Resilient Consensus Protocol in the Presence of Trusted Nodes

Waseem Abbas, Yevgeniy Vorobeychik, and Xenofon Koutsoukos
Institute for Software Integrated Systems
Vanderbilt University, Nashville, TN 37212, USA
{waseem.abbas, yevgeniy.vorobeychik, xenofon.koutsoukos}@vanderbilt.edu

***Abstract*—In this paper, we propose a scheme for a resilient distributed consensus problem through a set of trusted nodes within the network. Currently, algorithms that solve resilient consensus problem demand networks to have high connectivity to overrule the effects of adversaries, or require nodes to have access to some non-local information. In our scheme, we incorporate the notion of trusted nodes to guarantee distributed consensus despite any number of adversarial attacks, even in sparse networks. A subset of nodes, which are more secured against the attacks, constitute a set of trusted nodes. It is shown that the network becomes resilient against any number of attacks whenever the set of trusted nodes form a connected dominating set within the network. We also study a relationship between trusted nodes and the network robustness. Simulations are presented to illustrate and compare our scheme with the existing ones.**

***Index Terms*—Resilience, consensus, graph robustness, adversary, dominating set.**

I. Introduction

In distributed networks, individual nodes interact locally to accomplish some global objective. Abnormal or malicious behavior by a subset of nodes can affect the global behavior, and may prevent the system from achieving the overall objective. To overcome the adversarial effects, design and analysis of resilient algorithms for distributed networks is required, and has been an active area of research. One of the integral problems in distributed networks and controls is distributed consensus, which finds a wide variety of applications in the domain of distributed optimization, clock synchronization, social networks, coverage, and formation control to name a few (see e.g., [1], [2]).

In this paper, we propose a scheme for the distributed consensus problem that is resilient to an arbitrary number of attacks. A key object in our scheme is the notion of *trusted nodes*, which is a subset of secured nodes deployed within the network. It is shown that all the normal nodes successfully achieve consensus within a network, even a sparse one, whenever trusted nodes form a *connected dominating set* within the network.

The issue of agreement among nodes in the presence of adversaries has been extensively studied in the area of distributed computing ([3], [4]). In the context of network control systems, resilient protocols for consensus in the presence of malicious, byzantine, non-colluding and other threat models have been studied recently, and it is observed that the resilience of various consensus algorithms is tied strongly to the underlying topology of networks.

In [5], it is shown that a network needs to be at least $(2F+1)$-connected[1] to overrule the effects of F malicious nodes under the local model of communication. Similar results have been reported in [6], in which algorithms are presented to identify and detect F byzantine attacks whenever the underlying graph of the network is at least $(2F+1)$-connected, and F non-colluding attacks whenever it is $(F+1)$-connected. Consensus under the generalized fault model using Iterative Approximate Byzantine Consensus (IABC) algorithm has been reported in [7]. Adversarial Robust Consensus Protocol (ARC-P), proposed in [8], guarantees consensus in the presence of malicious and byzantine adversaries in networks with fixed and time-varying topologies modeled by directed graphs. Recently, a more generalized form of ARC-P (and other related algorithms) has been studied in [9], which is known as the Weighted Mean-Subsequence-Reduced (W-MSR) algorithm. A new notion of robustness is introduced in [9], which completely characterizes the resilience properties of W-MSR in terms of the underlying graph structure of the network.

These algorithms, under certain constraints, ensure consensus in the presence of adversarial nodes, and offer resilience within networks. However, there are certain limitations to them that become crucial in various practical scenarios. For instance, these schemes may require individual nodes to have a global knowledge of the network, or some information that is beyond their local neighborhood, for instance as in [5], [6]. Moreover, algorithms that require only local information by the nodes to update their states, may pose high connectivity constraints on the network that are hard to verify. For instance, under W-MSR algorithm [9] (or ARC-P [8]), to handle F total attacks, a graph needs to satisfy a certain robustness property ($(F+1,F+1)$-robust)[9], [11]. In other words, it is not possible to make sparse networks (like tree) resilient against adversarial attacks using these protocols.

In this paper, we offer a scheme for resilient consensus in networks (even sparse) in the presence of any number of adversaries by incorporating the idea of trusted nodes,

[1] A graph is κ-connected if at least κ vertices are required to be deleted from the graph to make it disconnected.

978-1-4799-4187-2/14/$31.00 ©2014 IEEE

which are the subset of nodes that cannot be compromised by adversarial attacks. In other words, these are the nodes with higher security. Similar concept has been used in [10], in which trusted nodes are incorporated in tree networks. Simulations in [10] show the usefulness of this concept for certain cases, although analysis is not provided. Similarly, in [12], a two-layer hierarchical framework, which integrates trust evaluation mechanism for the purpose of detecting adversaries, is presented. In this paper, we characterize constraints on trusted nodes to guarantee distributed consensus in the presence of any number of adversarial attacks and even for sparse networks.

The rest of the paper is organized as follows: in Section II, problem formulation is given; resilient consensus protocol with trusted nodes is presented in Section III, and analyzed in Section IV, in which constraints on trusted nodes are stated to ensure consensus. The protocol is illustrated through examples in Section V. In Section VI, we comment on the relationship between the notion of robustness and trusted nodes. The paper is concluded in Section VII.

II. Problem Formulation

A network of agents is modeled by an undirected graph $G(V,E)$, in which the vertex set V represents agents and the edge set E corresponds to the information exchange among agents. An edge between node i and j is represented by $(i,j)=(j,i)$. The neighborhood of node i is defined as $\mathcal{N}(i)=\{j\in V:\ (i,j)\in E\}$, and the *closed* neighborhood is $\mathcal{N}[i]=\mathcal{N}(i)\cup\{i\}$. The cardinality of $\mathcal{N}(i)$ is called the *degree* of node i.

There are three types of nodes that exist within the network, *normal nodes*, *trusted nodes*, and *adversaries*, as defined below. Each node i has a state value at a given time instant k, denoted by $x_i(k)$. This value can be a sensor measurement, position variable, optimization parameter, opinion, or any other observation. For simplicity, we assume $x_i(k)\in\mathbb{R}$. All these results are easily extendable if $x_i(k)\in\mathbb{R}^d$, for $d>1$.

A. Normal Nodes

All nodes have initial values $x_i(0)$'s, and they update them by synchronously interacting with their neighbors. In fact, each node i updates its value according to an update rule that depends only on the state values of nodes in $\mathcal{N}[i]$, i.e.,

$$x_i(k+1)=f(\{x_j(k)\}),\ \ j\in\mathcal{N}[i] \tag{1}$$

A normal node, at each time step, computes its value according to this pre-defined update law, by receiving values from its neighbors that might also contain adversaries. Also, note that a node does not have a knowledge about the identities of its neighbors except the the trusted nodes.

B. Adversaries and Threat Models

An adversary is a node that does not follow the update rule (1) to update its state value, and therefore, might prevent the network from achieving the required objective. If an adversarial node sends the same value to all of its neighbors, then it is commonly called a *malicious* attack. On the other hand, if a misbehaving node sends different values to different nodes in the neighborhood, the term *byzantine* is typically used. We will call these nodes collectively as *adversaries*.

Moreover, the scope of threat is typically defined in terms of the maximum number of attacks (adversarial nodes) that can occur within the system. Let F be the number of maximum attacks, which can occur at nodes within the network (except the trusted nodes). In the current literature, resilient consensus protocols require to have a knowledge of F in some form. One of our objectives is to ensure the consensus in the presence of adversaries even if nodes do not have this information.

C. Trusted Nodes

Trusted nodes are the normal nodes that cannot be compromised by an adversarial attack. It simply means that attacking these nodes is sufficiently difficult (e.g. because of their high security investment), and we can safely assume that they cannot be attacked. Furthermore, trusted nodes also update their value according to the update rule (1).

D. Main Objective – Resilient Consensus

The objective is to design an update rule (1) (resilient consensus protocol) for the normal nodes, a subset of which consists of trusted nodes, so that they all achieve a common state value even in the presence of F adversaries, where F can be *any* number. In other words, for *any* number of adversarial attacks, we want to achieve the following,

(i) as $k\to\infty$, $x_i(k)=x_j(k)=x$ for all the normal nodes i,j.

(ii) Let $x_{\min}(0)$ and $x_{\max}(0)$ be the minimum and the maximum of the initial values of the normal nodes respectively, then $x_{\min}(0)\le x_i(k)\le x_{\max}(0)$, for all k and for any normal node i.

(iii) For any number of adversaries F, determine necessary and sufficient conditions on the number, location and connectivity of trusted nodes to achieve (i) and (ii).

Conditions (i) and (ii) are referred to as the *agreement* and *safety* conditions respectively in [9]. It is to be mentioned here that existing algorithms for resilient consensus require the network be highly connected, even if the number of adversaries is small. Thus, we want to have a scheme that is resilient even for the networks that are sparse.

III. Resilient Consensus Algorithm

A. Consensus Algorithm

A wide variety of algorithms exist in literature to achieve consensus among the nodes in a distributed set up. *Linear consensus protocol* has been extensively studied

and adapted in the context of various applications. In its simplest form, in a network consisting of n nodes, every node i updates its value according to the following update rule,

$$x_i(k+1) = \sum_{j \in \mathcal{N}[i]} w_{ij}(k) x_j(k) \tag{2}$$

Let $\alpha \in \mathbb{R}$, and $0 < \alpha < 1$, then $w_{ij}(k)$ in (2) satisfy $w_{ij}(k) \geq \alpha$, $\forall j \in \mathcal{N}[i]$ and for all times k. Moreover, $\sum_{j=1}^{n} w_{ij}(k) = 1$, $\forall i$, k. Linear consensus protocol in (2) has been extensively studied and convergence conditions have been found for many different types of network including time-varying, directed, continuous and discrete-time (e.g., [16], [2]). However, it has also been shown that (2) is not resilient against adversarial attacks: even a single misbehaving node can make the network not achieve consensus, thus leading to the study of resilient consensus protocols. We present a scheme that guarantees consensus for any number of adversaries, even in sparse networks, given a sufficient number of trusted nodes.

B. Resilient Consensus Protocol with Trusted Nodes (RCP-T)

The algorithm is described as follows:

Step 1: A node i receives state values from its neighbors, and arrange them as a sorted list.

Step 2: (a) If node i has at least one trusted node in $\mathcal{N}[i]$, then let $\mathcal{T}_i$ be the set of trusted nodes in $\mathcal{N}[i]$. If t_M^i and t_m^i be the maximum and minimum values in $\mathcal{T}_i$ at time k respectively, then we define $r_M^i = \max\left(x_i(k), t_M^i\right)$, and $r_m^i = \min\left(x_i(k), t_m^i\right)$. Moreover, let

$$\mathcal{R}_i(k) = \{j \in \mathcal{N}[i] : \, r_m^i \leq x_j(k) \leq x_M^i\}$$

(b) If node i does not have any trusted node in $\mathcal{N}[i]$, then i removes F largest values that are greater than $x_i(k)$ from the sorted list in step 1. If the number of values that are greater than $x_i(k)$ is less than F, then all values greater than $x_i(k)$ are removed from the list. Similarly, if the number of values smaller than $x_i(k)$ is less than F, i removes all values smaller than $x_i(k)$, otherwise F smallest values are removed from the list. Let $\mathcal{R}_i(k)$ be the set of nodes corresponding to the values that are left in the updated list.

Step 3: Each normal node i updates its value according to the following rule,

$$x_i(k+1) = \sum_{j \in \mathcal{R}_i(k)} w_{ij}(k) x_j(k) \tag{3}$$

Here, $w_{ij}(k)$ satisfy the above conditions. It is to be noted that if every node in the network is connected to at least one trusted node, then step 2(b) is never needed. We want to analyze under what conditions on the set of trusted nodes, RCP-T guarantees consensus even in the presence of *any* large number of adversaries.

C. RCP-T and W-MSR Algorithm

If none of the nodes is connected to a trusted node, then RCP-T is same as the W-MSR algorithm in [9], thus, RCP-T is different from W-MSR in step 2(a). Moreover, under the W-MSR algorithm, it is shown in [9] that the network is resilient against F number of total attacks if and only if the network satisfies the so called (r,s)-robustness property with $r = s = F+1$. (r,s)-robustness is defined below.

Definition 3.1: [9] Given a graph $G(V,E)$. Let $S \subseteq V$, and $\mathcal{X}_S^r$ be the set of vertices in S that are adjacent to at least r neighbors in $(V-S)$, i.e.,

$$\mathcal{X}_S^r = \{i \in S : | \, \mathcal{N}(i) - S \, | \geq r\}$$

A graph $G(V,E)$ is said to be (r,s)-robust if for any two nonempty, disjoint subsets, say S_1 and S_2, at least one of the following is true,

$$\begin{array}{ll} \text{(i)} & | \, \mathcal{X}_{S_1}^r \, | = | \, S_1 \, | \\ \text{(ii)} & | \, \mathcal{X}_{S_2}^r \, | = | \, S_2 \, | \\ \text{(iii)} & | \, \mathcal{X}_{S_1}^r \, | + | \, \mathcal{X}_{S_2}^r \, | \geq s \end{array}$$

Under the W-MSR algorithm, acquiring resilience against F adversaries require the network to have a large connectivity. For instance, a network that is resilient against $F \geq 2$ adversarial attacks, is at least $\left(\left\lceil \frac{3(F+1)}{2} \right\rceil - 1\right)$-connected. Similarly, a tree network cannot handle a single attack. Moreover, every normal node needs to have a correct information of F. Thus, our objective is to develop a scheme that can make the network resilient against *any* number of adversaries, even if the network is sparse and do not have high connectivity. RCP-T achieves this objective by introducing the notion of trusted nodes.

IV. Analysis

In this section, we analyze RCP-T. Let $x_{\min}(k)$ and $x_{\max}(k)$ be the minimum and maximum values of normal nodes at time k. Since, a normal node i updates its value by taking the convex combination of values in $\mathcal{R}_i(k)$ in (3), which lie in the range $[x_{\min}(k) \;\; x_{\max}(k)]$, thus $x_i(k+1) \in [x_{\min}(k) \;\; x_{\max}(k)]$, which implies that condition (ii) (safety condition) in Section II-D is satisfied.

Next, we need to find out under what constraints, RCP-T converges. In other words, we have to find out the necessary and sufficient conditions on the trusted nodes that guarantee consensus in the presence of any number of adversaries.

A. Sufficiency Condition on Trusted Nodes for Resilient Consensus

First, we show that all trusted nodes reach a consensus under RCP-T if they are connected. Second, every normal node converges to a value of a trusted node if it is connected to at least one trusted node. Thus, if every normal node is connected to at least one trusted node, and all trusted nodes induce a connected subgraph, RCP-T

converges irrespective of the number of adversarial nodes present within the network. To prove these statements, we use the techniques employed in [9].

Lemma 4.1: Under RCP-T, all trusted nodes reach consensus in the presence of any number of adversarial nodes, if the set of trusted nodes induce a connected subgraph.

Proof: Let $\mathcal{T}$ be a set of trusted nodes that induce a connected subgraph. Assume $M(k)$ and $m(k)$ to be the maximum and minimum values of trusted nodes at time step k. Since every trusted node updates its value by taking the convex combination of values in the interval $[m(k)\ M(k)]$, both $m(k)$ and $M(k)$ are bounded and monotone functions of time k, with limiting values, say D_m and D_M respectively.

For consensus among trusted nodes, it suffices to show $D_M = D_m$. Suppose that $D_M \neq D_m$, then there exists a constant ϵ_0 such that $D_M - \epsilon_0 > D_m + \epsilon_0$. Moreover, let $\mathcal{X}_M(k,\epsilon_\ell) = \{i \in \mathcal{T} :\ x_i(k) > D_M - \epsilon_\ell\}$, and $\mathcal{X}_m(k,\epsilon_\ell) = \{j \in \mathcal{T} :\ x_j(k) < D_m - \epsilon_\ell\}$. Note that $\mathcal{X}_M(k,\epsilon_\ell)$ and $\mathcal{X}_M(k,\epsilon_\ell)$ are disjoint. Fix $\epsilon < \frac{\alpha^N}{1-\alpha^N}\epsilon_0$. Let k_ϵ be such that $M(k) < D_M + \epsilon$ and $m(k) > D_m - \epsilon$, $\forall\ k > k_\epsilon$. Now, consider nonempty, disjoint sets $\mathcal{X}_M(k_\epsilon,\epsilon_0)$ and $\mathcal{X}_m(k_\epsilon,\epsilon_0)$. Since, nodes in $\mathcal{T}$ induce a connected subgraph, at least one of the following is always true irrespective of the number of malicious nodes,

(i) $\exists\, x_i \in \mathcal{X}_M(k_\epsilon,\epsilon_0)$ that is connected to some trusted node in $(\mathcal{T} - \mathcal{X}_M(k_\epsilon,\epsilon_0))$

(ii) $\exists\, x_j \in \mathcal{X}_m(k_\epsilon,\epsilon_0)$ that is connected to some trusted node in $(\mathcal{T} - \mathcal{X}_m(k_\epsilon,\epsilon_0))$.

Without loss of generality, assume that (i) is true. Note that the maximum value of any node in $(\mathcal{T} - \mathcal{X}_M(k_\epsilon,\epsilon_0))$ is $D_M - \epsilon_0$, and the maximum value of a node in $\mathcal{X}_M(k_\epsilon,\epsilon_0)$ is $M(k_\epsilon)$. Thus, we get the following

$$\begin{aligned} x_i(k_\epsilon+1) &\leq (1-\alpha)M(k_\epsilon) + \alpha(D_M - \epsilon_0) \\ &\leq (1-\alpha)(D_M+\epsilon) + \alpha(D_M - \epsilon_0) \\ &= D_M - (\alpha\epsilon_0 - (1-\alpha)\epsilon) \\ &= D_M - \epsilon_1 \end{aligned}$$

Here, $\epsilon_1 = \alpha\epsilon_0 - (1-\alpha)\epsilon$, and satisfies $0 < \epsilon < \epsilon_1 < \epsilon_0$. Similarly, if (ii) is true, for some $x_j \in \mathcal{X}_m(k_\epsilon,\epsilon_0)$, we will get $x_j(k_\epsilon+1) \geq D_m + \epsilon_1$. Note that $\mid \mathcal{X}_M(k_\epsilon+1,\epsilon_1) \mid < \mid \mathcal{X}_M(k_\epsilon,\epsilon_0) \mid$ owing to the fact that at least one trusted node in $\mathcal{X}_M(k_\epsilon,\epsilon_0)$ decreases its value to at least $D_M - \epsilon_1$. Similarly, $\mid \mathcal{X}_m(k_\epsilon+1,\epsilon_1) \mid < \mid \mathcal{X}_m(k_\epsilon,\epsilon_0) \mid$.

We define ϵ_ℓ recursively as $\epsilon_\ell = \alpha\epsilon_{\ell-1} - (1-\alpha)\epsilon$, and note that $\epsilon_{\ell-1} < \epsilon_\ell$. Repeat the same analysis for time steps $(k_\epsilon + \ell)$, until for some $\ell = K$, either $\mathcal{X}_M(k_\epsilon + K,\epsilon_K) = \emptyset$, or $\mathcal{X}_m(k_\epsilon + K,\epsilon_K) = \emptyset$. Such K exists, as there are finite number of trusted nodes. Observe that

$$\begin{aligned} \mathcal{X}_M(k_\epsilon+K,\epsilon_K) = \emptyset \ &\Rightarrow\ x_i(k_\epsilon+K) \leq D_M - \epsilon_K,\ \forall\, i \in \mathcal{T} \\ \mathcal{X}_m(k_\epsilon+K,\epsilon_K) = \emptyset \ &\Rightarrow\ x_i(k_\epsilon+K) \geq D_m + \epsilon_K,\ \forall\, i \in \mathcal{T} \end{aligned}$$

Now, if $\epsilon_K > 0$, we will have a contradiction that the largest value of any trusted node converges to D_M, or the smallest value converges to D_m. Here, $\epsilon_K = \alpha\epsilon_{K-1} - (1-\alpha)\epsilon$, and $\epsilon = \frac{\alpha^N}{1-\alpha^N}\epsilon_0$. It has been shown in [9] (Theorem 1) that this ϵ_K and ϵ yield $\epsilon_K > 0$. Thus, we get a contradiction. Hence, ϵ_0 must be 0, and $D_m = D_M$. ∎

Lemma 4.2: Under RCP-T, if a normal node i is connected to at least one trusted node, and the set of trusted nodes induce a connected subgraph, then $x_i(k) \to D$, as $k \to \infty$, even in the presence of any number of adversary nodes. Here, D is the limit value of trusted nodes.

Proof: Let $\mathcal{T} \subset V$ be the set of trusted nodes. Since, nodes in $\mathcal{T}$ are connected, x_j's are converging, i .e., $x_j \to D$, $\forall j \in \mathcal{T}$ as per Lemma 4.1. Moreover, let k_ϵ be such that $D - \epsilon < x_j(k) < D + \epsilon$, $\forall k > k_\epsilon$. Furthermore, let τ be a trusted node in $\mathcal{N}[i]$ that has the minimum value in $\mathcal{N}[i] \cap \mathcal{T}$, if $x_i(k_\epsilon) > D + \epsilon$. If $x_i(k_\epsilon) < D - \epsilon$, then let τ be a node with the maximum value in $\mathcal{N}[i] \cap \mathcal{T}$. Without loss of generality, we assume the former case. Since, normal node i updates its state by taking the convex combination of values in the range $x_i(k)$ and $x_\tau(k)$, x_i is converging and a monotone function, and therefore, has a limit X. We have to show that $X = D$.

Lets assume that $X \neq D$, which implies that $X - \epsilon_0 > D + \epsilon_0$ for some constant ϵ_0. Further, we fix $\epsilon < \frac{\alpha}{1-\alpha}\epsilon_0$. Let k_ϵ be some time such that $x_\tau(k) > D - \epsilon$ and $x_i(k) < X + \epsilon$, for all $k > k_\epsilon$. Note that such a k_ϵ exists owing to the convergence of x_i. Since, all nodes involved in the computation of $x_i(k_\epsilon + 1)$ have values not greater than $x_i(k_\epsilon) < X + \epsilon$. Also, observe that x_τ has a value not greater than $X - \epsilon_0$. Thus, we get

$$\begin{aligned} x_i(k_\epsilon+1) &< (1-\alpha)(X+\epsilon) + \alpha(X - \epsilon_0) \\ &= X - \alpha\epsilon_0 + (1-\alpha)\epsilon = X - \epsilon_1 \end{aligned}$$

Thus, $x_i(k_\epsilon+1) < X - \epsilon_1$, where $\epsilon_1 = \alpha\epsilon_0 - (1-\alpha)\epsilon > 0$, with $\epsilon < \frac{\alpha}{1-\alpha}\epsilon_0$. This contradicts the fact that x_i converges to X, which provides the desired contradiction. Thus, it must be the case that $\epsilon_0 = 0$ and $X = D$. ∎

Thus, if every normal node is connected to at least one trusted node and the set of trusted nodes induce a connected subgraph, consensus will always be achieved among the normal nodes (including the trusted nodes) irrespective of the number of malicious attacks. We can relate these conditions to the notion of *connected dominating set* of the underlying graph defined as below,

Definition 4.1: Let $G(V,E)$ be an undirected graph. A set $S \subseteq V$ is a connected dominating set whenever,

(i) For any $i \in V$, there exists some $s \in S$ such that $s \in \mathcal{N}[i]$.

(ii) The vertex set S induce a connected subgraph.

The cardinality of the minimum connected dominating set is known as *connected domination number*, denoted by γ_c. Moreover, if a set $S \subseteq V$ satisfies only (i), then it is called a *dominating set* of G.

Thus, in a network containing at least γ_c trusted nodes, which are arranged to form a connected dominating set, consensus can always be achieved in the presence of any number of adversaries. This is useful as the notion of

connected dominating sets has been widely studied in the graph theory and sensor network literature. Further, we get a *sufficient condition* on the number of trusted nodes required to ensure consensus in the presence of any number of malicious agents. Thus, using Lemma 4.1, and Lemma 4.2, we can directly state the following main result.

Theorem 4.3: (*Sufficient Condition*) Under RCP-T, a network of agents achieve consensus in the presence of any number of adversaries if there exists a set of trusted nodes that form a connected dominating set in the network.

B. Necessary Condition on Trusted Nodes for Resilient Consensus

Now, we will show that if a set of trusted nodes do not form a connected dominating set, then there always exists a scenario, in which consensus cannot be achieved within a network in the presence of adversaries.

Lemma 4.4: If the set of trusted nodes do not form a connected dominating set, there always exists a scenario, in which consensus cannot be achieved by the normal nodes implementing RCP-T, in the presence of adversarial nodes.

Proof: Let a set of trusted nodes do not form a dominating set and there exists a normal node i that is not connected to any trusted node. Moreover, assume that all of the neighbors of i are adversarial nodes, and $x_j \in [a\ b]$, $\forall j \in \mathcal{N}[i]$, where a and b are any two real numbers. If $x_\ell < a$ (or $x_\ell > b$), $\forall \ell \in V - \mathcal{N}[i]$, then node i will never update its value, and rest of the normal nodes will not converge to the value of node i, thus consensus will not be achieved among normal nodes.

For the connectivity condition on trusted nodes, let there be $\sigma > 1$ connected components in the subgraph induced by the trusted nodes. Assume that all trusted nodes in the i^{th} connected component have value c_i, and $c_i \neq c_j$, $\forall i,j \in \{1,2,\cdots,c_\sigma\}$. c_j is the state value of all the trusted nodes in the j^{th} connected component. In such a scenario, none of the trusted nodes will update their values under RCP-T. Since, trusted nodes in different connected components have different values, consensus will not be achieved. ■

Using Lemma 4.4 and the fact that any node in the network can be an adversarial node (except trusted node), we can state the following result directly.

Theorem 4.5: *(Necessary Condition)* For an arbitrary number of adversaries, connected dominating set of trusted nodes is a necessary condition for consensus among normal nodes.

It is to be mentioned here that the notion of connected dominating set in graphs has been extensively studied in both graph theory (e.g., [13]), and sensor network literature (e.g., [14], [15]), wherein a wide variety of applications along with various distributed algorithms for constructing small sized dominating sets have been reported.

Remark: It is to be mentioned here that if trusted nodes form a connected dominating set, a much simpler strategy for resilient consensus can be derived as follows: In the update stage (Step 3) of RCP-T, instead of incorporating all the values in $\mathcal{R}_i(k)$, a node i just considers the value of its trusted neighbor while ignoring the rest of the values in $\mathcal{R}_i(k)$. However, this approach is not prudent as $\mathcal{R}_i(k)$ may also contain state values of other normal neighbors of i, and ignoring their state values means a useful information is being discounted and not employed during the update step. This might result in a consensus value that is far away from the true average of the initial values of the normal nodes.

V. Simulations

In this section, we illustrate RCP-T through a couple of examples and also compare it with W-MSR algorithm [9]. In our first example, we illustrate the algorithm for a network that has a tree graph topology. Building area networks, which are an important concept in smart grid, are often modeled by such network topologies [10]. In fact, we use the building area network example in [10].

A tree network with 26 nodes is shown in Fig. 1. Values inside nodes indicate node id's, whereas the initial state values $x_i(0)$ are shown alongside the nodes. The node set $\{3,7,12,18,26\}$ forms a connected dominating set and therefore, chosen as a set of trusted nodes. We assume that nodes in $\{5,10,15,17,21\}$ are adversaries such that $\forall k \geq 0$

$$
\begin{aligned}
x_5(k+1) &= 50\\
x_{10}(k+1) &= 100\ \sin(0.1\pi k) + 100\\
x_{15}(k+1) &= 25\ \sin(0.04\pi k) + 150\\
x_{17}(k+1) &= 170 - 0.2k\\
x_{21}(k+1) &= 210
\end{aligned}
$$

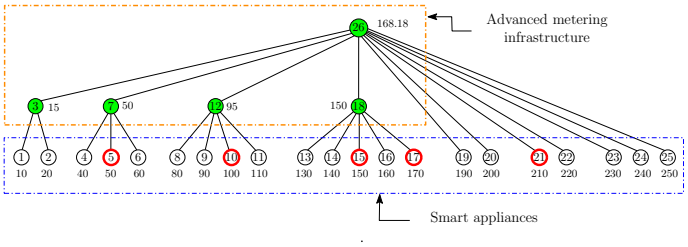

Fig. 1. A building area network represented by a tree graph.

If all the normal nodes implement RCP-T, then consensus is achieved among the normal nodes as shown in Fig. 2. If W-MSR algorithm is implemented, consensus is not achieved even if only one adversary exists in the network as shown in Fig. 2. The primary reason is that the network does not have 'enough connectivity', or more precisely, it is not $(2,2)$-robust. However, in the presence of trusted nodes that form a connected dominating set, any number of adversarial attacks can be handled.

Tree graphs are examples of sparse networks, thus we select a more connected network as our second example. The graph shown in Fig. 3 is $(2,2)$-robust [9]. Node id's are shown inside the nodes, whereas initial values are shown near them. Nodes 3 and 7 are trusted nodes and they form a connected dominating set. Nodes 4 and 6 are attacked by adversaries. In the first plot, all normal

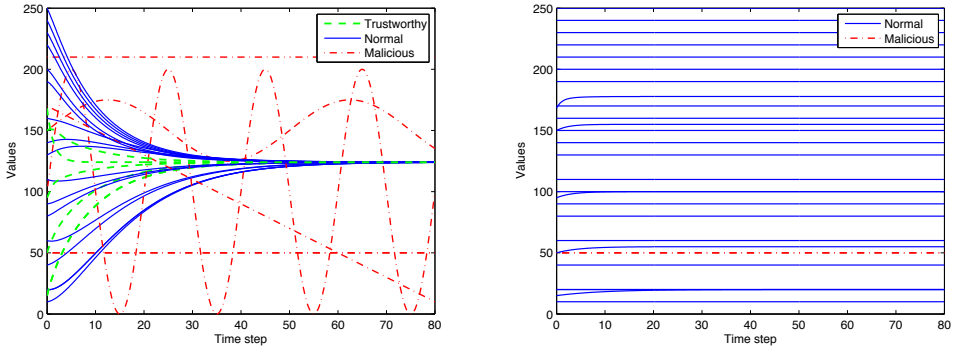

Fig. 2. In the first plot, consensus is achieved among the normal nodes implementing RCP-T in the presence of five adversaries. In the second plot, nodes implement W-MSR, and consensus is not achieved in the presence of a single adversary.

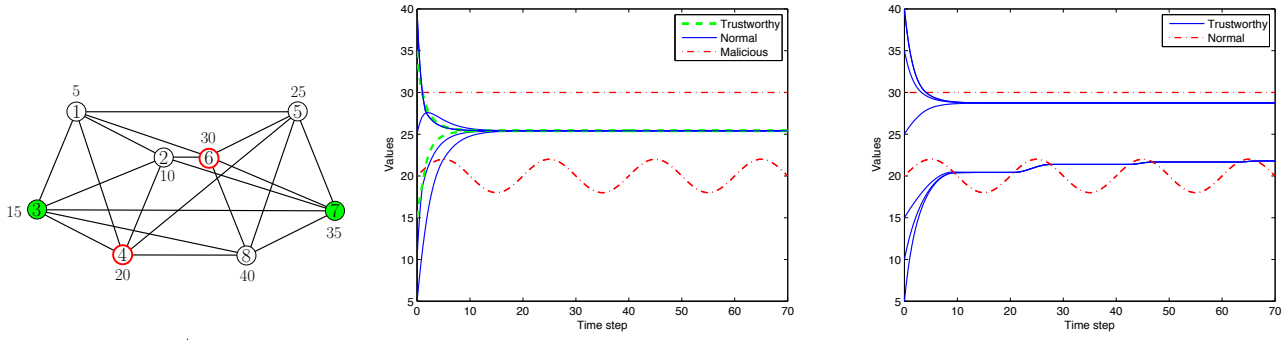

Fig. 3. (a) A $(2,2)$-robust graph. (b) Consensus is achieved under RCP-T. (c) Normal nodes implement W-MSR, and fail to achieve consensus.

nodes achieve consensus in the presence of trusted nodes while implementing RCP-T. In the second plot, nodes implement W-MSR, and consensus is not achieved in the presence of two adversaries. It should be noted that in the case of RCP-T, consensus will be achieved in the presence of any number of adversaries.

VI. Number of Trusted Nodes and Robustness

It is shown above that the sufficient number of guards required to overcome the effect of *any* number of adversaries, under RCT-P is equal to the connected domination number (γ_c) of the underlying graph. Moreover, if no trusted nodes are present, a network can achieve consensus in the presence of at most F adversaries if and only if the graph is $(F+1,F+1)$-robust. So, an interesting question can be asked in a more general setting as follows,

For a given network that can handle F adversarial attacks, what is the minimum number of trusted nodes that need to be added to achieve consensus in the presence of $F' > F$ adversaries?

For instance, for a given $(F+1,F+1)$ robust graph, what is the minimum number of trusted nodes that need to be in the network so that consensus can be achieved in the presence of $F' = F+1$ adversaries. It turns out that the answer is not straight forward. An interesting observation in this regard is that in certain cases, adding as many as $(\gamma_c - 1)$ trusted nodes is not enough to make $(F+1,F+1)$ robust network resilient against $F+1$ adversaries. Two such examples are shown in Fig. 4 and Fig. 5.

The graph in Fig. 4 is $(2,2)$ robust [9]. Thus, it is resilient against $F=1$ malicious attack. The connected domination dumber of the graph $\gamma_c = 4$. There is no way to make this graph resilient against any two malicious attacks by adding three trusted nodes. One example situation is illustrated in Fig. 4. Note that the set of trusted nodes form a dominating set (not a connected dominating set).

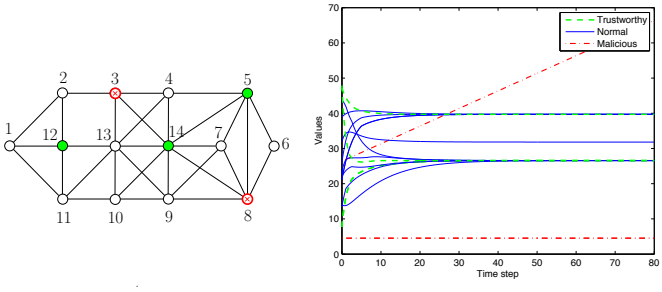

Fig. 4. (a) $\{5,12,14\}$ is the set of trusted nodes, and $\{3,4\}$ are malicious nodes. (b) Normal nodes fail to achieve consensus.

Another example in which any lesser than γ_c trusted nodes in an $(F+1,F+1)$-robust network, do not ensure consensus among normal nodes in the presence of $F+1$

attacks, is shown in Fig. 5. The graph is again $(2,2)$-robust [9], with $\gamma_c = 2$.

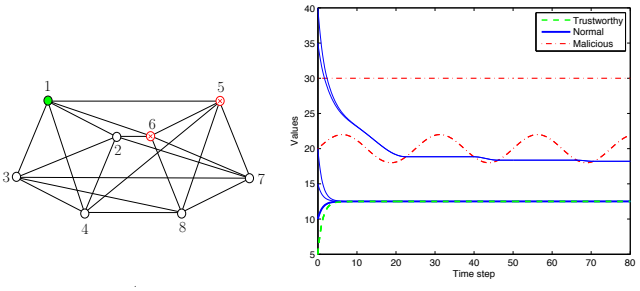

Fig. 5. (a) Node 1 is the set of trusted nodes, and nodes 5 and 6 are malicious nodes. (b) Normal nodes fail to achieve consensus.

Thus, it seems that for many networks, adding trusted nodes is useful when they have a sufficient existence within the network (i.e., γ_c), in which case resilience against any number of attacks is achieved. Furthermore, any number of trusted nodes, which is lesser than this threshold (γ_c), might not improve the resilience properties of such networks, as shown in above examples. Characterization of such networks will be an interesting problem.

At the same time, there are networks in which adding fewer than γ_c trusted nodes indeed improve their resilience property. The graph shown in Fig. 6 is not resilient even against a single malicious attack, as it is not $(2,2)$-robust. Here, $\gamma_c(G) = 3$, but having two trusted nodes make G resilient against one malicious attack.

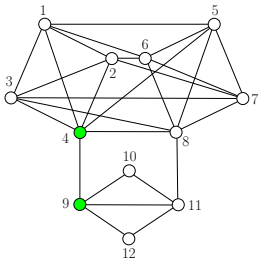

Fig. 6. A graph G. $\{4,9\}$ is the set of trusted nodes.

Thus, to completely understand the connection between trusted nodes and robustness, relationship between connected domination number of a graph and its (r,s)-robustness needs to be studied. An interesting problem can be to figure out the connected domination number of an (r,s)-robust graph for $s = r$.

VII. Conclusions

In a network, there may be nodes that are more secure than the others. This information can be used to relax connectivity constraints on the network topology to ensure consensus among nodes despite adversaries. In other words, it is possible to make a network resilient (for consensus purpose) against any number of adversarial attacks, even if the network is sparsely connected, by making a subset of nodes more secure. In this paper, we have shown that if a set of trusted nodes form a connected dominating set, consensus can be achieved despite any number of misbehaving nodes. Moreover, there is a relationship between the notion of (r,s)-robustness [9], which characterizes the resilience properties of a network, and trusted nodes. This relationship can be studied by figuring out the connected domination number of (r,s)-robust graphs. Moreover, instead of simply assuming that trusted nodes are completely secured from adversarial attacks due to higher security investments, it will be interesting to introduce varying level of trust among nodes, and then analyze the resilience of networks, which is also one of our future work.

VIII. Acknowledgments

This work is supported in part by the National Science Foundation (CNS-1238959, CNS-1035655) and National Institute of Standards and Technology (70NANB13H169).

References

[1] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Review*, vol. 53, no. 4, pp. 747–772, 2011.

[2] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, 2010.

[3] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *Journal of the ACM*, vol. 27, no. 2, pp. 228–234, 1980.

[4] N. A. Lynch, *Distributed Algorithms*, San Francisco, CA: Morgan Kaufman Publishers Inc., 1997.

[5] S. Sundaram and C. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Trans. Autom. Control*, vol. 56, no. 2, 2011.

[6] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach", *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 90–104, 2012.

[7] L. Tseng and N. H. Vaidya, "Iterative approximate byzantine consensus under a generalized fault model", *Intl. Conf. on Distributed Computing and Networking*, 2013.

[8] H. .J. LeBlanc and X. D. Koutsoukos, "Low complexity resilient consensus in networked multi-agent systems with adversaries," *Intl. Conf. on Hybrid Systems: Computation and Control*, Beijing, China, 2012.

[9] H.J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE J. Sel. Areas Commun*, vol. 31, no. 4, 2013.

[10] M. Yampolskiy, Y. Vorobeychik, X. Koutsoukos, P. Hovarth, H.J. LeBlanc, and J. Sztipanovits, "Resilient distributed consensus for tree topology," *ACM Intl. Conf. on High Confidence Networked Systems*, Berlin, Germany, 2014.

[11] H. Zhang and S. Sundaram, "Robustness of information diffusion algorithms to locally bounded adversaries," *IEEE American Control Conf.*, Montréal, Canada, 2012.

[12] X. Liu, P. Gao, and J. S. Baras, "Distributed consensus with byzantine adversaries," *Allerton Conf. on Communication, Control and Computing*, Monticello, IL, 2013.

[13] Y. Caro, D. B. West, and R. Yuster, "Connected domination and spanning trees with many leaves," *SIAM J. Discrete Math.*, vol. 13, no. 2, pp. 202–211, 2000.

[14] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," *Intl. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication*, Seattle, WA, 1999.

[15] P. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," *Mobile Networks and Applications*, vol. 9, pp. 141–149, 2004.

[16] W. Ren and R. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. Control*, vol. 50, 2005.