodes using the reachability information. Although offline provenance provides statistical measurements regarding past behaviour, it has high storage overhead.

Provenance can be stored in a centralized or decentralized storage system. Both systems have advantages and disadvantages depending on the application they will be deployed on. In a centralized storage, provenance data of the system is sent to a central storage system, whereas in the decentralized mode provenance is stored in a distributed manner in many locations such as in nodes in a network. In these systems, maintenance is easy but it is hard to search and query provenance information since there is not a systematic design behind. Furthermore, as transmitting data over long distances is expensive and sensor networks are

energy-constrained, decentralized approaches work better for sensor networks. In centralized systems, although there is a connection between the data and metadata, they are stored in different systems with different representations. Maintenance is more difficult in a centralized approach but it is easier to query and search provenance since the mechanism is designed keeping this requirement in mind [116].

Lange et al. build a provenance aware sensor networks for real-time data analysis. There are two kinds of storages in their system. First one is temporary working storage and it is usually stored in-memory. It is for storing window queries or caching. Second type is summary storage. It is for recording historical data, it is usually stored in disk since it is in huge amounts. Static storage keeps fixed metadata about sensors like their geographical location [73].

Provenance is used when constructing indexed, distributed repositories of sensor data. Ledlie et al. built provenance aware sensor data storage [74].

Provenance data can grow very large sometimes even larger than the actual data. Hence the way it is stored is important for its scalability. Inversion method described in the previous section is more scalable than the annotation approach. However metadata stored using the annotation approach can be reduced by several different methods. One method can be fixing the stored metadata to some specific kind and recursively traversing the ancestry path in order to obtain the full metadata history.

There are some methods proposed for solving storage problem in sensor networks. Park et al. propose a compression scheme for decreasing needed storage [102]. Provenance data can have redundancy as objects can share the similar provenance chain. For example, two objects with the same ancestor will have similar provenance data. Deduplication is used in provenance systems in order to eliminate redundant data and improve storage utilization

[21]. For instance, a part of the provenance data of the objects sharing the same ancestors are same hence provenance of the same ancestor is stored more than once in the system creating a storage overhead. Deduplication methods erase multiple entries of the same provenance chain. The other method is compressing data provenance.

When fine-grained provenance and coarse-grained provenance are compared, coarse-grained provenance is more scalable. Ways for controlling granularity of data provenance are proposed. For instance, in a geographical system, state level provenance (coarse granularity) can be used instead of city level provenance (fine granularity) as cities in the same state share the similar provenance[75].

3.6.4 Querying Provenance

In sensor networks data has multiple consumers. Data captured may be used not only by the node creating the data but by other nodes too. Both accessing and querying data should be flexible [74].

Although it is a fact that provenance is created more often than it is queried, provenance should be efficiently queried. In systems with a few number of objects, users will easily be able to track the provenance data. However efficient querying becomes crucial in systems with large number of objects where users do not have the precise knowledge of the objects they want to access. For example, a user might want to selectively lookup the ancestors of an object in a system where ancestry relations form long chains, at this point an efficient querying mechanism is mandatory. Otherwise the stored provenance data is inaccessible and is of reduced value [95].

Provenance queries aim to obtain the provenance of electronic data that is of interest to the user. A challenge in this is how to define precisely the data that user wants to obtain. As

provenance data is in the form of a graph, a query language should be capable of dealing with graphs in order to efficiently do querying. Provenance queries perform reverse graph traversal over the data flow DAG and terminates according to the query-specified scope; the query output is a DAG subset [91].

Using existing languages such as SQL, Prolog for querying has the advantage that they are known and accepted by many people. On the other hand these languages were not built for provenance models so they lack some important features which will make life easier for provenance query writers. Semantic Web languages are used for querying and representing models too. As they are capable of handling metadata, annotations, they can be a good choice but it is still unclear if they can handle large amounts of data.

Provenance differs from other metadata as it includes relations about objects. Moreover since most provenance data is a graph, any query language should be able to query graphs efficiently which is a challenging task. In the First Provenance Challenge, seven out of nine queries included paths which proves this claim.

A query language for provenance data should at least contain the following features [54, 74].

- regular expressions upon paths : It should be possible to describe paths and their elements (nodes/edges) by regular expressions.
- path pattern matching : Although whole path is not available before the query runs, pattern matching between paths should be doable. For example, we might be interested in nodes where their outgoing edges eventually touch another node no matter which nodes the path contains.
- paths as first class citizens : In query languages, set level operations should be doable

on sets and paths can be compared.

- recursive queries : Queries that are going backwards to find the origins and queries that are going forward to find the derived data are necessary.
- distributed queries : Data is stored near sensors, so the queries will be distributed.
- heterogeneous queries : There might be a need to query data from two communities traffic and weather.

An efficient evaluation of provenance queries remains an open problem. Although some techniques exist, they are bound to specific models. Some querying mechanisms are described below in order to give the reader an idea about the current approaches.

Patni et al. build a provenance architecture called Sensor Provenance Management System (PMS) that uses a domain specific provenance ontology called Sensor Provenance (SP). They develop a specific ontology called Sensor Provenance (SP) to annotate the sensor data [103]. They have grouped the annotation into three categories: spatial parameter, temporal parameter and domain parameter. They use SPARQL for querying provenance data. They can run provenance queries such as "Find all the sensors which have observations related to a blizzard occurring in Nevada on 24th August 2005 at 11 am".

Ledlie et al. have work on provenance aware sensor data storage [74]. In their paper, they try to characterize the ways sensor data can be queried. They consider the type of queries, a system should support. They give examples from the Document Versioning Systems. In these systems multiple programmers are working on the same program, they will be editing concurrently and independently. Typical queries on these systems could be, "show me the file as it is now, or as it was yesterday", "show me all changes to this file since last week", "show me when each line in this file was inserted", "find the person who removed this error

code". CVS systems support these queries, however document control systems involve operations done on many files in different directories. Mostly these queries should be handled manually in these systems. Some possible queries in an eScience system can be listed as follows: "Find all raw data from which this dataset was derived", "Show me what I need to reproduce this result", "Find an experiment that answers this question. "Sensor applications should support a variety of queries too. For instance, in a sensor-enabled ambulance team, sensors such as pulse oximeters, EKGs are placed on the patient. They record vital signs until patient reached to the hospital. Metadata is critical for patients, doctors and patient relatives. Queries can include "show me everything we have done for this patient", "show me the heart rate from moment of arrival until now", "show me the results of oxygen treatment". Based on these examples they make some derivations [74].

Kementsietsidis and Wang [66] look for a generic technique which will work with different models. They have built a system called Century to model provenance. In Century system, blood pressure readings and prehypertension alerts are stored. An alert is produced if in any of the readings the systolic pressure is larger than 135 mmHg. There are four readings in each 3-hour epoch. The TVC model [132] is used to establish relationship between inputs and outputs of the Century model. The provenance relationship between input and output can be described by some invariant primitives. These basic primitives are time, sequence and value. As an alert is created, a doctor should be able to issue a provenance query in order to figure out the readings which caused an alert. This is called a backward query. In a forward provenance query, a reading is given, alerts that are generated by this reading are found. Efficiency of provenance queries is very important as they are used in domains like health care, banking, finance which an on time reaction is very important. There are two alternatives. In one of them a table is maintained as alerts are created and simple SQL

queries are executed to find the provenance relation. Having persistent provenance data is the shortcoming of this approach. Because studies have shown that provenance data can be larger than the actual data, this causes space problems. In the other alternative, relationship between alerts and readings are encoded by backward/forward queries. The disadvantage of this approach is that in real environments the relationship between data and provenance data are very complex, this leads to very long, unreadable, unoptimizable queries. They propose an approach different than the ones stated above. As provenance is a binary relation between data items and processing nodes of a workflow or different data items, they argue for a structure that will index arbitrary binary relations. This index has the property called duality which means an index will work both on forward and backward queries.
Provenance should be validated in order to prevent spoofing of messages from malicious attackers. For instance, nodes can have digital signatures to validate the authenticity of the computed provenance [154]. There should be some kind of access control for securing data [73]. Making provenance records trustworthy is a challenge [55]. Cost of digital signatures and cryptography techniques is too high. Therefore light-weighted digital signature techniques should be used for handling security and privacy in data provenance systems [75].

3.7 Leveraging Provenance for Trust in Sensor Networks

In many sensor network applications, provenance can be used for assessing trust. Two examples are a battlefield monitoring system and a supervisory control data acquisition system. A battlefield system gathers target locations from multiple sources such as cameras, satellite images, vehicles, proximity sensors. Critical decisions are taken based on the data hence trustworthiness is a concern and can be assessed by using provenance. A Supervisory Control and Data Acquisition (SCADA) system collects real-time information from data

collection points such as sensors, based on this data it performs critical tasks. A failure can affect the whole system. Therefore provenance is a key in preventing failures beforehand by finding out untrusted sources [75].

3.8 Provenance Challenges

Provenance seems trivial when it is defined but it is very challenging to handle provenance of sensor networks. It is hard to define where to stop. We can think of a system recording everything and also recording everything about everything and so on. It has been included on InfoSec Council's Hard Problems List in USA [27] and British Computing Society has identified provenance as a Grand Challenge in the UK [117].

One of the challenges most provenance systems face is provenance collection. Extracting the provenance information from the APIs that are designed without considering the possibility of need for pulling provenance information is a hard task.

The other challenge is integrating provenance from different layers. When provenance information flows through layers, many challenges are faced. One of them is the naming inconsistency, as all layers have a different naming mechanism, it is difficult to find the correct matching between the names.

The other challenge is in making sure provenance flows with data. Provenance is metadata so it belongs to data hence should act together with data. On the other hand there can be objects present in one layer but missing in other layers, it will be confusing to which objects to connect these unmatching objects. Designing an interface coming over all these challenges will be very infeasible hence most provenance systems work in one layer and do not handle cross-layer applications [95].

Integration with existing domain systems is another challenge. Provenance is collected

from many systems with different natures. The same object can be represented differently in two systems and some objects have a definition in one system and no representation at all in another system.

It is also a challenge to build a generic provenance framework which can be used in various systems. There are many domain-specific provenance systems however it is still an open problem to build a system which will work for all domains.

Performance and storage is a challenge too. One interesting fact is that provenance data is often larger than the actual data, it gets larger as the application runs and it has to be stored for longer period of times. Although data is manipulated, provenance data of the old data has to be kept whereas the old data no longer exist. Performance becomes an issue too when provenance storage gets very big. Some solutions such as data deduplication, compaction exist but these solutions add computational complexity to the application. It is challenging to find a solution which will use application's resources efficiently and at the same time will answer the requirements.

Due to rapid and continuous nature of data streams there are challenges in handling the data provenance. One of the challenges is that due to limited network bandwidth provenance data should be small in size. The other challenge is performing the data provenance operations simultaneously with the data flow. As there is a huge data flow, the provenance creation becomes an issue [75]. When to record provenance is a question too. Provenance can be recorded every time a result is produced however meaningful situations to create provenance should be specified [73].

Chapter 4

Background on Sensors

A wireless sensor network is a collection of self-organized sensor nodes that form a temporary network. A centralized network administration or a network infrastructure is not predefined. Nodes communicate with each other via radio links. Radio links have limited transmission range, nodes far away from each other communicate via a multi-hop strategy. Each node acts as a router and as a host [90]. There is a very low data transmission capacity and bandwidth between nodes. One other property of wireless sensor networks is that they have a limited power supply and their energy is exhausted easily. Lastly, nodes join or leave a network at any given time and their position can change frequently, this results in a dynamic network topology. They can have the same challenges that a MANET has. In addition to challenges a MANET has such as absence of infrastructure, mobility, lack of connectivity, WSNs have also computational constraints. This is why a trust model for WSNs have to be designed [90].

A sensor node consists of four sub-systems [128].

- Computing sub-system (processor and memory) : controls the sensors and executes

communication protocols

- Communication sub-system (transceiver) : communicates with neighboring nodes
- Sensing sub-system (sensor) : links the node to outside world
- Power supply sub-system (battery) : supplies power

WSN technology is a newly emerging concept. Tiny and cheap nodes are employed in large numbers in difficult environments such as military fields for many purposes such as surveillance. Small low cost sensors collect and relay environmental data [6]. Originally WSNs were motivated by surveillance in battlefields for military however in time they were used in many areas [90]. Some examples of these areas are active volcano monitoring [138, 137], microclimate monitoring throughout the volume of redwood trees [32], building and bridge monitoring [49, 100], health-care monitoring [48], and many other applications[128, 6, 144, 19].

In some of the sensor networks, there is no centralized control over data as in the traditional databases, hence data is copied, moved, created, updated and deleted in an uncontrollable way. At this point provenance plays an important role in deciding about the qualities of the data such as trustworthiness, accuracy, and verifiability. Provenance management should be a concern in sensor networks in order to have an understanding of how the results are obtained for later use such as fault tolerance, troubleshooting, result reproduction and performance optimization.

Chapter 5

Related Work

5.1 Related Work on Provenance

Provenance research in eScience community has included work on different domains and applications such as sensor data access, analysis [8], provenance-based fault-tolerance mechanisms [28, 127], provenance-based trust assessment [108].

The database community has also addressed the issue of provenance. Cui et al. [31] were among the first researchers to formalize provenance of data in the context of relational databases called lineage of a tuple t. They associate each tuple t present in the output of a query with a set of tuples present in the input and they call it lineage of t. Basically the lineage of t is defined as the input data that contributed to t. Later based on this intuition, data provenance research has made a distinction between ”where”, ”why” and ”how” provenance [17]. There is also extensive research in database community on efficient provenance storage, collection and query models [93]. Database community has also done work on managing accuracy of the data through provenance [139]

Although not as extensive as research done in database and eScience community, there has been research done on provenance in sensor networks community too. But this area of research is still very new and open to new directions. Although sensor networks contribute to all science fields by their feasible characteristics, provenance management should be a concern too in order to have an understanding of how the results are obtained. Having this motivation, provenance should be more deeply studied by sensor networks community. Some research has leveraged provenance in fault recovery and success validation [122], in troubleshooting and bug finding [67, 69]. In some research, provenance information associated with sensor data has been used in answering domain specific complex queries [103, 102]. There is also research done on provenance aware sensor data storage [74].

Nature of provenance in sensor networks is different from eScience and database community in several ways [102], this is why research on provenance in sensor network community should be done more extensively to make robust provenance systems for sensor networks. he differences of provenance in sensor networks and eScience can be listed briefly as follows. One difference is that in scientific workflows data is taken from static databases whereas sensor networks have live data feeds. Second, generally ownership and access are not handled in scientific workflows whereas reporting sensor's identity and user privileges are important in sensor networks. Thirdly, scientific computations can last for many days on supercomputers but sensor data is more lightweight, in turn provenance of sensor data is lightweight too [102].

The points where sensor network provenance research differs from provenance in database research is as follows. Firstly, in database research, provenance is purely related to metadata and data transactions whereas in sensors there might be a need to capture the executable codes and process chains (workflow provenance). Secondly, data provenance is

computed only for the current state of the tables however in sensor networks old snapshots of the data is valuable too and there is a need to maintain explicit timestamp on data provenance of old results [102].

5.2 Related Work on Trust

Trust has been a research area in social sciences for a long time however it is a new area in computing motivated by trust models for e-commerce [83].

Trust in WSNs is an open and challenging research area. Although extensive efforts have been carried out for trust management in Ad-hoc and P2P networks, very little has been done on trust management in WSN domain [41]. Some of the reputation and trust systems in the context of sensor networks can be listed as follows [120, 30, 59, 22, 140, 29, 72, 115, 143, 57, 94, 81, 150, 142, 89]. Trust establishment in sensor networks is a must because the survival of a WSN depends on cooperative and trusting nature of its nodes. Due to resource limitation of sensor nodes, using traditional methods such as cryptography to generate trust is not possible [40]. Therefore new methods for secure communication and distribution of trust values between nodes are needed.

Ganeriwal and Srivastava were the first to introduce a reputation model for sensor networks [47]. Their system is called RFSN (Reputation-based Framework for High Integrity Sensor Networks). Their model uses beta distribution to represent and continuously update trust and reputation.Their notion of trust is binary. Their model classifies actions as cooperative and non-cooperative. Nodes use indirect second hand information to calculate trust. The second hand information coming from reliable nodes is given more weight. Expected value of reputation becomes trust of the nodes. If the trust is below a threshold, the node is marked as uncooperative. They only take into account positive feedback coming from

nodes. By doing so they eliminate the bad-mouthing attack but on the other hand bad experience is not shared among nodes.

DRBTS (Distributed Reputation-based Beacon Trust System) is a system that is primarily modeled for sensor networks which it is crucial to know the location of a sensor [120]. They use beacon nodes to find the nodes that are misreporting their places. Every beacon node is distributedly monitoring the 1-hop neighborhood for misbehaving nodes and updating the reputation of the misreporting nodes in the Neighbor-Reputation-Table (NRT). Sensor nodes use the information in NRT tables to decide about trustworthiness of a node based on a simple majority voting scheme.

Another prior work [34] presented the idea of estimating trust level of both data and data sources based on various factors such as data similarity, path similarity, data conflict and data deduction. Dai et al. calculate trust scores based on four factors: 1) path similarity 2) data similarity 3) data conflict 4) data deduction [34]. For data similarity, they calculate distance between two numerical values, distance between two categorical values and distance between two string values. However, this scheme will assign a high trust to a reported event if it is reported by multiple, different nodes and fail to identify the collusion attack. In a couple of recent efforts [52, 77], researchers attempted to dynamically evaluate the trustworthiness of the events reported by the data fusion system based on data similarity and path similarity. Firstly, if the same event is reported by multiple nodes, it is more likely to be true. Secondly, if the same event is reported along multiple different paths, that is more likely to be true. Moreover, in the prior schemes [52, 77], trustworthiness and reputation of individual nodes drops faster based on the rationale that reputation should be harder to build and easier to destroy.

Lim et al. also assess trust scores of streaming data based on provenance [75, 76]. Making use of provenance in the calculation of trust in streaming environments has similarities to our approach [76]. They use physical provenance (where the data item was produced) to compute trust. They store provenance in a database and do the trust assessment in a centralized manner, whereas our proposed architecture handles trust computations in a distributed manner. On the other hand in TIBFIT, a trust index based fault tolerance system, they keep a trust index as a quantative measure of fidelity of previous event reports [72]. Their approach is similar to ours in the sense that they keep historical correctness of nodes. However our work is more novel as our notion of trust is not just an error rate but a broader metric calculated using many values stored in reputation and provenance vectors.

IBM T.J. Watson Center has a work on a biomedical data stream system for online healthcare analysis. Their system called Century have medical sensors storing provenance information describing which data and processes contributed to an event in the system [85, 12]. Vijayakumar and Plale have proposed a system capturing provenance data for meteorology forecasting domain [130]. In our previous work we used provenance for network restructuring [39] and assessing trust [52]. The interest of our paper differs from all of the above as we are using provenance locally to detect data faults and to enhance trust using multiple trust metrics.

Bayesian network [135, 136] and game theory techniques [141] are used for building trust in networks [90]. Our model differs from all of the above as we use index based trust model supported by provenance for trust assessment in wireless sensor networks.

One of the most important breaches of sensor networks is that cluster heads can be malicious [30]. Garth et al. propose a distributed trust based framework for election of trustworthy cluster heads. Direct and indirect information coming from trusted nodes is used. Trust

is calculated as the weighted calculation of the packet drop rate, data packets and control packets. Every node is keeping a trust table of the nodes around it and they report to the cluster head upon request. The second-hand information is not used so the bad mouthing effect is prevented.

Considerable amount of work is done on the network trust and information security in general. An agent-based approach using Dempster-Shafer theory is being investigated in prior works [147, 146] to manage the trustworthiness in dynamic information sharing environments and to address the double counting problem in trust evaluation. One of the fundamental assumptions of Dampster-Shafer theory is that the observed evidences are independent and uncorrelated. Unfortunately, this is often not true where the failures are correlated (e.g., collection of malicious nodes report erroneous target together). In general, prior works focus on trust assessment on node level. Unfortunately, in a multihop, collaborative environment, the nodes may produce poor quality results only in certain execution context (e.g., magnetic sensor generates false alarm in windy condition) instead of all the time or due to a bad interaction.

A collusion attack takes place when multiple network nodes provide fake information with the intent of increasing the trust score of an event which is actually not true. Dai et al. proposed an approach to reflect the possibility of such attacks into the trust scores [33]. A majority rule is adopted to distinguish evidence items from malicious colluding parties and correct evidence items. Information trust assessment and node level trust assessment based on path and information similarity is proposed in [133]. A feedback mechanism is presented to adaptively adjust the trust value of nodes based on the information trustworthiness evaluated at the receiver.

Hur et al. has proposed a trust model for assessing trustworthiness of sensor data and to

remove the data from malicious nodes [59]. Their work has many similarities to a work of ours [39]. However their model does not make use of provenance, the historical data. Each node evaluates trustworthiness of its neighbor nodes by crosschecking the neighbor nodes redundant sensing data with its own result. More accurate results are found out by disregarding the data coming from malicious nodes.

Chen et al. [22] propose a reputation-based trust which borrows tools from probability, statistics and mathematics analysis. They have suggested a new term certainty used in trust system and they argued that the positive or negative outcomes for a certain event is not enough information to make a decision in WSNs. They build up a reputation space and trust space in WSNs, and define a transformation from reputation space to trust space. Finally, they discuss some important properties of them and point out some open problems in reputation system in WSNs.

Xiao et al. use a Trust Voting algorithm in their system called SensorRank. A sensor node consults with its neighbors to validate if its reading is true or not [140]. Faulty nodes do not participate in voting algorithm.

Tanachaiwiwat et al. [126] have built a trust routing model (TRANS) for sensor networks. In their model, nodes send probing messages to neighbours and wait for ACK messages. Nodes that are maliciously routing the message or dropping the message are blacklisted by the sink node. Traffic flow is from/to the sink.

One of the models using beta reputation system is Connected Dominating Set (CDS)-based reputation monitoring system by Srinivasan et al. [119]. The nodes obtain direct information about other nodes and store it as a beta distribution parameters tuple.

Momani et al. build a Gaussian Reputation System for WSNs [88]. Each node's reported data is evaluated by its neighbor nodes. They introduce a Bayesian probabilistic approach

for mixing second hand information from neighboring nodes with directly observed information.

RDAT uses a beta reputation system and base station evaluates trustworthiness of nodes based on sensing, routing and aggregation behaviors [99]. Reliability of the system is increased in presence of compromised nodes.

GTMS is a group-based trust management scheme developed by Shaikh et al. [115]. They combine centralized and distributed approaches. Their work has very similarities to our approach. However they do not consider faulty data sent by malicious nodes. Every group has a trust value, which is kept at a small database at the base station.

In Agent Based Trust and Reputation Management Sytem (ATRM) nodes store the trust and reputation information locally [13]. The network model is based on a clustered WSN with backbone where its core is a mobile agent system.

5.3 Related Work on Cognitive Networks

There is also some work on cognitive networks. Traditional networks only deal with amount of data transmitted however cognitive networks also deal with content of the information delivered. It is closely related to provenance in the sense that provenance can keep information about the content [86]. In cognitive networks, elements in networks have states that are changing based on the content of the information received. This idea is close to our idea of nodes in a network that are in specific states at a given time and are behaving according to a Finite State Machine [86]. A cognitive model takes the data and converts it into intelligent information. Apart from the provenance research, there have been many ideas of increasing the intelligence within a multihop network. Intelligence can mean a range of behaviors from a sensor that turns on a light to much more complicated

computing and actions. We cannot relate all possible uses of the term here, we use it in a broader sense meaning the capability of the network to provide an immediate and detailed data trust. There are several research threads that can be differentiated from the use in our architecture. One important common theme in making intelligent decisions within a network has been to better balance the traffic. Kelly provided a technique that makes use of local knowledge at a node to improve the traffic flow versus link capacity within the network [65]. Heo and Varshney made use of mobility to better position sensors in an area to improve coverage and energy efficiency [56]. Close to the ideas presented in our paper, Zahedi et al. have considered a two-tiered fault detection system for a sensor field that is collecting information [148]. Fusion node for a group of sensors weighs the usefulness of the inputs based on how accurate the result is compared to its likeliness for a misbehaved value. Our model is broader than the approaches listed as it is a general architecture applicable to different wireless network types. It is also more powerful as it is making use of provenance to create a distributed intelligence. Our approach is also novel in the sense that while storing rich trust and provenance information in vectors, we transmit one trust value over the network conserving network bandwidth utilization and reducing energy consumption. In addition, the two way communication (push and pull) between intermediate node and its children makes it possible to have an up-to-date trust picture of the network.

Chapter 6

Main Architecture

6.1 Definitions

Before describing the architecture, we will define the terms that are used throughout the thesis.

Trust: The trust of a node N, is the probability that N sends correct information. We call data trusted if its trust value is above a threshold value. For example, in a forest fire detection network, information reporting the correct coordinates of a fire within some tolerance can be trusted.

Reputation vector of a node: Every node has a reputation vector consisting of values that are used in trust calculation. Based on the domain of the network, different parameters might be needed for trust calculation and our architecture is flexible enough to support various parameters. For instance, for a wireless sensor network, minimum energy required can be a needed parameter in trust calculation. If the left energy is less than this value, the data created by this node can be given a low trust value.

Provenance vector of a node: Every node also has a provenance vector consisting of data

such as node id, group id, information on how much resources left in the node (e.g. battery life), average of the historically reported data, node location, etc. Some of this may be in the form of a statistical measure (mean or standard deviation). There is a relation between the fields of provenance vector and reputation vector.

Untrusted node: A node is named as untrusted if it has a trust value attached to its data lower than the threshold. There can be many causes for a low trust value. For instance, signal-to-noise ratio value can be higher than a threshold for the node it is sending data to. Another example is data value created by that node can be inconsistent with the data coming from other nodes.

Network restructuring: This refers to actions that change the network structure such as omitting or replacing a node, merging two groups, moving a receiving node to another group.

Trust assessment: In our model, trust assessment is done locally both at the node-level and the group-level. We assess trust based on vectors kept at the nodes. Group-level trust assessment is done based on similarity of information received from multiple nodes by an intermediate node.

6.2 Overview of ProTru

A possible scenario to show why we need to enhance trust in wireless sensor networks is as follows. In a temperature monitoring sensor network, many low cost sensors are used in order to measure temperature of the area. After the initial deployment of the network, since energy is drained from nodes, the nodes may die or start sending weak signals which results in a misreport. These nodes might not be trusted as others with better transmissions. This is only one example of a low trust value, however there can be other reasons for nodes

to have low trust values. For example, a node that is far away from the intermediate node can be considered untrusted due to possible noise during the transmission. In a traditional network, to take the decision of which nodes should be less trusted more provenance data needs to be transmitted to the central node and the decision could be too slow for comfort. However, in the architecture we are proposing, responsible intermediate node will detect the untrusted node by examining the trust value associated with the data coming from that node. Based on the actions defined in the algorithm housed in the intermediate node, the nodes with low trust values could be given lower weight or even may be omitted or replaced. As untrusted nodes are taken care of locally, data with higher trust values would be transmitted forward.

In our model, every intermediate node is aware of trust reputation of its neighbors which is an important characteristic for network integrity. Nodes in networks continuously perform same tasks and they should be consistent and cooperative. Being history-aware is one of the superior characteristics of our architecture. Furthermore, by keeping provenance and reputation information in local vectors at nodes, we decrease the data overhead marginally. In our network, the trust value which is a number computed based on these vectors is forwarded hence bandwidth is used efficiently and less transmission energy is consumed. One other superiority of our architecture is that it makes use of second hand information. Nodes do not adjust the trust of their data based on only their observations but they exchange information with intermediate nodes. Intermediate nodes oversee the whole group of nodes thus the second hand information exchanged between an intermediate node and a leaf node is the product of data and trust produced by every single node in the group.

Overwhelming the network by forwarding provenance information is avoided by passing only (data,trust) tuples. A parent node receives many (data,trust) tuples from its children, it

may take many possible actions based on the finite state machine housed in it. For example, after receiving the tuples, parent node can decide to remove a child with a low trust value from its communication path. Each of these possibilities will be discussed further below.

6.3 Architecture

We present a dataflow-oriented provenance model for wireless sensor networks that makes use of node-level trust-enhancing mechanisms. Our three level architecture consists of a graph that contains stationary leaf nodes, cluster heads and a base station that receives information from all sources. Our architecture works both with streaming environments and triggered node networks.

Below is the description of our architecture:

- *Leaf Node* : The source node (identified by a unique id) gathering data and triggering the network in case of an event e.g. data arrival.

- *Intermediate Node (Cluster Head)* : Computationally more powerful nodes receiving information from a group of nodes, doing calculations on the received information such as fusing, and transmitting the information for the group forward. The cluster-head aggregates the data and sends it to the base station or to the higher level cluster-head.

- *Base Station (Central Node)* : Top level of hierarchy which is a central station receiving values from intermediate nodes and calculating the final value. Base Station has a constant energy supply and has no energy constraints.

Nodes in the network are organized into clusters based on the algorithm that is briefly described below. The picture of the network topology is given in Figure 6.4. Each cluster

has a cluster head which is aware of cluster topology. Nodes in a cluster are within one communication hop of the intermediate node. Intermediate nodes transmit directly to base station.

6.4 Clustering Approach

We assume that N nodes are dispersed in a field and the assumptions in the above section hold. We will use the protocol HEED for clustering the network [145]. We will not present the details of the algorithm. Firstly, cluster heads are selected among the more powerful nodes in the network. Cluster heads communicate with nodes in their clusters (intra-cluster coordination) and they communicate with each other (inter-cluster communication). Each node v_i, where $1 \leq$ i $\leq$ N is then mapped to exactly one cluster c_j, where $1 \leq$ j $\leq N_C$, and N_C is the number of clusters ($N_C \leq$ N). After this step, nodes are assigned to clusters such that every node can communicate to its cluster head via a single hop.
The following requirements will be met by the HEED algorithm [145]:

1. Clustering is completely distributed. Each node independently makes its decisions based on local information.

2. Clustering terminates within a fixed number of iterations (regardless of network diameter).

3. Each node is either a cluster head, or a non-head node (which we refer to as regular node) that belongs to exactly one cluster.

4. Clustering should be efficient in terms of processing complexity and message exchange.

5. Cluster heads are well-distributed over the sensor field.

Messages from cluster heads are routed to a base station. Inter-cluster routing is used for clusters that are more than one hop away from the base station.

6.5 Network Model

We assume that a set of sensors are dispersed on an m*m operational area. Our network has the following properties:

- The nodes in the network are stationary.
- Nodes are location-aware, i.e. equipped with a GPS-capable antenna.
- Central (base) station has much more computational power, energy and communication resources.
- Nodes are left unattended after deployment.
- Cluster heads and base station are non-faulty.

Our motivation for this work is the fact that we have to consider trust values and restructure the network to enhance trust because values in reputation and provenance vectors can change over time.

Let L be the set of all leaf nodes and l_i be a leaf node in set L. As stated earlier a leaf node could be a sensor in the field, a text gathering node, etc. The nodes in the first level that collect information from the leaf nodes (cluster heads) are named as N_j and the node that collect information from the first level nodes is base station. All nodes in our network as well as leaf nodes have vectors of reputation and trust. $trust_{accuracy}$ and $trust_{freshness}$

values computed using the values in the vectors are forwarded along with the data while provenance and reputation vectors are kept at the nodes.

When l_i obtains a signal of a measurement, that value may not be a perfect value for many reasons. For instance, if it is a sensor network, Zahedi et al. characterize a sensor measurement to be in one of several states including normal, noisy, spike, frozen, saturation, bias, spike, oscillation [148]. l_i computes its best estimate of the true value and assigns a trust value to it. It will output a data value and its trust $(d_i, t_{accuracyi}, t_{freshnessi})$ onto the cluster head N_j. The accuracy of information is improved by making use of reputation and trust vectors to create a trust value that is sent along with each data item.

$(d_i, t_{accuracyi}, t_{freshnessi})$ tuple is sent to the cluster head N_j. Thus, at the receiving end of the network, a trust value, which may be modified along the path from source to destination, is received along with the data item. The fusion node uses its reputation and provenance vectors to first determine its revised trust value. If a value is unexpectedly different than the other received values, the fusion node may take one of several actions. Some possible actions can be as follows. Less weight can be given to that data value during computation and fusion. A message can be sent back to the node that sent the unexpected value to revise a parameter such as its state, a value in its trust vector. The fusion node may wake up a sleeping node and/or put the node with questionable data to sleep. Besides these actions, the cluster head calculates its trust in its fused/computed value before transmitting the new (data, trust) tuple to a base station. Finally, base station fuses all the data sent by intermediate nodes and its input becomes intelligence for the user.

Leaf Nodes

Leaf nodes collect and then disseminate information but do not receive information from other nodes.

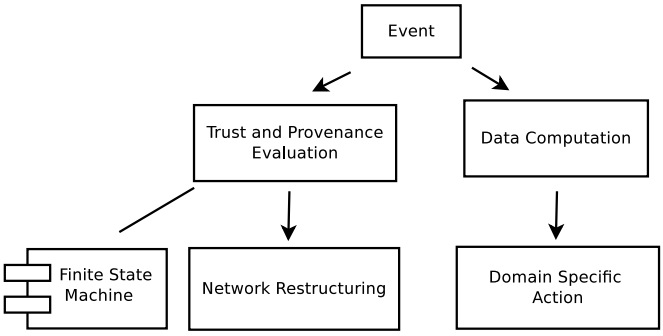

Figure 6.1: Main picture of the tool

$$\rho(\vec{t}) = \begin{bmatrix} \text{node id} \\ \text{group id} \\ \text{message id} \\ \text{creation time} \\ \text{state of the node} \\ \text{left energy} \\ \text{node locx} \\ \text{node locy} \\ \text{average error} \\ \text{mean} \end{bmatrix}, \tau(\vec{t}) = \begin{bmatrix} \text{acceptable value interval} \\ \text{minimum energy required} \\ \text{accuracy threshold} \\ \text{freshness threshold} \\ \text{v accuracy} \\ \text{v freshness} \end{bmatrix}$$

Figure 6.2: Example provenance and reputation vectors

Intermediate Nodes

Intermediate nodes are identified by a unique id and they are the leaders of a group of leaves. They take decisions related to the group such as increasing or decreasing the trust value of a leaf node, omitting a node. Groups can be formed in several ways depending on the domain that the network is deployed. For instance, they can be formed based on locations so that every node has to send data to a short distance reducing overall energy consumption or they can be formed based on the type of nodes such as camera nodes forming one group.

Intermediate nodes receive a trust value along with the data. They do computations, for example, fusion, on the received data and calculate the trust value or values of the data using the received value. They pass on the data and its calculated trust value up the stream. Thus instead of a network where data and provenance vectors travel the whole path from the source to the destination, nodes themselves contain sufficient provenance information preventing excessive transmission overhead of provenance information. This fusion and merging process also serves as a data filtering according to node credentials. For example, central node does not receive unnecessarily detailed data. Besides as intermediate nodes are spread over to the network, computations are done in a distributed manner. This is where the distributed computational nature of our architecture comes from.

Intermediate nodes receive two trust values along with the data; $trust_{accuracy}$ and $trust_{freshness}$. They do computations such as fusion, on the received data and calculate the adjusted trust values of the data. They pass on the data and its calculated trust value up the stream. Thus instead of a network where data and provenance vectors travel the whole path from the source to the destination, nodes themselves contain sufficient provenance information preventing the excessive transmission overhead of provenance information.

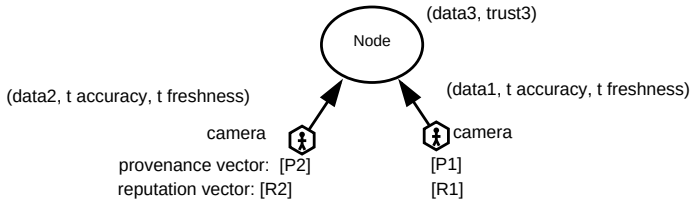

Figure 6.3: Fusion in intermediate node

As the information is computed and fused, new trust values are generated by the intermediate node as the result of comparisons of several (data,$t_{accuracy}$,$t_{freshness}$) tuples received. For instance, an intermediate node can fuse image data coming from two cameras, let's say $camera_1$ sends the tuple (i_1, $t_{accuracy1}$,$t_{freshness1}$) and $camera_2$ sends the tuple (i_2,$t_{accuracy2}$,$t_{freshness2}$) with trust values very close to each other as illustrated in Figure 6.3. However the fusion node might realize that quality of i_2 is much better than i_1 and it can send a *decrease your trust value* message to the $camera_1$.

Central Node (Base Station)

Intermediate nodes will send computed and fused data and the corresponding trust value to central node. Due to distributed nature of our architecture, a central node does not make decisions; it calculates the final result by taking the weighted average of incoming (data,trust) tuples.

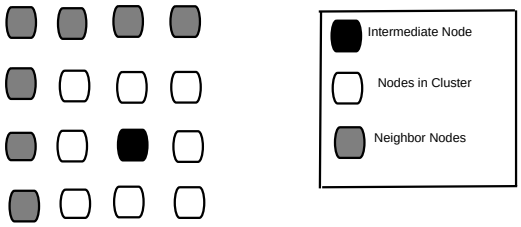

Figure 6.4: Network topology

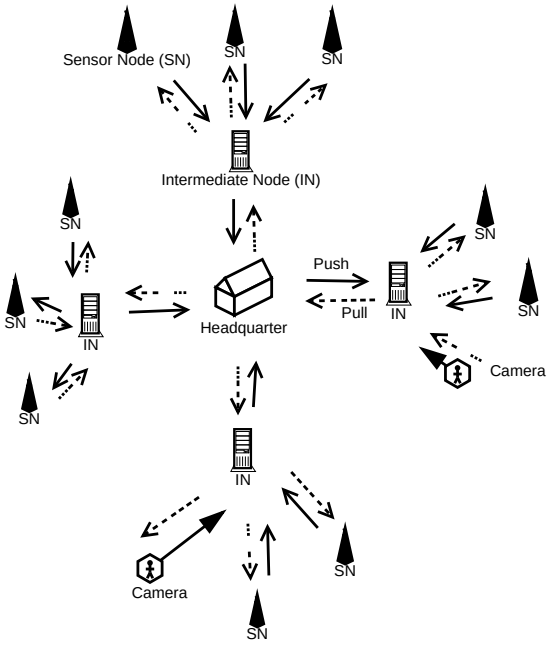

Figure 6.5: Network architecture

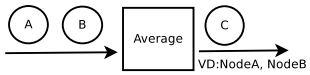

Figure 6.6: Node C depends on values of node B and node A

6.6 Provenance Model

Provenance is used to take network snapshots at time intervals. As we keep node id and belonged group id, we can track the dataflow in the network. When doing network restructuring, knowing the network picture at that time interval is very helpful. For instance, if the trust value of the group decreases below a threshold, intermediate node can decide to merge its group with a trusted group. Another example can be if the trust of a node is less than the threshold, the intermediate node will add a node with a high trust value to its group. To decide which node to add, it will analyze the network picture at the time. It will pick the most trusted node in the closest group.

We will use Open Provenance Model to model dataflow provenance in our network. The Open Provenance Model has been developed as a standard to facilitate provenance interoperability. In it, nodes represent objects, edges represent information flow between the source object (ancestor) and the destination object as illustrated in Figure 6.7. There are five predefined causal relationships *used, wasGeneratedBy, wasTriggeredBy, wasDerivedFrom, wasControlledBy* and edges are labeled with one of these causal relationships.

Provenance information of "on which leaf nodes does data depend" is referred as data dependency [28]. In our model, value dependencies will be stored to capture the network snapshot. An example of a value dependency is *"Intermediate node C depends on data coming from node A and B"* as shown in Figure 6.9. The node id and group id information in the provenance vector in Figure 6.9 form the data dependency.

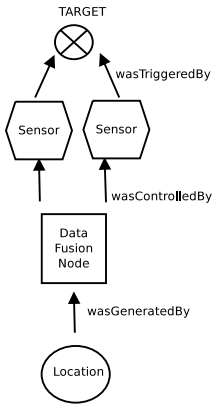

Figure 6.7: An example of a provenance graph

Provenance vector at the leaf node keeps these values:

- **node id:** Each leaf node is given a node id.
- **group id:** Each intermediate node is given an id.
- **message id:** Each message is given an id.
- **state of the node:** State of the leaf node such as sleeping, awake, transmitting.
- **creation time :** Time the data is sensed (created).
- **left energy:** Energy left at the nodes.
- **node loc x:** x coordinate of the location of the node.
- **node loc y:** y coordinate of the location of the node.
- **average error:** Average of the error in past data. This value is set to 0 at deployment because network is unaware of the right value, it is updated after first run.

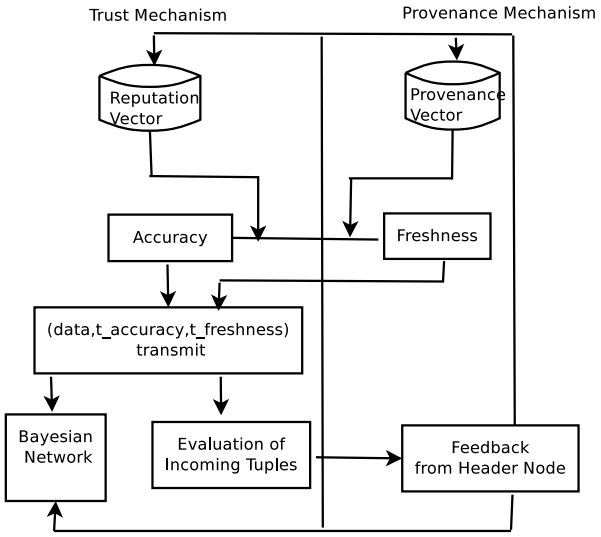

Figure 6.8: Functionality of the reputation and provenance mechanism

- **mean:** Average of the past data. After first time interval, the forwarded data value is copied into this vector entry. As data is sensed, this entry is being updated.

6.7 Trust Model

Trust management methods for ad-hoc networks can be classified into two broad categories: 1) certificate-based 2) behavior-based. In certificate-based models, trust is built based on the provision of a valid certificate by an authority or by other nodes. In behaviour-based models, a node monitors the behaviors of its neighbors in order to decide whether or not to trust them [41]. In our model intermediate node is monitoring the transmitted (data, $t_{accuracy}$, $t_{freshness}$) tuples by the leaf nodes over time, this makes our model behavior-based.

Trust History: Sensor nodes should be aware of trust history of themselves and their

neighbourhood [41]. This is done through provenance by keeping the $v_{accuracy}$ value in the provenance vector.

Historical Data: We keep bias and average of the data of each node for previous time intervals in provenance vector.

We characterize trust through attaching a decimal value between 0 and 1 to the data. In our model, the higher trust value is, more trusted the data is. Data trust depends on the source of the information. Trust value is assigned by inspecting the provenance and reputation vectors kept at the node. Nodes are first given some initial values of trust at the deployment time. Later these values are updated by the node itself and by the intermediate node of their group.

6.7.1 Trust Metrics

Trust is context-specific (audio, video, binary) and multi-faceted (freshness, accuracy) and dynamic (increase or decrease with experience) [136]. As trust is a multi-faceted value, trust models have to support multiple trust metrics such as accuracy, freshness, speed, consistency, security. Intermediate node may be interested in various aspects of trust such as it might want to know whether the node sends timely data with good quality which involves the node's capabilities in two aspects, quality and freshness. For some applications one trust metric is not sufficient for deciding about trust of the data. For instance, a fire detection wireless sensor network has to be very timely in reporting a fire otherwise it might be too late to stop the fire. In such a system, both data accuracy and data freshness should also be taken into account to make sure a node is not resending old, irrelevant data.

In our model for the sake of simplicity, we are considering only two trust metrics; data

accuracy and data freshness. However our model is flexible enough to support different trust metrics. Data can be 1) Bad 2) Good based on how accurate it is and it can be 1) Stale 2) New based on how fresh it is.

6.7.2 Reputation Mechanism

Reputation is a peer's belief in another peer's capabilities based on recommendations received from other peers [136]. In our model, we make use of both trust and reputation. When intermediate node collects data and trust tuples, it decides about trust of a leaf node by comparing its data and trust values with other tuples. This decision becomes reputation in our system and our reputation mechanism is built with the feedback coming from intermediate nodes.

Reputation vector keeps these values:

- **temp interval lower:** The lowest temperature value a leaf node can report.
- **temp interval upper:** The highest temperature value a leaf node can report.
- **min energy required:** The minimum energy value leaf nodes have to have in order to transmit.
- **accuracy threshold:** The threshold used for accuracy check at leaf nodes. If the $t_{accuracy}$ value is less than this threshold, leaf node does not send the data.
- **freshness threshold:** The threshold used for freshness check at leaf nodes. If the $t_{freshness}$ value is less than this threshold, leaf node does not send the data.
- $v_{accuracy}$**:** It is decreased or increased based on the error rate in the reading and used for $t_{accuracy}$ calculation. This value works as a placeholder for historical trust data.

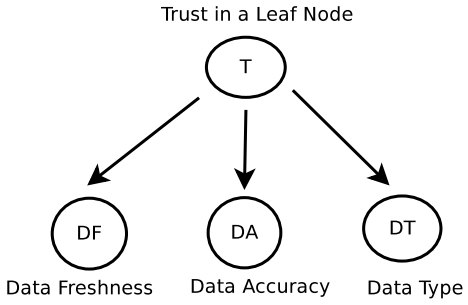

Figure 6.9: Trust dependency model

- $v_{freshness}$**:** It is the difference between the data creation time and data collection time as given in Equation 6.8.1.

6.8 Framework for Trust Enhancement

We propose a framework for trust assessment and enhancement based on data fusion and provenance. We use a probability-based trust model and weighting approach which weights both direct and indirect information (both good and bad). We describe efficient algorithms for data fusion, trust enhancement and network restructuring.

Main components of our framework can be listed as follows:

At leaf nodes:

- trust value computation
- data forwarding
- updating vectors

At intermediate nodes:

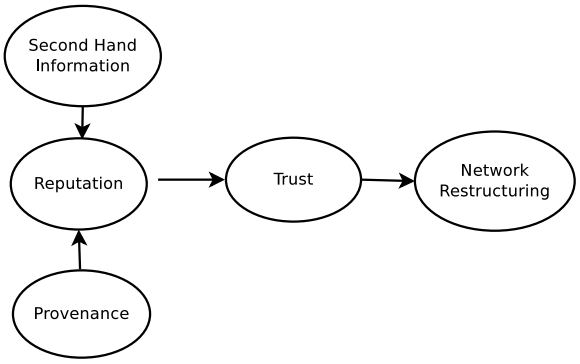

Figure 6.10: Trust mechanism of the tool

Figure 6.11: Event chain at the leaf node

- data fusion
- trust enhancement
- network restructuring
- transmitting fused data to central node

As illustrated in Figure 6.8 we make use of Second Hand Information which is the feedback coming from Intermediate nodes, the reputation information kept as a vector and

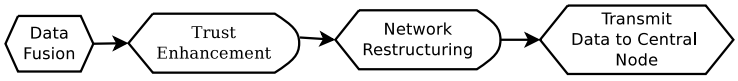

Figure 6.12: Event chain at the intermediate node

provenance data in computing trust. Based on the computed trust values, network restructuring is done as explained in below.

Provenance vectors keep historical data in the form of error and mean for accuracy. Error value keeps track of the error rate. After the first iteration by comparing the data values received, network will have a guess of the true value and the error value will be updated accordingly. Every group will have a guess of the right value. Intermediate node will send the fused and averaged data value to the leaf nodes and leaf nodes will update their error values accordingly.

Mean keeps the weighted average historical data. While calculating the trust value, these two values are used. Moreover it becomes easy to detect if the node sends faulty data by comparing the data to historical data at the leaf node level. This becomes an early check at the leaf node level before data is transmitted to the intermediate node increasing distributed local nature of our network. When the network is deployed, these values are set to null but they are updated after each transmission. he values in vectors are changing in time. Here are the situations when the vectors change:

- When a message is sent, the energy of the node is decreased by a constant value.
- v value changes as leaf node finds out that the data is very different than the average of the past data or the error is very big compared to average error.
- v value changes as a message is received from intermediate node when intermediate node finds out that node is faulty based on FINDDISSIMILAR algorithm.

6.8.1 Freshness Adjustment at the Leaf Nodes

We make use of synchronized time in our system for assessing data freshness. We assume that time is consistent and synchronized throughout the network to calculate data freshness. Intermediate node is waiting for all leaf nodes to transmit their reading at time $time_1$. We assume that all trusted leaf nodes transmit at the specified time interval. However based on the length of the time interval, they have different freshness values depending on the creation time of the data. For instance, if the cameras are sending pictures every 10 minutes, an image that was taken at the first minute of the 10 minute interval is fresher than the image that was taken at the fifth minute of the 10 minute interval. At time $time_1$, we assume that we have a non-empty set of data and trust values coming from trusted nodes.

$$v_{freshness} = time_{collection} - time_{creation} \tag{6.8.1}$$

where

- $time_{collection}$: The time Intermediate Node pulls data from leaf nodes. This value is not stored anywhere, it is read from the node's clock once the pull message is received from the Intermediate Node.
- $time_{creation}$: The time leaf node creates data. It is stored in provenance graph and is overwritten every time data is created.

$$t_{freshness} = e^{-\lambda v_{freshness}} \tag{6.8.2}$$

where

- $t_{freshness}$: is the $trust_{freshness}$
- λ : is a proportionality constant that is application dependent. It represents forgetfulness in the system. It determines how fast the system should remember past experiences.
- $v_{freshness}$: is a value kept at reputation vector that is used for calculating $t_{freshness}$. It is always greater than 0.

if ($t_{freshness} > \epsilon_{freshness}$)
data is new, send
else data is stale, ignore

6.8.2 Accuracy Adjustment at the Leaf Nodes I

In our model, trust is adjusted both locally (at the leaf nodes) and distributedly (by intermediate nodes). $t_{freshness}$ is a trust value that is calculated based on the creation time of the data hence there is no need to refer to historical behaviour for $t_{freshness}$. If the data is consistent with the previously created data (the average of previous data) then it is given a greater $t_{accuracy}$ value. If the data differs from the mean of the historical data within an allowed bound mean $\pm$ interval*σ, then it is considered good data. If data is good, $v_{accuracy}$ is decremented resulting in a higher $t_{accuracy}$ else it is incremented resulting in a lower $t_{accuracy}$ based on equations 6.8.3 and 6.8.4. Below is the algorithm used in deciding if the data is bad or good at the leaf nodes:

if ((mean-interval*σ) <data <(mean+interval*σ))
then data is good
else data is bad

where

- σ is a percent. The bigger it is, the system is more tolerable to errors.

$trust_{accuracy}$ value is calculated using $v_{accuracy}$ value kept at Reputation Vector of each node. It is decremented and incremented exponentially. We use a similar method to the Krasniewski et al.'s trust index calculation method in adjusting $trust_{accuracy}$ value [72]. We use the equation 6.8.3 in calculating f_r value which is used in $v_{accuracy}$ adjustment. $v_{accuracy}$ value is used in Equation 6.8.4 for $t_{accuracy}$ calculation.

$$f_r = \frac{|mean - data_{current}|}{interval} \tag{6.8.3}$$

where

- $data_{current}$ is the current data value of the leaf node.
- mean is the historical average data kept at the provenance vector of the node.
- interval is the possible data interval length.

if (bad data) $v_{accuracy}$ = $v_{accuracy}$ + f_r* δ
if ((good data) $\wedge$ ($v_{accuracy}$ >0)) $v_{accuracy}$ = $v_{accuracy}$ - f_r* δ
if ((good data) $\wedge$ ($v_{accuracy}$ == 0)) $v_{accuracy}$ = $v_{accuracy}$

$$t_{accuracy} = e^{-\lambda v_{accuracy}} \tag{6.8.4}$$

where

- $t_{accuracy}$: is the $trust_{accuracy}$
- λ : is a proportionality constant that is application dependent. It represents forgetfulness in the system. It determines how fast the system should remember past experiences.
- $v_{accuracy}$: is a value kept at reputation vector that is used for calculating $t_{accuracy}$. It is always greater than 0.

if (($t_{accuracy} > \epsilon_{accuracy}$) $\wedge$ (left energy $>$ energy required))
send data else ignore data

6.8.3 Failure Model at the Intermediate Node

At every time snapshot we have two categories of sensing nodes 1) Nodes sending correct data (inliers) 2) Nodes sending faulty data (outliers). Correct nodes are not accurate %100 however they make errors within a specified low bound (less than a threshold). We assume that intermediate nodes are reliable %100 of the time.

Although leaf nodes have some initiative for deciding about qualities of their data, intermediate node makes the final decision about data being accurate and fresh by making comparisons of data similarity during fusion. The result of the evaluation of intermediate node, "data is similar" (inlier) or "data is dissimilar"(outlier) is used to update the leaf node's trust. After making the decision at each transmission snapshot, by giving feedback to the leaf nodes, intermediate node adjusts the vectors of untrusted leaf nodes.

Overhead of data collection process should be as lightweight as possible [41]. For this purpose, we keep provenance and reputation information locally at the vectors in the nodes and we only transmit two decimal values $t_{accuracy}$, $t_{freshness}$ with the data. Our aim is

$$\vec{\varrho} = \begin{bmatrix} \text{trust threshold } \beta \\ \text{state of the intermediate node} \\ w_{accuracy} \text{ for image data} \\ w_{accuracy} \text{ for numeric data} \\ w_{freshness} \text{ for image data} \\ w_{freshness} \text{ for numeric data} \end{bmatrix}$$

Figure 6.13: Provenance vector kept at intermediate node

to conduct error detection using only local information during sensor fusion without increasing the communication overhead. Intermediate node is waiting for all leaf nodes to transmit their reading at time $time_1$. At time $time_1$, we have a non-empty set of data and trust values. Trust of the received $(d_i, t_{accuracyi}, t_{freshnessi})$ tuples is calculated using the Equation 6.8.6 below.

$$\begin{gathered} (d_1, d_2, d_3, ..., d_n) \\ (t_1, t_2, t_3, ..., t_n) \end{gathered} \tag{6.8.5}$$

$$\begin{aligned} t_i = t^i_{accuracy} * w_{accuracy} + t^i_{freshness} * w_{freshness} \\ w_{accuracy} + w_{freshness} = 1 \end{aligned} \tag{6.8.6}$$

where

- $t_{accuracy}$: is the $trust_{accuracy}$
- $t_{freshness}$: is the $trust_{freshness}$
- $w_{accuracy}$: is the weight value used in calculating trust of a data item.
- $w_{freshness}$: is the weight value used in calculating trust of a data item.

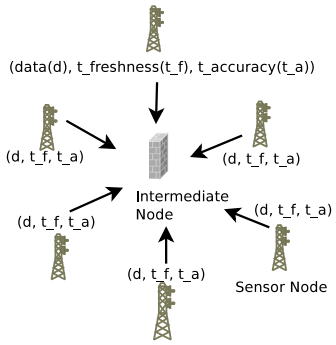

Figure 6.14: Intermediate node receives data from leaf nodes

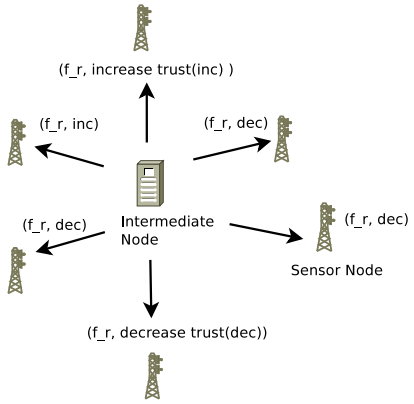

Figure 6.15: Intermediate node sends message to leaf nodes

After finding t_i values for all data using the Equation 6.8.6, intermediate node drops the data that has $\bar{t}$ value less than ϵ.

if $t_i < \beta$ drop the data
else fuse

6.8.4 Outlier Detection

After calculating t_i values for incoming data, intermediate node finds the outliers. At the intermediate node, a dissimilar value is found and the $v_{accuracy}$ value of the node which sends the dissimilar value is increased (decreasing $t_{accuracy}$) so that next time either it does not send any data or its data is given less weight. The difference of our approach than the previous research is that our algorithm will not solely based on data dissimilarity, it will also consider trust and provenance. Besides it is different than traditional outlier detection approaches because instead of pure data values we consider data*trust values. For instance, a node which has dissimilar data can be given more importance because it has a greater trust value.

As we want to consider trust values in addition to data values, we take the weighted average of the tuples. The weighted average equation is as follows:

$$\bar{d} = \frac{\sum_{i=1}^{n} d_i.t_i}{\sum_{i=1}^{n} t_i} \tag{6.8.7}$$

where

- $\bar{d}$ is the aggregated data value.
- d_i is the data value reported by $node_i$
- t_i is the trust value of data d_i calculated by intermediate node using the $t_{accuracy}$ and $t_{freshness}$ values reported by $node_i$

FINDDISSIMILAR algorithm which we gave below the pseudocode for works as follows. Between the values received, the weighted outlier values are selected. Weighted average of

the other remaining values are calculated. Depending on the data set many different algorithms can be used for finding the dissimilar data. Our model is flexible enough to support different outlier detection techniques. We use RANSAC algorithm that is developed by Fischler et al. [42]. RANSAC depends on the idea that captured data is interpreted in terms of a predefined model. In our case, the model is a line that is parallel to y-axis if the (d,t) values are plotted on a 2D graph. The inliers will have very close d values with varying t values creating a vertical line as illustrated in Figure 6.16 whereas the outliers will occur away from the line created by inliers.

FINDDISSIMILAR Algorithm:

Input: (d_i, t_i) tuples coming from leaf nodes n_i

Output: Indices of the outlier nodes $index_{out1}$,.., $index_{outn}$

begin

$(index_{outlier1}, .., index_{outliern}) = ransac(d_1 * t_1, ..., d_n * t_n)$

comment: excluding the outlier tuples in data fusion;

$\bar{x} = \frac{\sum_{i=1, i \neq out1,..,outn}^{n} d_i.t_i}{\sum_{i=1, i \neq out1,..,outn}^{n} t_i}$

end

$$f_r = \frac{|\bar{x} - (d_i * t_i)|}{interval} \tag{6.8.8}$$

where

- d_i is the data value of the outlier node.
- t_i is the trust value of the outlier node.
- interval is the possible data interval length.

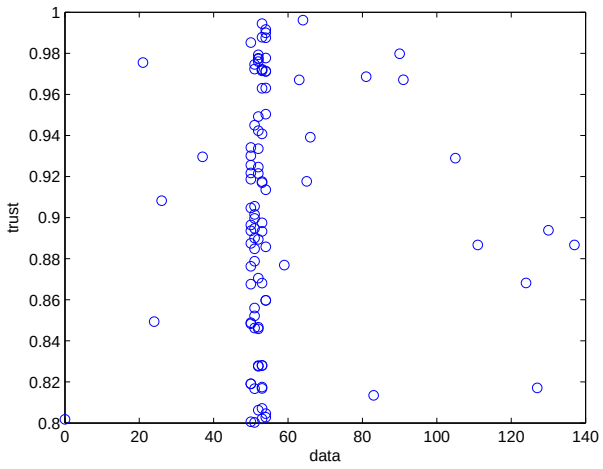

Figure 6.16: Data pattern fitting to line model in RANSAC algorithm

6.8.5 Data Fusion

Multi sensor data fusion has many advantages over single source data such as increased accuracy [36]. In our model, data from several sensors is combined at intermediate nodes where data fusion takes place. In a forest fire detection network, there are many sensors detecting the fire, in this case it will be less definite where exactly the fire is. In our model, fire detection data coming from sensors will be fused in intermediate nodes and more trusted results will be produced.

One significant innovation of our research related to data fusion is that the fusing scenario is dynamic. The group of sensors that are being fused are changing as the network is self-adjusting using the historical provenance data. On the other hand data fusion helps in getting more accurate results compared to a model where all data is collected at a central node without any intelligence.

The intermediate node calculates weighted average of all tuples excluding the outlier tuple

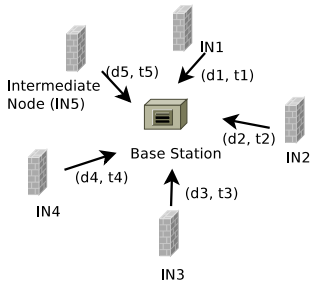

Figure 6.17: Intermediate nodes transmit to base station

based on Equation 6.8.8. Intermediate nodes use weighted average instead of arithmetic average because in arithmetic mean all data contribute equally to the result. However in weighted mean, data with higher trust values contribute to the result more than data with lower trust values. Intermediate node transmits the $\bar{d}$ value to the Central Node. Central Node takes the arithmetic average of the incoming $\bar{d}$ values and it becomes the calculated temperature value for the system.

if data is inlier
data is good, fuse data and send increase trust message
else data is bad
ignore data and send decrease trust message

Different weights are given to $t_{accuracy}$ and $t_{freshness}$ based on the data type during data fusion. The weights are kept in provenance vector of the intermediate node. For example, for a fire detection system, data freshness might be more important than data accuracy for image data since the fire can be detected from the image arriving timely although it is not accurate e.g. it has some corrupted pixels. However it will be totally useless if the image arrives half an hour later fire started.

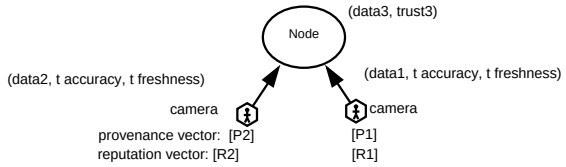

Figure 6.18: Fusion in intermediate node

6.8.6 Accuracy Adjustment at Leaf Nodes II

After finding out outlier nodes, intermediate node sends a decrease your trust message to outlier nodes and increase your trust message to inlier nodes along with the f_r value. $v_{accuracy}$ is adjusted at the leaf nodes based on equation below.

if (bad data) $v_{accuracy}$ = $v_{accuracy}$ + f_r* δ

if (good data $\wedge$ $v_{accuracy}$ >0) $v_{accuracy}$ = $v_{accuracy}$ - f_r* δ

if (good data $\wedge$ $v_{accuracy}$ = 0) $v_{accuracy}$ = $v_{accuracy}$

6.8.7 Network Restructuring

The restructuring works as illustrated in the flowchart in Figure 6.19. The network does re-clustering when a group's trust value is under a threshold. The clustering algorithm HEED that we are using supports efficient re-clustering [145]. Using this procedure network re-structures itself until it comes to a point where local arrangements cannot be done such that all intermediate nodes have computed trust values less than threshold. After this point central node is notified and central intervention will be done such as a new deployment, new resource allocation.

RESTRUCTURENETWORK Algorithm:

Input: tper, percentage of trusted nodes in group

Input: per_1, per_2 the boundary percentages

Output: restructured group

begin

if $(per_2 < tper < per_1)$ **then call** $recluster$ **fi**;

if $(tper < per_2)$ **then call** $omitgroup$ **fi**;

end

6.9 Logical Framework of ProTru

Distributed Intelligence refers to a system of entities working together to reason, plan and solve problems. Use of distributive intelligence is increasing in many domains such as automotive industry, robotic systems, gaming technologies. Most information networks have centralized intelligence e.g. a headquarters, a central station, however distributed intelligence is superior in many ways. Our work is unique in using provenance to have distributed intelligence. As intermediate nodes make decisions, our network has distributed intelligence. When healing process is done in a distributed manner in intermediate nodes, it is faster and more efficient compared to the centralized approach.

Information and its trust value flow up in the network to improve cognitive decisions of the nodes. A new trust value for a node may flow down the network; so it can control information. There are various decisions based on different conditions. For example, an intermediate node receiving data from a group of sensor nodes can make decisions about which nodes to wake up based on the incoming data and trust. If the received data is not sufficient to carry out the computations, it can simply send a *wake-up* signal to some of the

sleeping child nodes.

One possible simple intelligence structure that is developed further below is to have decisions made in an intermediate node based on a finite state machine housed in the node. Under this scheme, data items, trust values from leaf nodes are collected by an intermediate node. The possible actions that can be taken will be determined by the Finite State Machine in Figure 6.19. As Finite State Machine clearly show, our architecture makes use of provenance to support network restructuring and to improve its information gathering.

In a self-adjusting information network, dataflow produces more accurate results and with these improvements, network specific tasks can be done more quickly and more precisely. Besides trust is enhanced in our architecture by the network restructuring that takes place. Every intermediate node observes its leaf nodes' trust values and maintain the whole group's trust value bigger than a threshold. Networks with our architecture will create more trusted results.

6.10 Attack Model

When the information is forwarded to a cluster head, some attackers may refuse to attach their trust value to their data so that they can get away from the malicious node detection and at the same time they can create considerable damage in the decision making.

We assume that every node should sign the (data, trust) tuple with its private key when they attach the trust value to the data. When the other node receives the information it can read the meta data and find out who has sent the information from the node ID which is in the meta data. Now the other node can verify the integrity of the meta by verifying the signature using the corresponding public key of the previous node. Here the node cannot alter the meta data or trust value of the previous node because if they alter the meta data

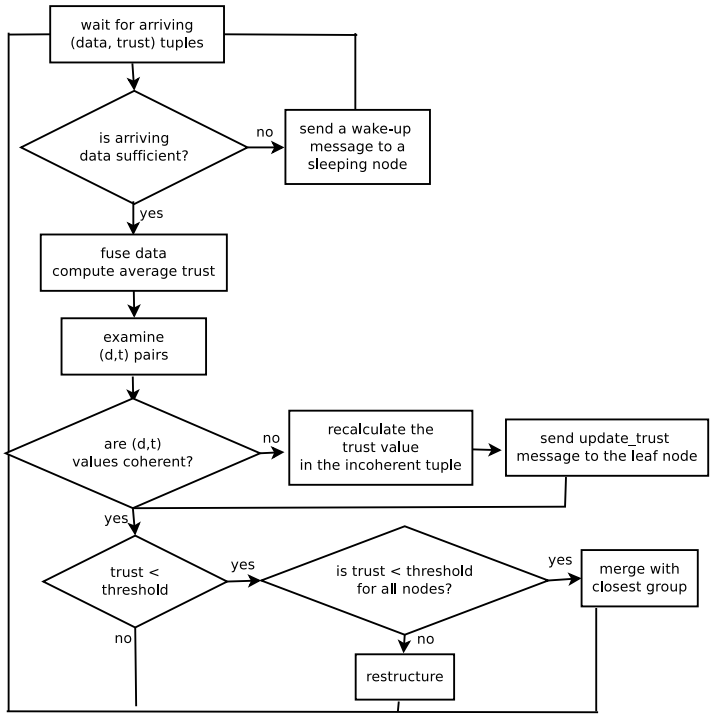

Figure 6.19: Finite state machine housed in intermediate node

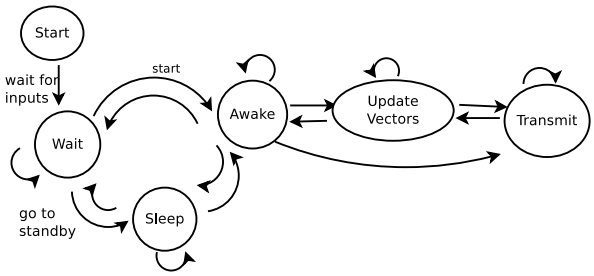

Figure 6.20: Finite state machine housed in leaf lode

it has to sign the meta data with its private key. Hence integrity of the provenance can be ensured. In addition the nodes can never forge or fake provenance. Due to the cryptography if any other nodes change the provenance in an unauthorized way, or if any nodes provide a fake ID within their own provenance, it can be detected by the recipient.

After considering this, we list the following attack model:

- **On-off attack:** Dishonest nodes can take advantage of it and behave well and badly alternatively.
- **New comer attack:** A node can throw away its bad reputation by creating a new identity.

Our system is resilient to on-off attacks as we make use of historical correctness. Reputation builds over time in our system. A dishonest node who has behaved well in past with a good trust value will be caught when it sends out faulty data by our FINDDISSIMILAR algorithm. Its trust value will be decreased and its data will be given less weight meaning it will effect the overall result less.

Our architecture can also defend itself against new comer attack. Public key and private keys are assigned to nodes by the central station at the deployment time . As cryptography is handled centrally, when a new identity is created maliciously, network will be aware of the malicious node.

Chapter 7

Experiments

This section presents an experimental study of our provenance leveraging trust enhancing system ProTru on a synthetic data set. We used MATLAB for our simulation.

As a case study, we are deploying our architecture on an ad hoc multi-hop sensor network which is set up to do surveillance of an area. The area to be watched has temperature sensors creating data. Readings are taken from temperature sensors every t time units. The system is looking for readings that deviate from normal as an alert. The information from the sensors are sent to a hub (intermediate node) that collects all the sensor information. In addition, the provenance data is saved locally at the nodes as vectors. Intermediate nodes are used for moving the data to the central hub. They have broadcast capacity, processing capability and limited memory, allowing sensor data fusion and storage of received input. An example of a system of this type is a forest fire surveillance system that has temperature sensors [149].

We will simulate a temperature monitoring system for fire detection, a sensor network deployed in a national forest. We observe error rate in final results in traditional approach and in our approach. Although our architecture is designed to support different sensor nodes

action	required energy
receive	8 ma
send	12 ma

Figure 7.1: Energy requirements for the sensor nodes in the simulation

(camera, temperature, solar) and data types (image, numeric, voice, video), for the sake of simplicity we are considering only temperature nodes in our simulation. If the temperature is above a threshold, it means the forest is very hot and a fire alarm is created.

We assume a static environment where sensors are randomly deployed in a region of size 100*100 units and all sensors have a common transmission range. The environment has a static temperature which is randomly assigned at each run. Nodes are measuring the constant temperature of a static environment. The temperature reading range is [25, 275].

We aim to show that our system can assess trust in a distributed manner and trust of the network will be kept greater than a traditional network. For our simulations, we use a network of sensor nodes sensing temperature. Our architecture is applicable to sensor network domain as it answers the concerns of a sensor network such as autonomy, decentralization, initialization, data overhead [41].

Data: We experimented on a synthetic dataset, which consists of 10 clusters, each with 10 nodes, making a total of 100 nodes. There is an intermediate node for every group. Nodes are not mobile. Each node has provenance and reputation vectors with the attributes listed in previous section. Data is generated for many time intervals. At each time interval, untrusted nodes which are forwarding faulty data are found and their trust is decreased. Data is generated by a random number generator function and faulty data (outliers) is injected.

Good Data: Inliers will fall into same small interval around the true temperature value.

Bad Data: Outliers are again generated by random number generator but without interval constraints meaning they can be any value within a large specified interval. A noisy reading

(referred to as a faulty reading) is randomly biased from the normal reading generated.

Trust: In our system trust values are not kept but computed on the fly using the feedback coming from intermediate nodes (second hand information) and the information in provenance and reputation vectors. It is attached to the data before transmission. A node's trust can range from 0 to 1.0. We applied algorithms described above for computing data trust. If the trust is below a threshold, the node is marked as faulty and isolated.

Initial trust: We assume that all nodes are trusted at the deployment and assign a random trust value between 0.8 and 1. Later FINDDISSIMILAR algorithm finds out the untrusted nodes distributedly and trust updating algorithms update (increase/decrease) the trust of the nodes.

Energy: Power management scheme is not used for sensor nodes. A simple energy model is used in simulations. We have to consider energy efficiency as we are working with sensors. The sensor nodes we are simulating have AA batteries that can provide nearly 1800 ma. An initial energy of 1800 ma is assigned to each sensor node. Receiving data costs approximately 8 ma, transmitting costs approximately 12 ma which makes 20 ma in total. Every time a node transmits data, we decrease its energy by 20 in our simulation.

In our simulation we implemented the architecture which we have described in the section above.

Based on the values of the constants, the system behaves differently. With different experiments, we monitored the behavior of our system.

7.1 Experiment I

The environment is synthetic where sensors are deployed in a 100 by 100 field to monitor temperatures. Moreover, unusual readings are randomly generated in the monitored field.

id	data	trust	data*trust
1	32	0.886	32*0.886=28.352
2	56	0.798	56*0.798=40.698
3	52	0.888	52*0.888=46.176
4	55	0.826	55*9.826=45.43
5	57	0.82	57*0.82=46.74
6	51	0.728	51*0.728 =40.768
7	57	0.734	57*0.734 =42.572
8	59,	0.748	59*0.748=44.132
9	58	0.78	58*0.78=44.46
10	115	0.852	115*0.852=97.98

Figure 7.2: Values received at an intermediate node

Type of Event	Monitoring Temperature
Faulty Data Percentage	20%
Size of group	10 nodes, 1 IN
Size of network	10 IN (10 groups), 100 nodes
Type of nodes	Temperature Nodes (Temperature Data)
β (trust threshold)	0.1
$v_{initial}$	rand(0,2)
δ	50
λ	0.1

Figure 7.3: Parameters for experiment 1

The faulty sensor rate (abbreviated as faulty rate) is the ratio of the number of faulty sensors and the total number of sensors deployed.

Figure 7.3 lists the values of the parameters of Experiment I. For each error percentage, we repeated the same experiment 20, 80, 800 and 1000 times and calculated the average value. Figure 7.2 illustrates data fusion with values at an intermediate node at an instant. In each simulation, temperature has been selected randomly. In the figure, both designs have increasing errors with more faulty sensors. In a traditional network the untrusted nodes are not found out and they continue contributing to the network making the computed values

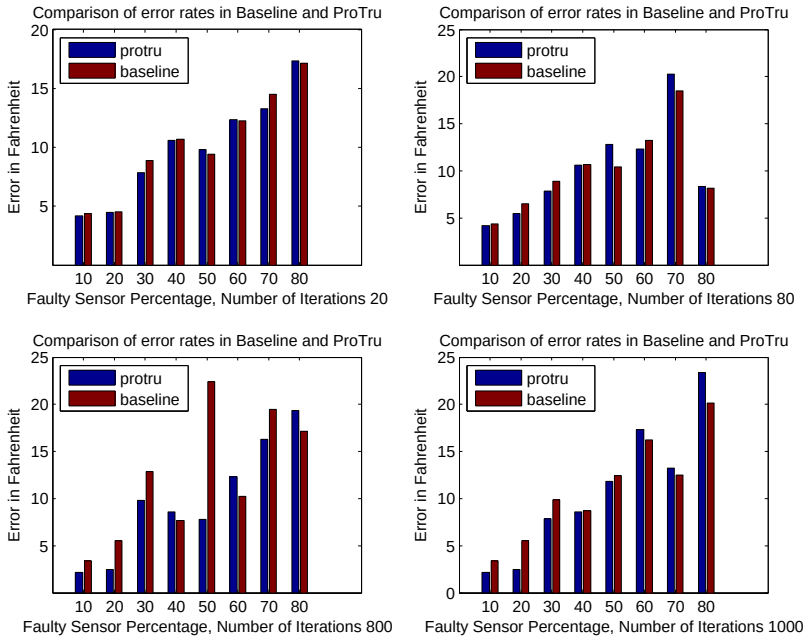

Figure 7.4: Errors in the temperature monitoring sensor network

faulty. Figure 7.4 shows the errors (distance between the real temperature of the environment and the estimated temperature by the network) in the traditional design (denoted as "Baseline") and our design (denoted as "ProTru"). As seen in the Figure 7.4, our architecture outperforms Baseline after many number of iterations. Our architecture shows much reliable results in time since it can filter out potentially faulty sensing data by gradually decreasing the trust of faulty nodes.

7.2 Experiment II

In this experiment we calculated the average trust value of the network changing over time. Nodes are omitted from the group when their trust reach a threshold value. We did not

consider the trust values of the omitted nodes while calculating trust of the network. Trust values of the nodes are increased when they report correct data and decreased when they report faulty data.

In Baseline architecture, all nodes are fully and equally trusted. Therefore average trust of the network always stays at the same value which is deceiving. In ProTru architecture, the trust of the network changes over time. Based on the network parameters and circumstances, trust of the network can increase, decrease or show a changing pattern. The reasons for decay are as follows. One reason is that as energy is drained from the nodes, they become less trusted. The other reason is that trust of faulty nodes is decreased. Trust values of non-faulty nodes are increased as they report correct data.

In our setup, we observed the change in trust for faulty percentage rates of 10-80 with the parameters in Table 7.3. The change in trust in ProTru is illustrated in Figure 7.5. Figure 7.6 illustrates the change in trust in a wireless sensor network without a trust architecture. In Baseline, the trust of the nodes do not change as there is no trust model deployed. The network continues to believe all the nodes equally as they continue transmitting. However in ProTru, trust of the nodes change in time as seen in Figure 7.5.

We have ran the experiment for 20, 80, 800 and 1000 iterations. When the network is deployed, an initial trust value between 0.9 and 1 is assigned. As the nodes begin transmitting, trust values of the outliers are decreased and trust values of the inliers are increased. When we iterate it for 20 times or 80 times, we see a decrease in trust. This means that decrease in trust is more than the increase. For 800 and 1000 number of iterations, we see that trust is fluctuating. Based on the behavior of the nodes, trust of the network increase and decrease over the time.

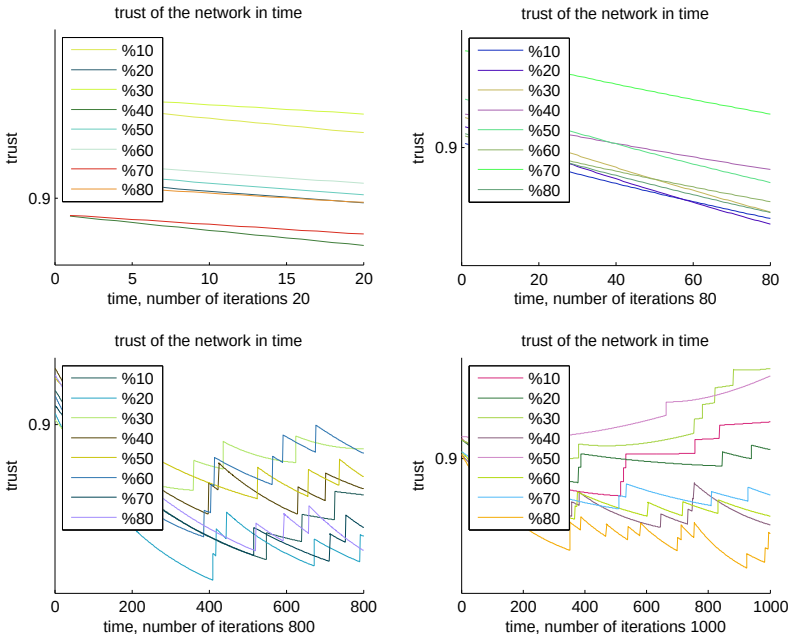

Figure 7.5: Changes in trust in ProTru

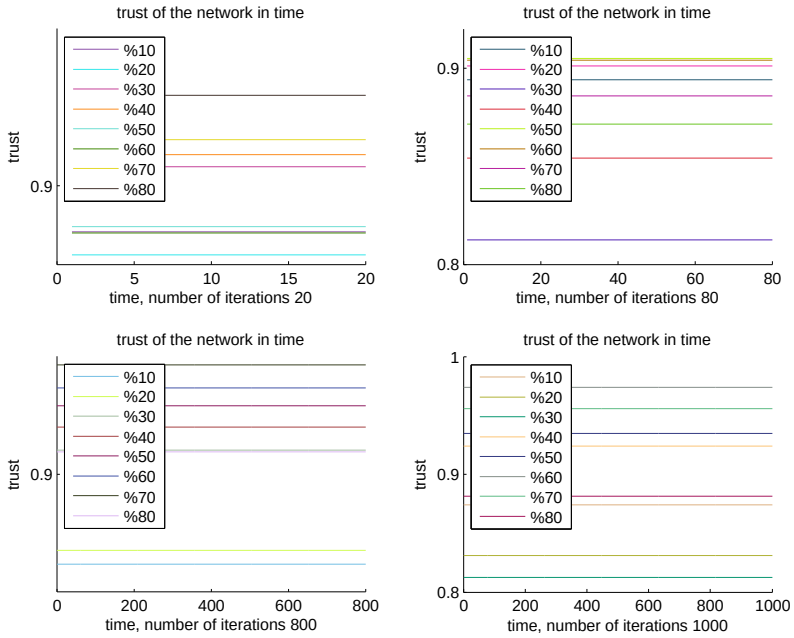

Figure 7.6: Changes in trust in baseline

7.3 Experiment III

In this experiment, we calculated the error for different δ values. While δ value is changing, the other parameters are kept constant as listed in Table 7.3. The outlier percentage for this experiment is 20%. Figure 7.7 shows the errors (distance between the real temperature of the environment and the estimated temperature by the network) in the traditional design (denoted as "Baseline") and our design (denoted as "ProTru").

We have presented the detailed explanation of the usage of δ in Chapter 6. Trust is calculated using the δ value. As the Equation 6.8.4 is an exponential formula, a big δ value makes the v value greater and t value smaller. δ is a positive constant determining how fast we untrust; the bigger it is, the quicker a node gets untrusted, the slower a node gets trusted. In Figure 7.7, ProTru performs better with larger δ values and bigger number of iterations. As δ values get bigger, the network ignores the values coming from untrusted node quickly and creates more accurate results. However as nodes get untrusted immediately, network lifetime will be shorter. We recommend a smaller δ value for a healthier network.

Our architecture performs better when it iterates more. As time passes and the number of iterations increase, the trust of the faulty nodes decrease and reach the trust threshold. Once they reach the trust threshold, the values sent by them are ignored and have no effect on the final result.

7.4 Experiment IV

In this experiment we calculated the error for different λ values. λ appears in Equation 6.8.4. λ is a positive constant determining how fast we untrust; the bigger it is, the

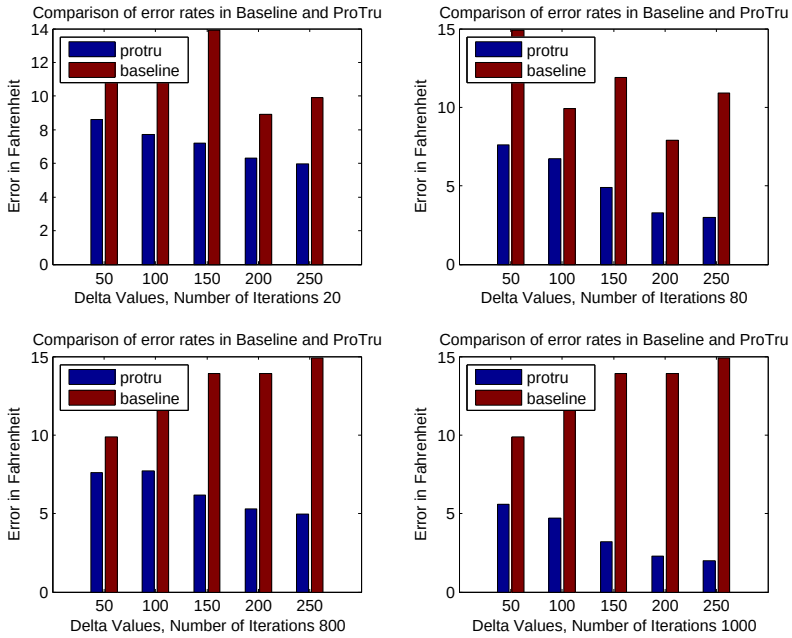

Figure 7.7: Error for different delta values

quicker a node gets untrusted. It represents the system's rewarding and punishing mechanism. If it is a bigger value, the system rewards and punishes easily. On the other hand, if it is a small value, the system rewards and punishes slowly.

While λ value is changing, the other parameters are kept constant as listed in Table 7.3. Figure 7.8 shows the errors (distance between the real temperature of the environment and the estimated temperature by the network) in the traditional design (denoted as "Baseline") and our design (denoted as "ProTru").

The behavior of λ and δ are very similar as they appear in the same equation. As the v value is multiplied directly with λ value, it has a greater impact. However in Figure 7.8 we do not see an obvious pattern. It might be due to the fact that the change in λ is not very marginal. One other reason for not seeing a decreasing pattern in error is as follows. The values are selected randomly in simulation, final results are affected by this randomness. For example, if the faulty nodes create values with large bias then the final result fluctuates more. The other reason is that when λ is large, even one faulty behavior of a trusted node is not forgiven and trust value of the trusted node is decreased abruptly. This might cause network to create faulty results because trusted nodes are omitted easily and cannot contribute to the final result.

7.5 Experiment V

In this experiment we calculated the error for different trust threshold values. Figure 7.9 shows the errors (distance between the real temperature of the environment and the estimated temperature by the network) in the traditional design (denoted as "Baseline") and our design (denoted as "ProTru").

As trust threshold gets smaller, the tolerance of the network to faulty nodes increase. Data

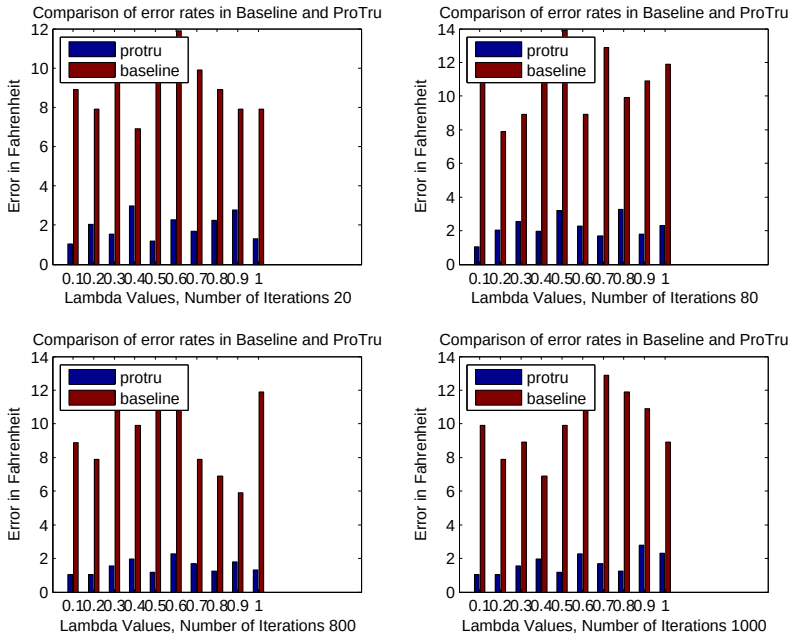

Figure 7.8: Error for different lambda values

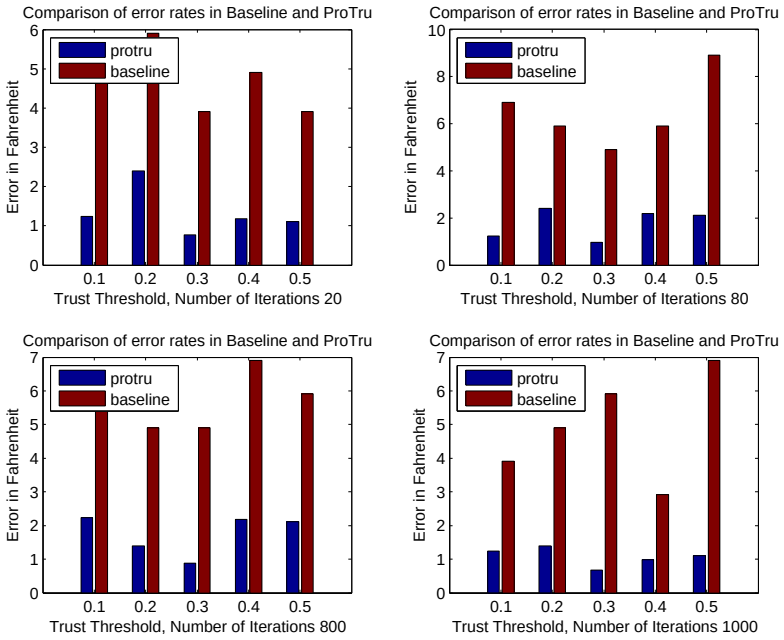

Figure 7.9: Error for different trust thresholds

coming from the nodes are ignored when their trust reach the trust threshold. The bigger the trust threshold is, the easier nodes are omitted. However if the trust threshold is very large, even trusted nodes can hit to the threshold easily and get omitted. It is more efficient for our network to have a trust threshold which is not very big or small. As seen in Figure 7.9 our network creates better results when trust threshold is 0.3.

We can also observe that the network creates better results as it iterates more because iterations help the network to identify faulty nodes.

Chapter 8

Conclusion

While the way information is created, shared and modified has changed a lot with sensor networks, we are exposed to serious new risks. Revolutionary ways are needed in order to overcome the newly introduced challenges. Provenance will have an essential part in this revolution, providing data integrity, authenticity, trustworthiness and availability [25].

We have presented an architecture which uses provenance to enhance trust by restructuring the network in a distributed manner. We designed this architecture and carried simulations to assess trust of the network.

Our system is superior to traditional sensor networks in creating more accurate results. Sensor networks get untrustworthy in time after deployment due to many reasons such as decreasing energy, exhausting resources. As trust is monitored and network is continuously restructured, our network remains trustworthy for a longer time. This becomes a very important benefit for mission critical networks. Our architecture keeps the trust of a wireless sensor network high for a longer time compared to traditional networks by decreasing the trust of the nodes creating faulty data. Using provenance for trust assessment is a novel method with a low communication overhead compared to other approaches such

as cryptography. Moreover trust enhancement is done locally and transmitted data is kept small making the model lightweight and our model efficient for wireless sensor networks. In addition support for multiple trust metrics makes the architecture flexible for different wireless sensor network domains such as industry, military and health-care.

Provenance research in sensor network community is rapidly improving. Once we are at a point where the nature of cause and trust is clearly understood in networks, the answers to fundamental questions such as “what is provenance”, “how do we know when we have enough provenance for an application”, “how can we assess trust using provenance” will be given. Then a robust provenance model for assessing trust in sensor networks can be developed and used as a standard.

Chapter 9

Future Work

One future research direction is adapting this architecture to wireless sensor networks for fault-tolerance including energy management problems, an approach which we believe is new. We argue that most of the faults faced by sensor networks will be easily detected and healed by our distributed architecture which we will explore in more detail in the future.

Next research direction we will take is to add centralized provenance storage scheme to our simulation, to build a real-life network with this design and to test our architecture on a real-world data set. We will experiment our system on the New York City Taxi and Limousine Commission (TLC) data. The TLC project focuses on the extraction of important information as well as trend in the dataset obtained from taxi and limousine cabs that ply the five boroughs of New York city. The dataset is estimated to comprise approximately of 147 million records with a negligible fraction of corrupt record.

We believe that the nature of our architecture suits well to the unpredictable and centrally uncontrollable nature of sensor networks. Our one other future research direction is adapting this architecture to wireless sensor networks for fault-tolerance including energy management problems, an approach which we believe is new. We argue that most of the faults

faced by sensor networks will be easily detected and healed by our distributed architecture which we will explore in more detail in the future. For instance, based on provenance and trust, intermediate nodes will do energy management by turning sensor nodes on and off for a longer network lifetime.

As a future work, we plan to extend the cognitive abilities of the architecture in terms of decisions taken by the nodes. In the present model, an intermediate node can ignore the data coming from a node or a node might stop sending data realizing its low $t_{accuracy}$. However in the extended model, we want to add more complex actions and decisions such as restructuring the network by omitting a whole group, merging groups, putting nodes in overlapping areas.

Bibliography

[1] A. Abdul-Rahman and S. Hailes. Using recommendations for managing trust in distributed systems. In *IEEE Malaysia International Conference on Communication*, volume 97, 1997.

[2] A. Abdul-Rahman and S. Hailes. A distributed trust model. In *Proceedings of the workshop on new security paradigms*, pages 48–60. ACM, 1998.

[3] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pages 9–pp. IEEE, 2000.

[4] K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 310–317. ACM, 2001.

[5] E. Aivaloglou, S. Gritzalis, and C. Skianis. Trust establishment in sensor networks: behaviour-based, certificate-based and a combinational approach. *International Journal of System of Systems Engineering*, 1(1):128–148, 2008.

[6] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.

[7] D. Balfanz, D. Smetters, P. Stewart, and H. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proceedings of the 9th Annual Network and Distributed System Security Symposium (NDSS)*, pages 7–19, 2002.

[8] D. Barseghian, I. Altintas, M. Jones, D. Crawl, N. Potter, J. Gallagher, P. Cornillon, M. Schildhauer, E. Borer, E. Seabloom, et al. Workflows and extensions to the kepler scientific workflow system to support environmental sensor data access and analysis. *Ecological Informatics*, 5(1):42–50, 2010.

[9] B. Bhargava, L. Lilien, A. Rosenthal, M. Winslett, M. Sloman, T. Dillon, E. Chang, F. Hussain, W. Nejdl, D. Olmedilla, et al. The pudding of trust [intelligent systems]. *Intelligent Systems, IEEE*, 19(5):74–88, 2004.

[10] M. Blaze, J. Feigenbaum, and A. Keromytis. Keynote: Trust management for public-key infrastructures. In *Security Protocols*, pages 625–625. Springer, 1999.

[11] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *IEEE Symposium on Security and Privacy*, pages 164–173. IEEE, 1996.

[12] M. Blount, J. Davis, M. Ebling, J. Kim, K. Kim, K. Lee, A. Misra, S. Park, D. Sow, Y. Tak, et al. Century: Automated aspects of patient care. In *13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 504–509. IEEE, 2007.

[13] A. Boukerch, L. Xu, and K. El-Khatib. Trust-based security for wireless ad hoc and sensor networks. *Computer Communications*, 30(11-12):2413–2427, 2007.

[14] S. Bourne. The Jacobson radical of a semiring. *Proceedings of the National Academy of Sciences of the United States of America*, 37(3):163, 1951.

[15] S. Bowers and B. Ludascher. Actor-oriented design of scientific workflows. *Conceptual Modeling*, pages 369–384, 2005.

[16] S. Buchegger and J. Le Boudec. Performance analysis of the confidant protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 226–236. ACM, 2002.

[17] P. Buneman and W. Tan. Provenance in databases. In *Proceedings of the ACM SIGMOD international conference on management of data*, pages 1171–1173. ACM, 2007.

[18] V. Cahill, E. Gray, J. Seigneur, C. Jensen, Y. Chen, B. Shand, N. Dimmock, A. Twigg, J. Bacon, C. English, et al. Using trust for secure collaboration in uncertain environments. *Pervasive Computing, IEEE*, 2(3):52–61, 2003.

[19] E. Callaway. *Wireless sensor networks: architectures and protocols*, volume 3. CRC press, 2004.

[20] J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named graphs, provenance and trust. In *Proceedings of the 14th international conference on World Wide Web*, pages 613–622. ACM, 2005.

[21] A. Chapman, H. Jagadish, and P. Ramanan. Efficient provenance storage. In *Proceedings of the ACM SIGMOD international conference on Management of data*, pages 993–1006. ACM, 2008.

[22] H. Chen, H. Wu, X. Zhou, and C. Gao. Reputation-based trust in wireless sensor networks. In *International Conference on Multimedia and Ubiquitous Engineering.*, pages 603–607. IEEE, 2007.

[23] J. Cheney, L. Chiticariu, and W. Tan. Provenance in databases: Why, how, and where. *Foundations and Trends in Databases*, 1(4):379–474, 2009.

[24] J. Cheney, L. Chiticariu, and W. Tan. Provenance in databases: Why, how, and where. *Foundations and Trends in Databases*, 1(4):379–474, 2009.

[25] J. Cheney, S. Chong, N. Foster, M. Seltzer, and S. Vansummeren. Provenance: a future history. In *Proceeding of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, pages 957–964. ACM, 2009.

[26] J. Cho, A. Swami, and R. Chen. A survey on trust management for mobile ad hoc networks. *Communications Surveys & Tutorials, IEEE*, 13(4):562–583, 2011.

[27] I. R. Council. Infosec hard problem list, 2005.

[28] D. Crawl and I. Altintas. A provenance-based fault tolerance mechanism for scientific workflows. *Provenance and Annotation of Data and Processes*, pages 152–159, 2008.

[29] G. Crosby and N. Pissinou. Cluster-based reputation and trust for wireless sensor networks. In *Proc. 4th Consumer Communications and Networking Conference (CCNC 2007)*, 2007.

[30] G. Crosby, N. Pissinou, and J. Gadze. A framework for trust-based cluster head election in wireless sensor networks. In *Dependability and Security in Sensor Networks and Systems, 2006. DSSNS 2006. Second IEEE Workshop on*, pages 10–pp. IEEE, 2006.

[31] Y. Cui, J. Widom, and J. Wiener. Tracing the lineage of view data in a warehousing environment. *ACM Transactions on Database Systems (TODS)*, 25(2):179–227, 2000.

[32] D. Culler, D. Estrin, and M. Srivastava. Overview of sensor networks. *Computer Journal*, pages 41–49, 2004.

[33] C. Dai, H.-S. Lim, E. Bertino, and Y.-S. Moon. Assessing the trustworthiness of location data based on provenance. In *Proceedings of the 17th ACM SIGSPATIAL*

International Conference on Advances in Geographic Information Systems, pages 276–285. ACM, 2009.

[34] C. Dai, D. Lin, E. Bertino, and M. Kantarcioglu. Trust evaluation of data provenance. Technical report, Technical Report CERIAS TR 2001-147, Purdue University, 2008.

[35] J. Darn and R. Listernick. The Health Insurance Portability and Accountability Act of 1996 (HIPAA). *Pediatrics: just the facts*, page 167, 2004.

[36] J. L. David Hall. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[37] T. Deelmann and P. Loos. Trust economy: aspects of reputation and trust building for smes in e-business. In *Eighth Americas Conference on Information Systems*, pages 2213–2221, 2002.

[38] L. Ding, P. Kolari, T. Finin, A. Joshi, Y. Peng, and Y. Yesha. On homeland security and the semantic web: A provenance and trust aware inference framework. In *Proceedings of the AAAI Spring Symposium on AI Technologies for Homeland Security*, 2005.

[39] G. Dogan, T. Brown, K. Govindan, M. Khan, T. Abdelzaher, P. Mohapatra, and J. Cho. Evaluation of network trust using provenance based on distributed local intelligence. In *Military Communications Conference*. IEEE, 2011.

[40] L. Eschenauer, V. Gligor, and J. Baras. On trust establishment in mobile ad-hoc networks. In *Security Protocols*, pages 47–66. Springer, 2004.

[41] M. Fernández-Gago, R. Roman, and J. Lopez. A survey on the applicability of trust management systems for wireless sensor networks. In *Security, Privacy and Trust in Pervasive and Ubiquitous Computing, 2007. SECPerU 2007. Third International Workshop on*, pages 25–30. IEEE, 2007.

[42] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[43] I. Foster, J. Vockler, M. Wilde, and Y. Zhao. Chimera: A virtual data system for representing, querying, and automating data derivation. In *International Conference on Scientific and Statistical Database Management*, pages 37–46. IEEE, 2002.

[44] J. Freire, D. Koop, E. Santos, and C. Silva. Provenance for computational tasks: A survey. *Computing in Science & Engineering*, 10(3):11–21, 2008.

[45] J. Freire, C. Silva, S. Callahan, E. Santos, C. Scheidegger, and H. Vo. Managing rapidly-evolving scientific workflows. *Provenance and Annotation of Data*, pages 10–18, 2006.

[46] D. Gambetta. Can we trust trust. *Trust: Making and breaking cooperative relations*, pages 213–237, 2000.

[47] S. Ganeriwal, L. Balzano, and M. Srivastava. Reputation-based framework for high integrity sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 4(3):15, 2008.

[48] T. Gao, D. Greenspan, M. Welsh, R. Juang, and A. Alm. Vital signs monitoring and patient tracking over a wireless network. In *27th Annual International Conference of the Engineering in Medicine and Biology Society*, pages 102–105. IEEE, 2006.

[49] S. Glaser. Some real-world applications of wireless sensor nodes. In *Proceedings of SPIE Symposium on Smart Structures and Materials/NDE*, page 344, 2004.

[50] B. Glavic, K. Esmaili, P. Fischer, and N. Tatbul. The case for fine-grained stream provenance. In *BTW Workshops*, volume 11, pages 58–61, 2011.

[51] J. Golbeck. Combining provenance with trust in social networks for semantic web content filtering. *Provenance and Annotation of Data*, pages 101–108, 2006.

[52] K. Govindan, W. X., M. Khan, G. Dogan, G. Zeng, K. Powell, T. Brown, T. Abdelzaher, and P. Mohapatra. Pronet: Network trust assessment based on incomplete provenance. In *Military Communications Conference, 2011. MILCOM 2011. IEEE*. IEEE, 2011.

[53] T. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 31–40. ACM, 2007.

[54] A. Happe. Agile provenance. Master's thesis, Technical University of Vienna, March 2010.

[55] R. Hasan, R. Sion, and M. Winslett. Preventing history forgery with secure provenance. *ACM Transactions on Storage (TOS)*, 5(4):1–43, 2009.

[56] N. Heo and P. Varshney. An intelligent deployment and clustering algorithm for a distributed mobile sensor network. In *International Conference on Systems, Man and Cybernetics*, volume 5, pages 4576–4581. IEEE, 2003.

[57] K. Hung, K. Lui, and Y. Kwok. A trust-based geographical routing scheme in sensor networks. In *Wireless Communications and Networking Conference*, pages 3123–3127. IEEE, 2007.

[58] M. Huq, A. Wombacher, and P. Apers. Inferring fine-grained data provenance in stream data processing: Reduced storage cost, high accuracy. In *Database and Expert Systems Applications*, pages 118–127. Springer, 2011.

[59] J. Hur, Y. Lee, H. Yoon, D. Choi, and S. Jin. Trust evaluation model for wireless sensor networks. In *Advanced Communication Technology, 2005, ICACT 2005. The 7th International Conference on*, volume 1, pages 491–496. IEEE, 2005.

[60] R. Ikeda and J. Widom. Panda: A system for provenance and data. In *Proceedings of the 2nd USENIX Workshop on the Theory and Practice of Provenance (TaPP10)*, 2010.

[61] A. Jøsang, E. Gray, and M. Kinateder. Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems*, 4(2):139–161, 2006.

[62] A. Josang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[63] A. Jøsang and S. Presti. Analysing the relationship between risk and trust. *Trust Management*, pages 135–145, 2004.

[64] A. Jsang and R. Ismail. The beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*, pages 41–55, 2002.

[65] F. Kelly and R. Williams. Dynamic routing in stochastic networks. *IMA Volumes in Mathematics and its Applications*, 71:169–169, 1995.

[66] A. Kementsietsidis and M. Wang. On the efficiency of provenance queries. In *International Conference on Data Engineering*, pages 1223–1226. IEEE, 2009.

[67] M. Khan, T. Abdelzaher, J. Han, and H. Ahmadi. Finding symbolic bug patterns in sensor networks. *Distributed Computing in Sensor Systems*, pages 131–144, 2009.

[68] M. Khan, H. Ahmadi, G. Dogan, K. Govindan, R. Ganti, T. Brown, J. Han, P. Mohapatra, and T. Abdelzaher. Dustdoctor: A self-healing sensor data collection system. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 127–128. IEEE, 2011.

[69] M. Khan, L. Luo, C. Huang, and T. Abdelzaher. Snts: Sensor network troubleshooting suite. *Distributed Computing in Sensor Systems*, pages 142–157, 2007.

[70] M. Kinateder, E. Baschny, and K. Rothermel. Towards a generic trust model–comparison of various trust update algorithms. *Trust Management*, pages 119–134, 2005.

[71] E. Kotsovinos and A. Williams. Bambootrust: Practical scalable trust management for global public computing. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1893–1897. ACM, 2006.

[72] M. Krasniewski, P. Varadharajan, B. Rabeler, S. Bagchi, and Y. Hu. Tibfit: Trust index based fault tolerance for arbitrary data faults in sensor networks. In *International Conference on Dependable Systems and Networks.*, pages 672–681. IEEE, 2005.

[73] R. Lange. *Provenance aware sensor networks for real-time data analysis*. PhD thesis, University of Twente, Netherlands, 2010.

[74] J. Ledlie, C. Ng, and D. Holland. Provenance-aware sensor data storage. In *21st International Conference on Data Engineering Workshops*, pages 1189–1189. IEEE, 2005.

[75] H. Lim, Y. Moon, and E. Bertino. Research issues in data provenance for streaming environments. In *Proceedings of the 2nd SIGSPATIAL ACM GIS 2009 International Workshop on Security and Privacy in GIS and LBS*, pages 58–62. ACM, 2009.

[76] H. Lim, Y. Moon, and E. Bertino. Assessing the Trustworthiness of Streaming Data. Technical report, Technical Report TR 2010-09, CERIAS, 2010.

[77] H. Lim, Y. Moon, and E. Bertino. Provenance-based trustworthiness assessment in sensor networks. In *Proceedings of the Seventh International Workshop on Data Management for Sensor Networks*, pages 2–7. ACM, 2010.

[78] Y. Liu, J. Futrelle, J. Myers, A. Rodriguez, and R. Kooper. A provenance-aware virtual sensor system using the open provenance model. In *Collaborative Technologies and Systems (CTS), 2010 International Symposium on*, pages 330–339. IEEE, 2010.

[79] Y. Liu, D. Hill, A. Rodriguez, L. Marini, R. Kooper, J. Myers, X. Wu, and B. Minsker. A new framework for on-demand virtualization, repurposing and fusion of heterogeneous sensors. In *Collaborative Technologies and Systems, 2009. CTS'09. International Symposium on*, pages 54–63. IEEE, 2009.

[80] J. Lopez, R. Roman, I. Agudo, and C. Fernandez-Gago. Trust management systems for wireless sensor networks: Best practices. *Computer Communications*, 33(9):1086–1093, 2010.

[81] R. Ma, L. Xing, and H. Michel. Fault-intrusion tolerant techniques in wireless sensor networks. In *Dependable, Autonomic and Secure Computing, 2nd IEEE International Symposium on*, pages 85–94. IEEE, 2006.

[82] D. McKnight and N. Chervany. The meanings of trust. Technical Report WP9604, University of Minnesota Management Information Systems Research Center, 1996.

[83] D. McKnight and N. Chervany. Conceptualizing trust: A typology and e-commerce customer relationships model. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, pages 10–pp. IEEE, 2001.

[84] S. Miles, P. Groth, M. Branco, and L. Moreau. The requirements of using provenance in e-science experiments. *Journal of Grid Computing*, 5(1):1–25, 2007.

[85] A. Misra, M. Blount, A. Kementsietsidis, D. Sow, and M. Wang. Advances and challenges for scalable provenance in stream processing systems. *Provenance and Annotation of Data and Processes*, pages 253–265, 2008.

[86] S. Misra, S. Misra, and I. Woungang. *Selected Topics in Communication Networks and Distributed Systems*. World Scientific Pub Co Inc, 2009.

[87] R. Molva and P. Michiardi. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. *Institute EurecomResearch Report RR-02-062*, 2001.

[88] M. Momani, K. Aboura, and S. Challa. Rbatmwsn: Recursive bayesian approach to trust management in wireless sensor networks. In *3rd International Conference on Intelligent Sensors, Sensor Networks and Information.*, pages 347–352. IEEE, 2007.

[89] M. Momani, J. Agbinya, G. Navarrete, and M. Akache. A new algorithm of trust formation in wireless sensor networks. In *Proceedings of the 1st IEEE International Conference on Wireless Broadband and Ultra Wideband Communications*, 2006.

[90] M. Momani and S. Challa. Survey of trust models in different network domains. *Arxiv preprint arXiv:1010.0168*, 2010.

[91] L. Moreau, P. Groth, S. Miles, J. Vazquez-Salceda, J. Ibbotson, S. Jiang, S. Munroe, O. Rana, A. Schreiber, V. Tan, et al. The provenance of electronic data. *Communications of the ACM*, 51(4):52–58, 2008.

[92] L. Moreau, B. Ludascher, I. Altintas, R. Barga, S. Bowers, S. Callahan, J. Chin, B. Clifford, S. Cohen, S. Cohen-Boulakia, et al. Special issue: The first provenance challenge. *Concurrency and Computation: Practice and Experience*, 20(5):409–418, 2008.

[93] L. Moreau, B. Ludäscher, I. Altintas, R. Barga, S. Bowers, S. Callahan, G. Chin Jr, B. Clifford, S. Cohen, S. Cohen-Boulakia, et al. Special issue: The first provenance challenge. *Concurrency and Computation: Practice and Experience*, 20(5):409–418, 2008.

[94] J. Mundinger and J. Le Boudec. Reputation in self-organized communication systems and beyond. In *Proceedings from the 2006 workshop on Interdisciplinary systems approach in performance evaluation and design of computer & communications sytems*, page 3. ACM, 2006.

[95] K. Muniswamy-Reddy. *Foundations for Provenance-Aware Systems*. PhD thesis, Harvard University Cambridge, Massachusetts, 2010.

[96] J. Myers, C. Pancerella, C. Lansing, K. Schuchardt, B. Didier, N. Ashish, and C. Goble. Multi-scale science: supporting emerging practice with semantically derived provenance. In *Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data*. Citeseer, 2003.

[97] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis & defenses. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 259–268. ACM, 2004.

[98] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Greenwood, T. Carver, M. Pocock, A. Wipat, and P. Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 2004.

[99] S. Ozdemir. Functional reputation based reliable data aggregation and transmission for wireless sensor networks. *Computer Communications*, 31(17):3941–3953, 2008.

[100] J. Paek, O. Gnawali, K. Jang, D. Nishimura, R. Govindan, J. Caffrey, M. Wahbeh, and S. Masri. A programmable wireless sensing system for structural monitoring. In *4th World Conference on Structural Control and Monitoring (4WCSCM)*, 2006.

[101] P. Papadimitratos and Z. Haas. *Securing mobile ad hoc networks*. CRC Press, 2002.

[102] U. Park and J. Heidemann. Provenance in sensornet republishing. *Provenance and Annotation of Data and Processes*, pages 280–292, 2008.

[103] H. Patni, S. Sahoo, C. Henson, and A. Sheth. Provenance Aware Linked Sensor Data. In *2nd Workshop on Trust and Privacy on the Social and Semantic Web, Co-located with ESWC2010, Heraklion Greece*, 2010.

[104] A. Perrig, J. Stankovic, and D. Wagner. Security in wireless sensor networks. *Communications of the ACM*, 47(6):53–57, 2004.

[105] A. Pirzada and C. McDonald. Establishing trust in pure ad-hoc networks. In *Proceedings of the 27th Australasian conference on Computer science-Volume 26*, pages 47–54. Australian Computer Society, Inc., 2004.

[106] B. Przydatek, D. Song, and A. Perrig. Sia: Secure information aggregation in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 255–265. ACM, 2003.

[107] D. Quercia, S. Hailes, and L. Capra. B-trust: Bayesian trust framework for pervasive computing. *Trust Management*, pages 298–312, 2006.

[108] S. Rajbhandari, I. Wootten, A. Ali, and O. Rana. Evaluating provenance-based trust for scientific workflows. In *Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on*, volume 1, pages 365–372. IEEE, 2006.

[109] Y. Rebahi, V. Mujica-V, and D. Sisalem. A reputation-based trust mechanism for ad hoc networks. In *10th IEEE Symposium on Computers and Communications, year = 2005, pages = 37–42, organization = IEEE*.

[110] P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. *Advances in Applied Microeconomics*, 11:127 – 157, 2002.

[111] S. Ries, J. Kangasharju, and M. Mühlhäuser. A classification of trust systems. In *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, pages 894–903. Springer, 2006.

[112] B. Rothstein. Trust, social dilemmas and collective memories. *Journal of Theoretical Politics*, 12(4):477–501, 2000.

[113] C. Sar and P. Cao. Lineage file system. *Online at http://crypto. stanford. edu/cao/lineage. html*, 2005.

[114] J. Scott. Social network analysis. *Sociology*, 22(1):109–127, 1988.

[115] R. Shaikh, H. Jameel, S. Lee, S. Rajput, and Y. Song. Trust management problem in distributed wireless sensor networks. In *Embedded and Real-Time Computing Systems and Applications, 2006. Proceedings. 12th IEEE International Conference on*, pages 411–414. IEEE, 2006.

[116] Y. Simmhan, B. Plale, and D. Gannon. A survey of data provenance techniques. *Computer Science Department, Indiana University, Bloomington IN*, 47405, 2005.

[117] B. C. Society. Grand challenges in computing, 2004.

[118] B. Solhaug, D. Elgesem, and K. Stølen. Why trust is not proportional to risk. In *Proceedings of The 2nd International Conference on Availability, Reliability and Security (ARES)*, pages 11–18, 2007.

[119] A. Srinivasan, F. Li, and J. Wu. A novel cds-based reputation monitoring system for wireless sensor networks. In *Distributed Computing Systems Workshops, 2008. ICDCS'08. 28th International Conference on*, pages 364–369. IEEE, 2008.

[120] A. Srinivasan, J. Teitelbaum, and J. Wu. Drbts: Distributed reputation-based beacon trust system. In *Dependable, Autonomic and Secure Computing, 2nd IEEE International Symposium on*, pages 277–283. IEEE, 2006.

[121] F. Stajano and R. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Security Protocols*, pages 172–182. Springer, 2000.

[122] E. Stephan, T. Halter, and B. Ermold. Leveraging The Open Provenance Model as a Multi-Tier Model for Global Climate Research. In *Proc. of 3rd International Provenance and Annotation Workshop (IPAW10), Troy, NY*, 2010.

[123] S. Sultana, E. Bertino, and M. Shehab. A provenance based mechanism to identify malicious packet dropping adversaries in sensor networks. In *31st International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 332–338. IEEE, 2011.

[124] J. F. Susan B. Davidson. Provenance and scientific workflows: challenges and opportunities. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 345–1350, 2008.

[125] W.-C. Tan. Provenance in databases : Past, current, and future. *IEEE Data Engineering Bulletin*, 30:3–12, 2007.

[126] S. Tanachaiwiwat, P. Dave, R. Bhindwale, and A. Helmy. Location-centric isolation of misbehavior and trust routing in energy-constrained sensor networks. In *International Conference on Performance, Computing, and Communications*, pages 463–469. IEEE, 2004.

[127] E. A. L. Thomas Huining Feng. Real-time distributed discrete-event execution with fault tolerance. *Proceedings of IEEE Real-Time and Embedded Technology and Applications Symposium*, 2008.

[128] M. Tubaishat and S. Madria. Sensor networks: an overview. *Potentials, IEEE*, 22(2):20–23, 2003.

[129] J. Vazquez-Salceda, S. Alvarez, T. Kifor, L. Varga, S. Miles, L. Moreau, and S. Willmott. Eu provenance project: an open provenance architecture for distributed applications. *Agent Technology and e-Health*, pages 45–63, 2008.

[130] N. Vijayakumar and B. Plale. Towards low overhead provenance tracking in near real-time stream filtering. *Provenance and Annotation of Data*, pages 46–54, 2006.

[131] J. Walters, Z. Liang, W. Shi, and V. Chaudhary. Wireless sensor network security: A survey. *Security in distributed, grid, mobile, and pervasive computing*, page 367, 2007.

[132] M. Wang, M. Blount, J. Davis, A. Misra, and D. Sow. A time-and-value centric provenance model and architecture for medical event streams. In *Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*, pages 95–100. ACM, 2007.

[133] X. Wang, K. Govindan, and P. Mohapatra. Provenance-based information trustworthiness evaluation in multi-hop networks. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[134] Y. Wang, G. Attebury, and B. Ramamurthy. A survey of security issues in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 2006.

[135] Y. Wang and J. Vassileva. Bayesian network-based trust model. In *International Conference on Web Intelligence*, pages 372–378. IEEE, 2003.

[136] Y. Wang and J. Vassileva. Trust and reputation model in peer-to-peer networks. In *Third International Peer-to-Peer Computing Conference*, pages 150–157. IEEE, 2003.

[137] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh. Monitoring volcanic eruptions with a wireless sensor network. In *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, pages 108–120. IEEE, 2005.

[138] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh. Deploying a wireless sensor network on an active volcano. *Internet Computing, IEEE*, 10(2):18–25, 2006.

[139] J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. *Technical Report*, 2004.

[140] X. Xiao, W. Peng, C. Hung, and W. Lee. Using sensorranks for in-network detection of faulty readings in wireless sensor networks. In *Proceedings of the 6th ACM international workshop on Data engineering for wireless and mobile access*, pages 1–8. ACM, 2007.

[141] L. Xiong and L. Liu. A reputation-based trust model for peer-to-peer e-commerce communities. In *E-Commerce, 2003. CEC 2003. IEEE International Conference on*, pages 275–284. IEEE, 2003.

[142] Z. Yao, D. Kim, and Y. Doh. Plus: Parameterized and localized trust management scheme for sensor networks security. In *International Conference on Mobile Adhoc and Sensor Systems*, pages 437–446. IEEE, 2006.

[143] Z. Yao, D. Kim, I. Lee, K. Kim, and J. Jang. A security framework with trust management for sensor networks. In *Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks*, pages 190–198. IEEE, 2005.

[144] E. Yoneki and J. Bacon. A survey of wireless sensor network technologies. Technical Report 646, University of Cambridge, 2005.

[145] O. Younis and S. Fahmy. Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *Mobile Computing, IEEE Transactions on*, 3(4):366–379, 2004.

[146] B. Yu, S. Kallurkar, and R. Flo. A demspter-shafer approach to provenance-aware trust assessment. In *International Symposium on Collaborative Technologies and Systems.*, pages 383–390. IEEE, 2008.

[147] B. Yu, S. Kallurkar, G. Vaidyanathan, and D. Steiner. Managing the pedigree and quality of information in dynamic information sharing environments. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 207. ACM, 2007.

[148] S. Zahedi, M. Szczodrak, P. Ji, D. Mylaraswamy, M. Srivastava, and R. Young. Tiered architecture for on-line detection, isolation and repair of faults in wireless sensor networks. In *Military Communications Conference*, pages 1–7. IEEE, 2008.

[149] J. Zhang, W. Li, N. Han, and J. Kan. Forest fire detection system based on a zigbee wireless sensor network. *Frontiers of Forestry in China*, 3(3):369–374, 2008.

[150] Q. Zhang, T. Yu, and P. Ning. A framework for identifying compromised nodes in wireless sensor networks. *ACM Transactions on Information and System Security (TISSEC)*, 11(3):12, 2008.

[151] J. Zhao, C. Goble, R. Stevens, and S. Bechhofer. Semantically linking and browsing provenance logs for e-science. *Semantics of a Networked World*, pages 158–176, 2004.

[152] D. Zhou. Security issues in ad hoc networks. In *The handbook of ad hoc wireless networks*, pages 569–582. CRC Press, Inc., 2003.

[153] L. Zhou and Z. Haas. Securing ad hoc networks. *Network, IEEE*, 13(6):24–30, 1999.

[154] W. Zhou, E. Cronin, and B. Loo. Provenance-aware secure networks. In *24th International Conference on Data Engineering Workshop.*, pages 188–193. IEEE, 2008.

[155] W. Zhou, M. Sherr, T. Tao, X. Li, B. Loo, and Y. Mao. Efficient querying and maintenance of network provenance at internet-scale. In *Proceedings of the 2010 international conference on Management of data*, pages 615–626. ACM, 2010.

[156] T. Zia and A. Zomaya. Security issues in wireless sensor networks. In *International Conference on Systems and Networks Communications.*, pages 40–40. IEEE, 2006.