TSCO: Trust-based secure and cooperative opportunistic resource utilization networks

Abduljaleel Al-Hasnawi

College of Technical Al-Mussaib, Al-Furat Al-Awsat Technical University, Iraq

E-mail: abduljaleel.alhasnawi@atu.edu.iq

Abstract. An opportunistic resource utilization network (denoted as Oppnet) is a type of network that enables its node to cooperate in an opportunistic and ad hoc manner. Cooperation in oppnet includes sharing node computational resources, rather than only opportunistic communication in the traditional opportunistic networks. The key idea of oppnet is its ability to grow from seed node (initiator) to expanded oppnet via joining more nodes until getting the desired resources in a timely manner. This expansion raises the security risk of joining malicious nodes and threatens the oppnet including nodes and data. Besides, uncontrolled oppnet expansion might allow joining nodes with low computational capabilities or limited resources that have delayed responses, and hence affect the performance of oppnet. This study aims to mitigate these two problems by proposing a new oppnet paradigm (an enhanced version of original oppnet) called Trust-based Secure Cooperative Oppnet (TSCO) that utilizing a trust as a foundation for security and cooperation. Trust foundation is the major aspect of TSCO, which relies mainly on establishing trust value for each oppnet node and updating these values based on cooperation experience with those nodes. In TSCO, oppnet nodes have classified based on their trust values in which the highest trust value has the priority to join oppnet, if it exists within the specific range. Otherwise, the second-highest trust value takes priority, and so on. To test the feasibility of TSCO in realistic systems, a framework of using TSCO as an oppnet expansion control regarding security and cooperation is simulated. The experimental results show that TSCO has a better performance in terms of secure, success expansion, and cooperation. The success rate of TSCO is higher than the success rate of the original oppnet (our experimental baseline). In addition, regarding the time of satisfying oppnet tasks, TSCO shows less average latency than the baseline.

Keywords: Trust, Security, Cooperation, Opportunistic Networks

1. Introduction

Opportunistic resource utilization networks (oppnets) considered in this study are categories of ad hoc networks where diverse nodes, not originally designed as oppnet nodes, join it dynamically to perform certain tasks they have been called to participate in. Oppnets use opportunistically all kinds of resources including computation, communication, storage, sensing, actuation, etc. Trust considered in this study is an aspect of security and reliability. In other words, trust is formulated and integrated within oppnet architecture to assure that only secure nodes with a sufficient level of trust, resources, reliability, and past cooperation experience can join the oppnet. If a node does not has an experience, an initial trust value assigns for it, and then this value is updated based on the new experience with oppnet. This section provides a theoretical background about the major aspects and terms considered in this research including oppnet, trust, security, and cooperation. Section 2 reviews
the literature and outlines a selected set of works that have similarity with our work and distinguishes our work in the literature. Section 3 describes the proposed Trust-based Secure Cooperative Oppnet (TSCO) static and dynamic architectures. Section 4 presents an evaluation framework for testing the feasibility and efficiency of the proposed TSCO model. Section 5 shows the experimental results and discusses the results in terms of the satisfying the goals of the study. Finally, section 5 concludes the paper and draws the directions for future work.

2. Background

2.1. Opportunistic Resource Utilization Networks

Opportunistic resource utilization network (Oppnet) is a special kind of ad hoc networks that enables opportunistic use of all kinds of resources, services, or capabilities (incl. hardware, software, applications, etc.), and opportunistic communications which happen to be within the oppnet’s reach [1]. Oppnet starts as a relatively small network, known as seed network, which grows into an expanded oppnet. It should be noted that the more pervasive is the environment in which the oppnet operates the more benefits they can provide (the higher interoperability, with more heterogeneous software and hardware components). Oppnets can be consider collaborative distributed systems [2], since integration of resources can only be achieved through collaboration. Entities collaborate with an oppnet, by sharing diverse resources, to help it achieving a certain goal. Oppnets can also be consider ad hoc networks, peer-to-peer (P2P) systems, and self-organizing networks (SONs) since they share many of their capabilities and characteristics. In particular, they share node localization and self-organization qualities with ad hoc networks; growth-by-joining abilities with P2P systems and forming, organization, and management.

Oppnet consists of three major components: seed, Decentralized Command Center (DCC), helpers, and Oppnet Virtual Machine (OVM) [3]. Seed is a single node or multiple neighboured nodes that initiated the oppnet deployment (expansion) process due to a need for resources. DCC is a controller, which can include an arbitrary subset of seed, works on inviting foreign nodes to help in reaching oppnet goals. The initial seed oppnet grows into an expanded oppnet. Helpers are foreign or oppnet entities that accept the DCC invitation. The OVM is a set of primitives that create a universal platform for application-level communication within and among oppnets. This communication is the basis for opportunistic utilization of all resources. OVM primitives can be downloaded to any system or device to make it oppnet-enabled, that is, able to communicate and share resources with other oppnets or oppnet-enabled nodes. Oppnet operation, illustrated in Figure 1, initiated by the seed oppnet when it declares the need for help by other surrounding nodes. The DCC notified by this declaration, via command sent by the seed, and starts the growth process. The growth consists of three steps: discovering enough candidate helpers, admitting them into oppnet, and integrating their resources into oppnet. This process prepare helpers to cooperate. Once the cooperation has done and the goal of an oppnet has achieved, DCC releases helpers’ resources and finalizes the oppnet process. Note that the oppnet DCC presides over the operations of the oppnet throughout its lifecycle.

2.2. Trust

Trust is an abstract concept, which combines many complicated factors. It can be reflected by utility, availability, reliability, confidence, and quality of services, reputation, and other concepts [4]. Trust can be defined as a particular level of subjective probability which an agent will perform a particular action for a specified period within a specified context. In opportunistic networks, trust of a specific node is a subjective estimation done by another peer node on the reliability and accuracy of information received from or traversing through that node [5]. The Level of trust is a probability varying from 0 (complete distrust) to 1 (complete trust). In this sense, trustworthiness is a measure of the actual probability that the trustees will behave as expected. Trustworthiness defined as the objective probability that the trustee performs a particular action on which the interests of the trustor depend. If the trustor ignores the difference between the believed (i.e., trust) and the actual (i.e., trustworthiness) probability, there is a risk of trust miscalculation. There is misplaced trust when the perceived trust is higher than the actual trustworthiness. Misplaced trust increases risk and thus
enhances the chance of deceit as well. On the other hand, when the perceived trust is lower than the actual trustworthiness, the trustee is distrusted more than warranted. Hence, the trustor may lose potentially good opportunities to cooperate with partners with high trustworthiness [6].

2.2.1. Trust Properties. Due to the unique characteristics of oppnets environments and the inherent unreliability of the wireless channels, the concept of trust in oppnets should be carefully defined. The main properties of trust in such environments can be defined in many directions. Trust is dynamic: In mobile environments, information is typically incomplete and can change rapidly due to node mobility, so building trust in such environments is based on temporally and spatially local information [7]. In order to capture the dynamicity of trust, it should be expressed as a continuous variable, rather than as a binary or even discrete-valued entity. Trust is subjective: Different level of trust can be determined by a trustor node against the same trustee node due to different experiences derived from a dynamically changing network topology. Trust is not necessarily transitive: If A trusts B, and B trusts C, it does not guarantee A trusts C. Transitivity of trust between two entities to a third party requires two types of trust to be done by a trustor: trust in a trustee and trust in the trustee’s recommendation of the third party [8]. Trust is asymmetric: Node A with higher capability (e.g., more energy or computational power) may not trust node B with lower capability at the same level that node B trusts node A.

2.2.2. Establishment of Trust. Two common ways to establish trust are through using policies or reputation. Hard evidence trust used in policies, and the estimation of trust used in reputation systems [9]. Policies determine specifically the conditions necessary to obtain trust, and can predetermine the actions and outcomes if certain conditions are met. Using policies to establish trust is based on managing and exchanging credentials and enforcing access policies. Reputation is an assessment based on the history of observations of or interactions with an entity. This assessment can be done either directly with the evaluator (subjective experience) or as reported by others (recommendations or third party verification). Using reputation to establish trust, focusing on the past performance or interactions for an entity which are combined to assess its future behavior. English et al. [10] defines four steps for establishing trust: trust formation, trust evolution, and trust exploitation. Trust formation is the process of establishing the initial trustworthiness value of each cooperator, then using the available trust evidences of the principal to set the initial trust value. If no evidence available for a principal regarding the specific action, it is possible use available trust evidence from other actions with the principal. Trust evolution is the evolution of trust values over time. New evidence from experience can modifies the stored trust value directly, producing a new trust value. Trust update is the process of adding experience evidence to the store of trust after each evaluation. The store contains the set of all trust evidences, and it is possible to re-evaluate trust values based on the most recent evidence. Trust exploitation is the process of determining the expected behavior of cooperator based on trust values of that cooperator.

2.3. Security
One of the major challenges of an oppnet is to timely detect and identify the malicious devices, and to prevent them from joining the network and ensure the privacy and security of the oppnet. Based on the standard ISO 7498-24 [11], there are eight security services with acronym CIA-AANN-A that define security are confidentiality, integrity, availability, authentication, access control, non-repudiation, notarization, auditability. Confidentiality is the property that information is not made available or disclosed to unauthorized individuals, entities, or processes. Integrity is the property that data has not been altered or destroyed in an unauthorized manner. Availability is the property of being accessible and useable upon demand by an authorized entity. Authentication is the corroboration that an entity is the one claimed, and the source of data received is as claimed. Access control is the prevention of unauthorized use of a resource. It includes the prevention of use of a resource (by an authorized entity) in an unauthorized manner. Non-repudiation is the prevention of entities’ denial to be involved in all
or part of a communication. Notarization is the registration of data with a trusted third party that allows (to assure) the accuracy of its characteristics such as content, origin, time, and delivery. Auditability is the ability of a system to trace all actions related to a given asset.

2.4. Cooperation

Cooperation can be defined as a joint collaboration interaction between two or more principals (Ps), to perform one or more actions (As) on one of the principals’ resources. There are five services for cooperation process, outlined in Figure 2, are cooperation request analyzer, cooperative risk assessment, cooperation decision, cooperation monitoring, and cooperation evaluation [10]. Functionally, cooperation services process the cooperation request denoted by \( \text{Req}(P, A) \), where \( P \) determines which node has to participate, \( A \) determines which action has to be done by \( P \) in cooperative processing. Cooperation services work in a sequence manner to determine whether the action \( A \) can take place by using the principal’s \( P \) resources. In cooperation request analyzer service, the contents of the received cooperation request \( \text{Req}(P, A) \) is analyzed to determine whether \( P \) has the necessary resources to allow the action \( A \) to take place. The request analyzer provides cooperative risk assessment with the results of request analysis. Cooperative risk assessment determines whether the risks are acceptable to enable cooperation to proceed. It estimates the risk of interacting with a particular principal \( P \) for a particular action \( A \) based on the outcomes from cooperation request analyzer and trust exploitation. In cooperation decision service the decision about enabling cooperation process takes place. The cooperation decision made based on the outcomes from cooperative risk assessment. In cooperation monitoring service, the progress of the individual cooperation actions is monitored to ensure that the interaction is progressing towards the desired outcome, and a negative outcome predicted during the cooperative risk assessment for that interaction. In cooperation evaluation service, trust foundation is provided with the evaluation of the interactions. The evaluation based on the range of outcomes established during cooperative risk assessment.

**Figure 1.** Oppnet operation flow diagram.

**Figure 2.** Cooperation process flow diagram.
3. Literature Review
Message forwarding opportunistic networks are wider investigated in the literature than opportunistic resource utilization networks. Although, they share that same opportunistic principle unless the latter is more general and includes opportunistic communication. This section outlines researchers’ work in the literature regarding trust, security, and cooperation in opportunistic networks or opportunistic resource utilization networks and distinguishes our work from them.

In [3], the author proposed a virtual machine for opportunistic resource utilization networks as a universal standard for the application-level of oppnet resource sharing. Oppnet virtual machine (OVM) is defined as a set of primitives that are designed to create a middleware for application-level resource acquisition by oppnet-enabled devices and systems. Any computational entity can be oppnet-enabled by downloading and installing the OVM primitives that enables those entities acquire and communicate computational resources in an opportunistic manner. The author improved the original OVM primitives by developing non-monolithic OVM-based oppnet middleware to support a very broad and diverse set of applications. This work improved oppnet in terms of primitives and applications. While, our work improves oppnet in terms of security and cooperation. Wu et al. [12] provided a systematic description of security threats in opportunistic networks. They also proposed a general security architecture of opportunistic networks and analyzed many security aspects including access control and authentication, privacy protection, secure routing, cooperation mechanisms, and trust management. In addition, they surveyed various trust and security solutions for opportunistic networks. This work investigated security and trust management in opportunistic communication rather than opportunistic resource utilization, as we do. Yao et al. [13] proposed the incorporating social trust concept for the routing decision opportunistic networks. They presented a Trust Routing based on Social Similarity (TRSS) scheme based on the common interests or social similarities of nodes. Social features are extracted from the social history record of a given node in opportunistic network. They established social trust based on observed nodes’ trustworthiness and based on direct or indirect recommendations. This work utilized social similarities to establish trust. While, we utilize actual cooperation experience to establish and maintain trust values for nodes. Kumar et al. [14] proposed altruism-dependent trust-based data forwarding mechanism, denoted ATDTN. In ATDTN, altruism value associated with each node representing its trust in the opportunistic network and determining its participation in forwarding messages or sharing resources. This study relied on altruism, which is established by social interactions, as a major factor of trust. While, we classify oppnet nodes based on their calculated trust values and manage their task assignment in opportunistic recourse utilization based on the calculated trust. Alajeely et al. [15] provided an overview of the major security issues in opportunistic networks and surveyed various security protection approaches to mitigating these issues. They described different types of attacks that can specifically affect routing in opportunistic networks, such as Blackhole attacks Wormhole attacks, Sybil attacks, Selective Dropping attacks, and Selfish attacks. Then, they outlined the defense against each type of attacks in terms of trust as a factor used in the defense process. Trust calculated mutually by interactive nodes through direct and indirect reputation and recommendation. This study utilized trust for security only. While, we utilize trust for security and cooperation to assure secure, effective, and efficient oppnet.

4. The Proposed TSCO Model
This section presents the proposed Trust based Secure and Collaborative Oppnet (TSCO) by outlining its static and dynamic architectures as well as the way of establishing trust based on the helpers classification described in Ref. [16].
4.1. Static TSCO Architecture
Figure 3 illustrates the static TSCO architecture, which includes four major components: oppnet, cooperation blocks, security services, and trust foundation. TSCO architecture components built in the matter that follows the basic oppnet operations to keep the functionality of oppnet. Security block consists of eight sub blocks; each represents a single security services (as explained in Subsection 2.3). Cooperation block consists of five sub blocks; each represents a single cooperation service in the cooperation process (as explained in Subsection 2.4). Trust Foundation acts as a basis for both cooperation services and security services. It sends tv for P to cooperative risk assessment to help determining risk level of P and to ensure that P satisfies security services. It also update trust value of an admitted helper based on information from cooperative processing unit that includes cooperation monitoring and cooperation evaluation.

4.2. Dynamic TSCO Architecture
Figure 4 shows the basic TSCO operations. DCC can grow the oppnet into the expanded oppnet by admitting other nodes. After determining enough candidates by DCC, cooperation request analyzer can analyze each cooperation request Req(P, A) to determine whether the helper P has the required resources to participate doing the action A. Then it feeds this information to the cooperative risk assessment. In this stage, cooperative risk assessment estimates every helper’s risk with respect to its trust value; it also helps DCC to make decision about the admission of the candidate helpers. Once candidate helper admitted, this means it passed cooperative risk assessment, and it is expected to be cooperative. Then DCC integrates helper’s resources in oppnet to start cooperative process. In this stage, the admitted helper will be under cooperation monitoring and evaluation stages to provide the information needed for trust calculation. Based on the cooperation monitoring and evaluation information, trust foundation starts calculating trust value for the admitted helper, and supplies cooperative risk assessment with this information. All the security, cooperation, and trust information about a specific helper stored in the DCC record for future references. In Figure 4, DCC controls every single block. However, control interactions by CC are not shown to avoid “crowding the picture.” The solid arrows indicate control flow between architecture blocks, and broken lines indicate data flow.
4.3. Establishment of Trust in TSCO

In TSCO system, DCC maintains trust value (Tv) for each helper P based on the set of all evidence values (EV) that related to the helper P including first-hand experience (EX) values and second-hand reputation (RP) values. Each of these sets of evidence has two subsets, positive and negative. For instance, EX represents the set of possible experience values for Helper P. Experience values consists of two subsets, EX\textsubscript{pos} (positive experience) and EX\textsubscript{neg} (negative experience) for the interactions. If there is no evidence related to the admitted helper P, DCC has to establish an Initial Trust Value (ITv) for P. Establishing an ITv in TSCO system depends on the helper type, based on the classification of helpers [16]. Table 1 illustrates estimate interval values for each helper type that can be establish as ITv. The evolution of a stored Tv value over time is important in TSCO system to ensure accuracy of Tv. For this reason, Trust Evolution takes a new piece of evidence and modifies the stored trust value Tv directly, producing a new trust value Tv'. After cooperative processing, new evidence outcome is an experience evidence that should be added to the stored trust information. This information contains all trust evidence, EV. When new ev'\in EV arrives as the latest experience of the helper, Trust Update
modifies EV to produce an updated set of trust evidence, EV’ [10].

Table 1. Initial and evolution of helpers’ trust values.

| Helper Type   | Initial tv Range | tv Evolution | tv Change | Resulting tv Range |
|---------------|------------------|--------------|-----------|--------------------|
| Private Unknown | 0 - 50%          | Up / Down    | Large     | 0 - 70%            |
| Public Unknown | 50 - 70%         | Up / Down    | Medium    | 30 - 90%           |
| Trusted Known Oppnet Reservists | 70 - 90%      | Relatively Constant | Small | 60 - 100%           |
|               | 90-100%          | Constant     | None      | 90 - 100%          |

5. Evaluation Framework
This section presents an evaluation framework for testing the feasibility and efficiency of the proposed TSCO model. The evaluation framework including experimental setup, simulation parameters, and evaluation metrics.

5.1. Experimental Setup
Simulation experiments are conducted for original oppnet model, which considered as a baseline in which a performance of TSCO compared with, and TSCO model. For the purpose of accurate comparison, experiments are designed to be implemented for both models (baseline and TSCO) simultaneously and within the same network configuration. The simulation is conducted using SimPy, a process-based discrete event simulation package based on standard Python. The simulation is implemented in the PyCharm IDE (Integrated Development Environment). The simulation runs on an Intel Core i7-4710 HQ 250 GHz processor with 8 GB RAM. To test TSCO performance on different network topology sizes, we have varied the size of expanded oppnet by considering five levels of scalable oppnet, namely, \( L_1 \), \( L_2 \), \( L_3 \), \( L_4 \), and \( L_5 \). These levels starting from small scale to a larger scale and so on. We consider these levels to simulate the realistic oppnet scenarios. Note that, within each level, many oppnet tasks can be done locally at the edge level. While, other oppnet tasks might require more communications to be done at fog or cloud levels [17].

5.2. Simulation Parameters
Three major types of nodes are defined and configured in the simulation are SEED, DCC, and HLPR. These nodes are the major actors in the simulation where the SEED (defined in Subsection 2.1) initiates oppnet growth process, the DCC (defined in Subsection 2.1) manages the growth and the cooperation processes, HLPR (helper) participates in the oppnet cooperation process. As a network node, we define two simulation parameters, CPU speed and the number of processing cores. The CPU speed is the maximum number of instructions in which a node’s CPU can process per second. CPU speed defined in MIPS (Million Instructions Per Second) [18]. Note that DCC has higher computation power that both SEED and HLPR nodes due to the management tasks required by it. In addition, typically, SEED nodes have the lowest computation power in oppnet environment since they always need resources from others so they initiate oppnet to get help. Helper resources are defined in the simulation using two parameters: capacity, which define the number of computational resource a helper has; and queuing type, which defines how helper node handle multiple help request based on network queuing theory [19]. Three major types of packets are defined in the simulation are Help Request (HR), Normal Help (NH), and TSCO Help (TH). These packet types are configured using two parameters: packet size, which simulated as a random variable with Poisson distribution [20]; and packet processing efforts, which simulated as a random variable with Binomial distribution [21]. The type of carried packet into three types defines the network links: Help Request Link (HR-Link),
Normal Help Link (NH-Link), and TSCO Help Link (TH-Link). These link types are configured using two parameters: bandwidth, which is the maximum number of bits per second that can be transmitted via the link, defined by Mbps (Megabits per second) [22]; and propagation delay, which is a period from the instant when the last bit of a packet is placed onto a transmission link by the sending node till the instant when the last bit of the packet is received by the receiving node [23]. Trust value for each type of oppnet helpers (outlined in Table 1) is defined in the simulation for four classes of oppnet helpers: Private Unknown Helper (PrUH), Public Unknown Helper (PuUH), Trusted Known Helper (TKH), and Oppnet Reservists Helper (ORH). Each helper class is configured in the simulation using two parameters: Trust Value Range, which determines the minimum and maximum trust value this helper class can have; and Trust Value Probability, which simulated as a random variable with Uniform distribution [24]. Table 2 lists the simulation the abovementioned parameters.

Table 2. Simulation Parameters.

| Parameter     | Instance      | Type            | Value | Type            | Value  
|---------------|---------------|-----------------|-------|-----------------|--------
| Node          | SEED          | CPU Speed (MIPS)| 4000  | Number of Cores | 2      
|               | DCC           |                 | 10000 |                 | 4      
|               | HLPR          |                 | 6000  |                 | 2      
| Resource      | HLPR_REC      | Capacity        | 2     | Queuing Type    | FIFO   
| Resource      | HR            |                 |       |                 |        
| Packet        | NH-Link       | Bandwidth (Mbps)| 50    | Propagation Delay| 2      
| Packet        | TH-Link       |                 | 100   |                 | 4      
| Link          |                |                 | 120   |                 |        
| Trust         | PrUH          | Trust Value     | 0-49  |                 | l = 0, h = 0.49 
| Trust         | PuUH          | Probability     | 50-69 |                 | l = 0.5, h = 0.69 
| Trust         | TKH           |                   | 70-89 |                 | l = 0.7, h = 0.89 
| Trust         | ORH           |                   | 90-100|                 | l = 0.9, h = 1    

5.3. Evaluation Metrics
Two metrics are considered in the simulation experiments are latency and success rate. The latency (LAT) for an oppnet task is defined as the period between the moment of submission of help request by seed node and the moment when competing the oppnet task. More specifically, LAT is the period from the instant when the first bit of HR is sent by a given SEED for a task till the instant when either the last bit of NH packet is delivered to SEED or the last bit of TH packet for this task is delivered to the SEED oppnet via DCC-HLPRs hope of communications. The success rate is defined as a number of successfully performed oppnet tasks to the number of total tasks in the experiment. We conducted two ways to accurately calculate success rate for oppnet task. First, we define a threshold in term of response time by helper to help request. The motivation for that is the mobility nature of oppnet helper, if a helper A within the seed oppnet reach at time $t_1$, and it receives the help request, there is no guarantee that at time $t_{1+\tau}$, the helper A is still in the reach of oppnet or it successfully forwarded the help request to other candidate helper. Second, we use the hypergeometric distribution [25], which is a discrete probability distribution that describes the probability of successes within a scale of events. Where uniform distribution used to calculate the probable range of success helps for the baseline and TSCO models and similarly the probable range of fail helps. Then, the binomial distribution used to calculate the probability of good samples (successes) and bad samples (fails) from the total sample size. Finally, hypergeometric distribution utilized to find the success rate of an experiment.

6. Results and Discussion
This section shows and discusses the experimental simulation results of using the proposed TSCO model as compared with the baseline (original oppnet model outlined and illustrated in Section 2.1 and Figure 1, respectively). According to our evaluation metrics (Subsection 5.3), experimental results have shown and discussed for each metric separately in a separate subsection.

### 6.1. Success Rate

Figure 5 shows the simulation experimental results regarding the absolute success rate. The simulation experiments are conducted for five levels of expanded oppnet are L₁, L₂, L₃, L₄, and L₅. For each level, the chart shows the total number of helps happened at that level of network configuration as well as the number of normal (baseline) success helps, which is used as the base in which the number of TSCO success helps compare with. The results show that with increasing the scale of oppnet from L₁ to L₅, the number of helps increase as well as the number of successes for both baseline and TSCO models. This is a normal phenomenon since based on the nature of oppnet structure and operation, the victims increases the probability of helps happened and hence increases the number of successes, accordingly. However, TSCO shows a higher number of success rate than the baseline for all levels. In the baseline oppnet, only 47% of total helps happened in the system are succeeded. While, in the TSCO system, 75% of total helps happened in the system are succeeded. Hence, TSCO shows a better performance than the baseline with 28% regarding the absolute success rate.

Figure 6 shows the simulation experimental results regarding the relative success rate. For each level, the chart shows the relative success rate at that level both the baseline and TSCO models. The experimental results show that with increasing the scale of oppnet from L₁ to L₅, the success rate is relatively stays stable with slightly increasing for both baseline and TSCO models. This is also a normal phenomenon since the success rate calculated relatively with the scale of victims. However, TSCO shows a higher success rate than the baseline for all levels. The average success rate for baseline is 44%. While, The average success rate for TSCO is 76%. Hence, TSCO over performances the baseline with 32% regarding the relative success rate.

![Figure 5. Absolute success rate.](image1)

![Figure 6. Relative success rate.](image2)

### 6.2. Latency

Figure 7 shows the simulation experimental results regarding the absolute latency. The simulation experiments are also conducted for five levels of expanded oppnet are L₁, L₂, L₃, L₄, and L₅. For each level, the chart shows the absolute latency at that level for both the baseline and TSCO models. The results show that with increasing the scale of oppnet from L₁ to L₅, the absolute latency increases slightly for both baseline and TSCO models. This is a normal phenomenon since the latency in this experiment is calculated as an accumulative quantity. Hence, increasing the scale of oppnet victims
increases the absolute latency, accordingly. However, TSCO shows a lower latency than the baseline for all levels. In average, TSCO has 35% less absolute latency than the baseline.

Figure 8 shows the simulation experimental results regarding the relative latency. For each level, the chart shows the relative latency at that level for both the baseline and TSCO models. The experimental results show that with increasing the scale of oppnet from L1 to L5, the relative latency is sharply decreasing for both baseline and TSCO models. This is a normal phenomenon since the latency is calculated relatively with the scale of victims. The absolute latency divided by the number of helps happened at each level and with increasing the scale by levels, the relative latency decreases accordingly. TSCO shows a lower relative latency than the baseline for all levels. In average, TSCO has 26% less relative latency than the baseline.

7. Conclusions
Opportunistic resource utilization networks can be significantly optimized by increasing their performance in terms of security and cooperation. Adding security control to the oppnet architecture assures the protection from selfish or malicious helpers. Furthermore, adding cooperation control improves the reliability of oppnet performance. In our perspective, the common key to improve both security and cooperation in oppnet is Trust.

In this paper, we proposed a new oppnet paradigm denoted TSCO, which integrate security and cooperation controls based on trust with the original oppnet architecture. A theoretical background on the major research terminologies, including oppnet, trust, security, and cooperation, is provided. Then, we distinguished our work in the literature. A conceptual design is presented for the proposed TSCO model including its static and dynamic architectures.

To prove the feasibility and efficiency of the proposed TSCO model, we have introduced a framework of performance evaluation including simulation experimental setup, simulation parameters, and evaluation metrics. Experiments are conducted for both TSCO and the original oppnet, simultaneously. The experimental results show that TSCO over performed the original oppnet (as a baseline) regarding absolute and relative success rate with 28% and 32%, respectively. TSCO also have shown a better performance than the original oppnet regarding absolute and relative latency with 35% and 26%, respectively.

References
[1] M. A. Alduailij and L. T. Lilien, (2015). A collaborative healthcare application based on opportunistic resource utilization networks with OVM primitives. International Conference on Collaboration Technologies and Systems (CTS), Atlanta, GA, 426-433.
[2] Yasmin, S., Qayyum, A., & Rais, R. N. B. (2017). Cooperation in opportunistic networks: an overlay approach for destination-dependent utility-based schemes. Arabian Journal for Science and Engineering, 42(2), 467-482.

[3] Alduailij, M. A. (2015). Design of Oppnet Virtual Machine for opportunistic resource utilization networks: A universal standard for application-level resource sharing. A Dissertation at Western Michigan University.

[4] Ciobanu, R. I., Marin, R. C., Dobre, C., & Cristea, V. (2017). Trust and reputation management for opportunistic dissemination. Pervasive and Mobile Computing, 36, 44-56.

[5] Qin, X., Wang, X., Lin, Y., Wang, L., & Zhang, L. (2017, October). An efficient routing algorithm based on interest similarity and trust relationship between users in opportunistic networks. In China Conference on Wireless Sensor Networks (pp. 273-284). Springer, Singapore.

[6] Dhahanjayan, G., & Subbiah, J. (2016). T2AR: trust-aware ad-hoc routing protocol for MANET. SpringerPlus, 5(1), 995.

[7] Muhammad, S., Wang, L., & Yamin, B. (2017, October). Trust model based uncertainty analysis between multi-path routes in MANET using subjective logic. In China Conference on Wireless Sensor Networks (pp. 319-332). Springer, Singapore.

[8] Tong, X., Zhang, W., & Wang, Y. (2016, August). Research on the transitivity closure of trust network. In 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD) (pp. 2070-2074). IEEE.

[9] Luo, T., Kanhere, S. S., Huang, J., Das, S. K., & Wu, F. (2017). Sustainable incentives for mobile crowdsensing: Auctions, lotteries, and trust and reputation systems. IEEE Communications Magazine, 55(3), 68-74.

[10] English, C., Wagealla, W., Nixon, P., Terzis, S., Lowe, H., & McGettrick, A. (2003, May). Trusting collaboration in global computing systems. In International Conference on Trust Management (pp. 136-149). Springer, Berlin, Heidelberg.

[11] ISO, ISO/IEC 10181-2:1996, “Information technology - Open Systems Interconnection - Security frameworks for open systems: Authentication framework - Part 2,” Last Accessed Feb. 2015. Available at: https://www.iso.org/obp/ui/#iso:std:iso-iec:10181:-2:ed-1:v1:en

[12] Wu, Y., Zhao, Y., Riguidel, M., Wang, G., & Yi, P. (2015). Security and trust management in opportunistic networks: a survey. Security and Communication Networks, 8(9), 1812-1827.

[13] Yao, L., Man, Y., Huang, Z., Deng, J., & Wang, X. (2015). Secure routing based on social similarity in opportunistic networks. IEEE Transactions on Wireless Communications, 15(1), 594-605.

[14] Kumar, A., Dhurandher, S. K., Woungang, I., Obaidat, M. S., Gupta, S., & Rodrigues, J. J. (2017). An altruism-based trust-dependent message forwarding protocol for opportunistic networks. International Journal of Communication Systems, 30(10), e3232.

[15] Alajeely, M., Doss, R., & Ahmad, A. A. (2016). Security and trust in opportunistic networks–a survey. IETE Technical Review, 33(3), 256-268.

[16] Tamez, E. B., Woungang, I., Lilien, L., & Denko, M. K. (2009, August). Trust management in opportunistic networks: A semantic web approach. In 2009 World Congress on Privacy, Security, Trust and the Management of e-Business (pp. 235-238). IEEE.

[17] Al-Hasnawi, A., Carr, S. M., & Gupta, A. (2019). Fog-based local and remote policy enforcement for preserving data privacy in the Internet of Things. Internet of Things, 7, 100069.

[18] “Instructions per second,” Wikipedia. Accessed on Oct. 10, 2020, from: https://en.wikipedia.org/wiki/Instructions_per_second.

[19] Memon, R. A., Li, J. P., & Ahmed, J. (2019). Simulation model for blockchain systems using queuing theory. Electronics, 8(2), 234.

[20] “Poisson Distribution.” 2008–2018. Accessed on Oct. 15, 2020, from: https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.poisson.html#numpy.random.poisson.
[21] “Binomial Distribution.” 2008–2018. Accessed on Oct. 15, 2020, from: https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.binomial.html#numpy.random.binomial.

[22] Cohen, M. A., Dennett, D. C., & Kanwisher, N. (2016). What is the bandwidth of perceptual experience?. *Trends in cognitive sciences*, 20(5), 324-335.

[23] Appasani, B., & Mohanta, D. K. (2018). Co-optimal placement of PMUs and their communication infrastructure for minimization of propagation delay in the WAMS. *IEEE Transactions on Industrial Informatics*, 14(5), 2120-2132.

[24] “Discrete Uniform Distribution.” 2008–2018. Accessed on Oct. 15, 2020, from: https://docs.scipy.org/doc/scipy/reference/tutorial/stats/discrete_randint.html.

[25] “hypergeometric Distribution.” 2008–2018. Accessed on Oct. 20, 2020, from: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.hypergeom.html.