Evaluation of Network Trust Using Provenance Based on Distributed Local Intelligence

Gulustan Dogan*, Theodore Brown*, Kannan Govindan‡, Mohammad Maifi Hasan Khan†, Tarek Abdelzaher†,
Prasant Mohapatra‡, Jin-Hee Cho§,

gdogan@gc.cuny.edu, tbrown@gc.cuny.edu, gkannan@cs.ucdavis.edu, mmkhan2@illinois.edu, zaher@illinois.edu, prasant@cs.ucdavis.edu, jinhee.cho@us.army.mil
City University of New York †University of Illinois at Urbana Champaign ‡University of California, Davis §U.S. Army Research Laboratory

***Abstract*—Provenance can play a significant role in a military information system for supporting the calculation of information trust. 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. We introduce a node-level trust-enhancing mechanism for information networks using provenance. A unique characteristic of the proposed trust architecture presented here is the use of provenance through the path of the information from source to destination in determining the information trust. In this proposed architecture each node in the system has a trust and provenance vector. Each information item transmitted over the network has a trust value associated with it. Nodes reexamine and update the trust value associated with the information, creating a distributed system that is more flexible and more responsive. As our system allows reconfigurations, initiatives taken by the intermediate nodes such as replacement of untrusted nodes will enhance the network trust in mission critical situations faster than a centralized approach.**

***Index Terms*—Provenance, Trust, Information Networks, Distributed Intelligence.**

I. Introduction

Provenance has been defined broadly as the origin, history, chain of custody, derivation or process of an object. In disciplines such as art, archeology, provenance is crucial to value an artifact as being authentic and original [1]. However provenance has also become a crucial component in fields that rely on digital information. Provenance has grown in importance in its use in helping to understand how the digitally captured data is manipulated at the source and used at the destination as intelligence.

The literature often divides provenance into data and workflow provenance [9]. Data provenance gives a detailed record of the derivation of a piece of data that is the result of a transformation step [13] whereas workflow provenance is the information or metadata that characterizes the processing of information from input to output [4].

In the computational world, as all kinds of information can easily be changed, provenance becomes an important way of keeping track of alterations [1]. It answers questions such as "how was the object created", "on what other objects does this object depend", "how do the ancestries of these two objects differ" [10]. Provenance management should also be a concern, in order to have an understanding of how results are obtained for later use such as trust enhancement, fault tolerance, troubleshooting, result reproduction and performance optimization. In this paper, we will use provenance for its trust enhancement capability.

In a military information network trust assessment is a crucial task. Information trust may depend on several factors such as the path traveled by the data, the trust of the source, time elapsed after the transmission, e.t.c. As enhancing trust involves understanding causal chains of events, dataflow model is a solid reference of the phases data goes through [2]. 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. The military has to depend on accurate data as it should be able to perform well in harsh environments. Therefore keeping the trust of a data item as up-to-date as possible is a clear concern.

This paper introduces an architecture that makes use of provenance to support a dynamically configurable network. Our targeted environment is a stationary wireless network with no attackers considered. That is, we expect information transmitted not to be compromised by outside forces. This architecture would improve the accuracy of information received by a central node, for instance, a headquarters. To enable this network, each node keeps a trust vector and a provenance vector within its own memory. These vectors are information relevant to this one node. The initial values in the vectors can be specified by a network administrator after the first deployment or the nodes can initialize some values such as the amount of energy they have. One way that the accuracy of information is improved is by making use of these vectors to create a trust value that is sent along with each data item. 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 trust value is computed using provenance and trust vectors of the node by a trust algorithm. While trust evaluation algorithm is not in the scope of our paper, our architecture is flexible enough to handle different trust algorithms. Much of the provenance and trust is contained within the network. Overwhelming the network by forwarding provenance information is avoided, only (data,trust) tuple is passed. A parent node receives many (data, trust) tuples from its children, it may take many possible actions based on the its current state and an algorithm based on say a 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.

A possible scenario to show why we need to enhance trust in information networks is as follows. In a target localization sensor network, many low cost proximity sensors are used in order to localize a target trespassing 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. Hence in this paper, provenance can be used in order to restructure the network for maintaining the trust, an idea that we believe is new.

This paper is organized as follows. Related work is presented in Section 2. In Section 3 we describe our architecture, Section 4 concludes the paper.

II. Related Work

In eScience community there has been work on using provenance to assess trust in scientific worklow systems [12]. With a different orientation, the database community has also done work on managing accuracy of the data through provenance [15]. Besides this, there has been research on using provenance in inferring trust on a specific kind of network such as social networks [5], agent networks [14]. Making use of provenance in the calculation of trust in streaming environments, a work in progress, has similarities to our approach [8]. 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.

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 broad sense meaning the capability of the network to provide an immediate and detailed data trustworthiness. 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 within the network. Kelly provided a technique that makes use of local knowledge at a node to improve the traffic flow versus link capacity within the network [7]. Heo and Varshney made use of mobility to better position sensors in an area to improve coverage and energy efficiency [6]. 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 [16]. 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.

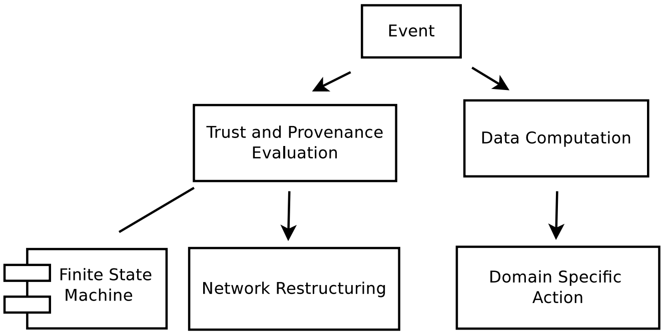

Fig. 1. *Main picture of the tool.*

The interest of our paper differs from all of the above as we are interested in using provenance to restructure the information network to keep the information as accurate as possible. To our knowledge, there has not been work done on provenance management in information networks using a distributed intelligence approach to enhance trust.

III. Architecture of the Intelligent Information Network Model

A. Main Model

We present a dataflow-oriented provenance model for information networks that makes use of node-level trust-enhancing mechanisms. Our three level architecture consists of a graph that contains stationary leaf nodes, intermediate nodes and a central node that receives information from all sources. We are considering an attack-free wireless network. Our architecture works both with streaming environments and triggered networks. Before describing the architecture, we will define the terms that are used throughout the paper.

All nodes in our network as well as leaf nodes have vectors of provenance and trust. A trust value computed using the vectors is forwarded along with the data while provenance and trust vectors are kept at the nodes.

Information trust: Information trust refers to the trust value placed on the information. We call an information trusted if its trust value is above a threshold value. One way of characterizing trust is through attaching a probability value to the information. For example in a target localization network, information reporting the correct coordinates of a target within some tolerance is trusted.

Trust vector of a node: Every node has a trust vector consisting of values in the provenance vector such as the accuracy of the GPS information, accuracy of the battery information, signal-to-noise ratio (SNR), the state of the node if the node can be in different states (e.g. sleeping), the trust value for this node. Some of this may be in the form of a statistical measure (mean or standard deviation). Nodes are first given some initial values of accuracy at the deployment time. Later these values are updated by the node itself or by the intermediate node for their group.

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), message id of its last transmitted message, etc. There is a correlation between what fields the provenance vector contains and the trust vector.

Trust value transmitted with the data: A trust value is computed using the provenance and trust vectors stored in the node using a trust algorithm. In this model, the trust value can be a probability value for example the probability of data being accurate.

Untrusted node: A node is named as untrusted if it has a low trust value attached to its data. Some causes of a low trust value can be listed as follows. Signal-to-noise ratio value can be higher than a threshold for the node it is sending data to or data value can be inconsistent with the other received data.

Network restructuring: 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.

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* : Computationally more powerful nodes receiving information from a group of nodes, doing calculations on its receiving information such as fusing, and transmitting the information for the group forward.
- *Root Node* : Top level of hierarchy which is a central station.

Leaf Nodes

Leaf nodes collect and then disseminate information but do not receive information from other nodes. A leaf node can be any entity such as a person, news article, sensor, etc. The trust and provenance vectors would be different for different types of leaf nodes.

$$\vec{\rho(t)} = \begin{bmatrix} \text{node id} \\ \text{group id} \\ \text{message id} \\ \text{timestamp} \\ \text{left energy} \\ \text{trust algorithm used} \\ \text{node location} \end{bmatrix}, \vec{\tau(t)} = \begin{bmatrix} \text{probability of accuracy} \\ \text{threshold} \\ \text{coordinate interval} \\ \text{minimum energy required} \end{bmatrix}$$

Fig. 2. *Example provenance and trust vectors.*

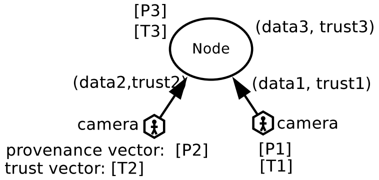

Fig. 3. *Fusion in intermediate node.*

Intermediate Nodes

Intermediate nodes are identified by a unique id and they are the leaders of a group of leaves or other intermediate nodes.. They can take decisions related to the group such as decreasing the trust value of a leaf node, omitting a node or adding others. Groups can be formed based on locations so that every node has to send data to a short distance reducing overall energy consumption.

A leaf node can belong to more than one group creating overlap. Overlapping is useful in unpredicted situations such as most nodes in one intermediate node's group may die or become dysfunctional producing untrusted results. 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 the 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 do 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.

As the information is computed and fused, a new trust value is also calculated in the intermediate node as the result of comparisons of several (data, trust) 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_1) and $camera_2$ sends the tuple (i_2, t_2) with trust values very close to each other as illustrated in Figure 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

Intermediate nodes will send computed and fused data and the corresponding trust value to central node. The central node will also makes decisions. For example, the central node can decide to omit or replace an intermediate node which is sending untrusted data.

B. Distributed Intelligence within the Network

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, at time t, data items, trust values from leaf nodes are collected by an intermediate node collecting data from them. The possible actions that can be taken at time (t+1) will be determined by the Finite State Machine in Figure 4. 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.

C. Overall Architecture

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 trust and provenance vectors can change over time.

Let L be the set of all leaf nodes. 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 are named as $N_{1,i}$ and the nodes that collect information from the first level nodes are named as $N_{2,k}$.

Let l_i be a leaf node in set L. When a leaf node 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 [16]. 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_i) onto the first level fusion node $N_{2,i}$. The trust value is computed by a function using the trust vector and provenance vector. We can denote the trust computation as follows. $< t_{1,i} >= f_{1,i}(d_i, P_i, T_i)$ If it is a sensor, since a sensor-type leaf node may have limited storage and processing power, the trust vector may have parameter values of a distribution e.g. a noise component that is $N(0, \sigma)$, with a pre-computed σ.

(d_i, t_i) tuple is sent to the first level fusion node. The fusion

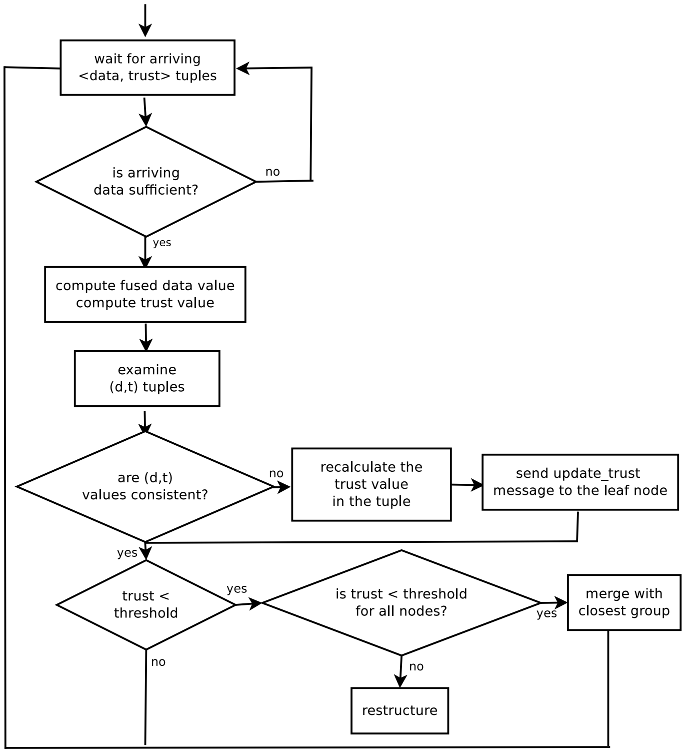

Fig. 4. *Finite state machine housed in intermediate node.*

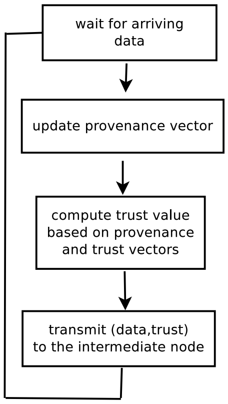

Fig. 5. *Finite state machine housed in leaf lode.*

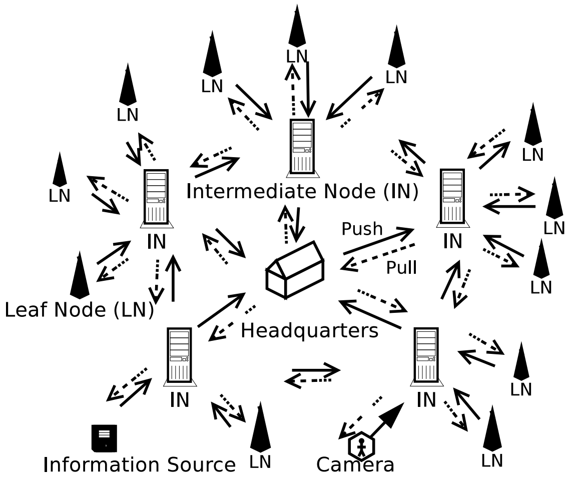

Fig. 6. *Overall architecture.*

node uses its trust and provenance vectors to first determine its revised trust value. It then compares trust values and data that arrives from its children based on the fidelity of the transmission path (source to destination) according to expectation of correlated values. 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 fusion node calculates its trust in its fused/computed value before transmitting the new (data, trust) tuple to a higher-level intermediate node.

The next level fusion node follows the same steps. It has a larger picture of the correctness of the data and so can influence either level below it. Finally the root node R can do all of the above and its input becomes intelligence for the user.

D. Dataflow Provenance, Network Restructuring

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, 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 8. There are five predefined causal relationships *used, wasGeneratedBy, wasTriggeredBy, wasDerivedFrom, wasControlledBy* and edges are labeled with one of these causal relationships.

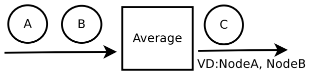

Fig. 7. *node C depends on values of node B and node A.*

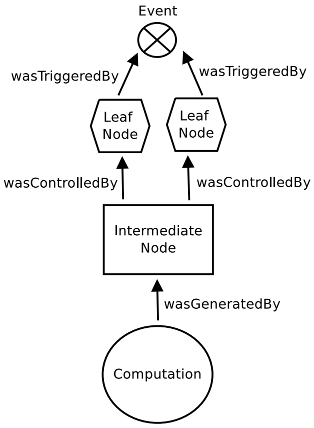

Fig. 8. *An example of a provenance graph.*

Provenance information of "on which leaf nodes does data depend" is referred as data dependency [3]. 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 7. The node id and group id information in the provenance vector in Figure 2 form the data dependency.

To illustrate our concepts, we make use of the example of a field of proximity binary sensors. In proximity-based wireless sensor networks, the likelihood of the target position is calculated using the binary values reported by proximity binary sensors. The sensors should be able to tell that there are k intruders and depending on the density give a reasonable location of each of them. A proximity sensor acts as a tripwire i.e. it reports a detection when a target close by triggers it. Examples of these sensors are seismic, acoustic, passive, infrared and they can be deployed in large numbers because of their low cost. The binary proximity behavior in sensors is achieved by implementing simple energy detection algorithms where the signal is compared to a threshold. If the signal exceeds the threshold, the sensor node reports a "1" meaning a detection, otherwise a "0" is reported for no detection. A network of such sensors can be used to localize and track targets[11]. This type of a network can differentiate the k intruders and locate them. Provenance data is captured in our architecture as a support for the trustworthiness of the target localization. The physical network contains sensors, fewer computation nodes and a central node but these are interspersed within the sensor field. Dataflow model of the target localization scenario is illustrated in the Figure 9 below based on Open Provenance Model standard.

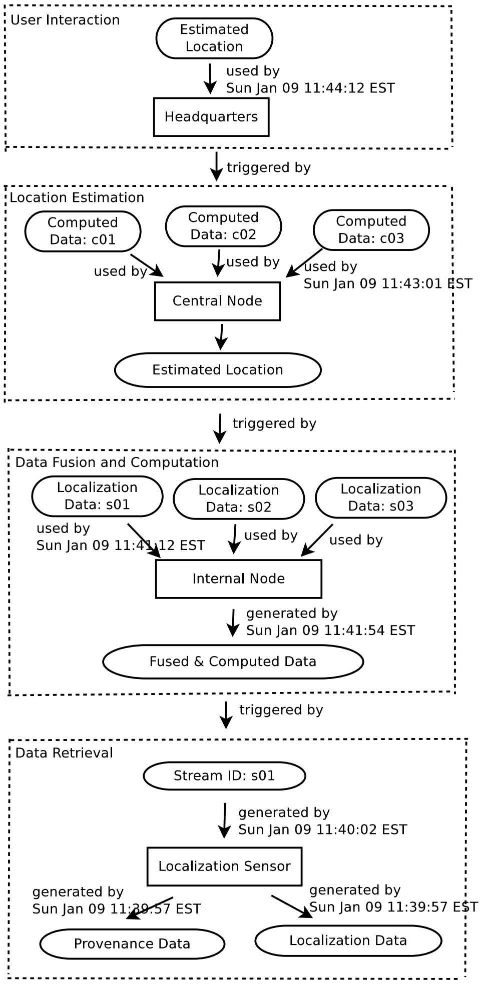

Fig. 9. *Open Provenance Model of the Target Localization.*

E. Benefit of Our System

Our system is superior to traditional information networks in creating more accurate results. Information 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.

IV. Conclusion

We have presented an architecture which uses provenance to enhance trust by restructuring the network in a distributed manner. Using provenance in trust assessment is a new research area with many open questions. We designed this architecture as the initial step of our work. Next research direction we will take is to build a real-life network with this architecture and to run simulations to assess trust of the network. We will compare the trust of our architecture with a real-life network system.

Acknowledgment

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official polices, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copy right notation here on.

References

[1] J. Cheney. Causality and the semantics of provenance. *Arxiv preprint arXiv:1004.3241*, 2010.

[2] 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.

[3] 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.

[4] S. Davidson and J. Freire. Provenance and scientific workflows: challenges and opportunities. In *SIGMOD Conference*, pages 1345–1350. Citeseer, 2008.

[5] J. Golbeck and A. Mannes. Using trust and provenance for content filtering on the semantic web. In *Proceedings of the Models of Trust for the Web Workshop*. Citeseer, 2006.

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

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

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

[9] B. L. Luc Moreau. The first provenance challenge. *Concurrency and Computation: Practice and Experience*, 2007.

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

[11] L. M. K. Qiang Le. Target localization using proximity binary sensors. *Aerospace Conference*, 2010.

[12] S. Rajbhandari, I. Wootten, A. Ali, and O. Rana. Evaluating provenance-based trust for scientific workflows. 2006.

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

[14] P. Victor, M. De Cock, C. Cornelis, and P. da Silva. Towards a provenance-preserving trust model in agent networks. In *Proceedings of Models of Trust for the Web, WWW2006 Workshop*. Citeseer, 2006.

[15] J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. Citeseer, 2005.

[16] 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, 2008. MILCOM 2008. IEEE*, pages 1–7. IEEE, 2008.