Trust based Fusion over Noisy Channels through Anomaly Detection in Cognitive Radio Networks*

Shameek Bhattacharjee
Department of EECS
University of Central Florida
Orlando, Florida-32816, USA
shameek@eecs.ucf.edu

Saptarshi Debroy
Department of EECS
University of Central Florida
Orlando, Florida-32816, USA
saptarsh@eecs.ucf.edu

Mainak Chatterjee
Department of EECS
University of Central Florida
Orlando, Florida-32816, USA
mainak@eecs.ucf.edu

Kevin Kwait
Air Force Research Laboratory
Information Directorate
Rome, NY-13441, USA
kevin.kwiat@rl.af.mil

ABSTRACT

Byzantine attacks have been identified as one of the key vulnerabilities in cognitive radio networks, where malicious nodes advertise false spectrum occupancy data in a cooperative environment. In such cases, the resultant fused data is very different from the actual scenario. Thus, there is a need to identify the malicious nodes or at least find the trustworthiness of nodes such that the data sent by malicious nodes could be filtered out. The process is complicated by presence of noise in the channel which makes it harder to distinguish anomalies caused by malicious activity and those caused due to unreliable noisy channels.

This paper proposes a scheme for trust based fusion by monitoring anomalies in spectrum usage reports advertised over unreliable channels by secondary nodes which leads to evaluation of *trust* of a node by its neighbors. The calculated trust is then used to determine if a neighboring node's advertised data could be used for fusion or not. We provide a heuristic trust threshold for nodes to disregard malicious nodes while fusing the data, which holds good for any probability of attack. A trust coefficient is calculated based on interactions with peers in a distributed manner. Results show that even at higher probabilities of attack (0.8 and above), 95% of the nodes generate fused data with accuracy as high as 84%. We compare our results of trust based fusion with blind fusion scheme and observe improvement in accuracy of fusion from individual nodes' as well as overall network's perspective. We also analyze an alternative weighted trust fusion technique and evaluate its performance. We find that at lower probabilities of attack a malicious node's contribution to the overall gain in cooperation is more than the damage done. We observe that above a critical value for probability of attack of 0.40, the overall gain in cooperation is compromised if the malicious nodes are considered in fusion. We also discover that an honest node's benefit due to cooperation depends on its relative position with respect to the spatial orientation of malicious nodes.

Categories and Subject Descriptors

C.2.0 [**Computer Communication Networks**]: General-*Security and Protection*; C.2.1 Network Architecture and Design - *Distributed Networks, Wireless communication*; C.4 [**Performance of Systems**]: *Fault Tolerance*

General Terms

Algorithms, Performance, Security, Theory

Keywords

Cognitive radio networks, attacks, anomaly detection, trust coefficient, fusion

1. INTRODUCTION

In cognitive radio (CR) paradigm, unlicensed (secondary) users opportunistically operate on parts of spectrum in absence of primary (licensed) users [10]. The primary regulatory aspect of this paradigm is that unlicensed CR nodes should relinquish their allocated channels and move to another available channel as soon as they are able to *learn or sense* the presence of licensed user on that channel. To reduce uncertainty in true spectrum map caused due to wireless characteristics like multi-path fading, shadowing, etc, these nodes engage in cooperative or collaborative spectrum sensing [3, 13], where nodes share their locally sensed reports, and the final inference on spectral occupancy is a fusion of multiple locally sensed spectrum reports. However, this feature introduces a vulnerability where the utility of cooperative sensing is crippled due to Byzantine attacks [1, 8], where malicious nodes advertise altered local sensing results. The motive behind such behavior may be selfish (gaining more spectrum resources) or malicious (crippling the operation of other nodes). Also locally sensed reports when send

*This research was supported by National Science Foundation, under award no. CCF-0950342, Air Force Office of Scientific Research (AFOSR), and National Research Council. Approved for Public Release; Distribution Unlimited: 88ABW-20113832, 11JUL11.

SIN'11, November 14–19, 2011, Sydney, Australia.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
NOV 2011	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Trust based Fusion over Noisy Channels through Anomaly Detection in Cognitive Radio Networks	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Department of EECS University of Central Florida Orlando, Florida-32816, USA**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADA553912. International Conference on Security of Information and Networks (4th) (SIN 2011) Held in Sydney, Australia on November 14-19, 2011. Approved for public release; U.S. Government or Federal Purpose Rights License., The original document contains color images.

14. ABSTRACT
Byzantine attacks have been identified as one of the key vulnerabilities in cognitive radio networks, where malicious nodes advertise false spectrum occupancy data in a cooperative environment. In such cases, the resultant fused data is very different from the actual scenario. Thus, there is a need to identify the malicious nodes or at least find the trustworthiness of nodes such that the data sent by malicious nodes could be filtered out. The process is complicated by presence of noise in the channel which makes it harder to distinguish anomalies caused by malicious activity and those caused due to unreliable noisy channels.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**8**	

to neighbors (in ad-hoc CR network) or to a central fusion center (in infra-structured CR network) may be altered due to channel noise regardless of malicious behavior. One way to track the occurrence of such attacks is to monitor anomalies in received spectrum reports, but this process is complicated by the fact that they are garbled by channel noise. Since such actions are hard to detect, either the malicious nodes remain in the network undetected, or honest nodes get penalized.

As far as countering Byzantine attacks are concerned, there are two approaches: isolating malicious nodes [7] or robust fusion [1]. The current literature mostly focusses on infra-structured CR network where a central fusion center fuses advertised local reports from individual nodes of the network. Anomaly detection is a feature that may be used for tracking possible malicious misbehavior. For example, anomalies in packet forwarding to unravel routing misbehavior [9]. Detected anomalies against a node can be used as an evidence that decides the trustworthiness of that node. The authors in [12], discuss a concept of moulding success and failure counts in packet forwarding into trust evidence from which *trustworthiness in relation to routing* is calculated. Presence or absence of packet forwarding anomalies provide success and failure counts which are used to build and manage trust in a traditional ad-hoc network. For trust management in CRN paradigm, authors in [2, 11], provide a framework that builds and maintains trust metrics based on beta distribution in a centralized CR network. Both of these works assume that success and failure counts are known, and do not provide a way to decide for success or failed cooperation which is utilized to calculate and maintain trust. The question on how to build a concrete trust evidence in cooperative CR networks is an unsolved problem. This motivates us to provide a comprehensive framework which gathers trust evidence based on monitoring anomalies in advertised local spectrum reports in an ad-hoc CR network, subsequently leading to calculation of a fair trust coefficient associated with each node. The trust is used as a metric that decides whether a particular node's advertisement is used in the fusion or not.

In this paper, we propose a trust based fusion model ensuring robust fusion for individual nodes in coexistence with malicious Byzantine nodes in an ad-hoc CR network. The nodes need not know the locations of other nodes which also eliminates the possibility of nodes advertising false location. All nodes share their individual sensing reports with their neighbors, and in the absence of a fusion center, each node is expected to fuse the reports advertised by its neighbors. Each node evaluates a trust coefficient for all its neighbors based on anomalies detected in sensing reports. This method provides upper and lower limits on possible received power levels on the channels of the operating spectrum. Then normalization criteria is used to build a predicted spectrum occupancy report for a particular node. This is then compared with that node's actually advertised binary spectrum report for mismatches which are termed as *anomalies*. The mismatches and matches form the trust evidence based on which fair trust coefficient is calculated. Higher trust value signifies more trustworthiness. To provide fairness due to anomalies caused by unreliable wireless channel properties, we define two models: an instantaneous fair trust model and a cumulative trust model. These provide two possible ways to provide fairness so that trustworthiness of honest nodes is not penalized. Furthermore, we provide a trust based fusion scheme that allows individual nodes to disregard malicious nodes data from fusion thus making it robust. It may be noted that the goal is not to isolate malicious nodes but to provide a robust and fair fusion in presence of malicious nodes.

To validate our proposed model, we perform simulation experiments on a customized simulator. We find that our proposed framework ensures that malicious nodes have trust values lesser than those of honest nodes which is independent on the intensity/severity of attack launched by malicious nodes. We provide a heuristic trust threshold for selective trust based fusion, below which a neighbors advertised report may be disregarded. We compare the results of the trust based fusion with blind fusion to establish the effectiveness of the proposed fusion scheme. We find that spatial distribution of malicious nodes may play a role which may cause exceptions for a few nodes. To validate the exceptions we analyze the performance both from individual node as well as overall network perspective, and show that most of the discrepancies are caused by certain spatial distribution of malicious nodes. We also find that at lower probabilities of attack the contribution of a malicious node to the entire network in cooperative sensing is more than the damage done. Thus, including them for fusion does more good than harm to the network overall. We also observe that beyond a critical probability of attack (0.40), the overall gain in cooperation decreases if malicious nodes are included in fusion.

The rest of the paper is organized as follows. Section 2 describes the system model and the assumptions. Section 3 discusses the proposed framework for monitoring cooperation behavior, calculation of trust coefficient and subsequently proposes a trust based fusion scheme. Simulation model and results are discussed in Section 4. Conclusions are drawn in the last section.

2. SYSTEM MODEL

We assume all secondary nodes continuously undergo spectrum sensing to decide whether a channel is occupied by primaries or not. Let us assume secondary node i constructs its observed occupancy vector as: $B^i_{act} = [d_1, d_2, \cdots, d_n]$, where d_k is 1 or 0 depending on whether the channel is occupied or unoccupied, and n is the number of channels being monitored. Once this binary vector is created, a secondary node would broadcast this information to its neighboring nodes. Similarly, a secondary node would also hear broadcast messages (binary occupancy vectors) from its neighbors. Based on the vectors a node receives, the node will employ a fusion technique to obtain a better estimate about the spectrum usage that can significantly improve the performance of spectrum sensing [1, 4]. Such cooperative sensing has other benefits such as mitigating the shadowing and multi-path effects.

We consider that the malicious nodes do not report their occupancy vectors truthfully; rather they inject errors in their occupancy vectors by flipping the bits in the vector. Flipping 0 to 1 implies that the channel is occupied when in reality it is unoccupied. Flipping 1 to 0 implies that an occupied channel is reported as unoccupied. We denote probability of attack P_{attack}, as the percentage of channels that a malicious node changes from its actual observed vector.

2.1 Assumptions

1. We consider an ad-hoc secondary network with N nodes with γ_{mal} fraction of nodes being malicious; H is the set of honest nodes and M malicious/dishonest nodes. We assume $\eta(M) < \eta(H)$, since in a realistic network, the number of malicious nodes is less than number of regular honest nodes. The secondary network has no dedicated central fusion center or allocation authority, and each individual node fuses the spectral sensing data it receives from other nodes from which it can hear from and forms its opinion on the availability of spectrum.
2. The nodes are not aware of the geographical coordinates of other nodes involved in cooperation. We assume the transmit power level of all secondary nodes are same. Knowledge of the transmitter output power, channel losses, and antenna gains with the appropriate path loss model allows us to find the distance between the two nodes using Received Signal Strength through localization or lateration. The location or identity awareness is not required. It prevents the nodes lying about location, and also reduces maintenance overhead.
3. Unlike in [7], which discusses a more restrictive fusion model (AND fusion rule), we use majority voting fusion rule, which gives more flexibility towards errors committed by nodes and/or malfunctioning nodes.
4. Each primary transmitter *whether it chooses to transmit or not*, transmits only on one channel; so the channel associated with a primary transmitter is known. The primary transmitter that transmits on channel k, is referred as T_k, and since it is fixed, its coordinates (x_{T_k}, y_{T_k}) are known to the nodes.
5. We assume nodes use some interference aware channel access framework, so that they do not interfere with other secondary nodes who are using the same channels sensed as unoccupied. Interference awareness is outside the scope of this paper.
6. We consider that reporting of spectrum sensing data, take place over noisy links, hence the observation and monitoring processes under imperfect conditions.
7. We consider independent attacks and do not consider collaborative Byzantine attacks. We do not delve into rationale of attacks. No matter what the motive is, we call them malicious nodes.
8. The outcome of local sensing is raw energy values which are converted into a binary vector of 0's and 1's, where 0 represents absence of primary and 1 represent primary's presence.
9. The probability of false alarm is the probability that a channel which is actually empty (H_0) is erroneously detected by a node to be occupied, and is denoted by P_f or $P(H_1|H_0)$. Similarly, the probability of missed detection is the probability that a channel which is occupied (H_1) is not detected by a node and is denoted by P_m or $P(H_0|H_1)$. It is traditionally the channel between the primary and the CR node. There is body work that deals with calculation of such probabilities [5].

3. TRUST BASED FUSION MODELS

In this section we discuss gathering of trust evidence based on unraveling anomalies in spectrum sensing data reported by neighbors of a node in an ad hoc CR network. We predict the bounds of received power and use a normalization criterion to predict occupancy vector, which is then compared with advertised vectors sent by neighbors j of node i. This comparison is recorded as the trust evidence which is then modeled into a numerical trust coefficient, reflecting the trustworthiness of a neighbor. However observations might be erroneous due to noise, and to certain extent anomalies may be caused by unreliable channel conditions like noise rather than malicious behavior. We provide two approaches of computing fair trust coefficient to counter noise. Subsequently we propose a trust based filtered fusion scheme for robust and secure cooperative spectrum sensing.

Table 1: Notations

Symbol	Meaning
N_i	Neighbor set of node i
H	Set of honest nodes
M	Set of malicious nodes
γ_{th}	Common threshold used to normalize power vectors
$s_{T_{ik}}$	Dist. between node i and tower T_k for channel k
d_k	Binary Decision on a channel k, $d_k \in 0, 1$
j	Set of all neighbors of i, $j \in N_i$
P^i	Measured power vector on n channels at node i
B^i_{act}	Actual binary occupancy vector formed at i
B^i_{adv}	Advertised binary occupancy vector by node i
$P^{ij}_{predict}$	Vector of power ranges for neighbor j predicted by i
D^j_i	Binary occupancy of node j, predicted by i
$d^j_k\|_{predict}$	Predicted decision on any channel k, for D^j_p
$(\alpha, \beta, X)^j$	Three tuple trust evidence
$E^j_{trust_i}$	Reputation or trust of neighbor j calculated by node i
BF^i_{blind}	Fusion result at node i, when all j are included
TBF^i	Fusion result due to trust based selective inclusion of j

3.1 Bounds on Power Levels and Anomaly Detection

Suppose node i measures the power vector

$$P^i = \{\gamma^i_1, \gamma^i_2, \cdots, \gamma^i_n\},$$

where γ^i_k is the power received on channel k and n is the number of channels. Each node i forms its binary vector B^i_{act} $= \left[\ d^i_1, d^i_2, ...d^i_n\ \right]$ from it's power vector P^i by comparing γ^i_k with threshold γ_{th}, where

$$d^i_k \begin{cases} = 1 & \text{when } \gamma^i_k \geq \gamma_{th} \\ = 0 & \text{when } \gamma^i_k < \gamma_{th} \end{cases} \quad (1)$$

Each node i, advertises a public binary vector B^i_{adv}.

$$B^i_{adv} \begin{cases} = B^i_{act} & \text{if node } i \in H \\ \neq B^i_{act} & \text{if node } i \in M \end{cases} \quad (2)$$

Just the way node i advertises its binary vector, it also hears similar advertisement from its neighbors. For a neighboring node $j \in N_i$, node i estimates its possible power vector using their mutual distance and received signal strength (RSS) localization [14]. Though it is difficult for node i to accurately predict the power vector of node j, nevertheless it can always estimate the lower and upper bounds using RSS models. Let us describe how node i estimates the upper and lower bounds of power vector as $P^{ij}_{predict}$.

Assuming transmit power of all nodes are same, node i calculates the distance between the node j and itself whenever it receives a signal (vector) from it as s_{ij}. Based on the distance s_{ij}, node j may be anywhere on the circle with node i at the center. We draw a straight line from the center of the circle to the primary transmitter T_k located at (x_{T_k}, y_{T_k}) as shown in Fig. 1. Under ideal conditions, the

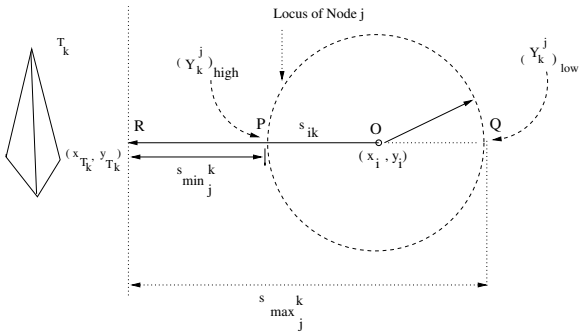


Figure 1: Calculation of Max and Min RSS range on channel k of neighbor node j

RSS due to T_k will be maximum on the circle that is closest to T_k, i.e., on point P and minimum at a point Q that is farthest from T_k. We denote the power levels at these two locations as $[\gamma_k^j]_{high}$ and $[\gamma_k^j]_{low}$ at distances $s_{min_j^k}$ and $s_{max_j^k}$, respectively. For all locations on the circle, the RSS on channel k varies between $[\gamma_k^j]_{high}$ and $[\gamma_k^j]_{low}$.

Using commonly used model for RSS [6], we get

$$\gamma_k^i = P_k \times \frac{A^2}{s_{ik}^{\alpha}}; \tag{3}$$

where A = frequency constant, α is path loss factor, s_{ik} is the distance between T_k and node i, and P_k is the transmit power of T_k. We get the bounds as:

$$[\gamma_k^j]_{high} = P_k \times \frac{A^2}{s_{min_j^k}^{\alpha}}; \tag{4}$$

$$[\gamma_k^j]_{high} = P_k \times \frac{A^2}{s_{max_j^k}^{\alpha}}; \tag{5}$$

Now we divide Eqn. 3 with Eqn. 4 and Eqn. 5. Since s_{ik}, $s_{min_j^k}$ and γ_k^i are known to node i, it is easy to find $[\gamma_k^j]_{high}$ and $[\gamma_k^j]_{low}$. Node j is somewhere on the circular locus. Now the predicted power vector of node j is a 2-tuple vector

$$P_{predict}^{ij} = \left[([\gamma_1^j]_{low}, [\gamma_1^j]_{high}), \cdots, ([\gamma_n^j]_{low}, [\gamma_n^j]_{high})\right].$$

With the estimated power vector being known, the inference drawn by a node j on channel k is,

$$d_k^j|_{infer} = \begin{cases} 0 & \text{if both } [\gamma_k^j]_{low} \text{ and } [\gamma_k^j]_{high} \leq \gamma_{th}; \\ 1 & \text{if both } [\gamma_k^j]_{low} \text{ and } [\gamma_k^j]_{high} \geq \gamma_{th}; \\ X & \text{otherwise} \end{cases} \tag{6}$$

Eqn. 6 is the normalization criterion. When both the lower and higher predicted power levels on a channel are less than γ_{th}, it implies that channel i is not being used by any primary transmitter, i.e., channel is unoccupied. So in this case $d_k^j|_{infer}$ is inferred as 0. Similarly, if both the lower and higher predicted power levels are greater than γ_{th}, $d_k^j|_{infer}$ is inferred as 1. Such inference can be drawn for the above two scenarios. However, no inference can be drawn when one of $[\gamma_k^j]_{low}$ and $[\gamma_k^j]_{high}$ is above γ_{th} and the other is below γ_{th}. We denote such cases as X. Now node i compares $D_i^j = [d_1^j|_{infer}, ..., d_n^j|_{infer}]$ with received $B_{adv}^j = [d_1^j, ...d_n^j]$ on corresponding channels k for matches and mismatches.

A match occurs when on channel k, $d_k^j|_{infer} = d_k^j$. A mismatch occurs if $d_k^j|_{infer} \neq d_k^j$. If $d_k^j|_{infer} = X$ no inference can be drawn. For each neighbor j, node i computes the number of matches, mismatches, and no inference with B_{adv}^j as α_j, β_j, X_j respectively. This forms the trust evidence.

3.2 Instantaneous Fair Trust Model

The logic behind the trust evidence is out of all inferences (0,1) advertised on all channels how many have achieved trustworthiness (α), how many inferences have *not* succeeded to achieve trust (β), and how many inferences are undecided. To account for the X channels where no inference could be drawn, we consider them in the ratio of $\alpha : \beta$. Thus the proportion of matches is updated as $\alpha^j + \frac{X_j}{\alpha^j+\beta^j} \times \alpha^j$. But this does not account for the proportion that may be lost due to mismatches caused due to channel noise. Thus we provide a fair instantaneous trust coefficient.

To account for channel noise, shadowing and fading, we define probability of false alarm as $P_f = P(H_1|H_0)$, probability of missed detection as $P_m = P(H_0|H_1)$, and channel error probability due to noise as P_e.
P_f and P_m probabilities due to sensing inaccuracies when nodes are not able to detect the presence or absence of primary transmission. Thus, local sensed reports are modified. Moreover, when local sensed reports are advertised to the neighbors, they may be altered due to noise between the relevant CR nodes. Considering the channel error probability we define modified false alarm probability for node j vector at the node i as

$$P_{fe}^{'} = (1 - P_f).P_e + P_f.(1 - P_e)$$

and modified missed detection probability, as

$$P_{me}^{'} = (1 - P_m).P_e + P_m.(1 - P_e)$$

$P_{fe}^{'}$ is the probability that a 0 in node $j's$ advertised vector will reach as 1 at node i, irrespective of malicious behavior of node j. Similarly, $P_{me}^{'}$ is the probability that a 1 in node $j's$ advertised vector will reach as 0 at node i, in spite of any malicious behavior. We need to discount these mismatches caused by P_m, P_f and P_e to achieve a fair trust coefficient. Let the actual number of $0's$ and $1's$ in ideal case for any received vector is $x_0^{ideal_j}$ and $x_1^{ideal_j}$ respectively. Let the number of $0's$ and $1's$ in received vector from j be $H(0)^{received}$ and $H(1)^{received}$, which are known. Therefore,

$$x_0^{ideal_j} - P_{fe}^{'} \times x_0^{ideal_j} = H(0)^{received} \tag{7}$$

$$x_1^{ideal_j} - P_{me}^{'} \times x_1^{ideal_j} = H(1)^{received} \tag{8}$$

From Eqn. 7 and Eqn. 8, we find $x_0^{ideal_j}$ and $x_1^{ideal_j}$, the other parameters being known. The total mismatch from ideal scenario caused due to channel uncertainty is

$$P_{fe}^{'} \times x_0^{ideal_j} + P_{me}^{'} \times x_1^{ideal_j} = \alpha_{noise}^j \tag{9}$$

where α_{noise}^j accounts for the mismatches that occur due to unreliable channels conditions. Thus we need to add it to the matches to account for fairness. So the modified instantaneous fair trust coefficient is

$$E_{FairTrust_i}^j = \frac{\alpha^j + (\frac{X_j}{\alpha^j+\beta^j} \times \alpha^j) + \alpha_{noise}^j}{\alpha_j + \beta_j + X_j} \tag{10}$$

where $0 < E_{FairTrust_i}^j < 1$.

3.3 Cumulative Fair Trust Model

The instantaneous trust model considers only one observation, i.e., α and β are computed based on transmission on a single time slot while taking care of mismatches caused by noise. Another approach is a continuous trust model where it is possible to make multiple observations of neighbors before concluding a trust. To minimize such effects of channel noise that may prevail over multiple slots, we define cumulative trust model where the computation of the trust is done over multiple time slots. Doing so, allows to discount (but not eliminate) some errors that occur randomly on certain slots. We observe the transmissions over a decision window of l time slots and do an averaging of the parameters. Thus, α_j, β_j, and X_j computed over l time slots are given as:

$$\alpha_j^l = \frac{1}{l}\sum_{k=1}^{l}\alpha_j^k \tag{11}$$

$$\beta_j^l = \frac{1}{l}\sum_{k=1}^{l}\beta_j^k \tag{12}$$

$$X_j^l = \frac{1}{l}\sum_{k=1}^{l}X_j^k \tag{13}$$

where α_j^k, β_j^k, X_j^k are the observed value at the kth time slot of a particular window. Using these at the end of l slots the trust at a particular decision window u is given by

$$E^{l^u}_{window_i^j} = \frac{\alpha_j^l \times (1 + \frac{X_j^l}{\alpha_j^l + \beta_j^l})}{\alpha_j^l + \beta_j^l + X_j^l} \tag{14}$$

where u, is the current window number. Though the trust is computed over l time slots, it is necessary to consider a longer history of a node to truly capture its trustworthiness. To do so, we propose an exponential weighted moving average. If is the trust value computed in the current decision window, u, then the moving average is updated as:

$$E^{l^u}_{FairTrust_i^j} \leftarrow k_1(E^{l^{u-1}}_{FairTrust_i^j}) + k_2(E^{l^u}_{window_i^j}) \tag{15}$$

where $0 < k_1, k_2 < 1$ and $k_1 + k_2 = 1$. Such moving averages takes into consideration longer histories but with exponentially decaying weights for old observations. To illustrate, let us consider a node that experienced bad channel conditions during an entire decision window, and it's trust may be reduced. However, when the channel conditions change it gets a chance to redeem its trust. Similarly, a malicious node that gathers trust by being honest at the beginning will loose its trust quickly when it decides to launch attacks.

3.4 Trust based Fusion Scheme

Using the computed trust coefficients, we study the performance of two fusion schemes: blind fusion and trust-based filtered fusion. We justify the effectiveness of the trust based fusion in Section 4.

3.4.1 Blind Fusion

For blind fusion, node i considers all its neighbors to be honest and includes B_{adv}^j from all its neighbors along with its own B_{act}^i. We formally define Blind Fusion as $BF_{blind}^i = \nabla[B_{adv}^j \oplus B_{act}^i]$, $j \in N_i$ where ∇ is the operator for majority voting rule. Majority voting is a popular fusion rule where final fused inference on a channel is based on what at least half the neighboring nodes advertise with all the nodes treated equally. $\oplus$ is the operator for combination.

3.4.2 Trust-Based Fusion (TBF)

We propose a fusion scheme whereby we only consider neighboring nodes whose $E^j_{trust_i}$ is higher than some trust threshold, Γ_{opt}. (Later in Section 4, we show how to find the optimal threshold). Thus, for trust-based fusion, node i only considers those neighbors whose $E^j_{trust_i} \geq \Gamma_{opt}$. In effect, the fusion is done with information from trusted nodes only.

$$If E^j_{trust_i} \begin{cases} \geq \Gamma_{opt} & \text{Node } j \text{ is trusted ;} \\ < \Gamma_{opt} & \text{Node } j \text{ is not trusted} \end{cases} \tag{16}$$

We define Trust based Fusion as $TBF^i = \nabla[TFS_i \oplus B_{act}^i]$, where TFS_i is the trusted fusion set of binary vectors accumulated by node i using Eqn. 16, which includes B_{adv}^j of trusted nodes only.

Although the nodes are not aware of the ideal scenario, we are aware of what would have been the ideal fusion result, which is the case when for all node $j \in N_i$, $B_{act}^j = B_{adv}^j$, so we define Ideal Fusion for node i, $BF_{ideal}^i = \nabla[B_{act}^j \oplus B_{act}^i]$. This is later used for comparing the performance of various fusion schemes by measuring deviation from ideal result.

3.4.3 Weighted Trust Based Fusion (WTBF)

We consider another possible alternative fusion scheme where the final decision on a channel is based on the binary decision's trusts associated with it. Here we consider *all* broadcasts, but weigh their binary vectors in proportion to their trust values. Unlike traditional voting, this fusion considers weights of the nodes that broadcast 1 for a channel and those broadcast that broadcast 0 for the same channel. As the weighing factor we simply consider their trust values. On channel k, for all nodes that have advertised 1 on channel k, their trusts are added. Let this be $E^j_{trust}(d_k = 1)$, and similarly for those who advertised 0 be $E^j_{trust}(d_k = 0)$. Thus, the WTBF result is 1 if $\sum E^j_{trust}(d_k = 1) > \sum E^j_{trust}(d_k = 0)$ and 0 if $\sum E^j_{trust}(d_k = 1) < \sum E^j_{trust}(d_k = 0)$, $\forall k = (1, n)$. This scheme unlike TBF is not exclusionary but gives weight to the most trusted decision. However our performance evaluation shows that trust based fusion (TBF) is the more robust, reliable and effective scheme than weighted trust fusion. Weighted trust fusion may work well but not under all conditions. We justify this in Section 4.

4. SIMULATION MODEL AND PERFORMANCE ANALYSIS

In this section, we study the performance of our proposed technique and its effectiveness in capturing cooperation misbehavior. First, we provide results for the average trust values of different nodes, which reflect that our scheme is successful in capturing cooperation misbehavior through trust coefficients. Results show that the malicious nodes have a trust distribution significantly lower than honest nodes. Secondly, we show how we use the trust based fusion scheme to filter out advertised data from malicious nodes. We show trust based fusion is robust than blind fusion. We also show that Weighted Trust based Fusion is *not a stable alternative for robust fusion.*

In the simulation, we evaluate performance metrics, by a parameter 'Mismatch'. For a particular case of fusion

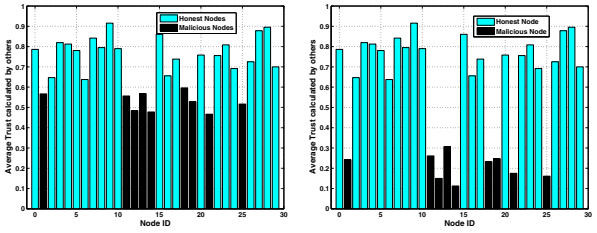
scheme, Mismatch is the difference between BF^i_{ideal} and the corresponding fusion schemes TBF and WTBF. Mismatch reflects deviation of *Cooperative* sensing accuracy of nodes in coexistence with malicious nodes. We compare the 'Mismatch' for each fusion scheme to show that trust based fusion works better than blind fusion, and also show the impact of the spatial orientation of nodes hampering cooperation gain. We compare average mismatch of blind fusion and trust based fusion from individual node's perspective and overall network's perspective.

4.1 Simulation Set up

For simulation, we consider a network of 60x60m grid, with 30 randomly scattered nodes out of which 9 nodes are programmed to be malicious. All nodes continuously scan 50 channels, record the signal power on each of them, and create the binary occupancy vector which they then advertise. The malicious nodes attack (i.e., change the bits in the channel occupancy vector) with a probability between 0.2 and 0.8. It is to be noted that high probability of attack facilitates easy detection and at the same time very low attack probability do not significantly effect the network. The probabilities P_f, P_m and P_e are assumed as 0.03,0.002 and 0.01 respectively. Transmission range of all nodes is considered to be 20m, thus a node hears broadcasts from all the nodes that are in a 20m radius.

4.2 Trust Measurement

In Fig. 2, we see the difference in trust distribution between honest and malicious nodes for both low intensity and high intensity of attack. We measure the average trust of each node as evaluated by all its other neighboring nodes. We observe that malicious nodes have significantly low trust values than the honest nodes for both the low attack probability in Fig. 2(a) and high attack probability in Fig. 2(b). In Fig. 3, we plot all malicious node's trust against various probabilities of attack and observe that the node's trust is lowered as it becomes more aggressive in attacks i.e., the more it attacks the more damage it does to its trustworthiness. We observe that the average trust for set of honest nodes does not change given all other P_f, P_e, and P_m remain constant, but average trust for malicious nodes decreases with increase in probability of attack.



(a) (b)

Figure 2: Trust distribution at: (a) Probability of attack = 0.3; (b) Probability of attack = 0.7

4.3 Cumulative Trust Measurement

From Fig. 4, we observe the distribution of trust values for cumulative trust model for particular node (Node 2). The sample values are plotted alongside the cumulative exponentially weighted moving average trust coefficient. We observe that as the noise and bad channel conditions may effect the instantaneous sample values, the cumulative model is able to redeem its trust value after a certain length of time. For simulation we used weight factors $k_1 = 0.8$ and $k_2 = 0.2$. We consider a length of 240 windows.

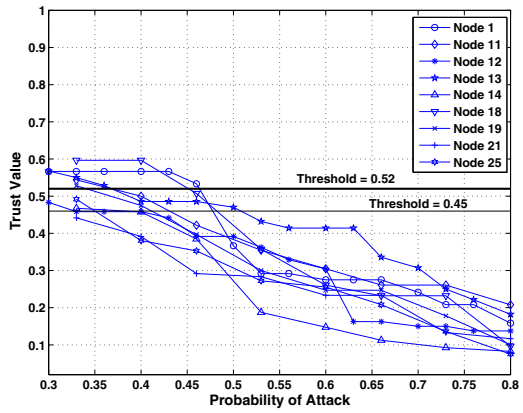


Figure 3: Trusts of malicious nodes at various probability of attack

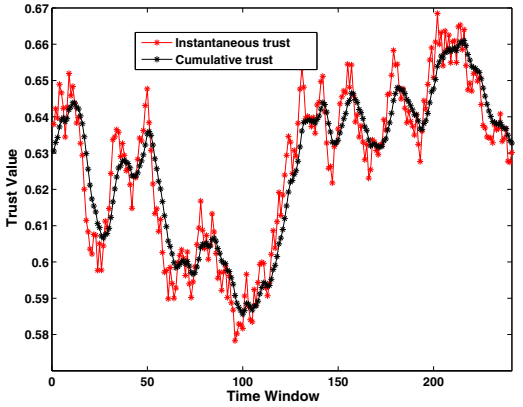


Figure 4: Trust distribution of node 2: Instantaneous and cumulative values

4.4 Heuristic Trust Threshold for Trust-based Fusion

From Fig. 5, we obtain value of trust threshold Γ_{opt} for trust based fusion. We perform a trust based fusion with Γ ranging from 0.2 to 0.8 and compare how 'mismatch' varies from ideal results. Fig. 5, shows that for very low values of threshold there are more mismatches since most of the malicious nodes are included for the purpose of fusion. However, as we increase the threshold, malicious nodes start getting discarded and mismatch decreases. When the threshold is very high (above 0.6), the mismatch again increases as high threshold means we are also discarding the honest nodes along with the malicious nodes. Since our goal is to minimize the total average mismatch from ideal scenario, we notice a range of threshold values from 0.45 to 0.52, where the average mismatch is the least for different probabilities

of attack. From individual node's perspective, the idea is to exclude maximum number of malicious nodes from fusion. We choose Γ_{opt} as 0.52, because we can exclude malicious nodes even at lower probabilities of attack. This is evident from Fig. 3 containing average trusts for malicious nodes for different probabilities of attack. Now observe the horizontal lines for the two possible candidates of trust threshold. If Γ_{opt} was 0.45, the nodes are successful in excluding majority of malicious nodes from fusion for higher probabilities of attack but not for lower probabilities of attack because malicious nodes have lower trust values when they attack less aggressively. If we take 0.52 as threshold, then more malicious nodes are excluded even for lower probability of attack. However, if we take $\Gamma_{opt} = 0.45$, we allow a few malicious nodes in fusion for lower probability of attack.

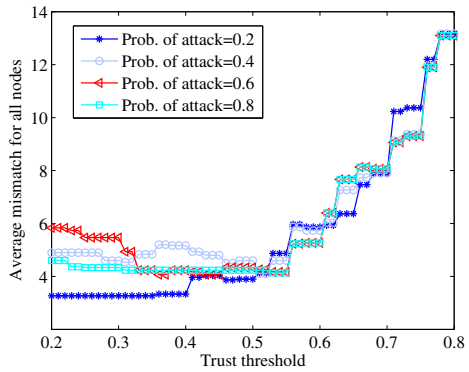


Figure 5: Choosing threshold for trust based fusion

4.5 Performance of Trust based Fusion and Weighted Trust Fusion

Using $\Gamma_{opt} = 0.52$, we compare the trust-based fusion with blind fusion and weighted trust fusion for 'mismatch' as shown in Fig 6. The graph clearly shows that for most nodes, mismatches are far less when trust based fusion is used. However there are a few nodes which are exceptions, e.g., node numbers 7, 27 and 28. This reveals an interesting effect of spatial orientations which explains why these nodes are exceptions (discussed in the next subsection). Since Fig. 6 can be misleading for a few nodes we provide Fig. 7 for total number of mismatches for all nodes that shows the efficiency of trust based fusion. Also from Fig. 8(a) and Fig. 8(b), it is evident that by using trust based fusion the nodes 10 and 23 have mismatches significantly less than if they had blindly fused occupancy data without trust filtration. To measure what percentage of nodes are benefited using our framework of TBF, even at high probabilities of attack of 0.8, 90% of the nodes have average mismatches less than 8 shown in Fig. 9(a). The results reflect the effectiveness of trust based fusion over blind fusion.

We also notice that weighted trust based fusion(WTBF), although works well for a fraction of individual nodes is not a stable for other nodes and the network overall. This is because, often malicious nodes with same false advertisement, their trusts add up and vote out the true advertisement. Hence we do not compare WTBF in other graphs. We are better off using TBF which follows a selective filter based policy.

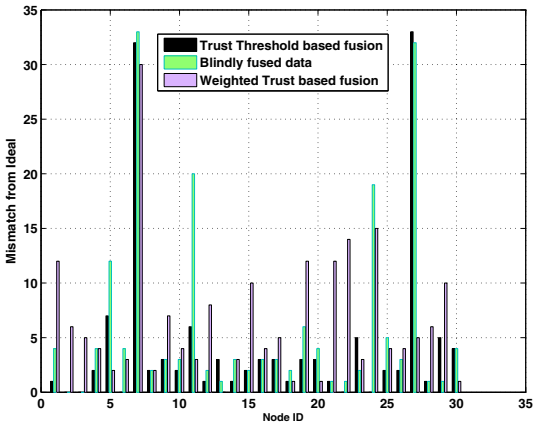


Figure 6: Comparison for selective fusion schemes at Probability of attack = 0.5

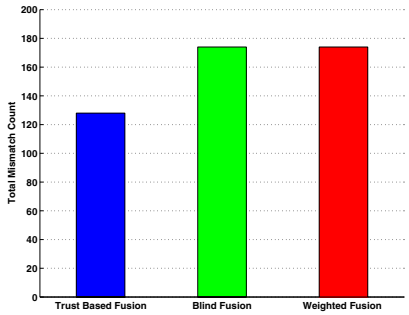
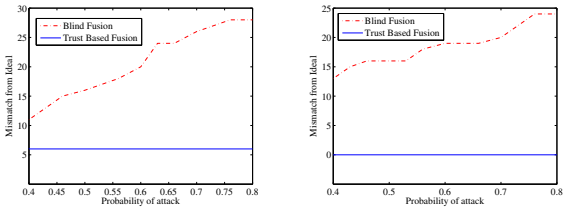


Figure 7: Total mismatch count at probability of attack = 0.5

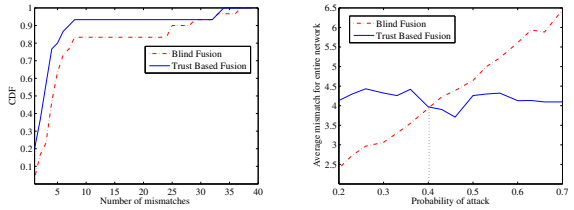


(a) (b)

Figure 8: Blind fusion vs. trust based fusion: (a) Performance of node 10; (b) Performance of node 23

4.6 Effect of Spatial Orientation on Cooperation Gain

The relative position of an honest node with respect to malicious nodes plays an important role in determining the outcome of the fusion process. If an honest node has many malicious nodes in its neighborhood, then trust based fusion scheme will not yield good estimates for the channel occupancy; as a matter of fact no fusion scheme will be effective. However this does not mean that these nodes will not obtain a channel to use, but the gain from cooperative sensing is diminished and such nodes are better off using their own binary vector. In such cases filtering out neighbors will not help to reach anywhere close to ideal result. For e.g., node 7 had only 2 neighbors both being malicious. Similarly for Node 27, two out of three neighbors were malicious. So being in close proximity of many malicious nodes negates the benefit from cooperation. Recall that difference between ideal result and result of applied fusion scheme is an indirect measure of cooperation gain in spectrum sensing accuracy. Out of the 125 mismatches in Fig. 7 for Trust Based Fusion, more than half the mismatches are due to nodes 7 and 27. So, in particular we see when the number of honest neighbors is less than or equal to malicious neighbors, there is a rapid loss in cooperation gain for any fusion scheme. This is the reason for the few exceptions in Fig. 6.



(a) (b)

Figure 9: (a) The CDF at probability of attack = 0.8 (b) Average mismatch for the entire network vs. probability of attack

4.7 Blind and Trust based Fusion: Overall network perspective

An interesting comparison for average mismatch for all nodes in the network, reveal how overall cooperative sensing accuracies are effected. We see that for very lower probabilities of attack, if all nodes still allow advertisement from those malicious nodes in the fusion, the average mismatch is slightly less than trust based filtered fusion although the difference is not much as shown in Fig. 9(b). This is because, lower attack probability means malicious nodes choose to attack less channels and cooperate on more channels. That is why the damage done to the network is less if those nodes are considered in fusion. However as the malicious nodes increase their attack probability, we see marked increase in mismatch for blindly fused data, while the mismatch for trust based fusion scheme is much less. In our simulation (Fig 9(b)) we see for $P_{attack} > 0.40$, the trust based fusion has very low overall mismatch from ideal, while for blind fusion the mismatches continually increases. So from malicious nodes perspective, they will have to employ a probability of attack to cause significant harm to the entire network.

5. CONCLUSIONS

We proposed a framework for evaluating trust through monitoring false advertisements of neighboring nodes, and a trust based fusion schema to address the problem of coexistence with malicious nodes in an ad-hoc CR network. We provided an optimal trust threshold that may be used by each node to disregard advertisements of possible malicious node from the fusion. We are able to identify malicious nodes even at lower probabilities of attack. We also observed that for lower probability of attack, malicious nodes may also contribute to cooperation. So from malicious nodes perspective in order to significantly harm the network, it has to employ a probability of attack higher than critical probability of attack 0.40.

6. REFERENCES

[1] R. Chen, Jung-Min Park, K. Bian ,"Robust Distributed Spectrum Sensing in Cognitive Radio Networks", *Proc. IEEE INFOCOM*, 2008.

[2] K. Chen, P. Chen, N. Prasad, Y. Liang and S. Sun,"Trusted Cognitive Radio Networking", *Journal Wireless Communications and Mobile Computing*, Vol. 10 Issue 4, April 2010.

[3] G. Ganesan and Y. Li,"Cooperative spectrum sensing in cognitive radio, Part II: Multiuser networks," *IEEE Trans. Wireless Commun., Vol. 6*, pp.2214-2233, 2007.

[4] A. Ghasemi and E. S. Sousa, Collaborative spectrum sensing for opportunistic access in fading environments, *The 1st IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN*, Nov. 2005.

[5] Jorg Hillenbrand, Timo A. Weiss, and Friedrich K. Jondral, "Calculation of Detection and False Alarm Probabilities in Spectrum Pooling Systems", *IEEE Communications Letters, VOL. 9, NO. 4,* April 2005.

[6] S.-W. Jeon, N. Devroye, M. Vu, S.-Y. Chung and V. Tarokh, "Cognitive networks achieve throughput scaling of a homogeneous network", in *IEEE Trans. Inform. Theory*, March 2008.

[7] H. Li and Z. Han, "Catching attacker(s) for Collaborative Spectrum Sensing in Cognitive Radio Systems: An Abnormality Detection Approach", *Proc of IEEE DySpan*, 2010

[8] S. Marano, V. Matta and Lang Tong, "Distributed Detection in Presence of of Byzantine Attacks in Large Wireless Sensor Networks", *In proc. of MilCom*, 2006.

[9] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks", in*Proc. of ACM/1EEE International Conference on Mobile Computing and Networking (MOBICOM)*, 2000.

[10] J. Mitola, "Cognitive Radio: An integrated Agent Architecture for Software Defined Radio", *Ph.D. Thesis, KTH, Stockholm* 2000.

[11] S. Parvin, S. Han, L.Gao, F.Hussain and E.Chang, "Towards Trust Establishment for Spectrum Selection in Cognitive Radio Networks", *In Proc. of IEEE International Conference on advanced Information networking and Applications*, 2010.

[12] Y. Sun, Z. Han, and K. J. Ray Liu, "Defense of Trust Management Vulnerabilities in Distributed Networks", *IEEE Communications Magazine, Special Issue, Security in Mobile Ad Hoc and Sensor Networks*, Vol. 46, Issue 2, pp.112-119, Feb 2008.

[13] J. Unnikrishnan and V. V. Veeravalli, "Cooperative Sensing for Primary Detection in Cognitive Radio," IEEE Journal of selected topics in signal processing, Vol. 2, no. 1, pp. 18-27, February, 2008.

[14] http://www.cisco.com/en/US/docs/solutions/Enterprise/Mobility/wifich2.html#wp1049544.