# Using Trust Measures to Optimize Neighbor Selection for Smart Blockchain Networks in the IoT

Giancarlo Fortino, *Fellow Member, IEEE*, Fabrizio Messina, Domenico Rosaci, Giuseppe M. L. Sarnè, *Senior Member, IEEE*

This is a post-print version of the paper published at DOI: 10.1109/JIOT.2023.3263582 in the IEEE Internet of Things Journal, Volume 10, Issue 24, Pages 21168 - 21175.

***Abstract*—The blockchain paradigm is currently modifying all the major Internet of Things (IoT) application domains, giving the opportunity of constructing decentralized environments in which trustful and anonymous activities can be efficiently performed. Blockchain proposes an approach which assures data saved on a distributed ledger would be continuously synchronized and in such a way the ledger will remain consistent. The distributed ledger has the responsibility to maintain a higher level of consistency. Given a connected network, the Optimum Neighbor Selection (ONS) of paths can be obtained by finding the Minimum Spanning Tree of the network. However, none of past approaches proposed in the literature to construct ONS considered the problem of having nodes with low levels of reliability in the network and even malicious or fraudulent nodes, that is a situation very common in an IoT environment. In this paper, we propose an optimized blockchain ONS algorithm, called Trust-based ONS (TONS), which allows the the miners to communicate with a globally-optimized selection of neighbors. The algorithm can also guarantee that these nodes are the most reliable miners. We also describe an experimental simulation campaign we have performed to evaluate the effectiveness and the efficiency of our approach.**

***Index Terms*—Blockchain, Internet of Things, Trust and Reputation.**

## I. Introduction

In the last years, the potential of the Internet of Things (IoT) to deliver important services in many different domains [1]–[3], like from social e-business to smart cities or from industry to intelligent transportation systems, mainly derives from the possibility of interconnecting heterogeneous devices. In turn, IoT devices more and more are provided with different functionalities, into a machine- and human-centric network, in order to meet the most recent requirements of the aforementioned domains.

However, the overwhelming number of connected devices and the often large traffic of data lead to a bottleneck in reaching the required levels of quality-of-service, due to the well-known constraints of the IoT devices in terms of bandwidth, storage and computational capabilities [4], [5]. The relatively recent paradigm of Blockchain [6] (BC), is currently modifying all the major IoT application domains, giving the possibility of constructing decentralized environments in which trustful and anonymous activities can be efficiently performed. The IoT world largely benefits from the combination with the BC technology, due to the decentralized management, the lower operational costs and, first of all, the robustness against malicious attacks, making convergence of IoT and BC as one of the most significant challenges for realizing the IoT of the future [7]–[9].

All the advantages that the IoT obtains, in consequence of the introduction of the BC, derive from the possibility of exploiting a Distributed Ledger (DL) to manage the transactions. A DL is a digital system in which the transactions of assets and their details are contemporary recorded on several nodes. Differently from traditional databases, into DLs there are not central data stores or global administration needs, since each node processes and autonomously verifies every data item, thereby generating a record of each item and creating a consensus about its validity.

One of the most important problem in a DL environment is represented by the *consistency* of the Ledger [10]. Several situations can generate different readings, e.g. the continuous alteration of data and the fluctuating transmission delay between nodes. The BC is a technology that, although it did not directly solve the consistency problem, proposes an approach which assures data saved on the DL would be continuously synchronized and, in such a way, the DL remains consistent.

In this context, many past studies proposed in the literature show that in a BC-based DL we have higher level of consistency when there is both a small number of neighbors per miner and a low delivery time rates between neighbors [11], [12]. Moreover, the possible presence in the system of miners having low levels of reliability, or even showing malicious or fraudulent behaviors, introduces another issue to address their impact on the consistency of the DL.

Therefore, given a connected network, the Optimum Neighbor Selection (ONS) of paths can be obtained by finding the Minimum Spanning Tree (MST) of the network. Past approaches proposed in the literature as, for instance, [12], compute the MST of a given BC network, thus obtaining both enhanced DL consistency and enhanced data finality. However, at the best of our knowledge, none of these past approaches considered the problem of having nodes with low levels of reliability in the network and even malicious or fraudulent nodes that, conversely, is a situation potentially very common in an IoT environment.

To this purpose, in this paper, we propose an optimized

G. Fortino is with the Department DIMES, Univ. of Calabria, Via P. Bucci, 87036 Rende (CS), Italy, (e-mail: giancarlo.fortino@unical.it).

F. Messina is with the Department of Mathematics and Computer Science, University of Catania, Italy (e-mail: messina@dmi.unict.it).

D. Rosaci is with the Department DIIES , University of Reggio Calabria Via Graziella, Loc. Feo di Vito, 89122, Reggio Calabria, Italy (e-mail: domenico.rosaci@unirc.it).

G.M.L. Sarné is with the Department of Psychology, University of Milano Bicocca, P.za dell'Ateneo Nuovo 1, 20126 Milano (MI), Italy, (e-mail: giuseppe.sarne@unimib.it).

BC networking algorithm, called **T**rust-based **ONS** (TONS), which allows the miners to communicate with globally-optimized selection of neighbors, and also guaranteeing that these nodes are the most reliable miners possible. We highlight that the main contributions introduced by our proposal are the following: (*ii*) our algorithm is based on a trust and reputation model used to evaluate the trustworthiness of the blockchain nodes, with the purpose to effectively model the environment in which the miner nodes operate, giving the possibility to consider the presence of misbehaving nodes; (*ii*) our algorithm computes the optimal neighbor selection of the network, taking into account both delivery time rates and reputation of the nodes.

In order to validate our approach, we have performed several simulations of networks built following the well known Barabási–Albert (BA) Erdős–Rényi (ER) and network models [13], [14], showing that our proposed algorithm outperforms the classical RNS solution, used in most cases to compute ONS in blockchain networks, both in terms of efficiency (time and number of exchanged messages to build ONS) and effectiveness (percentage of misbehaving nodes included in the ONS). Our evaluation therefore shows that TONS is particularly suitable to be applied in the IoT domain, where the issue of detecting and neutralizing misbehaving nodes is currently a key challenge.

We also notice that although our algorithm introduces additional time cost to compute the trust measures, however, this cost does not affect the efficiency of the algorithm in constructing the ONS, since the two threads are performed by the node independently to each other.

The paper is organized as follows. In Section II, some related work is presented. Section III introduces some technical preliminaries and our research problem. The core of our proposal, that is the algorithm to compute the MST, is described in Section IV, while Section V is devoted to describe some experiments we have performed to evaluate the effectiveness and the efficiency of our approach.

## II. Related Work

~~IoT is a pervasive technology which has transformed our lives by connecting people and smart objects (equipped with sensing, processing, actuation and communication features) introducing intensive interactions and new services. Usually, IoT devices hold limited computational, storage and power capabilities, while IoT applications require connectivity and power for a long time also to handle the large volumes of data they usually generate. Moreover, IoT systems are vulnerable to privacy and security threats and, also in large IoT networks, the presence of a single failure point could lead to the crash of the whole system.~~

Blockchain technology is computational expansive and it is aimed to realize a distributed ledger on a peer-to-peer (P2P) architecture, organized in a chain of data blocks [15]. The adoption of a distributed consensus protocol enables BCs to work both in presence of unreliable actors and in absence of trusted third parties. As a result, BCs are commonly considered secure against attacks and threats (although several vulnerabilities have been identified [16]), behind to be transparent, immutable, provided with both privacy-preserving properties and able to keep a complete and public transaction history [17].

~~For integrating these two complementary technologies several issues must be addressed. However, the task of representing a comprehensive background about the large number of issues posed by IoT and BC integration is beyond our aims. Therefore, we will discuss only those proposals that, in our opinion, come closer to the main aspects characterizing our approach. In particular, we will focus on the adoption of trust and reputation systems and the neighbors searching in IoT and BC-based systems. In particular, providing the involved actors with trust and reputation information is essential to mitigate uncertainty, risk perception and to foster acceptance and consumption of IoT services and applications.~~

Usually, IoT devices hold limited computational, storage and power capabilities, while IoT applications require connectivity and power for a long time also to handle the large volumes of data they usually generate [18]–[21].

Trust and reputation systems enable qualified services enriched by context-aware intelligence also in order to safeguard user privacy and information security [22], [23], while recently BCs technology has been exploited to support trust management in IoT networks [24].

More specifically, synergies between trust (i.e., reputation) systems and BCs are useful in tackling a wide variety of attacks made easier by both the characteristics of devices, such as the heterogeneity and limited resources of smart objects, and the high dynamism of IoT networks. For instance, in [25] a BC-based architecture for a decentralized model to exchange trusted data is proposed. ~~To this aim, the IoT data exchange requirements have been classified on three categories (i.e., trusted trading, trusted data access and trusted privacy preservation), while the trustworthiness is provided by the joined BC, which also adopts a high-efficiency consensus algorithm. An Ethereum BC has been exploited to realize a prototype of this system.~~

Another interesting proposal is TrustChain [25] which safeguards from sybil attacks[1] by replacing the Proof-of-Work with a mechanism able to determine the transactions validity and integrity. ~~TrustChain contributes to build an immutable chain of temporally ordered transactions signed by each agent and computes agents' trust scores provided of Sybil-resilience in an online IoT community.~~

In federated dynamic social IoT environments, IoT devices can migrate across the different environments. Therefore, in each environment, a share of IoT devices could be mutually untrusted, making potentially risky the choice for a partner with which interacting and cooperating. To tackle this issue, in [26] multi-agent and BC technologies are leveraged to form in each federated IoT environment some groups of trusted IoT devices (each associated with a software agent) on the basis of their reputation score. ~~To disseminate trusted and certified device/agent reputation scores in each federated environment, the authors relied on a BC. Experiments have~~

[1] The sybil attack consists in gaining a large influence within the community by creating many fake identities to increase or decrease the reputation of some agents.

~~shown as this approach can detect a variety of deceptive behaviors and support agents in forming effective groups of trusted devices able to promote honest behaviors.~~ A different BC-based trust model for IoT is described in [27]. This system adopts a multilevel approach to compute a reputation score in a decentralized manner for both new and known entities by dynamically aggregating the information sources. Smart contracts and BC ensure the trustworthiness of scores and their security. The authors of [28] designed a distributed trust model for IoT exploiting existing trust domains connected for creating trusted relationships between devices in an end-to-end manner. The BC supports this trust system through a new cryptographic primitive called obligation chain. ~~This system is suitable to a wide range of scenarios and business models without introducing the typical latency of BCs. Experimental tests prove the benefits of the proposed solution and its scalability in presence of a large number of IoT devices.~~

An important aspect of our proposal is the computation of the Minimum Spanning Tree[2] (MST) search. A number of centralized, semi-distributed or distributed MST algorithm have been proposed in the literature [30]–[33]. Even though a distributed MST is the most suitable in a fully distributed permissionless BC scenario because they can work without requiring a prior knowledge of network topology or disclosing the peer identity or exploiting trusted third parties Different proposals there exist in the literature that in presence of IoT devices and BC include a MST search step.

In this respect, DONS [27] (Dynamic and Optimized Neighboring Search) is a hybrid architecture where an elected peer manages the network topology. The leader exploits the neighbor lists to compute the best neighbors for each peer by solving a MST on the basis of the message propagation delays. Consequently, each peer can know its optimum neighboring offering the best propagation delay. ~~A test campaign has verified an improvement in transactional throughput about the security and the privacy of this approach. However, the neighbor list of all peers and the topology calculation must be repeated if, for some reason, the current leader becomes unavailable. In this case suitable countermeasures can be implemented to avoid the repetition this task. Note as the computation time of the MST obviously increases as the number of peers increases.~~

Synchronizing BC miners in adding transaction blocks can be critical due to propagation delays of the broadcast messages. To such a purpose, in [34] an adaptive broadcasting mechanism for BC authentication, authorization, and accounting services is proposed by combining transmission (for resending lost messages) and computational (due to data verification) times. ~~To obtain secure transmission, and subsequent miners synchronization, a MST problem to minimize processing and transmission delays is solved. Computational experiments demonstrate the effectiveness of the proposed solution.~~ Similarly, in [35] to minimize the time gap between the propagation of a winning block and the start of the next mining competition, a message propagation mechanism is designed for taking advantage from exploiting the closest neighbors on the basis of the message latency (named Round Trip Time). ~~In other words, the lower the latency, the closer the neighbor. Simulations have shown as this method also can evaluate the influence of simultaneously established connections.~~

For IoT scenarios, in [36] edge computing is adopted to reduce system latency and bandwidth limitation, while BC obviates the problem of edge computing security through a three-tier network model, called blockchain-based mobile edge computing system (BMEC), leveraging a solution using artificial neural networks. A MST search is performed on a weighted indirect graph, formed by edge-based blocks to improve the BC transaction speed, constructed based on predefined rules of priority, application type, and past behaviors of edge devices. Both low transaction speed and latency can limit the efficiency of cryptocurrency transactions.

A possible solution to optimize the number of payments between untrusted peers is the use of payment channels by leveraging sets of intermediate nodes that forward and load payments. Therefore, a possible solution for cryptocurrency payments, between untrusted peers, to avoid limitations due to transaction speed and latency can be that to perform payment over channels that exploit intermediate nodes to forward and charge payments. Most of the existing solutions rely on just choosing the shortest path, but this may cause a rapid saturation of that canal, as well as the inability to make multiparty payments simultaneously on the same channel. Differently, the proposal of Chen et al. [37] allows the same intermediate channel to be used simultaneously for multiple payments as well. For this purpose, a MST is calculated to select those intermediate nodes that are more robust, and give higher guarantees of non-saturation, by using channel transmission quality as the weight of the tree.

## III. Technical preliminaries

Let $NET = \langle N, A, W\rangle$ be a blockchain network, consisting of a undirected, weighted and connected graph, where $N$ is a set of miner nodes in $NET$ and $A = a_{i,j}$ with $i, j \in N$ is the set of bidirectional communication lines between the miners.

Each arc $a_{i,j} \in A$ is associated with a pair $w_{i,j} = (t_{i,j}, \tau_{i,j}) \in W$, called *weight*, where $t_{i,j}$, called *time-weight*, is the transmission time needed to deliver 1 byte of data from miner $i$ to miner $j$ (or vice versa), while $\tau_{i,j}$, called *trust-weight*, represents the mutual trustworthiness between miners $i$ and $j$. Moreover, we call *overall weight* of the arc $a_{i,j}$ the positive value $ow_{i,j} = t_{i,j} \cdot \tau_{i,j}$, i.e. the product between time-weight and trust-weight. We also define the *global overall weight* of the entire network $NET$ ($gow$) as the sum of the overall weights of all its arcs.

We assume that each node $i \in NET$ knows its own neighbors $\nu_i = (\nu_{i,1}, .., \nu_{i,l})$ and the related weights. In our model we build the adjacency matrix, which is a matrix of size $|N \times N|$, where each element of the matrix is equal to $ow_{i,j}$ if the arc $a_{i,j}$ exists in the network. A sub-network of $NET$ is a graph $NET^*(N^*, A^*, W^*)$, such that $N^* \subseteq N$ and $A^* \subseteq A$,

[2] A Minimum Spanning Tree is a minimum-weight tree that in a graph, connected and with undirected arcs, contains only that subset of the arcs necessary to connect all vertices with each other by one and only one path, is a problem widely addressed in the literature [29], [30].

and $NET^*$ is also undirected. Moreover, $NET^*$ will inherit the weigths of the adjacency matrix of the graph $NET$.

In such a context, a *Spanning Tree* $ST$ of $NET$ is a connected acyclic sub-graph of $NET$ where $N^* = N$ and $\mid A^* \mid = \mid A \mid -1$. Consequently, a *Minimum Spanning Tree* $MST$ of $NET$ is a unique Spanning Tree where the global weight of $MST$ is minimum compared to all Spanning Trees of $NET$.

### A. Research Problem

The problem we desire to solve is the *Optimum Neighbor Selection* ($ONS_i$), which is to find the subset $h_i = (h_1, ..h_n) \in \nu_i, \forall i \in N$, such that $a_{i,h} \in MST_{NET} \forall h \in h_i$.

## IV. Our proposed algorithm for computing the ONS

This section describes the algorithm to compute the ONS. As discussed before, our algorithm generates a global view of the underlying blockchain network, by computing the $MST$.

First of all, the algorithm must be executed periodically, because miners can be continuously attached and detached from the network, changing this global view. Secondly, we suppose that one of these miners (the *leader*) performs the MST computations on the behalf of all the others.

In our solutions, as we discuss in SectionV, we have used the approach described in [12]. In few words, the leader shall collect public local views from all peers, then it solves the MST problem of the network graph and, finally, broadcasts the information (in anonymous form) about the MST throughout the network.

Moreover, we do not deal with the selection of the leader, that can be obtained by any of the approaches proposed in the literature as, for example by [12], [38]. Our focus is instead on the construction of the $MST$, taking into account the trust aspect.

In particular, in the following of this section, we describe the main aspects of our trust-based ONS: *i*) feedback and assurances; *ii*) the trust model; *iii*) the procedure to compute the various components of the trust model.

### A. Feedback, recommendations and assurances

Let $a$ and $b$ two nodes that may interact in the blockchain network, such that $a$ may ask $b$ to provide data for the blockchain construction. In this situation, $a$ may require to a third node $c$ a "*recommendation*" (a value belonging to the real domain $[0, 1]$) about $b$. Otherwise, $a$ can ask $b$ itself for providing an auto-declaration of its expertise.

In our approach the interactions between nodes are conducted by exchanging a *proof* which synthetically describes the performed interaction. For this reason, the possible recommendation of node $c$ about node $b$ is based on the previous interactions of $c$ with node $b$. Our protocol is capable of generating, as an output, a $[0..1]$-real value of assurance (where $1$ is the maximum level of assurance), representing a trust measure of the recommendation that node $c$ sends to a requester node $a$ about a third node $b$. The protocol generates the level of assurance by evaluating the proof that $c$ is able to show to $a$ for ensuring the level of assurance about the provided recommendation.

The maximum level of assurance is reached when all the interactions between $b$ and $c$, on which $c$ produces its recommendation, are traced through messages whose authenticity and non-repudiation are based on a certification released by a reliable witness and recorded on the blockchain itself. Differently, if for some transactions a weaker mechanism is adopted, the level of assurance decreases. The minimum value is obtained in the case of all transactions with no proof.

If $a$ selects $b$, the latter provides data to $a$ and, finally, node $a$ assigns a "*feedback*" to $b$.

### B. Trust Model: Formal Definition

We define a a set of four mappings, denoted by $T_i$, $Rep_i$, $\beta_i$, and $P_i$. The four mappings are associated with each node $i \in N$ belonging to the blockchain network $NET = \langle N, A, W \rangle$ as defined in Section III. In particular, each mapping takes a node $j$ as input and yields as output a different trust measure that $i$ assigns to $j$. Each measure is represented by a real number belonging to the interval $[0, 1]$, where 0 (1, resp.) is the minimum (maximum, resp.) value of trust.

In particular:

- $T_i(j)$ represents the overall *trust* that $i$ assigns to the interactions with $j$.
- $Rep_i(j)$ represents the *reputation* that $i$ assigns to $j$. Reputation is a measure of trustworthiness that a node assigns to another node based on some recommendations coming from nodes of the blockchain.
- $tw_i(j)$, called *trust weight*, represents the *weight* that $i$ assigns to the reliability aspect w.r.t. reputation in evaluating $j$. In other words, when $i$ has to compute the overall trust score of a node $j$, it considers both the contribution of the trust $T_i(j)$ and the reputation $Rep_i(j)$. The percentage of relevance to assign to the trust with respect to the reputation is represented by the value $tw_i(j)$. In our framework, the mapping $tw_i$ is computed by the node $i$ based on the *assurance information*, provided together with the recommendations by the contacted nodes.
- $S_i(j)$ represents the overall "score" (preference) that $i$ assigns to $j$, based on both the reliability and reputation perceived by $i$.

Besides the four mappings described above, we define in our framework a mapping denoted by $RC_i$, representing the *recommendations* obtained by node $i$.

We define a *recommendation* as a pair $r = \langle v, l \rangle$, where $v$ and $l$ are two [0..1]-real numbers called *recommendation value* and *recommendation level of assurance*, respectively.

Formally, $RC_i(j, k)$ is a mapping that receives two nodes $j$ and $k$ , and yields as output a recommendation $RC_i(j, k)$ representing the recommendation that the node $j$ provided to the node $i$ about the node $k$, together with a measure of the level of assurance that can be associated with this recommendation.

### C. The updating process of trust measures

Each node is responsible for updating its own mappings. This updating process is described by the following phases:

- **Phase 1: Reception of the Recommendations**. The node $i$ receives some recommendations from the other nodes, in response to previous recommendation requests. Such recommendations are then encoded in the $RC_i$ mapping. In particular, each recommendation coming from a node $j$ and related to a node $k$ is contained in a *recommendation message* $m$, which is a tuple $\langle v, l\rangle$, whose elements are stored by $i$ in the mapping $RC_i(j,k).v$ (the recommendation value) and $RC_i(j,k).l$ (the recommendation level of assurance), respectively.
- **Phase 2: Computation of T mapping**. $i$ updates the T mapping for any node $j$ whenever $i$ has interacted with some node $j$ and, as a consequence, it has released one or more feedback for the contributions given by node $j$. These feedback are contained in a mapping $FEED_i^k(j)$, which is a real number belonging to $[0,1]$, representing the quality of the collaboration that the node $j$ provided to the node $i$ during the $k$th interaction. A feedback equal to 0 (1, resp.) means minimum (maximum, resp.) "quality of the service".

  Based on these feedback, the node $i$ updates its own mapping $T_i$, computing the current reliability shown by a node $j$ by averaging all the feedback concerning $j$. Therefore, denoting by $m$ the number of recent interactions between $i$ and $j$, the current trust $T_i(j)$ is computed as:

$$T_i(j) = \frac{1}{m}\sum_{k=1}^{m} FEED_i^k(j) \tag{1}$$

  At each new step, this current reliability is taken into account for updating the element $T_i$, averaging the value of $T_i$ at the previous step $t-1$ and the current reliability computed at the new step $t$, denoted by $T_i^t$. Thus:

$$T_i^t(j) = \alpha \cdot T_i^{(t-1)}(j) + (1-\alpha)\cdot T_i(j) \tag{2}$$

  where $\alpha$ is a real value belonging to $[0,1]$, representing the importance that $i$ assigns to the past evaluations of reliability with respect to the current evaluation.
- **Phase 3: Computation of Rep and $\beta$**. The recommendations contained in the mapping $RECC_i$ are used by the node $i$ to compute the reputations of the other nodes of the community. In particular, $i$ computes the reputation of another node $j$ as a weighted mean of all the recommendations received from the other nodes of the community concerning $j$ (let us denote by $AS$ this set), where the weight of each recommendation value is the corresponding level of assurance. Thus $Rep_i(j)$ is equal to:

$$\frac{\sum_{k\in AS, k\neq i} RECC_i(k,j).v \cdot RECC_i(k,j).l}{\sum_{k\in AS, k\neq i} RECC_i(k,j).l} \tag{3}$$

  where, we recall, $RECC_i(k,j).v$ (resp., $RECC_i(k,j).l$) is the value (resp., the level of assurance) of the recommendation that the node $k$ provided to the node $i$ about the node $j$.

  The $\beta$ coefficient associated with the node $i$ is recorded in the mapping $\beta_i$. The computation of the average level of assurance of the recommendations related to a node $j$, denoted by $\beta_i(j)$, is obtained by averaging the level of assurance associated with all the recommendations related to $j$. Thus:

$$\beta_i(j) = \frac{\sum_{k\in AS, k\neq i} RECC_i(k,j).l}{|AS|-1} \tag{4}$$

- **Phase 4: Computation of S**. The node $i$ finally computes the overall score $S_i(j)$ in the node $j$ by taking into account both the trust $T_i(j)$ and the reputation $Rep_i(j)$. In particular, the value of the mapping $\beta_i(j)$ is used to weight the importance of the service reliability with respect to reputation:

$$S_i(j) = \beta_i(j)\cdot T_i(j) + (1-\beta_i(j))\cdot Rep_i(j) \tag{5}$$

- **Phase 5: Sending the mapping S to the leader**. Each node $i$ sends its mapping $S$ to the leader l of the network.
- **Phase 6: Computation of ONS**. The leader L uses Prim's approach [39] to find the $MST$. To this purpose, the leader computes the trust-weight $\tau_{i,j}$ of the arc $a_{i,j}\in A$, as the arithmetic mean between the score measures $S_i(j)$ and $S_j(i)$. Finally, the Leader computes its own $ONS$ from the MST. Then, it sends the MST to the miners of the ONS. Each node, once it has received the MST, forward it to its own neighbors.

At each step, the node $i$ exploits the mapping $P$ to select the most suitable candidates to require a collaboration.

## V. Experiments

This section describes the experiments we performed to evaluate our algorithm in terms of effectiveness and efficiency. To estimate the advantages deriving from the use of trust measures, we compared our TONS algorithm with another version of the same algorithm, that we have called Classical ONS (CONS), where only the transmission time, as the weight associated with each network arc, is used. Moreover, we also compared TONS with the approach called Random Neighbor Selection (RNS) presented in [27].

Our experiments were carried out on a ASUS PC equipped with an Intel i7 CPU (8-Cores, 7GHz), 32 GB DDR4-SDRAM, 1 TB of SSD and Windows-11 OS. We realized several experiments by exploiting two random network models, namely Barabási–Albert (BA) Erdős–Rényi (ER) and network models [13], [14].

We varied the number of nodes to analyze the behavior of the algorithm in presence of networks having different sizes. Moreover, we also varied the percentage of misbehaving nodes randomly generated, where a misbehaving node in our simulation generates fraudulent messages.

In our simulations a BC network is built randomly and a miner selected randoly is initially marked as the node which has sent a block of data. The node then shares the generated data with its own neighbors; each of them will send this data with a group of its neighbors, and so on. The simulation will end when data has spread over the entire network, reproducing the common gossiping approach that is generally adopted in BC applications.

The three node selection methods compared in this work are adopted, in our experiments, for identical networks and same generated data. At the end of each simulated scenario, we computed (*i*) the total time, (*ii*) the number of exchanged messages and (*iii*) the percentage of misbehaving nodes selected in the ONS.

The complete set of parameters on which our experiments are based, as well as the results, for the algorithm TONS, CONS and RNS and for the two random models BA and ER, are presented in Table I and Table II, respectively. The results obtained by TONS, CONS and RNS algorithms in terms of effectiveness and efficiency are described in details in the next two sub-sections.

## A. Effectiveness in detecting misbehaving nodes

The obtained results clearly show that algorithm TONS significantly outperforms CONS and RNS in terms of percentage of detecting misbehaving nodes both for the networks built following the BA model and those generated accordingly to ER model.

In particular, the advantage introduced by TONS increases with the size of the network, as graphically highlighted in Figures 1 and 2 with respect to CONS (which is the second best performer) comes from a minimum of $25.1\%$ (resp. $27.7\%$) for a network size of $150$ nodes to a maximum of $65,8\%$ (resp. $65,0\%$) percent for a network size of $1000$ nodes, in the case of the BA (resp. ER) networks.

Such an advantage is evidently due to the adoption of trust measures in computing the weights of the networks arcs. Indeed, the advantage is particularly higher in the case of large-size networks depending it on the fact that in this case the misbehaving nodes are more difficult to be detected than in small-size networks without the use of trust information; differently, TONS algorithm is able to easily recognize malicious behaviors by using certificated reputation.

We also note that RNS is clearly the worst performer in all cases, with performances in terms of avoiding misbehaving nodes that are $28-46\%$ worse than those of TONS and $9-10\%$ worse than those of CONS.

| **number of nodes** | **150** | **300** | **500** | **1000** |
|---|---|---|---|---|
| **average neighbors** | **2** | **2** | **5** | **7** |
| TONS time | 20.81 | 37.43 | 60.21 | 101.66 |
| CONS time | 25.43 | 46.12 | 79.18 | 135.74 |
| RNS time | 27.17 | 49.55 | 84.29 | 154.12 |
| TONS number of messages | 711 | 3515 | 9881 | 17553 |
| CONS number of messages | 792 | 4243 | 12466 | 23602 |
| RNS number of messages | 853 | 4711 | 13861 | 26423 |
| TONS % mis. nodes identified | 96.7 | 94.2 | 91.1 | 89.7 |
| CONS % mis. nodes identified | 77.2 | 66.2 | 64.1 | 54.1 |
| RNS % mis. nodes identified | 69.9 | 58.1 | 57.0 | 48.6 |

TABLE I
RESULTS OF TONS, CONS AND RNS ALGORITHMS SIMULATION EXPERIMENTS ON THE BA RANDOM NETWORK MODEL WITH DIFFERENT SIZES

| **number of nodes** | **150** | **300** | **500** | **1000** |
|---|---|---|---|---|
| **connection probability** | **0.02** | **0.015** | **0.01** | **0.007** |
| TONS time | 21.73 | 35.29 | 58.17 | 100.62 |
| CONS time | 23.61 | 39.11 | 69.93 | 127.12 |
| RNS time | 26.11 | 44.57 | 77.54 | 145.31 |
| TONS number of messages | 757 | 4224 | 9147 | 18621 |
| CONS number of messages | 819 | 4677 | 10544 | 22477 |
| RNS number of messages | 920 | 5223 | 12111 | 25719 |
| TONS % mis. nodes identified | 97.15 | 95.67 | 92.44 | 90.91 |
| CONS % mis. nodes identified | 76.04 | 65.12 | 63.67 | 55.12 |
| RNS % mis. nodes identified | 65.11 | 58.27 | 55.67 | 44.21 |

TABLE II
RESULTS OF TONS, CONS AND RNS ALGORITHMS SIMULATION EXPERIMENTS ON THE ER RANDOM NETWORK MODEL WITH DIFFERENT SIZES

## B. Efficiency

The capability of detecting misbehaving nodes impacts also on the performances of the TONS algorithm in terms of efficiency, since considering not recognized misbehaving nodes in the blockchain activities slows down the construction of the ONS and generates a large number of messages.

Indeed, the obtained results show (see Figures 3 and 4) that TONS, in the case of a BA network, is $18\%$ quicker than CONS in building the ONS for a network having $150$ nodes, and about the $25\%$ percent quicker in the case of a network with $1000$ nodes.

Moreover, as shown in Figures 5 and 6 always in the case of a BA network, TONS generates a $10\%$ fewer messages than CONS for a network having $150$ nodes, and $25\%$ fewer messages than CONS for a network having $1000$ nodes. Very similar results are obtained for ER networks.

Similarly to the the effectiveness performances, we highlight that also in terms of efficiency RNS is clearly the worst performer in all cases, presenting time performances that are $30-50\%$ worse than those of TONS and $7-14\%$ worse than those of CONS, and showing performances in terms of generated messages that are $20-50\%$ worse than those of TONS and $8-12\%$ worse than those of CONS.

## VI. CONCLUSIONS

Blockchain systems are mainly dependent on the consistency of the Distributed Ledger, where the consistency state is

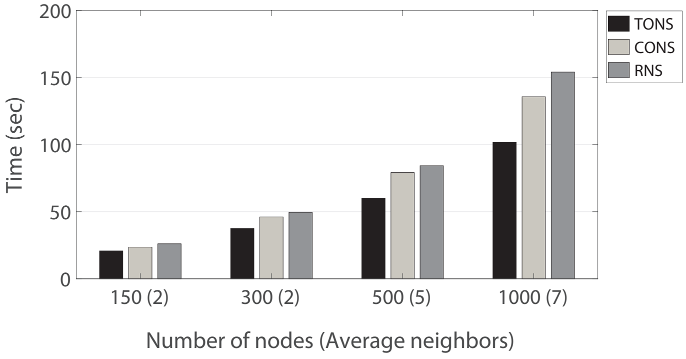


Fig. 1. Time to detect misbehaving nodes - Barabási–Albert (BA) network model

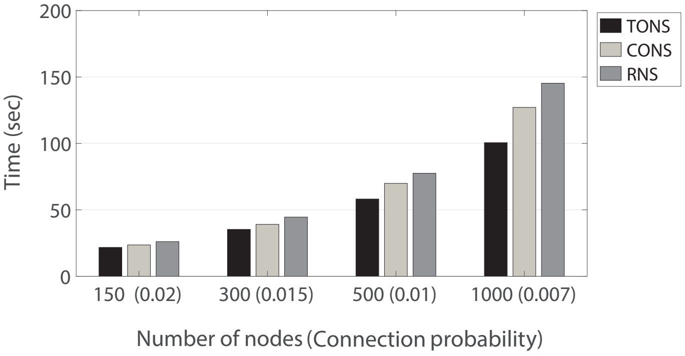


Fig. 2. Time to misbehaving nodes - Erdős–Rényi (ER) network model

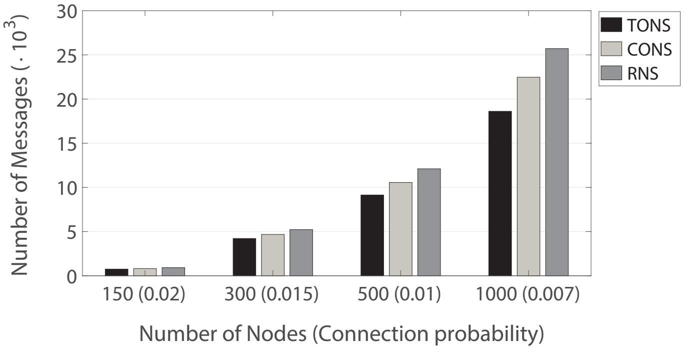


Fig. 4. Number of messages - Erdős–Rényi (ER) network model

generally obtained to propagate shared data, through shortest possible paths in the network, optimizing the Neighbor Selection for each node.

Previous works in the literature have a drawback particularly relevant in the IoT, i.e. the problem of having nodes with low levels of reliability in the network and even malicious or fraudulent nodes.

In this work, we address this issue by presenting an optimized BC networking algorithm, called **T**rust-based **ONS**, which allows the miners to communicate with globally-optimized selection of neighbors (with the minimum number of neighbors), and also assuring that these nodes are the most trustable miners, based on a trust and reputation model exploited to evaluate the trustworthiness of the blockchain nodes. The TONS algorithm computes the optimal neighbor selection, taking into account not only the delivery time rates but also the reputation of the nodes in the network.

Our experiments performed simulations of networks built by following the well known ER and BA network models and show that our proposed algorithm outperforms the classical RNS solution, used in most cases to compute ONS in blockchain networks, both in terms of efficiency (time and number of exchanged messages to build ONS) and effectiveness (percentage of misbehaving nodes included in the ONS). Therefore, the results of our experimental campaign show that TONS is particularly suitable to be applied in the IoT domain, where the issue of detecting and neutralizing misbehaving nodes is currently a key challenge.

We highlight that the usage of our algorithm introduces the time cost to compute the trust measures, that is generated during the transactions performed in the network. However, this cost does not affect the efficiency of the algorithm in constructing the ONS, since the two threads are performed by the node independently to each other. Differently, the power consumption to calculate trust measures is a cost that actually impact on the total power consumption associated with each node. In this study we do not have analysed such an aspect, which is currently the focus of our ongoing research.

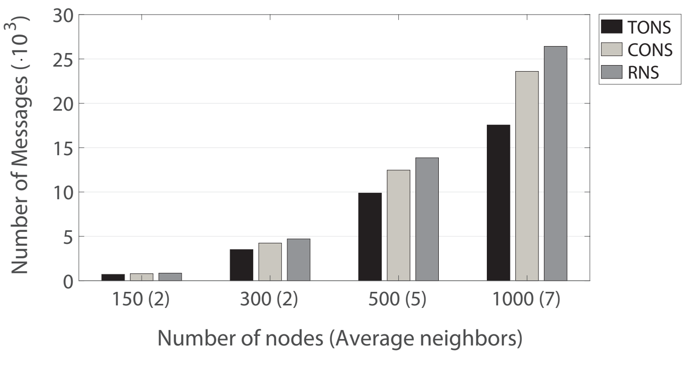


Fig. 3. Number of messages - Barabási–Albert (BA) network model

## Acknowledgment

This work was partially supported by the projects "T-LADIES" (PRIN 2020TL3X8X) granted by the Italian MUR, "Fluidware" (PRIN 2017KRC7KT) granted by the Italian MUR, and by Pia.ce.ri. 2020-2022 granted by the University of Catania.

## References

[1] P. Asghari, A. M. Rahmani, and H. H. S. Javadi, "Internet of things applications: A systematic review," *Computer Networks*, vol. 148, pp. 241–261, 2019.

[2] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," *Information systems frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[3] J. E. Ibarra-Esquer, F. F. González-Navarro, B. L. Flores-Rios, L. Burtseva, and M. A. Astorga-Vargas, "Tracking the evolution of the internet of things concept across different application domains," *Sensors*, vol. 17, no. 6, p. 1379, 2017.

[4] Y.-K. Chen, "Challenges and opportunities of internet of things," in *17th Asia and South Pacific design automation conference*. IEEE, 2012, pp. 383–388.

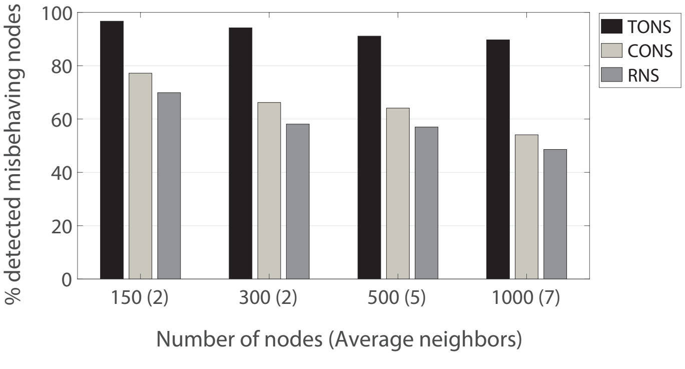


Fig. 5. Percentages of detected misbehaving nodes - Barabási–Albert (BA) network model

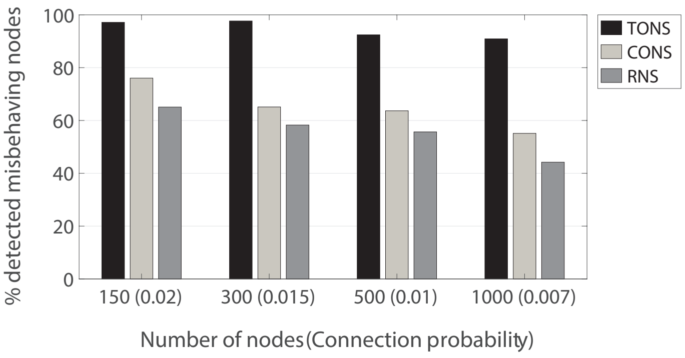


Fig. 6. Percentages of detected misbehaving nodes - Erdős–Rényi (ER) network model

[5] E. M. Migabo, K. D. Djouani, and A. M. Kurien, "The narrowband internet of things (nb-iot) resources management performance state of art, challenges, and opportunities," *IEEE Access*, vol. 8, pp. 97 658–97 675, 2020.

[6] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International journal of web and grid services*, vol. 14, no. 4, pp. 352–375, 2018.

[7] T. M. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the internet of things," *Ieee Access*, vol. 6, pp. 32 979–33 001, 2018.

[8] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, "Blockchain technologies for the internet of things: Research issues and challenges," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2188–2204, 2018.

[9] X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, and K. Zheng, "Survey on blockchain for internet of things," *Computer Communications*, vol. 136, pp. 10–29, 2019.

[10] L. Kiffer, R. Rajaraman, and A. Shelat, "A better method to analyze blockchain consistency," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 729–744.

[11] A. Kertesz and H. Baniata, "Consistency analysis of distributed ledgers in fog-enhanced blockchains," in *International European Conference on Parallel and Distributed Computing (Euro-Par 2021)*, vol. 27.

[12] H. Baniata, A. Anaqreh, and A. Kertesz, "Dons: Dynamic optimized neighbor selection for smart blockchain networks," *Future Generation Computer Systems*, vol. 130, pp. 75–90, 2022.

[13] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[14] S. Chatterjee and P. Diaconis, "Estimating and understanding exponential random graph models," *The Annals of Statistics*, vol. 41, no. 5, pp. 2428–2461, 2013.

[15] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[16] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020.

[17] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *Ieee Access*, vol. 8, pp. 16 440–16 455, 2020.

[18] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with iot. challenges and opportunities," *Future generation computer systems*, vol. 88, pp. 173–190, 2018.

[19] X. Zhou, W. Liang, I. Kevin, K. Wang, and L. T. Yang, "Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 171–178, 2020.

[20] X. Zhou, Y. Hu, J. Wu, W. Liang, J. Ma, and Q. Jin, "Distribution bias aware collaborative generative adversarial network for imbalanced deep learning in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 570–580, 2022.

[21] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, I. Kevin, and K. Wang, "Hierarchical adversarial attacks against graph-neural-network-based iot network intrusion detection system," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9310–9319, 2021.

[22] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for internet of things," *Journal of network and computer applications*, vol. 42, pp. 120–134, 2014.

[23] G. Fortino, L. Fotia, F. Messina, D. Rosaci, and G. M. L. Sarné, "Trust and reputation in the internet of things: State-of-the-art and research challenges," *IEEE Access*, vol. 8, pp. 60 117–60 125, 2020.

[24] R. Kumar and R. Sharma, "Leveraging blockchain for ensuring trust in iot: A survey," *Journal of King Saud University-Computer and Information Sciences*, 2021.

[25] Z. Huang, X. Su, Y. Zhang, C. Shi, H. Zhang, and L. Xie, "A decentralized solution for iot data trusted exchange based-on blockchain," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2017, pp. 1180–1184.

[26] G. Fortino, F. Messina, D. Rosaci, and G. M. L. Sarné, "Using blockchain in a reputation-based model for grouping agents in the internet of things," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1231–1243, 2019.

[27] B. Shala, U. Trick, A. Lehmann, B. Ghita, and S. Shiaeles, "Blockchain and trust for secure, end-user-based and decentralized iot service provision," *IEEE Access*, vol. 8, pp. 119 961–119 979, 2020.

[28] R. Di Pietro, X. Salleras, M. Signorini, and E. Waisbard, "A blockchain-based trust system for the internet of things," in *Proceedings of the 23nd ACM on symposium on access control models and technologies*, 2018, pp. 77–83.

[29] R. L. Graham and P. Hell, "On the history of the minimum spanning tree problem," *Annals of the History of Computing*, vol. 7, no. 1, pp. 43–57, 1985.

[30] P. C. Pop, "The generalized minimum spanning tree problem: An overview of formulations, solution procedures and latest advances," *European Journal of Operational Research*, vol. 283, no. 1, pp. 1–15, 2020.

[31] C. F. Bazlamaçcı and K. S. Hindi, "Minimum-weight spanning tree algorithms a survey and empirical study," *Computers & Operations Research*, vol. 28, no. 8, pp. 767–785, 2001.

[32] S. Ruzika and H. W. Hamacher, "A survey on multiple objective minimum spanning tree problems," in *Algorithmics of Large and Complex Networks*. Springer, 2009, pp. 104–116.

[33] G. Pandurangan, P. Robinson, M. Scquizzato *et al.*, "The distributed minimum spanning tree problem," *Bulletin of EATCS*, vol. 2, no. 125, 2018.

[34] F. Y.-S. Lin, C.-H. Hsiao, Y.-F. Wen, and Y.-C. Su, "Adaptive broadcast routing assignment algorithm for blockchain synchronization services," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2018, pp. 487–492.

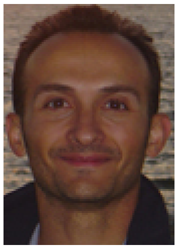
[35] W. Bi, H. Yang, and M. Zheng, "An accelerated method for message propagation in blockchain networks," *arXiv preprint arXiv:1809.00455*, 2018.

[36] G. Li, X. Ren, J. Wu, W. Ji, H. Yu, J. Cao, and R. Wang, "Blockchain-based mobile edge computing system," *Information Sciences*, vol. 561, pp. 70–80, 2021.

[37] Y. Chen, X. Li, J. Zhang, and H. Bi, "Multi-party payment channel network based on smart contract," *IEEE Transactions on Network and Service Management*, 2022.

[38] R. Antwi, J. D. Gadze, E. T. Tchao, A. Sikora, H. Nunoo-Mensah, A. S. Agbemenu, K. O.-B. Obour Agyekum, J. O. Agyemang, D. Welte, and E. Keelson, "A survey on network optimization techniques for blockchain systems," *Algorithms*, vol. 15, no. 6, p. 193, 2022.

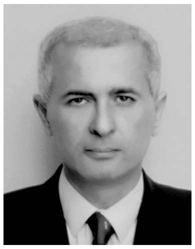
[39] R. C. Prim, "Shortest connection networks and some generalizations," *The Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.



**Giancarlo Fortino** Giancarlo Fortino (IEEE Fellow '22) is Full Professor of Computer Engineering at the Dept of Informatics, Modeling, Electronics, and Systems of the University of Calabria (Unical), Italy. He is Highly Cited Researcher 2002-2021 in Computer Science. His research interests include wearable computing systems, Internet of Things, and Cyber-security. He is author of 550+ papers in int'l journals, conferences and books. He is (founding) series editor of the IEEE Press Book Series on Human-Machine Systems and of the Springer Internet of Things series, and is AE of premier IEEE Transactions. He is cofounder and CEO of SenSysCal S.r.l., a Unical spinoff focused on innovative IoT systems. Fortino is currently member of the IEEE SMCS BoG and chair of the IEEE SMCS Italian Chapter. Contact him at giancarlo.fortino@unical.it

**Fabrizio Messina** Fabrizio Messina received his Ph.D. in Computer Science from the Department of Department of Computer Science and Mathematics at the University of Catania, Italy in 2009. He is currently serving as assistant professor with tenure track in the same department. His research interest includes Distributed systems, Simulation systems, Trust and reputation. Contact him at fabrizio.messina@unict.it.

**Domenico Rosaci** is Associated Professor of Computer Science at the Department of Information, Infrastructures and Sustainable Energy Engineering at the University Mediterranea of Reggio Calabria, Italy. In 1999, he took the PhD in Electronic Engineering. His research interests include distributed artificial intelligence, multi-agent systems, trust and reputation in social communities. He is a member of a number of conference PCs and he is Associate Editor of Journal of Universal Computer Science (Springer). Contact him at domenico.rosaci@unirc.it.



**Giuseppe M. L. Sarné** (IEEE Senior '22) is Associated Professor of Computer Science at the Department of Psychology at the University of Milan Bicocca, Italy. His main research interests include distributed artificial intelligence, multi-agent systems, trust and reputation systems. He is a member of a number of conference PCs and he is Associate Editor of E-Commerce Research and Applications (Elsevier) and member of the Editorial Board of Big Data and Cognitive Computing (MDPI). He is member of the IEEE technical committee on Hyper-intelligence. Contact him at giuseppe.sarne@unimib.it.