A Framework for Secure Knowledge Management in Pervasive Computing

Sheetal Gupta, Anupam Joshi, and Tim Finin
University of Maryland, Baltimore County
1000 Hilltop Circle
Baltimore, Maryland 21250
Email: {sheetal4, joshi, finin}@cs.umbc.edu

***Abstract*—A feature common to many pervasive computing scenarios is that devices acquire information about their environment from peers through short-range ad-hoc wireless connections and use it to maintain a model of their current context. A fundamental issue in such situations, is that knowledge obtained from peer devices may vary in reliability with devices providing incorrect data either inadvertently out of ignorance or other limitations or intentionally in pursuit of malicious or self-serving goals. We describe a heuristic based on a Bayesian approach to infer which of the received answers is most likely to be correct. The suggested answers and the reputation values of the sources themselves are used to determine the most likely answer. We have implemented the techniques and evaluated them in a prototype system using the Glomosim network simulator, and show that our scheme improves data accuracy in low trust environments.**

I. Introduction

In pervasive computing environments, devices can be both producers and consumers of data. A device cannot always rely on a central, trusted source for information and knowledge, but must rely on information available from peers connected over mobile ad-hoc networks (MANETs). Scenarios such as first responders reacting to an event, vehicular ad-hoc networks, and battlefield information management are classic examples. Several researchers have proposed techniques for managing data and information in such environments based on the idea that peer devices cache information and cooperate (see for instance [1], [2], [3])

In such situations, the information provided by peer devices may not be reliable. This could be due to the presence of malicious devices in the network or simply due to their ignorance. Peer-provided data cannot benefit from the security mechanisms available in a client-server environment. We note that in our scenarios, devices are moving, and the underlying network topology (and hence the peers of any given node) are changing over time. This makes the problem different from many of the agreement protocols commonly studied in distributed systems.

We propose a new technique to infer the most accurate data from the different versions of the same data provided by peers. Our approach is a heuristic based on probability theory. There is a growing body of work in reputation management schemes that seek to use the past behavior of a peer to establish its reputation, for instance [4], [5], [6]. The reputation value is an indicator of both the trustworthiness and capability of the device. We choose the data which is most likely to be correct using the provided data along with the reputations of their sources. This technique aims to reduce the risk of propagating incorrect data in the network.

We illustrate the applicability of our technique in a real-life scenario. Soldiers on the battlefront carrying mobile handheld devices with wireless capabilities is a scenario where it is useful to cache the latest information obtained from peer devices. This includes information about supplies, enemy strength, strategic planning etc. In such tactical environments a central trusted authority is lacking and connectivity is volatile. Using a validation technique to verify data ensures credibility of data.

II. Related Work

Jonker *et al.* [7] propose a formal framework for trust evolution. They propose a mathematical model for trust management in multi-agent systems. The trust function is based on intial trust, experiences and trust dynamics. Perich *et al.* [8] propose a distributed, mathematical model for trust and belief management in mobile ad hoc networks. The model categorizes devices as reliable and unreliable. Several trust learning functions are described based on experience and recommendations from peers. The devices perform information source discovery and combine the suggested answer accuracy degree and reputation of sources to decide on the final answer. The devices accept the answer whose accuracy level is above a threshold value and is the highest among the received answers. Simple ways of combining accuracies of different versions of an answer are used like taking the maximum, minimum and average accuracy. Our focus is not on how trust relations evolve, rather our approach works on top of a trust evolving mechanism. Moreover the model proposed by Perich *et al.* often concluded on the incorrect answer in highly dishonest environments. Our approach works reliably in such environments.

An approach is described by Patwardhan *et al.* [9] in which a few nodes are trusted a priori and data is validated either using agreement among peers or direct communication with a trusted node. Collaborative propagation of reliable data helps in improving the timeliness of data. Bad nodes are detected when the data they provide is invalidated by the validation algorithm. Consensus is achieved when the number of copies

agreeing is greater than a threshold value. The reputations of the devices are not considered when determining the consensus answer. We use this approach as a base line to compare the performance of our validation algorithm.

III. Background

We wish to infer the most accurate data from the different versions provided by our peers. A simple approach would be to accept the answer from the highest reputation node and reject the other answers that were received from lower reputation nodes. However, this has the disadvantage of excessive reliance on the reputation values. For example if we received answer 0 from a node with reputation 0.9 and the answer 1 from three other neighbors with reputations only slightly less, 0.83, 0.85 and 0.87. In this case, it is intuitive to believe the answer provided by three highly reputed nodes.

Another approach is to find the answer based on majority agreement as studied in [9]. An answer is accepted after the number of nodes that agree on an answer becomes greater than some threshold value. However, this approach completely ignores the reputations, leading to compromising on data reliability in low trust scenarios.

We propose an approach that takes into account the reputations as well as consensus to decide the most accurate answer. Simulation results show that the approach for data validation presented here performs very well in terms of accuracy in low trust scenarios. However the bayesian approach we use requires some assumptions.

A. Assumptions

Our approach makes a number of assumptions which we sketch and motivate here.

Finite answer sets. A node choose an answer from a finite set of possible answers. This assumption is reasonable when the answer is boolean, or from a small set ("Addresses of all Chinese restaurant within five miles"). It is not true for some class of queries, such as those where the answer is a real number ("What is the current temperature").

One right answer. Each query has only one correct response. It is also easy to imagine queries or applications where more than one distinct answer might be correct (e.g., "Addresses of three nearby restaurants") or there may be more than one way to encode an answer with the same meaning. We simplify the model by leaving this problem for future work.

Uniform probability of correctness. Each potential answer is equally likely to be the correct one. In real life scenarios, typically the initial probabilities are not known and this is a reasonable assumption to make. If the priors are known, they can be incorporated in the formula.

Reputation. A node's reputation value is used as a measure of the probability that it gives the right answer. This is a reasonable assumption, since the reputation evolution mechanism assigns reputation values based on how correct the node has been in the past. A higher value indicates a node having a positive history. Such a node is more likely to provide the correct answer in future.

No collusion. The nodes do not collude with each other, i.e., they respond to queries independently. Having a collusion with multiple participants is difficult to achieve in practise and is thus a rare occurrence. Also, the nodes validate data before propagating it further. Hence they will not propagate incorrect data obtained from malicious peers, which can be mistaken as collusion. Collusion is handled by our validation algorithm if the participating nodes have negative histories and thus low reputations.

IV. Proposed approach

Let the reputation value of node n_i be denoted by r_i. Let the actual answer be A_A and A_i be the answer returned by node n_i. Then we assume the source node reputation indicates the probability that the received answer A_i is equal to the actual answer, which has value equal to the constant c.

$$\Pr[A_i = c | A_A = c] \quad = \quad r_i$$

The probability with which node i gives the incorrect answer is equal to $1 - r_i$. There are two cases in calculating the probability that node n_i gives the particular incorrect answer A_i.

If the answer can take only binary values viz. either 0 or 1, then the probability that a node lies is given by $1 - r_i$. Thus,

$$\Pr[A_i \neq c | A_A = c] \quad = \quad (1 - r_i)$$

The second case makes a closed world assumption for input, so that the node can choose the answer from a set of k distinct answers. The answer range is given by $\{c_1, c_2, c_3, \ldots, c_k\}$. Then,

$$\Pr[A_i \neq c_x | A_A = c_x] \quad = \quad (1 - r_i)/(k - 1)$$

However for most queries in practise, it is not possible to get a reasonable estimate for k. Moreover, for some class of queries, a node can give the incorrect answer in infinite number of ways. The answer range size, $k \to \infty$. Thus,

$$\Pr[A_i \neq c | A_A = c] \to 0$$

We explored the first two cases of binary and finite answer range size and present the results in the simulation section. We assume that the possible answer space is discrete. They also belong to the same domain space since they are answers to the same question. Let A_1, A_2, A_3 be the answers returned by nodes n_1, n_2, n_3 and their reputations be r_1, r_2 and r_3 respectively.

Using Bayes theorem, probability that the actual answer is equal to A_1 given the received answers from our neighbors is,

$$\begin{aligned} & Pr[A_A = A_1 | (A_1, A_2, A_3)] \\ & = \frac{Pr[(A_1, A_2, A_3) | A_A = A_1] * Pr[A_A = A_1]}{Pr[(A_1, A_2, A_3)]} \end{aligned}$$

Then the ratio of the probabilities of the actual answer to be equal to A_2 to it being equal to A_1, given the replies from

our neighbors is,

$$\frac{Pr[A_A = A_2|(A_1, A_2, A_3)]}{Pr[A_A = A_1|(A_1, A_2, A_3)]}$$
$$= \frac{Pr[(A_1, A_2, A_3)|A_A = A_2]}{Pr[(A_1, A_2, A_3)|A_A = A_1]} * \frac{Pr[A_A = A_2]}{Pr[A_A = A_1]} * \frac{Pr[(A_1, A_2, A_3)]}{Pr[(A_1, A_2, A_3)]}$$
$$= \frac{Pr[(A_1, A_2, A_3)|A_A = A_2]}{Pr[(A_1, A_2, A_3)|A_A = A_1]} * \frac{Pr[A_A = A_2]}{Pr[A_A = A_1]}$$
$$= \frac{Pr[(A_1, A_2, A_3)|A_A = A_2]}{Pr[(A_1, A_2, A_3)|A_A = A_1]}$$

(assuming equal initial probabilities)

We define relative likelihood that the actual answer is equal to A_1 as,

$$R[A_A = A_1|(A_1, A_2, A_3)] = Pr[(A_1, A_2, A_3)|A_A = A_1]$$

A heuristic to determine which of the received answers is most likely to be the true answer is to compute the relative probabilities $R[A_A = A_1|(A_1, A_2, A_3)]$, $R[A_A = A_2|(A_1, A_2, A_3)]$ and $R[A_A = A_3|(A_1, A_2, A_3)]$ and choose the answer with the maximum relative correctness probability. Generalizing for $A_A = A_i$,

$$R[A_A = A_i|(A_1, A_2, A_3)] = Pr[(A_1, A_2, A_3)|A_A = A_i]$$

Assumption: Replies from the neighbors are conditionally independent i.e. they do not collude while replying. Then,

$$R[A_A = A_i|(A_1, A_2, A_3)]$$
$$= Pr[A_1|A_A = A_i] * Pr[A_2|A_A = A_i] * Pr[A_3|A_A = A_i]$$

Case 1: $A_A = A_1$ and $A_1 \neq A_2 \neq A_3$. Of the returned answers only A_1 equals the actual answer.

$$R[A_A = A_1|(A_1, A_2, A_3)]$$
$$= Pr[A_1 = A_A|A_A = A_1] * Pr[A_2 \neq A_A|A_A = A_1] * Pr[A_3 \neq A_A|A_A = A_1]$$
$$= r_1 * (1 - r_2) * (1 - r_3)$$

Using the approach where the probability with which a node gives the incorrect answer as $(1 - r_i)/(k - 1)$, we get

$$R[A_A = A_1|(A_1, A_2, A_3)]$$
$$= Pr[A_1 = A_A|A_A = A_1] * Pr[A_2 \neq A_A|A_A = A_1] * Pr[A_3 \neq A_A|A_A = A_1]$$
$$= r_1 * \frac{(1 - r_2)}{(k - 1)} * \frac{(1 - r_3)}{(k - 1)}$$

Case 2: $A_A = A_1$ and $A_1 = A_2 \neq A_3$.Of the returned answers both A_1 and A_2 agree on the actual answer.

$$R[A_A = A_1|(A_1, A_2, A_3)]$$
$$= Pr[A_1 = A_A|A_A = A_1] * Pr[A_2 = A_A|A_A = A_1] * Pr[A_3 \neq A_A|A_A = A_1]$$
$$= r_1 * r_2 * (1 - r_3)$$

Using the second approach,

$$R[A_A = A_1|(A_1, A_2, A_3)]$$
$$= Pr[A_1 = A_A|A_A = A_1] * Pr[A_2 = A_A|A_A = A_1] * Pr[A_3 \neq A_A|A_A = A_1]$$
$$= r_1 * r_2 * \frac{(1 - r_3)}{(k - 1)}$$

Case 3: $A_A = A_4$ and $A_1 \neq A_2 \neq A_3 \neq A_4$ This represents the case where none of the answers returned by the current neighbors is the actual answer.

$$R[A_A = A_4|(A_1, A_2, A_3)]$$
$$= Pr[A_1 \neq A_A|A_A = A_4] * Pr[A_2 \neq A_A|A_A = A_4] * Pr[A_3 \neq A_A|A_A = A_4]$$
$$= (1 - r_1) * (1 - r_2) * (1 - r_3)$$

Using the second approach,

$$R[A_A = A_4|(A_1, A_2, A_3)]$$
$$= Pr[A_1 \neq A_A|A_A = A_4] * Pr[A_2 \neq A_A|A_A = A_4] * Pr[A_3 \neq A_A|A_A = A_4]$$
$$= \frac{(1 - r_1)}{(k - 1)} * \frac{(1 - r_2)}{(k - 1)} * \frac{(1 - r_3)}{(k - 1)}$$

We compare the probabilities thus computed and choose the answer having the maximum probability value. If the probability that none of the returned answers is the actual answer, is the greatest, we wait till we get the actual answer from our future neighbors. We obtain the answer to a question from at least three neighbors before running the above calculation.

V. Illustrative Examples

We observed that with $(k - 1) = 1$, the algorithm gives more importance to the answer given by high reputation nodes $(r > 0.5)$. With $(k - 1) > 1$, the algorithm believes more in majority agreement.

Example 1: The answers returned by our neighbors are $A_1 = 1$, $A_2 = 0$ and$A_3 = 0$. And their reputations are

$r_1 = 0.25$, $r_2 = 0.75$ and $r_3 = 0.75$. Then we have,

$$\begin{aligned} R[A_A = 0|(1,0,0)] &= (1-r_1) * r_2 * r_3 \\ &= 0.75 * 0.75 * 0.75 = 0.421875 \\ R[A_A = 1|(1,0,0)] &= r_1 * (1-r_2) * (1-r_3) \\ &= 0.25 * 0.25 * 0.25 = 0.015625 \\ R[A_A = x|(1,0,0)] &= (1-r_1) * (1-r_2) * (1-r_3) \\ &= 0.75 * 0.25 * 0.25 = 0.046875 \end{aligned}$$

Here x is a value other than 0 and 1. Thus $A_A = 0$ with a higher probability. Intuitively the answer that is agreed upon by greater number of trusted nodes is chosen above the answer given by fewer distrustful nodes. We reach the same conclusion using $(k-1) = 2$.

Example 2: To illustrate how the validation algorithm behaves in general, we note that the difference in the reputations of the good device and the other devices needed for the good guy to win is a function of how high the reputations of the other devices in consideration are. If we are in a high trust neighborhood, viz. all answers are from devices having reputations > 0.5, then the higher the other devices' reputation, the greater must be difference in reputations between the correct guy and the other guys.

For example with $r_1 = 0.7$, $r_2 = 0.6$ and $r_3 = 0.6$, and $A_1 = 1$, $A_2 = 0$ and $A_3 = 0$, node 1 wins.

However for reputations $r_2 = 0.7$, $r_3 = 0.7$ and values $A_1 = 1$, $A_2 = 0$ and $A_3 = 0$, r_1 must be as high as 0.845 for node 1 to win. Thus the difference in reputations required increases in a high trust neighborhood.

By using $(k-1) = 2$, with $r_2 = 0.6$ and $r_3 = 0.6$, and $A_1 = 1$, $A_2 = 0$ and $A_3 = 0$, r_1 is required to be even higher, viz. $r_1 = 0.82$, for the answer $A_1 = 1$ to win. Intuitively having a k factor increases the importance of majority agreement.

Example 3: Values are $A_1 = 1$, $A_2 = 0$ and $A_3 = 0$. And their reputations are $r_1 = 0.75, r_2 = 0.25$ and $r_3 = 0.25$, then

$$\begin{aligned} R[A_A = 0|(1,0,0)] &= 0.015625 \\ R[A_A = 1|(1,0,0)] &= 0.421875 \\ R[A_A = x|(1,0,0)] &= 0.140625 \end{aligned}$$

Thus when we have one highly trusted neighbor and several low reputation neighbors, the answer given by the highly trusted node is chosen i.e. $A_A = 1$.

We observed that getting an answer from a distrustful node, that contradicts the answer given by a highly trusted node, actually makes it easier to converge on the correct answer given by the highly trusted node. To illustrate, if in the above example another node with a low reputation of $r_5 = 0.25$ gave the answer $A_5 = 1$ matching that from the highly reputed node 1. Then, the probabilities would be,

$$\begin{aligned} R[A_A = 0|(1,1,0)] &= 0.046875 \\ R[A_A = 1|(1,1,0)] &= 0.140625 \\ R[A_A = x|(1,1,0)] &= 0.140625 \end{aligned}$$

This makes the probability of $A_A = 1$ become equal to the probability that the actual answer is not received yet. Thus we wait for it further. So getting the correct answer from a low reputation node can delay converging on the correct answer. But will eventually lead to increasing its reputation, after the answer is proved correct.

On the other hand, with $(k-1) = 2$, we still converge on the correct answer in this case.

$$\begin{aligned} R[A_A = 0|(1,1,0)] &= \frac{(1-r_1)}{2} * \frac{(1-r_2)}{2} * r_3 \\ &= (0.25/2) * (0.75/2) * 0.25 \\ &= 0.01171875 \\ R[A_A = 1|(1,1,0)] &= r_1 * r_2 * \frac{(1-r_3)}{2} \\ &= 0.75 * 0.25 * (0.75/2) \\ &= 0.0703125 \\ R[A_A = x|(1,1,0)] &= \frac{(1-r_1)}{2} * \frac{(1-r_2)}{2} * \frac{(1-r_3)}{2} \\ &= (0.25/2) * (0.75/2) * (0.75/2) \\ &= 0.017578125 \end{aligned}$$

Thus the answer chosen is $A_A = 1$. Here in contrast to the previous case where $(k-1) = 1$, the probability that we do not know the answer yet, is less than that for $A_A = 1$. Intuitively an agreeing answer even from a low reputation node, contributes in choosing that answer.

Example 4: In a low trust neighborhood, even a single answer from a good guy, viz. with reputation > 0.5 is sufficient for the good guy to win. For the border case of $r_1 = 0.51$, $r_2 = 0.49$ and $r_3 = 0.49$, and $A_1 = 1$, $A_2 = 0$ and $A_3 = 0$, we have,

$$\begin{aligned} R[A_A = 1|(1,0,0)] &= 0.132651 \\ R[A_A = 0|(1,0,0)] &= 0.117649 \\ R[A_A = x|(1,0,0)] &= 0.127449 \end{aligned}$$

Thus even in this border case of reputations, the good guy wins i.e. $A_A = 1$, without having a huge difference in reputations above the bad guys.

In the approach using $(k-1) = 2$, the answer $A_A = 0$ wins because it was returned by two nodes having reputations only slightly less than the third node. Thus the emphasis on getting an answer from a node having $r > 0.5$ in order to believe it, is not true anymore. This can be good strategy to tolerate error in reputations. On the other hand the same answer from two low reputation nodes, wins over an answer from a high reputation node. This might lead to compromising on the credibility of data.

The answer validation scheme presented above relies on an reputation evolving scheme to provide the input reputation values. The reputation evolving scheme must give a fairly accurate estimate of peer reputations. As we saw in example 4, the proposed validation algorithm with $(k-1) = 1$, is sensitive to accuracy of reputation values around 0.5. Thus a correct distinction between devices as good $(r > 05.)$ or bad $(r < 0.5)$ is sufficient to yield the correct answer. In a high reputation environment, if the reputation values are incorrect and the peers provide different versions of answers,

TABLE I
SIMULATION PARAMETERS

Spatial Dimensions	700 m x 900 m
Simulation period	30 min
Mobiles devices	50,100,150,200
Stationary devices	38
Transmission range	99.472 m
Routing Protocol	AODV
Mobility pattern	Vehicular trace
Cache size	10

it might take longer to converge on the correct answer. It might even lead to concluding on the incorrect answer if a sufficient number of incorrectly, highly reputed nodes agree on the wrong answer.

VI. SIMULATIONS

We implemented our validation algorithm using the Glomosim simulator. The information in this section was generated using the simulation parameters mentioned in Table 1. The experiment was run with 50 mobile nodes and 38 "pre-trusted" anchor nodes, for a duration of 30 minutes. A mobility pattern of vehicular movement was chosen, with speeds ranging from 15 m/s to 25 m/s and pause times of 0 to 30 s. Each anchor node has a list of 5 answers that it will seed into the network every 2 minutes. Each mobile node has a set of 5 queries that it wishes to get answered and broadcasts to all nodes in range after every 1 minute. The mobile nodes are assigned queries and anchor nodes are assigned answers in a random uniform distribution pattern.

The percentage of bad nodes in the network was increased from 0% to 100% in steps of 10%. Reputation values are assigned to the nodes initially and do not change during the simulation. The good nodes are assigned reputation values > 0.5 and bad nodes are assigned values < 0.5. The bad nodes also run the validation algorithm to obtain the correct answer. However they push incorrect answers into the network. The good nodes too perform the validation step, but only push validated answers into the network. The effect on the number of correct answers obtained in the network, the answering latency and total traffic in the network was observed as the fraction of bad nodes is increased in the network. The performance of this validation algorithm is compared to that of the validation algorithm described in [9]. The reputation of the node represents the probability that it will answer correctly.

Figure 1 shows the average total number of correct and incorrect answers obtained in this setup. It shows that our validation algorithm performed consistently better in terms of total number of correct answers obtained than a threshold validation approach. The number of correct answers obtained dropped significantly as the percentage of bad nodes was increased beyond 60%. This is because in a low trust neighborhood, most of the received answers come from distrustful sources. Thus the validation algorithm calculated that probability of the answer not being known as the highest. Hence most

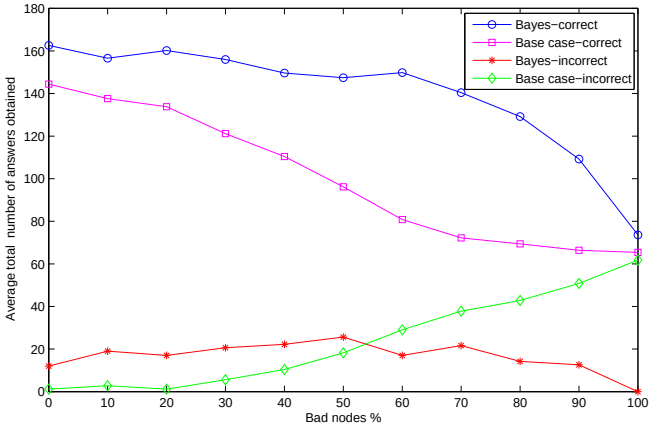

Fig. 1. Number of correct and incorrect answers obtained as the percentage of malicious nodes in the network is increased.

queries remained unanswered though they received multiple copies/versions of answers from the bad nodes.

It concluded on the wrong answer more often in high reputation environments, than the base case algorithm. This is because when a high reputation node gives the incorrect answer, the proposed algorithm will choose it as the correct answer, since it relies on correctness of reputation values. The base case algorithm does not consider the reputations at all, but relies on reaching a threshold agreement. So it results in fewer incorrect answers in high-trust scenarios. In such environments where most peers are highly trusted, majority agreement is probably better approach. However, our algorithm performed better in low-trust environments. The number of incorrect answers obtained was fewer than the base case in such environments and reached a value of 0 at 100% bad nodes. At 100% bad nodes, the only way to get an answer is by direct encounter with the source of data and not through the cache of any peer devices.

We implemented the second approach where the probability with which a node gives the incorrect answer is equal to $(1 - r_i)/(k - 1)$. Figure 2 shows the number of incorrect and correct answers obtained when there were 50% bad nodes in the network, for different values for $(k - 1)$. We varied $(k - 1)$ from 1 which is the same as in previously mentioned experiments, up to 10. We also tried to estimate k by counting the number of unique answers we obtain for a given query. We observe from the figure that the number of correct answers is the maximum at $(k-1) = 2$, and the number of incorrect answers is also the least in this case. Otherwise the performance for all other values of $(k - 1)$, is worse than the base case where we divide by 1, with more incorrect answers and fewer correct answers.

Figure 3 shows the number of correct and incorrect answers obtained using both the approaches as the number of bad nodes in the network are increased. We observe from figure that the performance is almost the same up to around 50% bad nodes.

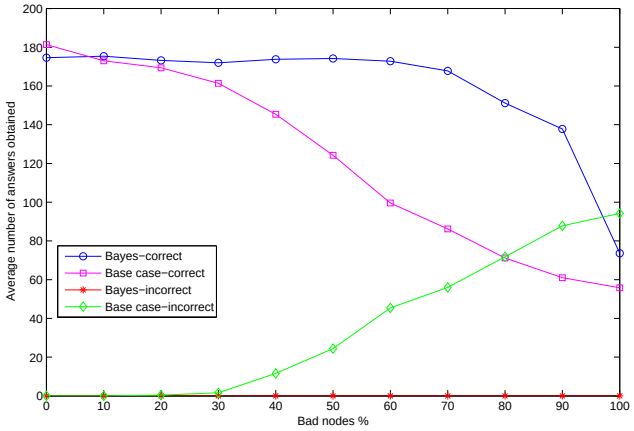

Fig. 2. Number of correct and incorrect answers obtained using the new setup.

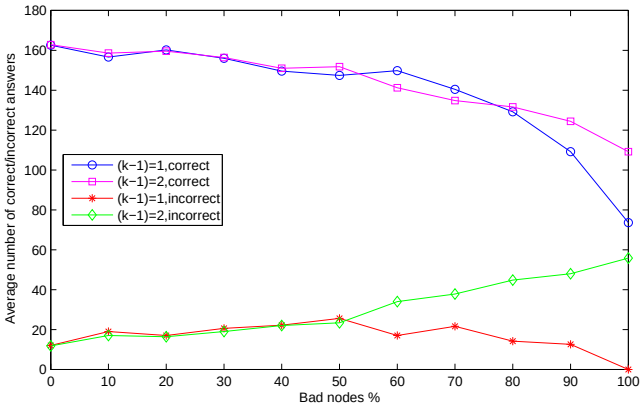

Fig. 3. Comparison of the number of correct/incorrect answers obtained at (k-1)=1 viz. base case and when (k-1)=2.

Greater number of correct answers are obtained using $(k-1)=2$ in highly distrustful environments. This is because in low trust environment, the bad nodes return the correct answers with a small probability. With $(k-1)=2$, the algorithm is able to conclude on the correct answer returned by bad nodes if sufficient number of nodes agree on the correct answer. With $k-1=1$, the algorithm rejects the right answer if it comes from a low reputation node. However, the number of wrong answers also increases for $(k-1)=2$ as compared to $(k-1)=1$. This is due to the same reason that if enough number of nodes agree on the wrong answer, the algorithm takes it to be the correct answer.

VII. Conclusion and Future Work

We proposed a data validation scheme based on Bayes theorem. The nodes consider the data and the data source reputation to determine the most correct answer. We looked at how the algorithm works for different cases of data values and reputations. We observed the performance for the two cases of binary and finite answer range size. Simulation results showed that the algorithm using $k-1=1$ performs very well in terms of reliability and accuracy of data in low trust environments ($> 50\%$ bad nodes). In high trust environments($< 50\%$ bad nodes), majority voting works best in terms of accuracy of data.

For future work we are investigating how to avoid the computation involved in validating other answers received by a device i.e. the answers that do not correspond to any of the queries that the device has. We can do that by assigning an accuracy level to each answer received without going through the validation algorithm. When propagating answers in the network, devices also push their confidence values. This scheme will help in making the right decisions about reputation evolution. The reputation evolution mechanism will not only consider the answer value but also its suggested accuracy while updating reputation values.

Acknowledgment

The authors would like to thank Dr. Suresh Purini and Dr. Yun Peng, UMBC for their help during discussions. Partial support for this work was provided by MURI award FA9550-08-1-0265 from the Air Force Office of Scientific Research.

References

[1] F. Perich, A. Joshi, T. Finin, and Y. Yesha, "On Data Management in Pervasive Computing Environments," *IEEE Transactions on Knowledge and Data Engineering*, May 2004.

[2] L. Yin and G. Cao, "Supporting cooperative caching in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 1, pp. 77–89, January 2006.

[3] B. Xu and O. Wolfson, "Data management in mobile peer-to-peer networks," *2nd International Workshop on Databases, Information Systems, and Peer-to-Peer Computing (DBISP2P'04)*, August 2004.

[4] P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Sixth IFIP conference on security communications, and multimedia (CMS 2002)*, 2002.

[5] J. L. B. S. Buchegger, "Performance analysis of the confidant protocol: Cooperation of nodes: Fairness in distributed ad hoc networks," in *Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, 2002.

[6] J. B. Laurent Eschenauer, Virgil D. Gligor, "On trust establishment in mobile ad-hoc networks," in *Proc. of the Security Protocols Workshop*, April 2002.

[7] C. M. Jonker and J. Treur, "Formal Analysis of Models for the Dynamics of Trust Based on Experiences," in *the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering (MAAMAW-99)*, F. J. Garijo and M. Boman, Eds., vol. 1647. Berlin: Springer-Verlag: Heidelberg, Germany, 30– 2 1999, pp. 221–231.

[8] F. Perich, J. L. Undercoffer, L. Kagal, A. Joshi, T. Finin, and Y. Yesha, "In Reputation We Believe: Query Processing in Mobile Ad-Hoc Networks," in *International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Boston, MA, August 2004.

[9] A. Patwardhan, A. Joshi, T. Finin, and Y. Yesha, "A Data Intensive Reputation Management Scheme for Vehicular Ad Hoc Networks," in *Proceedings of the Second International Workshop on Vehicle-to-Vehicle Communications*. IEEE, July 2006.