Improving Network Connectivity and Robustness Using Trusted Nodes with Application to Resilient Consensus

Waseem Abbas, Aron Laszka, and Xenofon Koutsoukos

***Abstract*—To observe and control a networked system, especially in failure-prone circumstances, it is imperative that the underlying network structure is robust against node or link failures. A common approach for increasing network robustness is redundancy: deploying additional nodes and establishing new links between nodes, which could be prohibitively expensive. This paper addresses the problem of improving structural robustness of networks without adding extra links. The main idea is to ensure that a small subset of nodes, referred to as the trusted nodes, remain intact and function correctly at all times. We extend two fundamental metrics of structural robustness with the notion of trusted nodes, network connectivity and r-robustness, and then show that by controlling the number and location of trusted nodes, any desired connectivity and robustness can be achieved without adding extra links. We study complexity of finding trusted nodes and construction of robust networks with trusted nodes. Finally, we present a resilient consensus algorithm with trusted nodes and show that, unlike existing algorithms, resilient consensus is possible in sparse networks containing few trusted nodes.**

***Index Terms*—Robust graphs, network connectivity, resilient consensus, dominating sets.**

I. Introduction

THE correct operation of most networked and distributed systems requires information exchange and cooperation between individual components. As a consequence, malicious attackers may try to disconnect and disrupt networked systems by impairing or tampering with components, for example, using denial-of-service type cyber-attacks, wireless jamming, or even physical attacks. Since it is virtually impossible to protect every node in a network against all possible attacks, networks operating in potentially adversarial environments must be designed to be structurally robust. In the literature, a wide variety of notions have been introduced for quantifying structural robustness (e.g., [2], [3], [4]) as well as an equally wide variety of approaches for increasing robustness. A common aspect of these approaches is that they aim to provide robustness through augmenting the network by deploying additional nodes and communication links. In other words, these approaches increase *robustness through redundancy*. For instance, network connectivity can be improved by strategically adding links between nodes, a technique commonly referred to as the connectivity augmentation (e.g., [5], [6]).

Some of the results appeared in preliminary form in [1].

W. Abbas is with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, USA (waseem.abbas@vanderbilt.edu). A. Laszka is with the Department of Computer Science, University of Houston, TX, USA (alaszka@uh.edu). X. Koutsoukos is with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, USA (xenofon.koutsoukos@vanderbilt.edu).

Although effective, improving structural robustness of the underlying network by adding further links between nodes may be impossible or prohibitively expensive in practice. For instance, to solve a distributed consensus problem by a group of nodes of which a small subset may act maliciously, various resilient consensus algorithms have been proposed. These algorithms guarantee consensus if the underlying network graph satisfies certain connectivity and robustness requirements. However, these requirements are typically overly restrictive in the sense that the network needs to be very highly connected and dense to override the effects of even a very small number of malicious nodes (e.g., [7], [8], [9]). This restriction limits the applicability of existing algorithms in sparse networks, or in scenarios wherein adversaries might exist in greater numbers.

In this paper, we study an alternative approach for increasing structural robustness, including connectivity. We consider improving the robustness of network structures through device hardening. The idea is to ensure the availability and operational integrity of a very small subset of nodes, which we call *trusted nodes*, at all times by protecting them from failures and attacks. While it is often impossible to protect most devices from attacks, we can typically afford to harden a small set of devices. For example, we can protect devices against physical compromise through tamper-proof hardware, and can protect them against cyber-attacks by hardening their software. Since device hardening is expensive, the set of trusted nodes must be small and carefully chosen.

We investigate the question of *how can a sparse network that has fewer connections but contains a small number of trusted nodes exhibit the structural attributes of a highly connected or robust network?* In this direction, we first consider network *connectivity* since it is a fundamental property of any network, and it is by far the most widely used metric of topological robustness. Secondly, we consider a recently introduced measure of structural robustness in graphs referred to as r*-robustness* [2]. This robustness notion is very useful in characterizing resilience of various dynamical processes over networks in adversarial environments [3], [10]. In fact, for a class of distributed consensus algorithms, the ability to guarantee consensus among nodes, some of which act maliciously, can be completely specified in terms of the r-robustness of the network graph (e.g., [11], [12], [13]).

A. Contributions and Organization

Our main contributions are as follows:

(1) We propose and characterize the notion of network connectivity and robustness with trusted nodes, and show that

these network properties can be significantly improved by selecting a small subset of nodes as trusted and without adding extra links. As a consequence, even sparse networks having few trusted nodes can behave as highly connected or more robust networks.

(2) For network connectivity with trusted nodes, we show that computing an optimal set of trusted nodes to achieve a desired connectivity is computationally hard. Then, we present a heuristic to compute a small subset of trusted nodes, and also present a numerical evaluation.

(3) For robustness, we show that deploying a trusted node is equivalent to deploying a certain number of non-trusted nodes, thereby comparing two alternative approaches to achieve desired robustness in networks. We also provide results regarding combination of smaller networks with a given robustness to construct bigger networks with the same robustness properties.

(4) Using the notion of r-robustness with trusted nodes, we study resilient consensus problem. In particular, we present necessary and sufficient conditions in terms of the robustness of underlying network graph to achieve consensus in the presence of malicious nodes. By controlling the number and locations of trusted nodes, the desired robustness, and hence, resilient consensus can be achieved even in sparse networks.

The rest of this paper is organized as follows: Section I-B gives a brief overview of the related work. Section II describes preliminaries and definitions that will be used throughout the paper. Section III presents the notion of network connectivity with trusted nodes, the complexity of finding an optimal set of trusted nodes, and presents heuristics along with numerical evaluations. Section IV introduces r-robustness with trusted nodes. Section V relates trusted nodes to other graph constructions that increase robustness. Section VI utilizes the notion of r-robustness with trusted nodes and presents a resilient consensus algorithm along with necessary and sufficient conditions for consensus. Finally, Section VII concludes the paper.

B. Related Work

To quantify changes in a network structure as a result of node or edge removals, and hence structural robustness, various measures have been reported in literature such as integrity, toughness [14], fragmentability [15], expansion ratio [16], and others (e.g., see [4], [17]). Recently, the notion of *r-robustness* in graphs, introduced in [18] and [2] has received much attention for its usefulness in characterizing resilience of dynamical processes over networks in adversarial environments (e.g., [13], [19], [20], [21]). Structural robustness, such as network connectivity, can be improved in networks by adding links between nodes. The problems related to adding the minimum number of edges to attain the desired network connectivity are referred to as the *connectivity augmentation* problems. In graph-theoretic terms, the issue was investigated in detail for the first time in [22], and extensively studied later. For a comprehensive list of papers in the area of connectivity augmentation, we refer readers to an earlier survey by Frank [23] and Chapter 8 in [5].

Moreover, there has been an increasing interest in utilizing game theory to create networks satisfying certain attributes [24], [25]. In a typical *network creation game*, the goal is to include or remove edges between nodes to optimize various network performance measures such as connectivity. A cost is associated with the creation of an edge, and the objective is to achieve a network with the desired attributes while minimizing the cost (e.g., [26], [27]).

Contrary to the approach of achieving desired structural robustness in graphs by strategically adding edges, we use the notion of trusted nodes. In fact, we show that by selecting a small subset of nodes as trusted, we can achieve any desired network connectivity and r-robustness. Our notion of *trusted nodes* is similar to that of *anchor nodes* used in [28] to maximize the size of k-core in graphs, which is often used to model users participation in social networking phenomena. In a recent work, Dziubiński and Goyal [29] explore trade-offs between the cost of adding links and defending nodes against attacks for network connectivity.

II. Preliminaries

We consider a network of agents that is modeled by an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, in which the vertex set $\mathcal{V}$ represents agents and the edge set $\mathcal{E}$ corresponds to the information exchange among agents. An (undirected) edge between nodes i and j is represented by ij. The neighborhood of node i is defined as $\mathcal{N}(i) = \{j \in \mathcal{V} : \ ij \in \mathcal{E}\}$, 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. A *path* $\mathcal{P}$ of length n is a non-empty graph with the vertex set $\mathcal{V} = \{u_0, u_2, \cdots, u_n\}$, and the edge set $\{u_0u_1, u_1u_2, \cdots, u_{n-1}u_n\}$. The vertices u_0 and u_n are the *end vertices*, whereas, all remaining vertices are the *inner vertices* of the path. In a graph $\mathcal{G}$, two paths are *independent* if they do not have any common inner vertex. We use the terms vertex and *node* interchangeably throughout the paper. If $W \subset V$, then $\mathcal{G} \setminus W$ is the subgraph induced by the remaining vertices and edges of $\mathcal{G}$. If $\mathcal{G} \setminus W$ has at least two components, then W *separates* $\mathcal{G}$. Similarly, if u and v belong to two different components, then W *separates* u and v. Such a set W is referred to as the *vertex cut*.

A. Network Connectivity

Connectivity is a fundamental graph property. A graph is *k-connected* if there does not exist a set of $k-1$ vertices whose removal disconnects the graph. *Vertex connectivity* or simply *connectivity* of $\mathcal{G}$, denoted by $\kappa(\mathcal{G})$, is the maximum value of k for which $\mathcal{G}$ is k-connected. The connectivity of a complete graph with n nodes is defined to be $n-1$ although no vertex cut exists. A classical theorem of Menger relates the notion of connectivity to the number of independent paths between any two nodes (e.g., see [30]). It states that if u and v are distinct, non-adjacent vertices of $\mathcal{G}$, then the minimum size of a vertex cut separating u and v is equal to the maximum number of independent paths between u and v. Consequently, for any $k \geq 2$, a graph is k-connected if and only if any two vertices have k independent paths between them.

B. Network Robustness

Several measures of network robustness exist in literature as discussed in Section I-B. Owing to its usefulness in characterizing the resilience of dynamical processes over networks, in this paper we consider the notion of r-*robustness* as introduced in [2], [18].

In a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, a subset $\mathcal{S}\subset\mathcal{V}$ is r-*reachable* if there exists a node in $\mathcal{S}$ that has at least r neighbors in $\mathcal{N}(i)\setminus\mathcal{S}$.

Definition *(r-robustness [2])* A graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is r-robust if for every pair of nonempty, disjoint subsets of $\mathcal{V}$, at least one of the subsets is r-reachable.

The notion of r-robustness can be further generalized as follows: let $r\in\mathbb{Z}^+$, then define $\mathcal{X}_{\mathcal{S}}^r\subseteq\mathcal{S}$ to be the subset of nodes in $\mathcal{S}$, each of which has at least r neighbors outside of $\mathcal{S}$, that is,

$$\mathcal{X}_{\mathcal{S}}^r=\{i\in\mathcal{S}:|\mathcal{N}(i)\setminus\mathcal{S}|\geq r\}. \tag{1}$$

Definition *((r,s)-robustness [2])* A graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is said to be (r,s)-robust for some $r,s\in\mathbb{Z}^+$, if for any pair of nonempty and disjoint subsets of $\mathcal{V}$, say $\mathcal{S}_1$ and $\mathcal{S}_2$, at least one of the following is true:

$$\begin{array}{ll}\text{(i)} & |\mathcal{X}_{\mathcal{S}_1}^r|=|\mathcal{S}_1|,\\ \text{(ii)} & |\mathcal{X}_{\mathcal{S}_2}^r|=|\mathcal{S}_2|,\\ \text{(iii)} & |\mathcal{X}_{\mathcal{S}_1}^r|+|\mathcal{X}_{\mathcal{S}_2}^r|\geq s.\end{array} \tag{2}$$

Note that r-robustness is the same as $(r,1)$-robustness. In Appendix B, we list all possible r-robust graphs with n nodes, where $n\in\{3,4,\cdots,9\}$.

The notion of (r,s)-robustness is very effective in characterizing the resiliency of a class of distributed consensus algorithms in the presence of malicious and misbehaving nodes. In comparison to the classical k-connectivity, the notion of r-robustness is more pertinent to quantify the *local* connectivity of nodes [3], and hence, is more suitable in characterizing network topologies in the context of local-information-based algorithms, including distributed consensus algorithms.

Next, we extend the connectivity and robustness notions to include trusted nodes. We begin with the notion of network connectivity with trusted in the next section.

III. Network Connectivity with Trusted Nodes

In the traditional k-connectivity notion, the idea is to ensure that the graph remains connected if *any* $k-1$ nodes are removed from the network. By adding more edges between nodes, vertex connectivity can be improved. However, if we fix a small subset of nodes such that they cannot be removed from the network, then the minimum number of nodes from the *remaining* set that are required to disconnect the network also increases. Thus, instead of adding more edges or links, we get an alternative way to improve network connectivity. A merit of this approach is that by making only a very small fraction of nodes insusceptible to removals, or as we call *trusted*, the overall node connectivity can be significantly improved. In practice, trustedness can be achieved by making such components more resilient and secure against physical attacks, tampering, and malicious intrusions through sophisticated security mechanisms. Next, we define the notion of connectivity with trusted nodes $\mathcal{T}$ as follows:

Definition *(k-Connected with $\mathcal{T}$)* – An undirected graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is said to be k-connected with $\mathcal{T}\subseteq\mathcal{V}$, if there does not exist a set of at most $(k-1)$ vertices in $\mathcal{V}\setminus\mathcal{T}$ whose removal disconnects the graph. The maximum value of k for which the graph is k-connected with $\mathcal{T}$ is denoted by $\kappa_{\mathcal{T}}(\mathcal{G})$ and is referred to as the *connectivity with* $\mathcal{T}$.

Analogous to the independent paths, we define the notion of *independent paths with* $\mathcal{T}$ as follows: If $\mathcal{T}$ is the set of trusted nodes, then two paths are independent with $\mathcal{T}$ if any inner vertex that is common in both paths is a trusted node. For instance, in Figure 1, paths $\{u_1u_2,u_2x,xu_3\}$ and $\{v_1v_2,v_2x,xv_3\}$ are independent with $\mathcal{T}=\{x\}$. A path consisting of only trusted nodes is referred to as the *trusted path*. If for any pair of nodes, there exists a trusted path between them, then $\mathcal{G}$ is referred to as *completely connected with* $\mathcal{T}$. If a graph is *completely connected with* $\mathcal{T}$, we define its $\kappa_{\mathcal{T}}=\infty$.

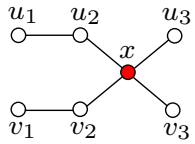


Fig. 1. Independent paths with $\mathcal{T}=\{x\}$.

Next, we compute and relate connectivity with trusted nodes to the traditional notion of connectivity. Let $\mathcal{G}'(\mathcal{V},\mathcal{E}')$ be a graph obtained from $\mathcal{G}(\mathcal{V},\mathcal{E})$ as follows: for every non-adjacent pair of nodes u and v in $\mathcal{G}$, if there exists a trusted node that is adjacent to both u and v, or if there is a trusted path connecting u and v, then add an edge uv in $\mathcal{G}'$. An example is shown in Figure 2, where $\{u,v,z\}$ is the set of trusted nodes in $\mathcal{G}$. Since nodes u and v induce a trusted path in $\mathcal{G}$, all neighbors of u and v are pair-wise adjacent in $\mathcal{G}'$. Similarly, neighbors of trusted node z are adjacent in $\mathcal{G}'$.

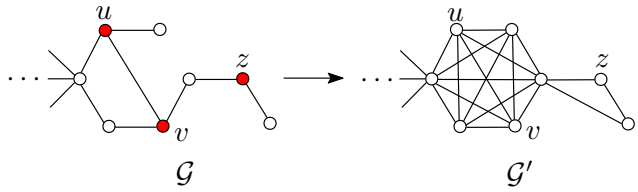


Fig. 2. $\mathcal{G}$ with the set of trusted nodes $\{u,v,z\}$ and the resulting $\mathcal{G}'$.

Proposition 3.1: Let $\mathcal{G}(\mathcal{V},\mathcal{E})$ be a graph that is not completely vertex connected with $\mathcal{T}$, then $\kappa_{\mathcal{T}}(\mathcal{G})=\kappa(\mathcal{G}')$.

Proof: Let $\kappa_{\mathcal{T}}(\mathcal{G})=k$. If nodes u and v are connected trough a trusted path in $\mathcal{G}$, then u and v are adjacent in $\mathcal{G}'$. Thus, if there is no subset of $k-1$ non-trusted nodes in $\mathcal{G}$ whose removal disconnects the graph, then there is no subset of *any* $k-1$ nodes in $\mathcal{G}'$ whose removal disconnects $\mathcal{G}'$. Similarly, we observe that every vertex-cut in $\mathcal{G}$ consisting of only non-trusted nodes is also a vertex-cut in $\mathcal{G}'$. ■

A direct consequence of the above proposition is the following Menger's type result.

Corollary 3.2: For a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ and $\mathcal{T} \subseteq \mathcal{V}$, the following statements are equivalent:

1) $\mathcal{G}$ is k-connected with $\mathcal{T}$.
2) For any two distinct, non-adjacent vertices $u, v \in \mathcal{V}$, either there exists a trusted path between u and v, or there exists at least k paths between u and v that are independent with $\mathcal{T}$.

As an example, consider a 2-connected graph in Figure 3, which becomes 4-connected with two trusted nodes $\mathcal{T} = \{6, 10\}$. To compute the connectivity of $\mathcal{G}$ with $\mathcal{T}$, we first obtain $\mathcal{G}'$ as above, and then can use any algorithm to compute the connectivity of $\mathcal{G}'$. There is an extensive literature on such algorithms [31]. A typical approach is to utilize the *max-flow-min-cut* theorem (e.g., see [5]).

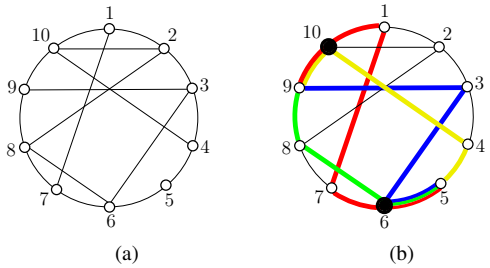


Fig. 3. (a) Graph is 2-connected. (b) The graph becomes 4-connected with $\mathcal{T} = \{6, 10\}$. Between nodes 5 and 9, there are four independent paths with $\mathcal{T}$, shown in red, blue, green and yellow.

A. Computing a Set of Trusted Nodes

In this section, we present heuristics to select a minimum set of trusted nodes $\mathcal{T}$ to achieve a desired connectivity.

1) Problem Complexity: First, we show that finding a minimum set of trusted nodes that achieve a certain connectivity is a computationally hard problem. We begin by formulating this as a decision problem.

Definition (*Trusted Connectivity Augmentation Problem (TCAP)*) Given a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, a desired connectivity k', and the number of trusted nodes T, determine if there exists a set of trusted nodes $\mathcal{T}$ of cardinality T such that $\mathcal{G}$ is k'-connected with $\mathcal{T}$.

Theorem 3.3: TCAP is NP-hard.

We show that TCAP is NP-hard using a reduction from a well-known NP-hard problem, the Set Cover Problem (SCP).

Set Cover Problem (SCP) Given a base set U, a family $\mathcal{F}$ of subsets of U, and a threshold size t, determine if there exists a subfamily $\mathcal{C} \subseteq \mathcal{F}$ of cardinality t whose union is U.

Proof: Given an instance of SCP, we construct an instance of TCAP as follows:

- For each element $u \in U$, create a node u. Similarly, for each member F of the family $\mathcal{F}$, create a node F.
- For each $u \in U$ and $F \in \mathcal{F}$, create an edge (u, F) if $u \in F$.
- For each $F_1, F_2 \in \mathcal{F}$, create an edge (F_1, F_2).
- Let number of trusted nodes be $T = t$, and let the desired connectivity be $k' = |\mathcal{F}|$.

It is clear that the reduction can be performed in polynomial time. As a consequence, we only need to show that TCAP has a solution if and only if SCP does.

First, let us suppose that there exists a set cover $\mathcal{C}$ of cardinality t. Then, let the set of trusted nodes be $\mathcal{T} = \mathcal{C}$. Since $\mathcal{C}$ is a set cover of U, every node corresponding to an element of U is connected to a trusted node in $\mathcal{T}$. Further, every node corresponding to a member of $\mathcal{F} \setminus \mathcal{C}$ is also connected to a trusted node in $\mathcal{T}$, and the trusted nodes are connected to each other. Consequently, $\mathcal{G}$ cannot be separated by the removal of any set of non-trusted nodes, which proves that $\mathcal{T} = \mathcal{C}$ is a solution for TCAP.

Second, let us suppose that there does not exist a set cover of cardinality t. Now, we will show that the graph $\mathcal{G}$ cannot be k'-connected with any set of trusted nodes $\mathcal{T}$ of cardinality $T = t$. Let $\mathcal{T}$ be an arbitrary set of trusted nodes of cardinality T, and consider the removal of all non-trusted nodes corresponding to members of $\mathcal{F}$. Since $\mathcal{T} \cap \mathcal{F}$ cannot be a set cover due to our supposition, there exists an element $u \in U$ that is connected only to non-trusted nodes. Consequently, the removal of the non-trusted nodes corresponding to members of $\mathcal{F}$ separates u from the remainder of the graph, which proves that $\mathcal{T}$ cannot be a solution for TCAP. ■

2) Heuristic: In a graph, complete connectivity with $\mathcal{T}$ is obtained whenever any two nodes are connected through a trusted path between them. A node trusted path exists between any pair of nodes if and only if $\mathcal{T}$ is a *connected dominating set*, which is defined as follows.

Definition (*Connected Dominating Set*) In a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, a subset of nodes $\Sigma \subseteq \mathcal{V}$ is a *connected dominating set* if for every $u \in \mathcal{V}$, either $u \in \Sigma$ or u is adjacent to some $v \in \Sigma$; and the nodes in Σ also induce a connected subgraph. The cardinality of the smallest connected dominating set is known as the *connected domination number*, denoted by $\gamma_{\mathcal{G}}$.

For any k, the minimum cardinality of trusted nodes that are required to achieve the desired k-connectivity with $\mathcal{T}$ is bounded by $\gamma_{\mathcal{G}}$:

$$|\mathcal{T}| \leq \gamma_{\mathcal{G}}. \tag{3}$$

Thus, starting from the set $\mathcal{T} = \Sigma$, we can iteratively reduce the cardinality of $\mathcal{T}$ to obtain a minimal set of trusted nodes with which the graph remains k-connected with $\mathcal{T}$. The notion of connected dominating set in graphs has been extensively studied in both graph theory and sensor network literature (e.g., see [32], [33]), wherein a wide variety of applications along with various distributed algorithms for constructing small connected dominating sets have been reported.

Let `V_Conn_Trust`$(\mathcal{G},\mathcal{T})$ denote the procedure for determining connectivity with a given $\mathcal{T}$, and let `Conn_Dom_Set`$(\mathcal{G})$ denote the procedure for finding a minimal connected dominating set. Then, a minimal $\mathcal{T}$ required to achieve a desired connectivity k' with trusted nodes can be obtained using Algorithm 1 given below. Starting from $\mathcal{T} = \Sigma$, in each iteration, a node is removed

if the resulting connectivity is greater than or equal to the desired connectivity with $\mathcal{T}$. [1]

Algorithm 1 Trusted Nodes for Connectivity

Input: $\mathcal{G}(\mathcal{V}, \mathcal{E})$, k'
Output: $\mathcal{T} \subseteq \mathcal{V}$
$\Sigma \leftarrow$ `Conn_Dom_Set`$(\mathcal{G})$
$\mathcal{T} \leftarrow \Sigma$
for $i = 1$ to $|\Sigma|$ **do**
  $v \leftarrow$ `V_Conn_Trust`$(\mathcal{G}, \mathcal{T} \setminus \{\Sigma(i)\})$
  if $v \geq k'$ **do**
    $\mathcal{T} \leftarrow \mathcal{T} \setminus \{\Sigma(i)\}$
  end if
end for

Note that in Algorithm 1, $O(|\Sigma|)$ calls are made to the subroutine that computes connectivity with trusted nodes.

B. Numerical Evaluation

We evaluate our results for three different types networks, including *Preferential attachment networks*, *Erdős-Rényi networks*, and *Random geometric networks*. These network are frequently used to model various networking phenomenon existing in nature and also for various engineering applications. The details of networks considered for our simulations are stated below.

- *Preferential attachment (PA) networks* with $n = 100$ nodes were obtained by adding nodes to existing networks one-by-one. Each new node was connected to $m = 3$ existing nodes such that the probability of connecting to an existing node was proportional to its degree.
- *Erdős-Rényi (ER) networks* consisting of $n = 100$ nodes were generated such that the probability of an edge between any two nodes was $p = 0.07$.
- *Random geometric (RG) networks* consisting of $n = 100$ nodes were generated by distributing the nodes uniformly at random in a unit square. An edge exists between any two nodes if the Euclidean distance between them is at most 0.18.

Every single point in the plots in Figure 4 is an average taken over thirty randomly generated instances. The minimum number of trusted nodes (computed by the Algorithm 1) sufficient to achieve the desired connectivity with $\mathcal{T}$ are plotted in Figure 4. In the case of the preferential attachment networks, connectivity without trusted nodes is 3. To increase the connectivity from 3 to 4, we observe a big jump in $|\mathcal{T}|$, which is almost equal to the size of the minimum connected dominating set. In the case of our preferential attachment networks, connectivity with $\mathcal{T}$ is exhibited as an 'all-or-nothing' type phenomenon, i.e., to increase connectivity even by one, the number of trusted nodes needed are sufficient to make the network completely connected with $\mathcal{T}$. However, in the cases of Erdős-Rényi and random geometric networks, we observe rather a continuous increase in $|\mathcal{T}|$. The plot of $\mathcal{T}$ is plateaued once $|\mathcal{T}|$ is equal to the size of the connected dominating set.

[1]The symbol $\leftarrow$ in the algorithm indicates that the value returned by the expression on the right side is assigned to the variable on the left side.

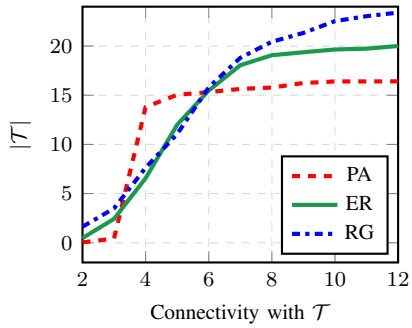


Fig. 4. Number of trusted nodes $|\mathcal{T}|$ as a function of connectivity with trusted nodes $\mathcal{T}$. For each connectivity value, a set $\mathcal{T}$ is found using Algorithm 1.

IV. Robustness with Trusted Nodes

In this section, we extend (r, s)-robustness in graphs (defined in Section II-B) by incorporating the notion of trusted nodes. We then show that by having a small number of trusted nodes, networks exhibit improved robustness that otherwise could be achieved only by adding extra edges.

Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, let $\mathcal{T} \subseteq \mathcal{V}$ be a set of trusted nodes. Then, for a non-empty subset $\mathcal{S} \subseteq \mathcal{V}$, we define $\mathcal{Y}_\mathcal{S}$ to be the subset of nodes in $\mathcal{S}$ each of which has at least one trusted neighbor outside of $\mathcal{S}$, that is,

$$\mathcal{Y}_\mathcal{S} = \{i \in \mathcal{S} : (\mathcal{N}(i) \setminus \mathcal{S}) \cap \mathcal{T} \neq \emptyset\}. \tag{4}$$

Recall that $\mathcal{X}_\mathcal{S}^r$ (defined in Equation (1)) is the subset of nodes in $\mathcal{S}$ each of which has at least r neighbors outside of $\mathcal{S}$. Next, for a given $\mathcal{S}$, we define $\mathcal{Z}_\mathcal{S}^r$ using (1) and (4) as follows:

$$\mathcal{Z}_\mathcal{S}^r = \mathcal{X}_\mathcal{S}^r \cup \mathcal{Y}_\mathcal{S}. \tag{5}$$

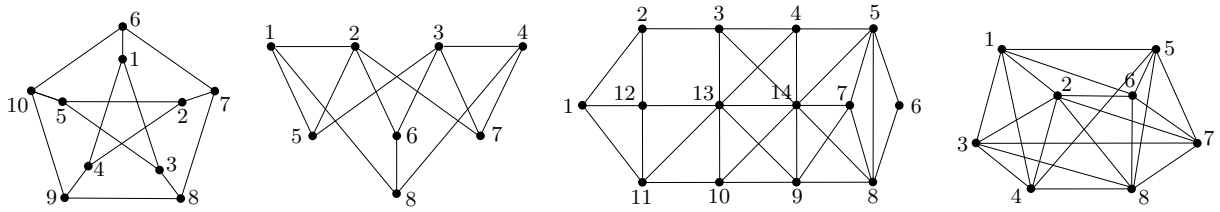
Note that $\mathcal{Z}_\mathcal{S}^r$ is simply the subset of nodes in $\mathcal{S}$ each of which has either at least r neighbors outside of $\mathcal{S}$, or has at least one trusted neighbor outside of $\mathcal{S}$. Moreover, we say that a set $\mathcal{S}$ is *r-reachable with trusted nodes* if the corresponding $\mathcal{Z}_\mathcal{S}^r$ is non-empty. Now, we define the notions of r-robust and then (r, s)-robust graph with trusted nodes as follows:

Definition A graph is r-robust with a set of trusted nodes $\mathcal{T}$ if for any pair of non-empty, disjoint subsets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$, at least one of the subsets is r-reachable with trusted nodes $\mathcal{T}$.

More generally, we define (r, s)-robustness with trusted nodes as follows:

Definition (*(r, s)-robustness with trusted nodes*) A graph is said to be (r, s)-robust with a set of trusted nodes $\mathcal{T}$ if for any pair of non-empty, disjoint subsets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$, at least one of the following conditions is true:

$$\begin{array}{ll} \text{(i)} & |\mathcal{Z}_{\mathcal{S}_1}^r| = |\mathcal{S}_1| \\ \text{(ii)} & |\mathcal{Z}_{\mathcal{S}_2}^r| = |\mathcal{S}_2| \\ \text{(iii)} & |\mathcal{Z}_{\mathcal{S}_1}^r| + |\mathcal{Z}_{\mathcal{S}_2}^r| \geq s \\ \text{(iv)} & (\mathcal{Z}_{\mathcal{S}_1}^r \cup \mathcal{Z}_{\mathcal{S}_2}^r) \cap \mathcal{T} \neq \emptyset \end{array} \tag{6}$$



(a) (b) (c) (d)

Fig. 5. (a) A Petersen graph with ten nodes. (b) A 2-robust graph. (c) A $(2,2)$-robust graph.(d) A 3-robust graph.

The condition (iv) above simply means that there exists a trusted node in $\mathcal{S}_1 \cup \mathcal{S}_2$ that has at least r non-trusted neighbors outside, or at least one trusted neighbor outside of its respective set. Note that an r-robust graph with $\mathcal{T}$ is equivalent to an $(r,1)$-robust graph with $\mathcal{T}$. It is shown in [2] that r-robustness implies r-connectivity. It can be shown easily that same result also holds in the case of trusted nodes, that is, an r-robust graph with trusted nodes is also r-connected with trusted nodes.

Examples: Petersen graph in Figure 5(a) is not 2-robust; for instance, consider $\mathcal{S}_1 = \{1,2,3,4,5\}$ and $\mathcal{S}_2 = \{6,7,8,9,10\}$, and note that neither of these sets contain a node with at least 2 neighbors outside of its respective set. However, the graph becomes 2-robust with any single trusted node. Moreover, the graph becomes 3-robust with any three trusted nodes that form a path, for instance, $\{1,4,9\}$. The graph in Figure 5(b) is 2-robust, but not $(2,2)$-robust; for instance, consider $\mathcal{S}_1 = \{1,2,5\}$ and $\mathcal{S}_2 = \{3,4,6,7,8\}$, and note that none of the conditions in (2) is satisfied by $\mathcal{S}_1$ and $\mathcal{S}_2$. However, the graph becomes $(2,2)$-robust with a single trusted node $\mathcal{T} = \{i\}$ where $i \in \{2,3,8\}$; and becomes 3-robust with two trusted nodes, for instance with $\mathcal{T} = \{j, 2j\}$, where $j \in \{1,3,4\}$. Similarly, the graph in Figure 5(c) is $(2,2)$-robust but not 3-robust [2], but becomes 3-robust with three trusted nodes, for instance with $\mathcal{T} = \{2,3,5\}$. Finally, the graph in Figure 5(d) is 3-robust [2]. However, the graph becomes 5-robust with a single trusted node $\mathcal{T} = \{i\}$ where $i \in \{1,2,5,8\}$.

The above examples illustrate the significance of trusted nodes in improving the robustness properties of graphs without adding extra links. We also note that in a network, a set of trusted nodes through which desired robustness can be achieved is not necessarily unique, and there could be multiple choices for such a set. For instance, the Peterson graph in Figure 5(a) becomes 2-robust if any single node is trusted. This can be useful as it allows switching between different sets of trusted nodes with time. Thus, a particular set of nodes do not have to remain trusted for the entire time, which allows for their repair and maintenance while ensuring that the network satisfies the desired robustness specification at all times.

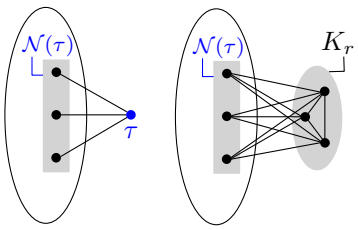
V. Comparing Trusted Nodes with Redundancy in Graphs

To better understand how trusted nodes increase robustness, we next relate trusted nodes to other constructions that increase robustness. More specifically, we show equivalence between making nodes trusted and deploying a certain number of additional nodes and links in a graph, thereby providing a clear comparison between these two alternative approaches, that is, between *trustedness* and *redundancy*. As a result, we compute the number of non-trusted nodes that could replace a trusted node while preserving the robustness property of the network. Later in this section, we present a way of growing a robust network with trusted nodes by adding new nodes to it one at a time. We also present an upper bound on the number of trusted nodes sufficient to achieve any desired robustness. First, we state the following lemma.

Lemma 5.1: If graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is r-robust with $\mathcal{T}$, then each trusted node in $\mathcal{T}$ is adjacent to at least r nodes in $\mathcal{V} \setminus \mathcal{T}$ or to at least one other trusted node.

Proof: See Appendix A.

Next, we show in the following result that making a single node trusted may increase robustness as much as replacing the node with an r-sized clique[2] does. The construction used in Theorem 5.2 is illustrated in Figure 6.



(a) $\mathcal{G}$ (b) $\mathcal{H}$.

Fig. 6. (a) An r-robust graph with a trusted node τ. (b) An r-robust graph obtained from $\mathcal{G}$ by replacing the trusted node by a clique K_r.

Theorem 5.2: Let $\mathcal{G}(\mathcal{V},\mathcal{E})$ be an r-robust graph with a set of trusted nodes $\mathcal{T} \subset \mathcal{V}$. Let $\mathcal{H}$ be a graph obtained from $\mathcal{G}$ by replacing each trusted node $\tau \in \mathcal{T}$ with a clique of r nodes, such that if a node u is adjacent in $\mathcal{G}$ to a trusted node $\tau \in \mathcal{T}$, then u is adjacent in $\mathcal{H}$ to all the nodes in the clique corresponding to the trusted node τ. Then, $\mathcal{H}$ is r-robust.

Proof: Let $\mathcal{V} = \mathcal{W} \cup \mathcal{T}$, where $\mathcal{W} = \{v_1, \cdots, v_n\}$ and $\mathcal{T} = \{\tau_1, \cdots, \tau_m\}$ are the disjoint sets of non-trusted and trusted nodes, respectively. In $\mathcal{H}$, each $\tau_i \in \mathcal{T}$ is replaced by a clique with nodes $\tau_i^1, \cdots, \tau_i^r$. The corresponding vertex set of $\mathcal{H}$ is $\mathcal{V}' = \mathcal{W} \cup \mathcal{T}'$, where $\mathcal{T}' = \{\tau_1^1, \tau_1^2, \ldots, \tau_1^r, \cdots, \tau_m^1, \tau_m^2, \cdots \tau_m^r\}$.

Consider two non-empty, disjoint subsets $\mathcal{S}_1', \mathcal{S}_2' \subset \mathcal{V}'$ in $\mathcal{H}$. We need to show that at least one of them is r-reachable. For this, consider $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$ in $\mathcal{G}$ such that for $x \in \{1,2\}$,

[2]An r-sized clique K_r is a graph consisting of r nodes with the property that all nodes are pair-wise adjacent.

$\mathcal{S}_x = (\mathcal{S}'_x \cap \mathcal{W}) \cup \{\tau_j : \tau_j^k \in \mathcal{S}'_x \text{ for } k \in \{1, 2 \cdots, r\}\}$. Note that $\mathcal{S}_1$ and $\mathcal{S}_2$ are not necessarily disjoint. Then, we have two cases.

Case 1: $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$: If a non-trusted node $v_i \in \mathcal{S}_x$, for some $x \in \{1, 2\}$, has at least r non-trusted or a single trusted neighbor outside of $\mathcal{S}_x$, then by the construction of $\mathcal{H}$, there is a v_i in $\mathcal{S}'_x$ having at least r neighbors outside $\mathcal{S}'_x$.

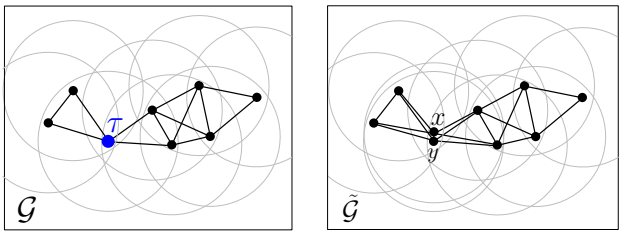
Case 2: $\mathcal{S}_1 \cap \mathcal{S}_2 \neq \emptyset$: Since only a trusted node can be common, let $\tau_i \in \mathcal{S}_1 \cap \mathcal{S}_2$. Then, there exist distinct $k, \ell \in \{1, \cdots, r\}$ such that $\tau_i^k \in \mathcal{S}'_1$ and $\tau_i^\ell \in \mathcal{S}'_2$ in $\mathcal{H}$. By Lemma 5.1, each trusted node in $\mathcal{G}$ is adjacent to (i) at least r non-trusted nodes or to (ii) at least one other trusted node.

Considering (i), let τ_i in $\mathcal{G}$ be adjacent to at least r non-trusted nodes, i.e., if $\mathcal{W}_i = \mathcal{N}(\tau_i) \cap \mathcal{W}$, then $|\mathcal{W}_i| \geq r$. This implies that in $\mathcal{H}$, $\mathcal{W}_i$ is also a subset of both $\mathcal{N}(\tau_i^k)$ and $\mathcal{N}(\tau_i^\ell)$. If $|\mathcal{W}_i \cap \mathcal{S}'_1| = a$, and $|\{\tau_i^1, \tau_i^2, \cdots, \tau_i^r\} \cap \mathcal{S}'_1| = b$, then τ_i^k has at least $(r-a)+(r-b) = 2r-(a+b)$ neighbors outside of $\mathcal{S}'_1$. They include $(r-a)$ non-trusted nodes in $\mathcal{W}_i$ and $(r-b)$ nodes in $\{\tau_i^1, \tau_i^2, \cdots, \tau_i^r\}$. Since $\mathcal{S}'_1 \cap \mathcal{S}'_2 = \emptyset$, this also implies that $\tau_i^\ell \in \mathcal{S}'_2$ has at least $a+b$ neighbors outside $\mathcal{S}'_2$. They include a nodes in $\mathcal{W}_i \cap \mathcal{S}'_1$ and b nodes in $\{\tau_i^1, \cdots, \tau_i^k\} \cap \mathcal{S}'_1$. If $a+b \geq r$, $\mathcal{S}'_2$ is r-reachable. If $a+b < r$, then $2r-(a+b) > r$, and $\mathcal{S}'_1$ is r-reachable, thus making $\mathcal{H}$ r-robust.

Considering (ii), let τ_i be adjacent to some other trusted node τ_x in $\mathcal{G}$. This means that in $\mathcal{H}$, both $\tau_i^k \in \mathcal{S}'_1$ and $\tau_i^\ell \in \mathcal{S}'_2$ are adjacent to r nodes in $\{\tau_x^1, \tau_x^2, \cdots, \tau_x^r\}$. If $|\{\tau_x^1, \tau_x^2, \cdots, \tau_x^r\} \cap \mathcal{S}'_1| = a$ and $|\{\tau_i^1, \tau_i^2, \cdots, \tau_i^r\} \cap \mathcal{S}'_1| = b$, then using the same argument as in (i), at least one of $\mathcal{S}'_1$ or $\mathcal{S}'_2$ is r-reachable, and hence, $\mathcal{H}$ is r-robust. ■

Example: The above result provides a way of replacing a trusted node with a certain number of non-trusted nodes while preserving the robustness property of the network. For instance, consider a unit-disk proximity network, in which an edge exists between any two nodes if and only if they are at most a unit (Euclidean) distance away from each other, as shown for example in Figure 7. Suppose that this network is r-robust with trusted nodes. By the above result, we can replace a trusted node with r number of non-trusted nodes while preserving the r-robustness of the resulting network. So, we can quantify the relationship between *trustedness* and *redundancy*, which here means having multiple non-trusted nodes instead of a single trusted node at a certain location. In Figure 7, $\mathcal{G}$ is 2-robust with a single trusted node τ. The network $\tilde{\mathcal{G}}$ obtained from $\mathcal{G}$ by replacing the trusted node with two non-trusted nodes x and y is also 2-robust. We note that the neighborhood of trusted node τ in $\mathcal{G}$ and the neighborhood of each non-trusted node replacing τ in $\tilde{\mathcal{G}}$ remains the same. In general, we can replace a trusted node in a proximity graph with r non-trusted nodes that are deployed at the same location as the trusted node was.

We also note that in Figure 7(a), if all nodes are non-trusted, then the network graph can be made 2-robust even by adding further links between nodes. This would require increasing the transmission ranges of the nodes resulting in more connections between them. Thus, we have multiple ways of improving r-robustness in such networks: by having trusted nodes, by replacing trusted nodes with other non-trusted nodes, or by adding more links between nodes through increasing the transmission ranges of nodes.



(a) (b)

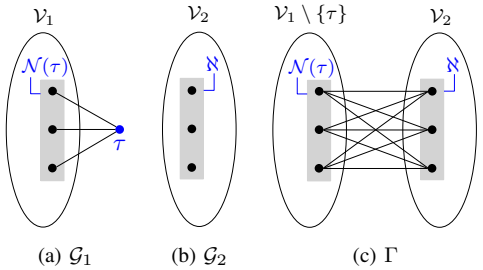
Fig. 7. (a) A unit-disk proximity network that is 2-robust with a trusted node τ. (b) A 2-robust network obtained by replacing the τ with two other nodes.

Next, in Theorem 5.4, we show how to replace a trusted node with another robust graph while preserving the robustness property of the overall network. First, we recall from Section II-B that an r-reachable subset of nodes is a subset in which there exists a node that has at least r neighbors outside of the subset. Thus, in a subset that is not r-reachable, each node has at most $r-1$ neighbors outside of the subset. An r-robust graph can have multiple subsets that are not r-reachable. For example, $\mathcal{G}_2$ in Figure 9(b) is 3-robust, and $\{a, b, c\}$, $\{a, d, e\}$, $\{a, b, e\}$ are examples of subsets that are not 3-reachable.

Lemma 5.3: Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be an r-robust graph, and $\aleph_1 \subset \mathcal{V}$ be a subset that is not r-reachable; then, $|\aleph_1| \geq r$. Moreover, for any two subsets $\aleph_1$ and $\aleph_2$ that are not r-reachable, we have $\aleph_1 \cap \aleph_2 \neq \emptyset$.

Proof: See Appendix A.

Now we relate a trusted node in a graph to replacing it with another robust graph. The construction used in the below theorem is illustrated in Figure 8.



(a) $\mathcal{G}_1$ (b) $\mathcal{G}_2$ (c) Γ

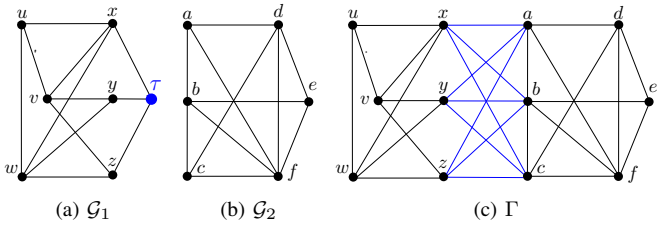
Fig. 8. (a) $\mathcal{G}_1$ is an r-robust graph with a trusted node τ. Nodes in $\mathcal{N}(\tau)$ are highlighted. (b) $\mathcal{G}_2$ is an r-robust graph with a subset $\aleph$ that is not r-reachable. (c) The graph Γ obtained by combining $\mathcal{G}_1$ and $\mathcal{G}_2$ is also r-robust.

Theorem 5.4: Let $\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1)$ be an r-robust graph with a trusted node $\tau \in \mathcal{V}_1$, and let $\mathcal{G}_2(\mathcal{V}_2, \mathcal{E}_2)$ be another r-robust graph with a subset $\aleph \subset \mathcal{V}_2$ that is not r-reachable. Let Γ be the graph obtained from $\mathcal{G}_1$ by replacing the trusted node τ with the graph $\mathcal{G}_2$ such that each vertex of $\mathcal{N}(\tau)$ in $\mathcal{G}_1$ is adjacent to all vertices of $\aleph$ in Γ. Then, Γ is also r-robust.

Proof: See Appendix A.

Example: In Figure 9, $\mathcal{G}_1$ is 3-robust with a trusted node τ, whose neighbors are $\{x, y, z\}$. On the other hand, $\mathcal{G}_2$ is also 3-robust (without any trusted node). The set of nodes $\aleph = \{a, b, c\}$ is *not* 3-reachable in $\mathcal{G}_2$. Γ is also a 3-robust graph that is obtained from $\mathcal{G}_1$ and $\mathcal{G}_2$ by removing τ, and by making every node in $\{x, y, z\}$ adjacent to every node in $\aleph$. Note that

we can use Theorem 5.4 to construct bigger networks from the smaller ones while preserving the r-robustness property.



(a) $\mathcal{G}_1$ (b) $\mathcal{G}_2$ (c) Γ

Fig. 9. (a) A 3-robust graph with $\mathcal{T} = \{\tau\}$. (b) A 3-robust graph $\mathcal{G}_2$. (c) A 3-robust graph Γ.

Next, we present an upper bound on the number of trusted nodes that are sufficient to achieve any desired robustness in the network. If the set of trusted nodes form a connected dominating set (as defined in Section III-A) in the network, then any desired (r, s)-robustness can be attained by such a set of trusted nodes as shown in the following result.

Theorem 5.5: For any given $r, s \in \mathbb{Z}^+$, a network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is (r, s)-robust with $\mathcal{T}$ if $\mathcal{T}$ is a connected dominating set of $\mathcal{G}$.

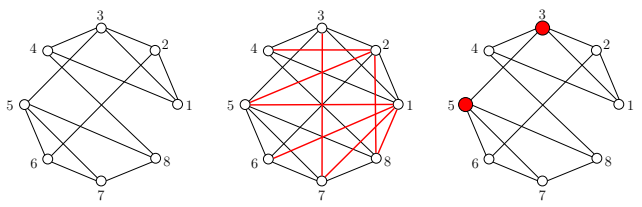
Proof: See Appendix A.

Consequently, the connected domination number $\gamma_{\mathcal{G}}$ becomes an upper bound on the minimum number of trusted nodes required to achieve desired network robustness with trusted nodes. For instance, consider the network graph in Figure 10(a), which is 1-robust. Since a graph with n nodes and containing no trusted node could be at most $\lceil \frac{n}{2} \rceil$ robust, we can make the graph 4-robust. The minimum number of extra edges required to achieve 4-robustness is eight as illustrated in Figure 10(b). However, if nodes 3 and 5—which also constitute a connected dominating set—are trusted, the graph becomes 4-robust with trusted nodes without adding any extra links. Thus, we can achieve the same robustness by creating eight more links or by making two nodes trusted. If the cost of creating extra links is higher, then trusted nodes provide a useful alternative for achieving robustness.

Finally, we present a way of growing a network by adding new nodes to it one at a time such that the robustness property of the network is preserved at every step.

Theorem 5.6: Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be an (r, s)-robust graph with trusted nodes $\mathcal{T} \subset \mathcal{V}$. Then, the graph $\mathcal{G}'$ obtained by adding a new vertex v_{new} to $\mathcal{G}$ is also (r, s)-robust with trusted nodes if v_{new} is adjacent to at least $r + s - 1$ non-trusted nodes or if it is adjacent to at least one trusted node.

Proof: See Appendix A.



(a) (b) (c)

Fig. 10. (a) A 1-robust graph. (b) The graph becomes 4-robust by including eight extra edges. (c) Nodes 3 and 5 constitute a connected dominating set, and are also trusted.

In the next section, we present a resilient consensus algorithm with trusted nodes and analyze its performance using the robustness of network graphs. We show that resilience of consensus algorithm against malicious nodes is significantly improved by having few trusted nodes within a network.

VI. Resilient Consensus Problem and Robustness with Trusted Nodes

Owing to a wide variety of applications in data aggregation, distributed optimization, parameter estimation, and flocking, consensus problem is of significant importance in distributed networks and cooperative control [34]. In a network of nodes (e.g., decision making components), the goal is to reach consensus on the quantity of interest by a local exchange of information. The resilient consensus problem deals with the situations wherein a subset of nodes become faulty or act maliciously, and the goal is to ensure consensus among the normal nodes.

A. Resilient Consensus Problem

1) System Model: We consider a network consisting of two basic types of nodes: *normal* nodes and *adversarial* nodes. Normal nodes have a special sub-class referred to as the *trusted nodes*. 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, or any other observation. For simplicity, we assume $x_i(k) \in \mathbb{R}$. However, our results can easily be extended to consider multi-dimensional state values.

Normal Nodes ($\mathcal{S} \subseteq \mathcal{V}$) – These nodes update their state values by synchronously interacting with their neighbors and following an update rule that depends only on the state values of neighbors. More specifically, $\forall i \in \mathcal{S}$,

$$x_i(k+1) = f(\{x_j(k)\}), \quad j \in \mathcal{N}[i]. \tag{7}$$

The neighborhood of a normal node i might contain adversarial nodes, whose identities are unknown to i.

Trusted Nodes ($\mathcal{T} \subseteq \mathcal{S}$) – These nodes are the sub-class of normal nodes that cannot be compromised by an adversarial attack (e.g. because of their high security investment, more resources, sophisticated hardware and software), and we can safely assume that they do not deviate from their normal behavior. Trusted nodes also update their value according to the update rule (7). Moreover, each normal node $i \in \mathcal{S}$ is aware of the identities of only trusted nodes in $\mathcal{N}(i)$.

Adversaries and Threat Models – An adversary is a node that does not follow the update rule (7) to update its state value, and therefore, might prevent the network from achieving consensus among normal nodes. If an adversarial node sends the same value to all of its neighbors, then it is commonly called a *malicious* attacker. On the other hand, if a misbehaving node sends different values to different nodes in its neighborhood, the term *byzantine* attacker is typically used. We call these nodes collectively the *adversaries*. Moreover, the scope of threat is typically defined in terms of the maximum number of attacks (i.e., adversarial nodes) that can occur within the system. Following [2], we consider two threat models: F-total and F-local. In the **F-total** model, there are at most

F adversarial nodes within the whole network. In the **F-local** model, there are at most F adversarial nodes in the neighborhood of each normal node.

2) Main Objective – Resilient Consensus in the Presence of Trusted Nodes: The objective is to design an update rule (7), for the normal nodes so that they all reach a common state value even in the presence of adversaries (under the F-total or F-local models). More precisely, we want to achieve the following:

(i) As $k \to \infty$, $x_i(k) = x_j(k) = x$ for all 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, $x_{\min}(0) \leq x_i(k) \leq x_{\max}(0)$ for all k and for any normal node i.
(iii) For a given network $\mathcal{G}$ and adversary model, determine necessary and sufficient conditions in terms of robustness of $\mathcal{G}$ with trusted nodes to achieve (i) and (ii).

Conditions (i) and (ii) are typically referred to as the *agreement* and *safety* conditions respectively. We mention here that existing algorithms for resilient consensus require the network to be highly connected, even if the number of adversaries F is small. We aim to provide a scheme that is resilient even in the case of sparse networks, in which the existence of few trusted nodes makes up for the typically high connectivity requirements.

B. Resilient Consensus Algorithm with Trusted Nodes

Now, we propose an algorithm, which we call the *Resilient Consensus Algorithm with Trusted Nodes (RCA-T)*, and describe it below.

Step 1: At each time step k, node i receives state values from its neighbors $\mathcal{N}_i(k)$.

Step 2: The nodes in $\mathcal{N}_i(k)$ are categorized into $\overline{\mathcal{N}_i}(k)$ and $\underline{\mathcal{N}_i}(k)$ as below.

$$\overline{\mathcal{N}_i}(k) = \{j \in \mathcal{N}_i(k) : x_j(k) > x_i(k)\}$$
$$\underline{\mathcal{N}_i}(k) = \{j \in \mathcal{N}_i(k) : x_j(k) < x_i(k)\}.$$

Next, we define $\overline{\mathcal{R}_i}(k) = \overline{\mathcal{N}_i}(k)$ if $|\overline{\mathcal{N}_i}(k)| < F$. Otherwise, $\overline{\mathcal{R}_i}(k)$ consists of the F nodes in $\overline{\mathcal{N}_i}(k)$ with the highest state values (ties are broken arbitrarily). Similarly, we define $\underline{\mathcal{R}_i}(k) = \underline{\mathcal{N}_i}(k)$ if $|\underline{\mathcal{N}_i}(k)| < F$. Otherwise, $\underline{\mathcal{R}_i}(k)$ consists of the F nodes $\underline{\mathcal{N}_i}(k)$ with the lowest state values (again, ties are broken arbitrarily). Finally, we define $\mathcal{R}_i(k) = \overline{\mathcal{R}_i}(k) \cup \underline{\mathcal{R}_i}(k)$.

Step 3: Let $\mathcal{T}_i(k)$ be the subset of trusted nodes in the neighborhood of node i at time step k, i.e., $\mathcal{T}_i(k) = \mathcal{N}_i(k) \cap \mathcal{T}$.

Step 4: Each normal node i updates its value according to the following rule:

$$x_i(k+1) = \sum_{j \in [(\mathcal{N}_i[k] \setminus \mathcal{R}_i(k)) \cup \mathcal{T}_i(k)]} w_{ij}(k) x_j(k). \quad (8)$$

Here, w_{ij} is the weight assigned to the value of node j by node i at time step k.[3] We note that if there is no trusted node in the network, then RCA-T is same as the Weighted-Mean-Subsequence-Reduced (W-MSR) algorithm in [2].

[3]For a discrete time linear consensus strategy as in (8), it is typically assumed that $w_{ij}(k) \geq \alpha$, $\forall j \in \mathcal{N}[i]$ and $\forall k$; where $\alpha \in \mathbb{R}$, and $0 < \alpha < 1$. Moreover, $\sum_{j=1}^{n} w_{ij}(k) = 1$ for a normal node i and $\forall k$ (e.g., see [34]).

C. Analysis

Next, we provide necessary and sufficient conditions for achieving consensus using RCA-T in the presence of malicious nodes.

Theorem 6.1: Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a time-invariant network, in which $\mathcal{T} \subset \mathcal{V}$ is a subset of trusted nodes and each normal node implements RCA-T algorithm. Then,

(i) under the *F-total malicious model*, consensus is achieved asymptotically if and only if $\mathcal{G}$ is $(F+1, F+1)$-robust with $\mathcal{T}$.
(ii) under the *F-local model*, to achieve asymptotic consensus, it is necessary that $\mathcal{G}$ is $F+1$-robust with $\mathcal{T}$, and is sufficient that $\mathcal{G}$ is $(2F+1)$-robust with $\mathcal{T}$.

The proof of Theorem 6.1 is given in Appendix A.

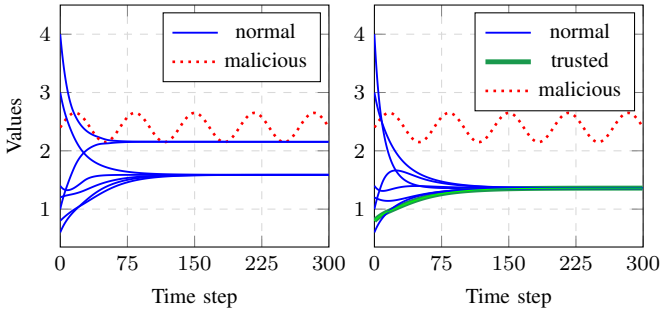
D. Simulation Results

Here, we illustrate the resilient consensus algorithm with trusted nodes and compare it with the W-MSR algorithm [2] with no trusted nodes. In the first example, we consider the *F-total model* for the network in Figure 5(b). We assume that $F = 1$, that is, there is only one malicious node (node 2) in the network that does not follow the state update rule (8). Without any trusted node, the graph is not $(2,2)$-robust, and hence does not satisfy the necessary condition for achieving resilient consensus. As a result, normal nodes fail to reach consensus using the W-MSR algorithm as shown in Figure 11(a). However, with node 8 as a trusted node, the graph becomes $(2,2)$-robust with a trusted node, and consensus is guaranteed using the resilient consensus algorithm with trusted nodes as illustrated in Figure 11(b).

Similarly, we consider the *F-local* model for the network in Figure 5(a). We assume that $F = 1$, that is, there can be at most one malicious node in the neighborhood of any node, and we let nodes 8 and 10 to be malicious. In the absence of any trusted node, the graph is not 2-robust. Since the necessary condition for resilient consensus is not satisfied, consensus is not achieved using the W-MSR algorithm as shown in Figure 12(a). However, with nodes $\mathcal{T} = \{1, 4, 9\}$ as trusted, the graph becomes 3-robust with $\mathcal{T}$, thus satisfying the sufficient condition for resilient consensus in Theorem 6.1. As a result, consensus is achieved in the presence of two malicious nodes (F-local model) using the resilient consensus algorithm with trusted nodes, as illustrated in Figure 12(b).

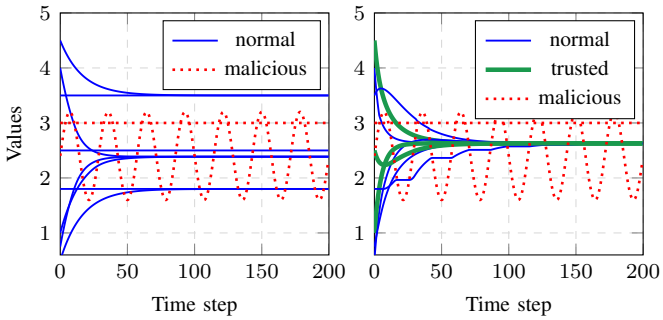
VII. Conclusions and Discussion

A typical approach to improving structural robustness is to add links in a strategic manner, that is, by "redundancy." We adapted a different approach to achieve desired structural robustness, defined in terms of network connectivity and r-robustness in this work. The basic idea was to make a small subset of nodes *trusted*, that is, insusceptible to failures. We then showed that existence of such nodes has an effect of having a higher network connectivity or an improved r-robustness property. We also presented heuristic to select a small subset of trusted nodes to achieve any desired value of network connectivity. Using this approach, even the sparse



(a) without any trusted node. (b) with a single trusted node.

Fig. 11. Resilient consensus with a single malicious node under the *F-total model*. (a) Consensus is not achieved without any trusted node. (b) Consensus is achieved with a single trusted node.



(a) without any trusted node. (b) with trusted nodes.

Fig. 12. Resilient consensus with two malicious nodes under the *F-local model*. (a) Consensus is not achieved without any trusted node. (b) Consensus is achieved with trusted nodes.

networks can be made structurally robust without adding edges. As an application, we illustrated that resilient consensus in the presence of malicious nodes can be achieved even in sparse networks containing a small number of trusted nodes.

Both approaches to improving structural robustness—adding extra links and making few nodes trusted—incur cost. The trusted nodes based approach is particularly useful in scenarios where adding links is not economical or simply infeasible. From an economic perspective, a thorough comparison of the two approaches would be an interesting direction for future work, along with determining conditions under which one approach is strictly better than the other. In future, we aim to apply the notion of trusted nodes to improve other metrics of structural robustness. Moreover, we would like to combine both approaches—redundancy and trustedness—to devise a more efficient strategy to improve structural robustness in networks.

ACKNOWLEDGMENTS

We thank the reviewers of our manuscript and of our previous conference contribution for their valuable comments and suggestions. This work was supported in part by the National Science Foundation (CNS-1238959), Air Force Research Laboratory (FA 8750-14-2-0180), and National Institute of Standards and Technology (70NANB17H266).

REFERENCES

[1] W. Abbas, A. Laszka, Y. Vorobeychik, and X. Koutsoukos, "Improving network connectivity using trusted nodes and edges," in *Proc. of the American Control Conference (ACC)*, Seattle, WA, 2017.

[2] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 766–781, 2013.

[3] H. Zhang, E. Fata, and S. Sundaram, "A notion of robustness in complex networks," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 3, pp. 310–320, 2015.

[4] J. Wu, M. Barahona, Y.-J. Tan, and H.-Z. Deng, "Spectral measure of structural robustness in complex networks," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 6, pp. 1244–1252, 2011.

[5] H. Nagamochi and T. Ibaraki, *Algorithmic Aspects of Graph Connectivity*. Cambridge University Press New York, 2008, vol. 123.

[6] G. Kortsarz and Z. Nutov, "Tight approximation algorithm for connectivity augmentation problems," *Journal of Computer and System Sciences*, vol. 74, no. 5, pp. 662–670, 2008.

[7] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1495–1508, 2011.

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

[9] N. H. Vaidya, L. Tseng, and G. Liang, "Iterative approximate byzantine consensus in arbitrary directed graphs," in *Proc. of the 2012 ACM Symposium on Principles of Distributed Computing (PODC)*, 2012.

[10] A. Mitra and S. Sundaram, "Secure distributed observers for a class of linear time invariant systems in the presence of byzantine adversaries," in *Proc. of the IEEE Conference on Decision and Control (CDC)*, 2016.

[11] S. M. Dibaji and H. Ishii, "Consensus of second-order multi-agent systems in the presence of locally bounded faults," *Systems & Control Letters*, vol. 79, pp. 23–29, 2015.

[12] Y. Wu, X. He, and S. Liu, "Resilient consensus for multi-agent systems with quantized communication," in *Proc. of the American Control Conference (ACC)*, 2016, pp. 5136–5140.

[13] H. LeBlanc and X. Koutsoukos, "Resilient first-order consensus and weakly stable, higher order synchronization of continuous-time networked multi-agent systems," *IEEE Transactions on Control of Network Systems*, 2017.

[14] D. Bauer, H. Broersma, and E. Schmeichel, "Toughness in graphs–a survey," *Graphs and Combinatorics*, vol. 22, no. 1, pp. 1–35, 2006.

[15] K. Edwards and G. Farr, "Fragmentability of graphs," *Journal of Combinatorial Theory, Series B*, vol. 82, no. 1, pp. 30–37, 2001.

[16] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bulletin of the American Mathematical Society*, vol. 43, no. 4, pp. 439–561, 2006.

[17] W. Abbas and M. Egerstedt, "Robust graph topologies for networked systems," in *6th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys)*, 2012, pp. 85–90.

[18] H. Zhang and S. Sundaram, "Robustness of information diffusion algorithms to locally bounded adversaries," in *Proc. of the American Control Conference (ACC)*, 2012, pp. 5855–5861.

[19] J. Zhao, O. Yagan, and V. Gligor, "On connectivity and robustness in random intersection graphs," *IEEE Transactions on Automatic Control*, vol. 62, no. 5, pp. 2121–2136, May 2016.

[20] E. M. Shahrivar, M. Pirani, and S. Sundaram, "Robustness and algebraic connectivity of random interdependent networks," in *Proc. of the 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys)*, 2015, pp. 252–257.

[21] L. Guerrero-Bonilla, A. Prorok, and V. Kumar, "Formations for resilient robot teams," *IEEE Robotics and Automation Letters*, vol. 2, 2017.

[22] K. P. Eswaran and R. E. Tarjan, "Augmentation problems," *SIAM Journal on Computing*, vol. 5, no. 4, pp. 653–665, 1976.

[23] A. Frank, *Mathematical Programming: State of the Art 1994*. Ann Arbor, MI: University of Michigan, 1994, ch. Connectivity augmentation problems in network design, pp. 34–63.

[24] D. Meng, M. Fazel, and M. Mesbahi, "Online algorithms for network formation," in *Proc. of the IEEE Conference on Decision and Control (CDC)*, 2016, pp. 135–140.

[25] E. M. Shahrivar and S. Sundaram, "The game-theoretic formation of interconnections between networks," *IEEE Journal on Selected Areas in Communications*, 2017.

[26] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker, "On a network creation game," in *22nd Annual Symposium on Principles of Distributed Computing (PODC)*, 2003, pp. 347–351.

[27] N. Alon, E. D. Demaine, M. T. Hajiaghayi, and T. Leighton, "Basic network creation games," *SIAM Journal on Discrete Mathematics*, vol. 27, no. 2, pp. 656–668, 2013.

[28] K. Bhawalkar, J. Kleinberg, K. Lewi, T. Roughgarden, and A. Sharma, "Preventing unraveling in social networks: the anchored k-core problem," *SIAM Journal on Discrete Mathematics*, vol. 29, no. 3, 2015.
[29] M. Dziubiński and S. Goyal, "Network design and defence," *Games and Economic Behavior*, vol. 79, pp. 30–43, 2013.
[30] O. R. Oellermann, "Menger's theorem," in *Topics in Structural Graph Theory*, L. W. Beineke and R. J. Wilson, Eds. Cambridge University Press, 2013, pp. 13–39.
[31] A.-H. Esfahanian, "Connectivity algorithms," in *Topics in Structural Graph Theory*, L. W. Beineke and R. J. Wilson, Eds. Cambridge University Press, 2013, pp. 268–281.
[32] X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du, "A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks," *Networks*, vol. 42, no. 4, pp. 202–208, 2003.
[33] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," *Mobile Networks and Applications*, vol. 9, no. 2, pp. 141–149, 2004.
[34] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

APPENDIX A
PROOFS

A. Proof of Lemma 5.1

Proof: For the sake of contradiction, suppose that $\mathcal{G}$ is an r-robust graph, but $\tau \in \mathcal{T}$ is such that $\mathcal{N}(\tau) \cap \mathcal{T} = \emptyset$ and $|\mathcal{N}(\tau)| \leq r-1$. Consider $\mathcal{S}_1 = \{\tau\}$ and $\mathcal{S}_2 = \mathcal{V} \setminus \mathcal{N}[\tau]$. Here, $|\mathcal{Z}^r_{\mathcal{S}_1}| = 0$, where $\mathcal{Z}^r_{\mathcal{S}_1}$ is defined in (5). At the same time, there is no node in $\mathcal{S}_2$ that is adjacent to τ. As a result, for each $v \in \mathcal{S}_2$, there are at most $r-1$ neighbors of v outside of $\mathcal{S}_2$. Thus, $|\mathcal{Z}^r_{\mathcal{S}_2}| = 0$, and $\mathcal{G}$ cannot be r-robust, which is a contradiction. ■

B. Proof of Lemma 5.3

Proof: For the sake of contradiction, let $|\aleph_1| < r$. Consider $\mathcal{S}_1 = \aleph_1$ and $\mathcal{S}_2 = \mathcal{V} \setminus \aleph_1$. Since $|\mathcal{S}_1| < r$, $\mathcal{S}_2$ is not r-reachable. As a result both $\mathcal{S}_1$ and $\mathcal{S}_2$, which are non-empty and disjoint subsets, are not r-reachable in an r-robust graph, which is not possible, hence $|\aleph_1| \geq r$.

Similarly, if $\aleph_1$ and $\aleph_2$ are non-empty, disjoint, and not r-reachable subsets, then by the definition of r-robustness, $\mathcal{G}$ is not r-robust, which is a contradiction. Hence, $\aleph_1 \cap \aleph_2 \neq \emptyset$. ■

C. Proof of Theorem 5.4

Proof: Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be a pair of non-empty disjoint subsets in Γ. The vertex set of Γ is $\mathcal{V} = (\mathcal{V}_1 \setminus \{\tau\}) \cup \mathcal{V}_2$. We have three cases:

Case 1: Both $\mathcal{S}_1$ and $\mathcal{S}_2$ have non-empty intersections with $\mathcal{V}_2$. Let $\mathcal{S}'_1 = \mathcal{S}_1 \cap \mathcal{V}_2$, and $\mathcal{S}'_2 = \mathcal{S}_2 \cap \mathcal{V}_2$. Since $\mathcal{G}_2$ is r-robust, at least one of $\mathcal{S}'_1$ and $\mathcal{S}'_2$ is r-reachable in $\mathcal{G}_2$. Without loss of generality, assume $\mathcal{S}'_1$ to be r-reachable in $\mathcal{G}_2$. Then, $\exists v \in \mathcal{S}'_1$ having at least r neighbors in $\mathcal{V}_2 \setminus \mathcal{S}'_1$, which directly implies that a node exists in $\mathcal{S}_1$ in Γ having at least r neighbors in $\mathcal{V} \setminus \mathcal{S}_1$, thus making $\mathcal{S}_1$ an r-reachable subset in Γ.

Case 2: Exactly one of the subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ has a non-empty intersection with $\mathcal{V}_2$. Again w.l.o.g, we assume that $\mathcal{S}_1 \cap \mathcal{V}_2 \neq \emptyset$ and $\mathcal{S}_2 \cap \mathcal{V}_2 = \emptyset$. Then, we have two subcases:

(a) $\mathcal{S}_1 \cap (\mathcal{V} \setminus \mathcal{V}_2) = \emptyset$: This simply means that $\mathcal{S}_1 \subseteq \mathcal{V}_2$ and $\mathcal{S}_2 \subseteq (\mathcal{V}_1 \setminus \{\tau\})$. If $\mathcal{S}_1$ is r-reachable in $\mathcal{G}_2$, then $\mathcal{S}_1$ is r-reachable in Γ. So, we assume that $\mathcal{S}_1$ is not r-reachable in $\mathcal{G}_2$. In this case, let $x \in \mathcal{S}_1 \cap \aleph$. Such an x exists by Lemma 5.3. Since x is adjacent to all nodes in $\mathcal{N}(\tau) \subset (\mathcal{V}_1 \setminus \{\tau\})$, and $|\mathcal{N}(\tau)| \geq r$ by Lemma 5.1, we deduce that $\mathcal{S}_1$ is r-reachable in Γ, which implies the r-robustness of Γ.

(b) $\mathcal{S}_1 \cap (\mathcal{V} \setminus \mathcal{V}_2) \neq \emptyset$: In this case, corresponding to $\mathcal{S}_1$ and $\mathcal{S}_2$ in Γ, consider two non-empty disjoint subsets $\mathcal{S}'_1$ and $\mathcal{S}'_2$ in $\mathcal{G}_1$. Here $\mathcal{S}'_2 = \mathcal{S}_2$; and $\mathcal{S}'_1 = \mathcal{S}_1 \setminus \mathcal{V}_2$ if $\mathcal{S}_1 \cap \aleph = \emptyset$ and $\mathcal{S}'_1 = (\mathcal{S}_1 \setminus \mathcal{V}_2) \cup \{\tau\}$ if $(\mathcal{S}_1 \cap \aleph) \neq \emptyset$. Since $\mathcal{G}_1$ is r-robust with a trusted node, at least one of $\mathcal{S}'_1$ and $\mathcal{S}'_2$ is r-reachable. We now show that the r-reachability of $\mathcal{S}'_1$ in $\mathcal{G}_1$ implies the r-reachability of $\mathcal{S}_1$ in Γ, and similarly the r-reachability of $\mathcal{S}'_2$ in $\mathcal{G}_1$ implies that $\mathcal{S}_2$ is r-reachable in Γ.
Let $\mathcal{S}'_2$ be r-reachable in $\mathcal{G}_1$. If a node $x \in \mathcal{S}'_2$ has r (non-trusted) neighbors outside of $\mathcal{S}'_2$, then it follows readily that a node exists in $\mathcal{S}_2$ in Γ with at least r neighbors outside of $\mathcal{S}_2$. If a node $x \in \mathcal{S}'_2$ in $\mathcal{G}_1$ is adjacent to a trusted node τ outside of $\mathcal{S}'_2$, then by the construction of Γ, a node exists in $\mathcal{S}_2$ in Γ that is adjacent to all nodes in $\aleph$ where $|\aleph| \geq r$ and $\mathcal{S}_2 \cap \aleph = \emptyset$. Thus, $\mathcal{S}_2$ is r-reachable in Γ if $\mathcal{S}'_2$ is r-reachable in $\mathcal{G}_2$. Similarly, r-reachability of $\mathcal{S}'_1$ in $\mathcal{G}_1$, which is defined as above, directly implies that $\mathcal{S}_1$ is also r-reachable in Γ.

Case 3: $(\mathcal{S}_1 \cap \mathcal{V}_2) = \emptyset$, and $(\mathcal{S}_2 \cap \mathcal{V}_2) = \emptyset$. Using a similar argument as in Case 2(b), we can show that at least one of $\mathcal{S}_1$ and $\mathcal{S}_2$ is r-reachable. ■

D. Proof of Theorem 5.5

Proof: Let $\mathcal{T}$ be a CDS, and $\mathcal{S}_1, \mathcal{S}_2$ be two disjoint, non-empty subsets of $\mathcal{V}$. Then, there are two cases:

(i) $\mathcal{S}_j \cap \mathcal{T} = \emptyset$ for both $j \in \{1,2\}$: Since $\mathcal{T}$ is a CDS, each node in such an $\mathcal{S}_j$ is adjacent to some trusted node outside $\mathcal{S}_j$. Thus, condition (i) or (ii) in (6) is satisfied.

(ii) $\mathcal{S}_j \cap \mathcal{T} \neq \emptyset$ for some $j \in \{1,2\}$: Assume w.l.o.g $\mathcal{S}_1 \cap \mathcal{T} \neq \emptyset$. If $\mathcal{S}_1 = \mathcal{T}$, then $\mathcal{Y}_{\mathcal{S}_2} = \mathcal{S}_2$ (where $\mathcal{Y}_{\mathcal{S}}$ is defined in (4)), and condition (ii) in (6) is satisfied. If $|\mathcal{S}_1 \cap \mathcal{T}| < |\mathcal{S}_1|$, then there is a trusted node in $\mathcal{S}_1$ that is adjacent to some trusted node outside of $\mathcal{S}_1$ (as $\mathcal{T}$ is a CDS). Consequently, $\mathcal{Y}_{\mathcal{S}_1} \cap \mathcal{T} \neq \emptyset$ and condition (iv) in (6) is satisfied.

■

E. Proof of Theorem 5.6

Proof: For any two non-empty, disjoint subsets $\mathcal{S}'_1$ and $\mathcal{S}'_2$, there are three cases: for some $i \in \{1,2\}$, (a) $v_{new} \notin \mathcal{S}'_i$, (b) $\{v_{new}\} = \mathcal{S}'_i$, (c) $v_{new} \in \mathcal{S}'_i$.

In case (a), since $\mathcal{G}$ is (r,s)-robust with trusted nodes, the conditions in (6) are satisfied directly by $\mathcal{S}'_1$ and $\mathcal{S}'_2$ in $\mathcal{G}'$.

In case (b), either condition (i) or (ii) in (6) is always satisfied under the condition of the theorem.

In case (c), assume w.l.o.g. that $v_{new} \in \mathcal{S}'_2$. Also let $\mathcal{S}_2 = \mathcal{S}'_2 \setminus \{v_{new}\}$ and $\mathcal{S}_1 = \mathcal{S}'_1$. Note that the subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ in $\mathcal{G}$ satisfy at least one of the conditions in (6) due to the (r,s)-robustness of $\mathcal{G}$ with trusted nodes. If $\mathcal{S}_1$ and $\mathcal{S}_2$ in $\mathcal{G}$ satisfy any of the conditions (i), (iii), or (iv) in (6), then the same conditions are satisfied by $\mathcal{S}'_1$ and $\mathcal{S}'_2$ in $\mathcal{G}'$. So, we assume that condition (ii), i.e., $|\mathcal{Z}^r_{\mathcal{S}_2}| = |\mathcal{S}_2|$ is satisfied. Note that

if $\mathcal{S}_2$ contains a trusted node and $|\mathcal{Z}^r_{\mathcal{S}_2}| = |\mathcal{S}_2|$, then (iv) is automatically satisfied. So, we assume that $\mathcal{S}_2$ consists of only non-trusted nodes. Since $|\mathcal{Z}^r_{\mathcal{S}_1}| + |\mathcal{Z}^r_{\mathcal{S}_2}| < s$ and $|\mathcal{Z}^r_{\mathcal{S}_2}| = |\mathcal{S}_2|$, there are at most $s-1$ nodes in $\mathcal{S}_2$. If v_{new} in $\mathcal{G}'$ is connected to at least one trusted node, it must be connected to a trusted node outside $\mathcal{S}'_2$. Similarly, if v_{new} in $\mathcal{G}'$ is connected to at least $r+s-1$ non-trusted nodes, then it must be connected to at least r nodes outside $\mathcal{S}'_2$. In both situations, $|\mathcal{Z}^r_{\mathcal{S}'_2}| = |\mathcal{S}'_2|$, thus satisfying the (r,s)-robustness condition with trusted nodes. ■

F. Proof of Theorem 6.1

We use a similar approach used in the proof of Theorem 1 in [2].

Proof: • *In the F-total model, $(F+1, F+1)$-robustness with $\mathcal{T}$ is a necessary condition*:

Let $\mathcal{G}$ be a graph that is not $(F+1, F+1)$-robust. Then there exist non-empty, disjoint subsets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$ that do not satisfy any of the conditions in (6). As a result, there are a maximum number of F nodes in $\mathcal{S}_1 \cup \mathcal{S}_2$ that are adjacent to at least one trusted node, or adjacent to at least $F+1$ non-trusted nodes outside of their respective sets, that is $|\mathcal{Z}^{F+1}_{\mathcal{S}_1}| + |\mathcal{Z}^{F+1}_{\mathcal{S}_2}| \leq F$. At the same time, none of the nodes in $\mathcal{Z}^{F+1}_{\mathcal{S}_1} \cup \mathcal{Z}^{F+1}_{\mathcal{S}_2}$ is trusted (as otherwise condition (iv) in (6) is satisfied). Thus, we assume that all nodes in $\mathcal{Z}^{F+1}_{\mathcal{S}_1} \cup \mathcal{Z}^{F+1}_{\mathcal{S}_2}$ are malicious. Since, $|\mathcal{Z}^{F+1}_{\mathcal{S}_i}| < |\mathcal{S}_i|$ for $i \in \{1,2\}$, there exists at least one normal node (either trusted or non-trusted) in $\mathcal{S}_1$, say x_1, and in $\mathcal{S}_2$, say x_2. Note that both x_1 and x_2 have less than $F+1$ neighbors outside of their respective sets, and are not connected to any trusted node outside of their respective sets. Now, consider that the state values for all nodes in $\mathcal{S}_1$ be a, for all nodes in $\mathcal{S}_2$ be $b > a$, and for all nodes in $\mathcal{V} \setminus (\mathcal{S}_1 \cup \mathcal{S}_2)$ be in the interval $(a\ \ b)$. Moreover, all malicious nodes keep their state values constant throughout. Since both x_1 and x_2 ignore all values (F or less) outside of their sets, and cannot consider a value of a trusted node outside of their set during the update phase, consensus cannot be achieved.

• *In the F-total model, $(F+1, F+1)$-robustness with $\mathcal{T}$ is a sufficient condition*:

Suppose $\mathcal{M} \subset \mathcal{V}$ is the set of malicious nodes, then the set of normal nodes (both trusted and non-trusted) in $\mathcal{G}$ is $\mathcal{V} \setminus \mathcal{M}$. We define $M(k) = \max_{i \in \mathcal{V}\setminus\mathcal{M}} x_i(k)$ and $m(k) = \min_{i \in \mathcal{V}\setminus\mathcal{M}} x_i(k)$. Since for all normal nodes $i \in \mathcal{V} \setminus \mathcal{M}$ and time steps k, $x_i(k+1)$ is a convex combination of values in $[m(k)\ M(k)]$, we deduce that both $m(k)$ and $M(k)$ are monotone and bounded functions of k. Consequently, by monotone convergence theorem, both $m(k)$ and $M(k)$ have some limit, say $\mathcal{D}_m$ and $\mathcal{D}_M$ respectively. For the consensus among normal nodes, we need to show that $\mathcal{D}_M = \mathcal{D}_m$.

On the contrary, suppose that $\mathcal{D}_M > \mathcal{D}_m$. Then, $\exists \epsilon_0 \in \mathbb{R}^+$ such that $\mathcal{D}_M - \epsilon_0 > \mathcal{D}_m + \epsilon_0$. Moreover, for any time step k and $\epsilon_i \in \mathbb{R}^+$, we define

$$\mathcal{S}_M(k, \epsilon_i) = \{j \in \mathcal{V} : x_j(k) > \mathcal{D}_M - \epsilon_i\}, \tag{9}$$

$$\mathcal{S}_m(k, \epsilon_i) = \{j \in \mathcal{V} : x_j(k) < \mathcal{D}_m + \epsilon_i\}. \tag{10}$$

Also, let $\mathcal{Z}^{F+1}_M(k, \epsilon_i) \subseteq \mathcal{S}_M(k, \epsilon_i)$ be the subset in which each node has either at least $F+1$ non-trusted neighbors in $\mathcal{V} \setminus \mathcal{S}_M(k, \epsilon_i)$, or has at least one trusted neighbor in $\mathcal{V} \setminus \mathcal{S}_M(k, \epsilon_i)$. Similarly, define $\mathcal{Z}^{F+1}_m(k, \epsilon_i) \subseteq \mathcal{S}_m(k, \epsilon_i)$ (see Figure 13 for illustration).

Now, assuming that V is the total number of normal nodes (trusted and non-trusted), we fix $\epsilon < \frac{\alpha^V}{1-\alpha^V}\epsilon_0$. Here, $\epsilon_0 > \epsilon > 0$. Note that there exists k_ϵ such that the maximum and minimum values of normal nodes at any time $k \geq k_\epsilon$ are bounded by $\mathcal{D}_M + \epsilon$ and $\mathcal{D} - \epsilon$.

Now, since $\mathcal{S}_M(k_\epsilon, \epsilon_0) \cap \mathcal{S}_m(k_\epsilon, \epsilon_0) = \emptyset$ and $\mathcal{G}$ is $(F+1, F+1)$-robust; one of the following conditions is satisfied, that is, either $|\mathcal{Z}^{F+1}_M(k, \epsilon_0)| + |\mathcal{Z}^{F+1}_m(k, \epsilon_0)| \geq F+1$, or $\big(\mathcal{Z}^{F+1}_M(k, \epsilon_0) \cup \mathcal{Z}^{F+1}_m(k, \epsilon_0)\big) \cap \mathcal{T} \neq \emptyset$. Since there are at most F malicious nodes, in either case there must exist a normal node (trusted or non-trusted) in $\mathcal{Z}^{F+1}_M(k_\epsilon, \epsilon_0) \cup \mathcal{Z}^{F+1}_m(k_\epsilon, \epsilon_0)$. Assume w.l.o.g that $i \in \mathcal{Z}_M(k_\epsilon, \epsilon_0)$ is one such normal node. Next, we show

Claim: $x_i(k_\epsilon + 1) \leq \mathcal{D}_M - \epsilon_1$, where $\epsilon_1 < \epsilon_0$.

To compute $x_i(k_\epsilon+1)$, node i ignores its F neighbors whose state values are lesser than its own value. Node i has at least $F+1$ neighbors with values lesser than its own, or has at least one trusted neighbor whose value is lesser than the node i's value. Thus, there is always a neighbor of i whose value is lesser than i, and is not ignored in computing $x_i(k_\epsilon + 1)$. Moreover, the maximum value of such a neighbor is $\mathcal{D}_M - \epsilon_0$ as it lies in the subset $\mathcal{V} \setminus \mathcal{S}_M(k_\epsilon, \epsilon_0)$. At the same time, the values of all other neighbors of i are bounded by $M(k_\epsilon)$. Since at each time step node i's state value is a convex combination of the state values of its neighbors and each coefficient in the combination is lower bounded by α, we have

$$\begin{aligned} x_i(k_\epsilon + 1) &\leq (1-\alpha)M(k_\epsilon) + \alpha(\mathcal{D}_M - \epsilon_0) \\ &\leq (1-\alpha)(\mathcal{D}_M + \epsilon) + \alpha(\mathcal{D}_M - \epsilon_0) \\ &\leq \mathcal{D}_M - \alpha\epsilon_0 + (1-\alpha)\epsilon \\ &= \mathcal{D}_M - \epsilon_1 \end{aligned} \tag{11}$$

Here, $\epsilon_1 = \alpha\epsilon_0 + (1-\alpha)\epsilon$, which satisfies $\epsilon_1 < \epsilon_0$. □

Similarly, if $i \in \mathcal{Z}_m(k_\epsilon, \epsilon_0)$, then we can show that

$$x_i(k_\epsilon + 1) \geq \mathcal{D}_m + \epsilon_1. \tag{12}$$

As a consequence of (11) and (12), at least one of the following is always true:

(i) $|\mathcal{S}_M(k_\epsilon+1, \epsilon_1) \cap (\mathcal{V} \setminus \mathcal{M})| < |\mathcal{S}_M(k_\epsilon, \epsilon_0) \cap (\mathcal{V} \setminus \mathcal{M})|$, i.e., the number of normal nodes in $\mathcal{S}_M(k_\epsilon+1, \epsilon_1)$ is strictly lesser than the normal nodes in $\mathcal{S}_M(k_\epsilon, \epsilon_0)$.

(ii) $|\mathcal{S}_m(k_\epsilon+1, \epsilon_1) \cap (\mathcal{V} \setminus \mathcal{M})| < |\mathcal{S}_m(k_\epsilon, \epsilon_0) \cap (\mathcal{V} \setminus \mathcal{M})|$.

Note that $\mathcal{S}_M(k_\epsilon+1, \epsilon_1)$ and $\mathcal{S}_m(k_\epsilon+1, \epsilon_1)$ are disjoint as $\epsilon_1 < \epsilon_0$. Next, we define $\epsilon_j = \alpha\epsilon_{j-1} - (1-\alpha)\epsilon$ for any $j \geq 1$. Note that $\epsilon_j < \epsilon_{j-1}$. Then, for any time step $k_\epsilon + j$, the above analysis can be repeated as long as $\mathcal{S}_M(k_\epsilon + j, \epsilon_j)$ and $\mathcal{S}_m(k_\epsilon + j, \epsilon_j)$ contain normal nodes. Since the number of normal nodes is finite, there exists a time step $k_\epsilon + K$ such that at least one of the following is always satisfied:

(a) $\mathcal{S}_M(k_\epsilon + K, \epsilon_K) = \emptyset$, which implies that the maximum value of any normal node at time step $k_\epsilon + K$ is upper bounded by $\mathcal{D}_M - \epsilon_K$, or

(b) $\mathcal{S}_m(k_\epsilon + K, \epsilon_K) = \emptyset$, which implies that the minimum value of normal nodes is lower bounded by $\mathcal{D}_m + \epsilon_K$.

If $\epsilon_K > 0$, then (a) implies a contradiction to the fact that largest value converges monotonically to $\mathcal{D}_M$, and (b)

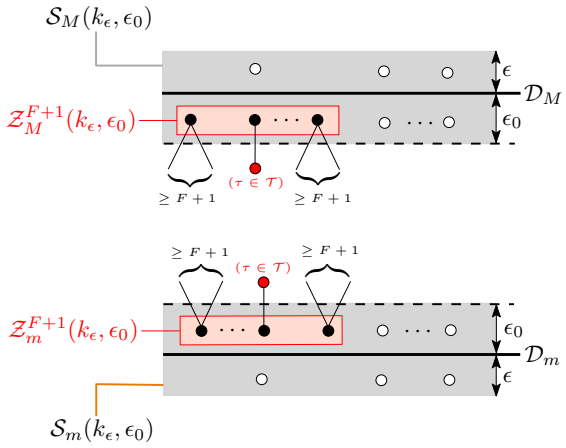


Fig. 13. Illustration of the proof of Theorem 6.1. Every node in $\mathcal{Z}_M^{F+1}$ and $\mathcal{Z}_m^{F+1}$ has at least $F+1$ neighbors outside, or at least one trusted neighbor outside the set containing the node.

contradicts to the fact that the smallest value converges monotonically to $\mathcal{D}_m$. Next, we show that $\epsilon_K > 0$.

$$\begin{aligned}\epsilon_K &= \alpha\epsilon_{K-1} - (1-\alpha)\epsilon = \alpha^K\epsilon_0 - (1-\alpha^K)\epsilon \\ &\geq \alpha^V\epsilon_0 - (1-\alpha^V)\epsilon.\end{aligned} \tag{13}$$

Since $\epsilon < \frac{\alpha^V}{1-\alpha^V}\epsilon_0$, we get $\epsilon_K > 0$, which gives the desired contradiction, thus proving that $\mathcal{D}_M = \mathcal{D}_m$.

• *In the F-local model, $F+1$-robustness with $\mathcal{T}$ is a necessary condition*:

Let $\mathcal{G}$ be a graph that is not $F+1$-robust, then there exist non-empty, disjoint subsets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathcal{V}$ such that each node in $\mathcal{S}_1$ and $\mathcal{S}_2$ has at most F neighbors outside of its respective set, that is $\mathcal{S}_1$ or $\mathcal{S}_2$. At the same time, there does not exist a node in $\mathcal{S}_1$ (and $\mathcal{S}_2$) that has a trusted neighbor outside of the set $\mathcal{S}_1$ (respectively $\mathcal{S}_2$). We assume state values of all nodes in $\mathcal{S}_1$ and $\mathcal{S}_2$ to be a and b respectively, where $a > b$. Moreover, each node in $\mathcal{V}\backslash(\mathcal{S}_1 \cup \mathcal{S}_2)$ has a value in the interval $(a\ b)$. Under this setting, each node in $\mathcal{S}_1$ would ignore the values of its neighbors outside of $\mathcal{S}_1$, and hence would never update its state value. Similarly, each node in $v \in \mathcal{S}_2$ would not change its value as v would not consider state values of the neighbors outside of $\mathcal{S}_2$. Consequently, consensus will not be achieved.

• *In the F-local model, $2F+1$-robustness with $\mathcal{T}$ is a sufficient condition*:

Using the same approach and arguments as in the *sufficiency proof of F-total model* above, we can show that $2F+1$-robustness with $\mathcal{T}$ is a sufficient condition to achieve consensus in the F-local model. ■

APPENDIX B
NUMBER OF r-ROBUST GRAPHS

The number of all possible r-robust graphs with n nodes is given in the following table. For each value of n and r in the above table, the adjacency matrices of all r-robust graphs with n nodes are available at
http://aronlaszka.com/data/r_robust_graphs.zip

TABLE I
NUMBER OF r-ROBUST GRAPHS

n	Total number of graphs	Number of r-robust graphs				
		$r=1$	$r=2$	$r=3$	$r=4$	$r=5$
3	4	2	1	0	0	0
4	11	6	2	0	0	0
5	34	21	8	2	0	0
6	156	112	45	8	0	0
7	1044	853	398	65	6	0
8	12346	11117	6372	1140	64	0
9	274668	261080	182859	44861	1977	26

Waseem Abbas is a postdoctoral research scholar in the Department of Electrical Engineering and Computer Science at Vanderbilt University, Nashville, TN. He received Ph.D. (2013) and M.Sc. (2010) degrees, both in Electrical and Computer Engineering, from Georgia Institute of Technology, Atlanta, GA. He was a Fulbright scholar from 2009 till 2013. His research interests are in the area of network control systems, graph-theoretic methods for large networked systems, and resilience of cyber-physical systems.

Aron Laszka is an Assistant Professor in the Department of Computer Science at the University of Houston, TX, USA. Previously, he was a Research Assistant Professor in the Department of Electrical Engineering and Computer Science at Vanderbilt University, and a Postdoctoral Scholar at the University of California, Berkeley between 2015 and 2016. He graduated summa cum laude with a Ph.D. in Computer Science from the Budapest University of Technology and Economics in 2014. In 2013, he was a Visiting Research Scholar at the Pennsylvania State University. His research interests broadly revolve around the security and resilience of cyber-physical systems and the Internet-of-Things, the economics of security, and game-theoretic modeling of security problems.

Xenofon Koutsoukos received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, USA, in 2000. He is a Professor with the Department of Electrical Engineering and Computer Science and a Senior Research Scientist with the Institute for Software Integrated Systems (ISIS), Vanderbilt University, Nashville, TN, USA. He was a Member of Research Staff at the Xerox Palo Alto Research Center (PARC) (2000–2002), working in the embedded collaborative computing area. His research work is in the area of cyber-physical systems with emphasis on formal methods, data-driven methods, distributed algorithms, security and resilience, diagnosis and fault tolerance, and adaptive resource management. Prof. Koutsoukos was the recipient of the NSF Career Award in 2004, the Excellence in Teaching Award in 2009 from the Vanderbilt University School of Engineering, and the 2011 NASA Aeronautics Research Mission Directorate (ARMD) Associate Administrator (AA) Award in Technology and Innovation.