IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 7, 2019, accepted July 1, 2019, date of publication July 9, 2019, date of current version July 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927729

COSTA: Composite Trust-Based Asset-Task Assignment in Mobile Ad Hoc Networks

JIN-HEE CHO[1], (Senior Member, IEEE), HAMID AL-HAMADI[2], (Member, IEEE), AND ING-RAY CHEN[1], (Member, IEEE)
[1]Department of Computer Science, Virginia Tech, Falls Church, VA 22043, USA
[2]Department of Computer Science, Kuwait University, Kuwait City 13060, Kuwait

Corresponding author: Jin-Hee Cho (jicho@vt.edu)

This work was supported in part by the U.S. Air Force Office of Scientific Research (AFOSR) under Grant FA2386-17-1-4076.

ABSTRACT In mobile ad hoc networks (MANETs), asset-task assignment problems have been explored with vastly different approaches. Considering the unique characteristics of MANET environments, such as no centralized trusted entity, a lack of resources, and high-security vulnerabilities, resource allocation is not a trivial problem particularly for situations where a mobile team aims to successfully complete a common mission. The existing approaches have studied asset-task assignment problems by best matching a node's functionality and requirements of a given task. In this paper, we propose a task assignment protocol using the concept of multidimensional trust, namely, CompoSite Trust-based Assignment (COSTA), aiming to maximize the completion ratio of a common mission consisting of multiple tasks by balancing trust and risk in executing them. Based on the core concept of trust defined as the willingness to take the risk in performing a given task, COSTA selects qualified nodes for a given task while meeting an acceptable risk level for executing multiple tasks contributing to successful mission completion. Given a mission consisting of dynamic multiple tasks, we model each task with importance, urgency, and difficulty characteristics and use them for selecting qualified members. In addition, we model a node's risk behavior (i.e., risk-seeking, risk-neutral, and risk-averse) and investigate its impact on mission performance where a payoff is given for member selection and task execution. We formulate an optimization problem for the task assignment using integer linear programming (ILP). Our simulation results validated with ILP solutions demonstrate the existence of an optimal acceptable risk level that best balances trust and risk so as to maximize the mission completion ratio. We conduct a comprehensive comparative analysis and show that COSTA achieves a higher mission completion ratio while incurring a lower communication overhead compared with non-trust-based counterparts.

INDEX TERMS Trust, risk, risk behavior, task assignment, mobile ad hoc networks.

I. INTRODUCTION

In tactical or service-oriented mobile ad-hoc networks (MANETs), a common mission is often assigned where it has multiple tasks. Efficiency and effectiveness of the asset-task assignment in such tactical contexts is considered as the key to successfully complete the given mission. In this work, a mission team is considered where the team is composed of different entities responsible for completing respective tasks to pursue a common mission in the tactical MANET environment. For example, such missions are given in situations of disaster management, personnel rescue, facility construction, surveillance / monitoring, target destruction, and so forth. Entities in a network are treated as "assets" to execute tasks contributing to completing a common mission. The assignment process of assets to tasks significantly impacts successful mission completion.

In this work, we measure the trust of entities in order to solve an 'asset-task assignment problem' as the so called 'soft security technique' to deal with malicious entities. The proposed trust-based mechanism has the goal of selecting qualified entities for each task characterized by trust-based requirements to ultimately lead to a successful mission completion. Accurate trust estimation of entities in a network is critical to making effective decision making, such as composing a task team with qualified, trustworthy members.

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Yan.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

The concept of trust is first discussed in social sciences and is often defined as a subjective opinion or belief regarding how an entity behaves based on certain criteria [1]. The asset-task assignment problem can be seen as a decision making process of a trustor node based on its peer-to-peer trust estimation about other trustee nodes.

Although trust has been vastly differently defined depending on an application domain [2], [3], the common key concept of trust has been identified as the "willingness to take a risk." We interpret trust as a decision making process under uncertain situations in which each entity does not have perfect knowledge about all other entities in a fully distributed environment.

In this work, a composite trust based asset-task assignment protocol is proposed, namely COSTA (CompoSite Trust-based Assignment), aiming to maximize the ratio of mission completion as well as only allowing an acceptable risk level by assigning qualified, trustworthy entities to a task. A team with sufficient members having high trust levels can meet the maximum acceptable risk level and can lead to successful task completion. We consider a node with vastly different capabilities so that it can participate in multiple tasks that arrive dynamically during its lifetime for maximizing its utilization. A node's active participation of task execution will lead to high incentives that enables the node to maintain high trust and continuously give more chances for active participation in mission activities.

We model dynamic tasks of a common mission and assume that entities including task leaders and members make decisions to achieve their own goals based on their risk behaviors (i.e., 'risk-averse, risk-neutral, or risk-seeking'). In addition, we analyze the effect of an acceptable level of risk to maximize the ratio of mission completion in the presence of uncooperative and malicious entities (see Section III-D for the definition). An example system would be a community of interest (CoI) system composed of heterogeneous entities (i.e., members) each contributing their resources to achieve the mission objective described as a set of tasks. The CoI would advertise mission tasks and gather sufficient members with adequate resources in order to carry out the tasks. Such a system would have numerous applications, including community search, spontaneous rescues missions, relief operations, and/or location-based data gathering. Another example would be for joined effort missions by different organizational entities, each contributing members (with their resources) to achieve a mission such as military operations consisting of joint forces and several humanitarian agencies. In both examples, entities could possibly be unknown to each other on an individual basis, be assembled on demand and in short notice to the necessity of the mission, and the assembled entities are heterogeneous in regards to capabilities, resources, and risk behaviors. Mission effectiveness is mainly influenced by: (a) a number of members executing a given number of tasks; and (b) performance of the selected members in completing the assigned task. A task may fail if it has a high standard in which case it may not find sufficient members to meet the high standard for task execution. On the other hand, a task may fail if it has a low standard in which case it may find sufficient members but the selected members cannot execute the task successfully due to high risk exposed by their untrustworthy behavior.

This work has the following **key contributions**:

1) We propose a novel 'task assignment protocol' that balances between trust and risk. That is, by controlling a degree of an acceptable risk level, we increase the assignment of more tasks to maximize mission completion.
2) We investigate and analyze the impact of intrinsic characteristics of tasks including importance, urgency, and difficulty, as well as "risk behavior" of nodes including 'risk-seeking, risk-neutral, and risk-averse behaviors' to the overall mission risk and the mission completion probability.
3) We adopt a context-dependent trust-based approach to guide entity allocation to task assignment. In this work, we consider 'task-dependent trust' where task requirements are key in entity evaluation.
4) We formulate the task assignment optimization problem with trust and risk management as an Integer Linear Programming (ILP) problem [4]. The mathematical formulation provides a theoretical basis and optimal solutions against which the performance of our task assignment protocol, with risk and trust management based on auction/bidding, may be evaluated for validity.

We structure this paper as follows. Section II discusses related work. Section III describes the system model in terms of network model, trust bootstrapping model, node behavior model, threat model, task model, and risk behavior model. Section IV explains our composite trust metric to evaluate the trustworthiness of mobile nodes based on multidimensional trust derived from communication and social networks. Section V provides the details of our proposed task assignment protocol, COSTA, with risk and trust management. Section VI gives the details of formulating the task assignment optimization problem as an ILP problem to yield optimal solutions against which the performance of our risk and trust management protocol is evaluated. Section VII presents comparative performance analysis based on the results obtained from our simulation experiments. Section VIII summarizes the key findings from this work along with future research directions.

II. RELATED WORK

A. TRUST MANAGEMENT IN MANETS

The term *trust management* is first coined by Blaze *et al.* [5] and identified as a distinct part of security services in networks. Trust management research has been explored with considerable attention because of its high importance and applicability in the process of decision making applications. The key characteristics of estimating trust in MANET environments have been discussed by considering:

(1) potential risks; (2) context-dependency; (3) interest of each party involved in a decision making; (4) cognitive learning process; and (5) system reliability. We consider the above characteristics in developing a trust-based asset-task assignment protocol with the special emphasis on the balance between trust and risk to maximize mission completion ratio.

The vital need of trust management in MANETs has been emphasized in terms of establishing a network consisting of nodes with an acceptable level of trust where the participating nodes do not have any prior knowledge to each other. Specifically trust management is critical to collecting and distributing evidence to estimate trustworthiness of nodes for successful task completion [6]. Many researchers have adopted the concept of trust in order to maintain or assess trust relationships among nodes in MANETs [7]–[12].

To estimate nodes' trust, two trust management methods have been popularly used: *evidence-based* and *monitoring-based* [13]. In *evidence-based trust management*, any credentials proving the trust relationships among nodes are used such as public key, address, identity, or any evidence that can be generated through a challenge and response process between two entities. On the other hand, *monitoring-based trust management* collects direct and/or indirect evidence based on observations (e.g., behaviors such as packet dropping and flooding) or recommendations from third parties (e.g., reputation). Our work uses the monitoring-based method based on observations and recommendations that are aggregated to derive a trust level of other nodes.

The relationship between trust and risk has been discussed in [14], [15]. When there exists high trust, risk is likely to be low. If a node does not take a risk, it may not have any gain. However, if taking a risk introduces a small gain or even a high penalty, a node would not take the risk. In this work, we identify the best balance between trust and risk in order to maximize the mission completion ratio which leads to the maximum payoff of a mission team.

B. TASK ASSIGNMENT IN MANETS

Task assignment problems have been studied to perform tactical operations in military MANETs. Cho *et al.* [16] used context-dependency to characterize trust and proposed a trust management scheme for maximizing mission success in tactical MANETs. The authors investigated a group member selection process for mission execution. They also proposed a combinatorial auction-based mission assignment algorithms for MANET environments and analyzed the merit of the proposed auction-based algorithm in communication cost and mission completion performance [17]. However, The above works [16], [17] did not address the effect of dynamically arriving tasks and risk behaviors.

Task assignment problems have been also explored in service-oriented MANETs. Wang *et al.* [18] proposed a trust management protocol for autonomous service-oriented MANETs with multiple conflicting objectives to effectively deal with malicious nodes exhibiting opportunistic service attacks and slandering attacks. In [19], the authors further investigated how trust-based service composition and binding protocol outperforms non-trust-based counterparts in terms of a user satisfaction level. Although the work is similar to our work, it did not investigate how an acceptable risk level for each task and an entity's risk behavior characteristics affect performance in service provision.

Auction-based approaches [20]–[27] have been actively employed for task assignment. Lee [20] proposed a resource-based task allocation algorithm for multi-robot systems. They considered an auction-based algorithm that uses remaining resources when performing task allocation. The proposed work, however, is limited to multi-robot systems without considering risk attitudes and behaviors. Schwarzrock *et al.* [21] studied a task allocation problem in cooperative systems using Unmanned Aerial Vehicles (UAVs). They used a swarm intelligence and multi-agent system approach to enable UAVs to individually decide which tasks to perform. But their work didn't consider malicious entities and is restricted to UAV operations. Tolmidis and Petrou [22] provided a solution to a multi-robot dynamic task allocation problem by leveraging an multi-objective optimization technique for task allocation.

Du *et al.* [23] proposed an auction-based approach to improve the sharing of data allowances among mobile users acting as data auctioneers and requesters. They considered mobile users' behaviors and their demands when optimizing for the sellers' incomes and needs, which finally determines the transfer of data allowances. In [24], they further investigated a similar model for mobile offloading but for mobile social platforms with the aim to balance data bidders and increase the income per unit time for sellers. However, their works [23], [24] didn't consider mission-oriented tactical environment requiring a level of trust and security to ensure mission completion under hostility. In addition, they only considered user's behaviors with regards to data spending without considering risk-based behavior modeling that significantly affects in their decision making process. Asghari and Shahabi [25] studied the problem of on-line task assignment in spatial crowdsourcing where the matching and scheduling responsibilities are divided between a spatial crowdsourcing server and workers. The authors focused on solving the bottleneck issues of using spatial crowdsourcing of task matching and task scheduling, and used an on-line auction-based framework.

Whitbrook *et al.* [26] extended the performance impact algorithm, which is a distributed auction-based task allocation algorithm, to allow dynamic online rescheduling and enhance its exploratory properties. Similar to our work, they considered dynamic task reassignment; however, their work mainly focused on the scheduling efficiency of the algorithm, and didn't consider the malicious entities in a given environment. Li *et al.* [27] examined how to protect the privacy of bidders in an auction-based mobile crowdsensing system. They provided the theoretical analysis and real-life tracing data simulations to prove the efficiency of the proposed mechanism. However, their work is limited to preserving privacy without considering a task allocation model. While the above works [20]–[27] tackled different, important aspects for a mission-oriented task allocation in MANETs, none of the above provide a holistic solution for such a system based on the concept of trust. Unlike these works, our work considers not only trust but also risk, and explores the trade-off between them for an optimal task assignment.

Unlike energy-aware performance metrics used in the literature [28]–[30], we use a performance metric called 'mission completion ratio' where a mission consists of multiple dynamic tasks, similar to the metric measuring the number of tasks completed in [31]. Unlike [29], our work considers and analyzes the effect of task importance on mission completion ratio.

Berg *et al.* [32] proposed a decision making model that allows three types of controls, namely, explicit incentives, monitoring, and reputation, to enhance confidence and trust in establishing initial interactions for delegation. Wang *et al.* [33] proposed a trust-based task scheduling mechanism for grid computing MANETs to maximize mission completion considering the required security and reliability in task assignment with minimum delay. Like [32], [33], we also took a trust-based approach; but we consider the risk behavior tendency of a decision maker and its impact on decision performance.

Unlike our previous works in [16]–[18], this work considers a node's risk behavior and aims to maximize the completion ratio of missions of multiple tasks by balancing trust and risk. Unlike these previous works, this work considers not only trust but also risk, and explores the trade-off between them for an optimal task assignment.

III. SYSTEM MODEL

A. NETWORK MODEL

In this work we consider a multi-hop MANET consisting of heterogeneous entities differing in functionality (e.g. sensing and actuating) and nature (i.e. machine or human). Thus entities include sensors, robots, unmanned vehicles, and humans (dismounted or aboard manned vehicles). We consider a mission with multiple tasks dynamically arriving where each task is a basic unit. Entities are responsible for carrying out tasks where each task has its own time frame (start and end time) and constraints (e.g. some tasks can run concurrently with tasks while others cannot).

We assume the use of a head leader (HL) responsible for governing and choosing task leaders (TLs) where each TL is responsible for leading a task team. TLs are chosen by the HL based on trustworthiness and node type matching with a given task. The TLs in turn choose members to carry out the allocated task. When a TL is not available and cannot lead a given task team due to its leave or being disconnected from the network, the HL selects a new TL among members available based on its type and level of its trustworthiness. A symmetric key, as a group key, can be used to prevent outside attackers from secure group communications between members. We use Group Diffie-Hellman (GDH) [34], an extension of the well-known two-party Diffie-Hellman (DH) key exchange protocol as a *contributory key agreement (CKA)* protocol, to generate a group key based on the agreement of group members as a shared secret key without having a secure channel.

When a node is disconnected from the mission group, the HL initiates running the GDH protocol and each node can use a new key based on the shares of other member nodes to maintain a valid, secret group key. Despite this key update, a group member may keep old trust information with non-member nodes (i.e., nodes that left the group) to be referred for interactions in the future. This way can prevent potential newcomer attackers (i.e., performing frequent rejoining to nullify their low trust in the past or current sessions). Further, the old trust information of non-members can be used as their initial trust upon their rejoin to the group. To this end, we use the authentication process when a node joins a network based on a public/private key pair. In the beginning of network deployment, each node is pre-loaded with a pair of public/private keys and other nodes' public keys. Upon rejoining a network, a node will regenerate a new pair of public/private keys based on old private/public keys respectively where other nodes are also able to generate new public keys of other nodes based on corresponding old public keys. Through challenge/response process using a private/public key pair of a node, the node's ID is authenticated and its old trust information is used to continue trust estimation upon the node's join.

B. TRUST BOOTSTRAPPING MODEL

We assume that when the network is initially deployed, there is no predefined node trust except for the HL that governs the mission group. At time of deployment, an entity's trust is computed based on limited direct observations, indirect third-party information, and challenge/response process authentication. A stronger trust level (i.e. with more confidence) is established as time goes on and the entity interacts more with other entities thus yielding more observations. Trust levels are computed at intervals and are based on interactions, thus without further updates or interactions between entities, trust decays over time. While node mobility can increase the chances of trust evaluation by bringing nodes into contact with one another, it may also hinder trust evaluation when the nodes physically out of reach. Mobility may occur when a node disconnects from its current group, leaves a group intentionally, or disconnects to save power. Involuntary disconnection may also occur due to physical location or terrain.

The motivation for a node to participate in task execution is to increase its trust level so that it has more chances to access network resources. Trust will decay when a node does not participate in task execution. Trust will decrease when a node fails to execute a task to completion due to misbehavior. Hence, a node will continuously participate in task assignment and select tasks with a reasonable chance of success for task execution so as to increase or at least maintain its trust level.

C. NODE BEHAVIOR MODEL

We consider M node types, $NT_1, \cdots, NT_M$, representing that a higher node type has higher capability than a lower node type. Furthermore, node types involving human interaction have more trust dimensions (i.e. QoS trust and social trust). This is shown in Table 1 where node types, their relevant characteristics, and trust dimensions are shown. Figure 1 shows an example mission group composed of a team leader (TL) and joined entities (i.e., bid-winners) to execute a task.

TABLE 1. **Characteristics of node types.**

Node type	Node characteristics	QoS trust	Social trust
NT_1	Stationary sensors	✓	
NT_2	Unmanned vehicles or robots carrying devices	✓	
NT_3	Human carrying devices	✓	✓
NT_4	Manned vehicles equipped with devices	✓	✓

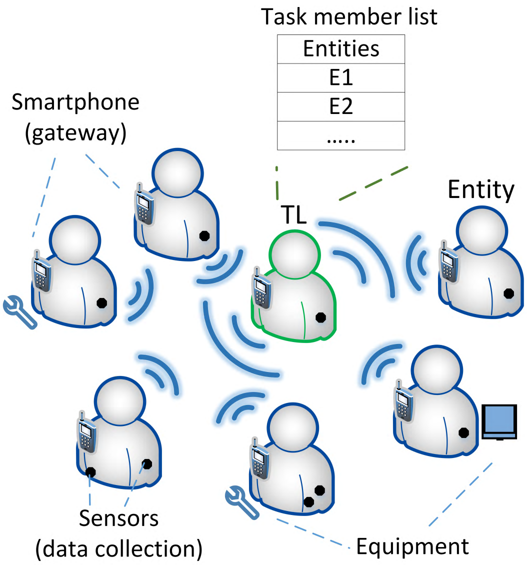


FIGURE 1. **An example mission group composed of a team leader (TL) and joined entities (i.e., bid-winners) for task execution.**

We consider both stationary entities, such as sensors, and mobile entities, including humans, robots, or vehicles. Prior to task assignment, nodes are assumed to follow their own mobility pattern (which we assume is random in this work). A node's mobility is influence by its TL and its assigned task, where a node stays within reach of the TL (and its group members) which assigned the task, and moves towards a new TL when it subsequently switches to a new task belonging to the new TL. Nodes have the freedom to leave and join the group. This happens with rates λ and μ respectively. Nodes have vastly different characteristics in terms of capabilities in speed, monitoring, and cooperation level (i.e., packet dropping). Furthermore, a node has monitoring capabilities which it uses to monitor neighboring node behaviors and actions. However, this monitoring and anomaly detection is not without error, and is characterized by false positive and false negative probabilities. We assume all nodes initially are benign but can be captured and converted into malicious nodes (see Section III-D for the definition). In this work, we do not assume the various distributions required by the protocol; we only investigate it in Section VII to show insights found in terms of the impact of the heterogeneity on performance. We summarize the parameters of a node considered in this work as follows:

- **Speed** (v_i): Node i moves randomly with speed v_i in between task assignments.
- **Detection error** (P_i^{fp}, P_i^{fn}): Node i's monitoring and detection is characterized by a false positive probability (misidentify a good node) and a false negative probability (fails to identify a bad node) when monitoring.
- **Group join and leave** (λ_i, μ_i): Node i may leave or join a group where the inter-arrival time of the events is exponentially distributed, with mean values λ_i and μ_i.
- **Cooperativeness** (P_i^C): Node i may drop a packet with the probability $(1 - P_i^C)$ based on its inherent characteristics of cooperativeness.
- **Reciprocity** (P_i^R): Node i may reciprocate the service received by other nodes with this probability based on its inherent characteristics of reciprocation.
- **Node compromise time** (σ_i): A node may be compromised with a certain rate, $\frac{1}{\sigma_i}$ where σ_i is selected from $[C_{min}, C_{max}]$ based on uniform distribution.

D. THREAT MODEL

Both uncooperative nodes and malicious nodes are considered in our system where uncooperative nodes exhibit selfish behavior and refrain from protocol participation in our system to selfishly hold on to their resources and maximize their individual gain. Thus, for example, an uncooperative node can choose to avoid relaying/transmitting packets in order to avoid energy consumption. Whereas malicious nodes aim to compromise and cause failure to the system. A malicious node thus performs packet jamming, good/bad mouthing attacks, forging and fabricating packets, in addition to packet dropping.

E. TASK MODEL

The system executes a mission where each mission is composed of multiple tasks where each task may be unique with regards to start time and duration, with the task duration of task m denoted as DT_m. Furthermore, each task will be matched by the TL with suitable members with respect to functionality (i.e., minimum NT_2 refers to a node with a node type equal to or above NT_2 as an eligible node) and trust level in each trust property X. We provide more details of our composite trust metric in Section IV.

Tasks arrive asynchronously and may start and end at different times. Each task has unique properties:

- **Required node type** Each task m is required to be executed by a node with a functionally compatible node type as specified by the TL (denoted by NT_m^{min}). Higher node types indicate higher compatibility, with human involvement further indicating high trust dimensions.
- **Task execution timeframe** (ET_m) refers to the start and end times of task m where the duration of task m, DT_m, is computed by the difference between the end and start time.
- **Minimum and maximum node population** (N_m^{min} and N_m^{max}) is needed for executing task m.
- **Minimum trust threshold** (T_m^{X-th}) is a threshold for each trust property X of task m.
- **Importance** (I_m) refers to the impact of task failure on mission completion with a higher value indicating more importance.
- **Urgency** (U_m) indicates how urgent a given task should be completed where a higher value means more urgent. The time allowed for task completion is urgency dependent where less urgent tasks may be allowed extra time for completion, beyond the normal end time.
- **Difficulty** (DF_m) represents task m's difficulty associated with an amount of required workload. This determines a minimum number of members; correspondingly it affects a maximum possible workload per time unit to be assigned to each member. A higher value refers to a more challenging task.

The concepts of urgency and difficulty are considered in estimating the risk level of executing a particular task while the concept of importance is used in calibrating the mission completion ratio. The level of an acceptable risk influences the degree of mission completion; thus, the three task properties naturally influences mission performance.

F. RISK BEHAVIOR MODEL

A node's risk propensity may affect its decision making particularly when the decision significantly affects its utility. We model three types of risk behaviors: risk-seeking, risk-neutral, and risk-averse [35]. We designate a node's risk behavior type based on its choice when multiple tasks are offered where a task with high importance brings high-payoff upon success but high-penalty upon failure. A node can choose a task with high importance as the TL will give the node highly positive trust recommendations if the node successfully completes the task, resulting in maximizing the mission completion ratio. However, when the node fails a high importance task, it may face risk as the TL will disseminate highly negative trust recommendations to other nodes. Therefore, we use an increase or decrease of a node's trust value as the reward (payoff) or penalty.

- **Risk-seeking**: A node tends to make a decision by taking a high risk in order to gain a high payoff. However, the node may face a high penalty upon task failure.
- **Risk-neutral**: A node tends to make a decision by taking a moderate risk in order to gain a moderate payoff.
- **Risk-averse**: A node tends to make a safe decision even if there is a low payoff.

We consider a node's risk behavior type in the decision making process during bidding, winner selection, and commitment. This is detailed in Section V.

IV. COMPOSITE TRUST METRIC

We define our proposed trust metric with two dimensions: *social trust* and *QoS (Quality-of-Service) trust*. Social trust means trust based on relationships between people such as 'friendship, familiarity, intimacy, honesty, or centrality (betweenness)' which are popularly used to enhance productivity based on those social relationships [2]. In the context of asset-task assignment, we leverage the concept of social trust to measure social connectedness and reciprocity, which are measured by:

- **Social Connectedness** (SC): Is a measure of social connections in a node's social circle [36]. A node's mobility pattern and a node's sociability effects social connectedness of a node over a measured period of time.
- **Reciprocity** (R): This is the degree of mutual giving and receiving [37]. When a node receives a favor from a giving node it is more likely to return the favor to the giving node. The degree of the reciprocity [37] can be estimated by the duration an entity returns for the past favor it received from another entity and the amount of net gain it returned. A node's reciprocity is dependent on its willingness to reciprocate (e.g., emotional status) and its expected future gain when returning the favor.

QoS trust is a measure of trust based on quality of service characteristics such as competence, availability, and reliability. We measure QoS trust in terms of competence and integrity which are captured by:

- **Competence** (C): This refers to an entity's capability to serve the received request, and is often called service availability. Competence may be affected by: (a) unintentional unavailability due to network or node conditions (e.g., node failure and disconnections); and (b) intentional nature of an entity (e.g., cooperativeness or willingness).
- **Integrity** (I): considers the selfishness and maliciousness behaviors of a node as an indication of a system attack which can be observed in both humans and machines.

A. OBJECTIVE TRUST

In this work, we model a node's ground truth trust (i.e., "objective trust") using a behavioral seed to represent its inherent, natural behavior. We use the objective trust to validate the accuracy of measured trust.

Objective trust of social connectedness in node j is based on node j's inherent sociability (P_j^{SC}) in the range of [0, 1], and the number of nodes encountered by it. Objective social connectedness trust of node j is defined by:

$$T_j^{SC} = \begin{cases} P_j^{SC} N_j^{enc} c & \text{if node } j \text{ is a member;} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where N_j^{enc} is the number of nodes node j encounters during a trust update interval and c is a normalizing parameter.

Objective trust of reciprocity in node j is modeled with a given initial seed behavioral relationship (P_j^R) as a real number in [0, 1]. We assume that node j' reciprocity trust in node i is based on node i's reciprocity trust in node j due to its nature of mutual interactions [32]. We assume the mutual favors between node i and node j in that if node i returns a favor to node j based on what node i received from node j. Objective trust of node j in reciprocity is estimated by:

$$T_j^{R} = \begin{cases} P_j^{R} & \text{if node } j \text{ is a member;} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Objective trust of node j in competence is estimated by node j's inherent cooperativeness (P_j^C) as a real number in $[GB_{min}, 1]$ and the link reliability based on network conditions (P_r) as:

$$T_j^{C} = \begin{cases} P_j^{C} P_r & \text{if node } j \text{ is a member;} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Objective integrity trust of node j is based on whether a node is compromised (i.e., 0 or 1) as:

$$T_j^{I} = \begin{cases} 1 & \text{if node } j \text{ is not compromised;} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

B. SUBJECTIVE TRUST

Each node performs peer-to-peer trust evaluation periodically, which is called "subjective trust" [38], using either direct evidence (i.e., direct observations) or indirect evidence. Nodes within the vicinity of one another (i.e., within wireless radio range) collect evidence which serves as a means for direct evaluation. This is done using installed monitoring mechanisms, where evidence is an indicator for the changes in trust (i.e., increasing or decreasing). The peer-to-peer trust evaluation is performed between nodes except the HL. Only the HL receives trust evaluation information about all TLs and regular nodes from TLs, and uses the average trust values to evaluate all nodes. The HL will use them for the selection of a new TL when the current TL is detected as untrustworthy. The HL will revoke the trust of an untrustworthy node (i.e., drop to zero) if the average trust value falls below a system tolerance level, denoted by T_{th}^{min}.

Node i's trust in node j for trust property X at time t, $T_{i,j}^X(t)$, is represented as a real number in [0, 1] where 1 indicates complete trust, 0.5 ignorance, and 0 distrust. The initial trust value is set to the ignorance value 0.5 as we do not assume trust is predefined in the network. When a trustor (node i) evaluates a trustee (node j) at time t in each trust property X, it updates $T_{i,j}^X(t)$ as follows:

$$T_{i,j}^X(t) = \alpha T_{i,j}^{D-X}(t) + (1-\alpha) T_{i,j}^{ID-X}(t) \tag{5}$$

$T_{i,j}^X(t)$ is based on both direct trust evidence, $T_{i,j}^{D-X}(t)$ (i.e., node i's direct observations or experiences), and indirect trust evidence, $T_{i,j}^{ID-X}(t)$, collected based on recommendations from third parties. α is a weight for direct evidence while $(1-\alpha)$ is a weight for indirect evidence where $1 < \alpha \leq 1$. The recommendations will be received from node i's 1-hop neighbors. Thus increasing the α increases the reliance on direct observations. In this work we follow [38] to find the best α yielding subjective trust values closest to the ground truth.

Direct trust of node i in node j on trust property X at time t, $T_{i,j}^{D-X}(t)$, is computed as:

$$T_{i,j}^{D-X}(t) = \begin{cases} P_{i,j}^{D-X}(t) & \text{if } HD(i,j) == 1; \\ \gamma T_{i,j}^{D-X}(t-\Delta t) & \text{otherwise,} \end{cases} \tag{6}$$

Where the direct trust is based on observations collected in period Δt, the periodic trust interval, when node i and node j are within a single hop distance. When $HD(i,j)$ is greater than a single hop, past trust experience (with applied decay γ) is used to derive the direct trust.

Below we show how to evaluate the direct trust value for each trust property when trustor node i encounters trustee node j. Note that a node may imperfectly observe evidence to derive trust values in each property.

Direct competence $P_{i,j}^{D-C}(t)$ is derived based on the number of replies received, $N_{i,j}^{rep}$, over the total number of requests, $N_{i,j}^{req}$ sent and is computed by:

$$P_{i,j}^{D-C}(t) = \begin{cases} \dfrac{N_{i,j}^{rep}}{N_{i,j}^{req}} & \text{for } N_{i,j}^{req} > 0; \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Direct integrity $P_{i,j}^{D-I}(t)$ is the ratio of number of messages received correctly, $N_{i,j}^{msg-crt}$ to the total number of messages received, $N_{i,j}^{msg-rcv}$, as:

$$P_{i,j}^{D-I}(t) = \begin{cases} \dfrac{N_{i,j}^{msg-crt}}{N_{i,j}^{msg-rcv}} & \text{for } N_{i,j}^{msg-rcv} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Note that detection error is taken into account.

Direct social connectedness $P_{i,j}^{D-SC}(t)$ is based on prior information about node j's sociability (P_j^{SC}) and the number of encounters with node j:

$$P_{i,j}^{D-SC}(t) = P_j^{SC} N_j^{enc} c \tag{9}$$

Notice that $P_{i,j}^{D-SC}(t)$ is computed in the same manner as the objective trust in Equation (1), but takes into account detection errors.

Direct reciprocity $P_{i,j}^{D-R}(t)$ is on the ratio of the number of services received by node j, $N_{i,j}^{svc}$, to the number of services provided by node i, $N_{j,i}^{svc}$, as:

$$P_{i,j}^{D-R}(t) = \frac{N_{j,i}^{svc}}{N_{i,j}^{svc}} \tag{10}$$

We also considered detection error in computing $P_{i,j}^{D-R}(t)$. For example, a node may mistakenly detect a positive experience as negative or a negative experience as positive with false positive or negative probability of a monitoring mechanism used by each node.

Indirect trust of node i in node j on trust property X at time t, $T_{i,j}^{ID-X}(t)$, is obtained by:

$$T_{i,j}^{ID-X}(t) = \begin{cases} \dfrac{\sum_{k \in R_i} T_{k,j}^{D-X}(t)}{|R_i|} & \text{if } |R_i| > 0; \\ \gamma T_{i,j}^{X}(t - \Delta t) & \text{otherwise.} \end{cases} \quad (11)$$

R_i is the set of 1-hop neighbors (whose trust is not revoked) of node i providing trustworthy recommendations towards node j. It calculates $T_{i,j}^{ID-X}(t)$ as the average of trustworthy recommendations. $T_{k,j}^{D-X}(t)$ is the direct trust evaluated by recommender node k towards node j. If R_i is an empty set, node i will use its past experience $\gamma T_{i,j}^{X}(t - \Delta t)$ with a decay factor, γ.

V. COSTA

The proposed COSTA is designed based on "a single item auction with multiple preferences" [39]. This auction type considers each bidder bidding on multiple items to select one in the end. This technique leads to the final assignment of a task to a node based on the mutual agreement between an auctioneer and a bidder, and is more likely to reduce potential rounds of auction processes. As seen in Figure 2, we have two layers of the auction process: between the HL and TLs and between TLs and members. The first auction is used to select TLs (with the HL being the auctioneer) while the second one enables the TL to recruit task members (with the TL being the auctioneer). In Sections V-B through V-H below, we describe the auction process for member selection between TLs and members in detail (see Figure 3). The auction process for TL selection between the HL and TLs can be conducted in a similar way with the HL being the auctioneer; it is briefly summarized in Section V-A.

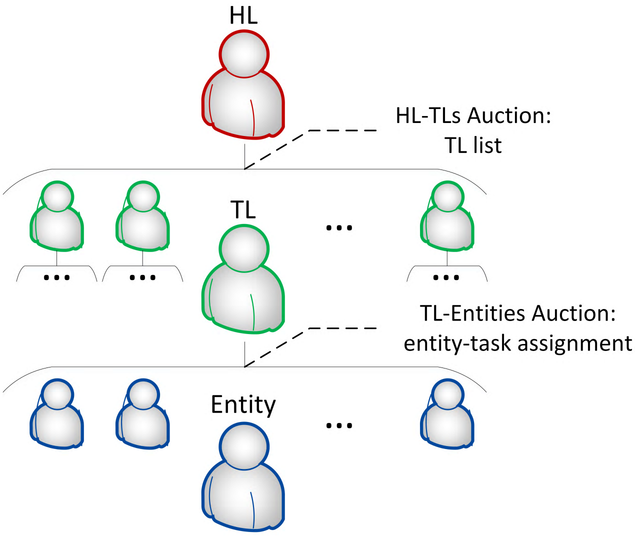


FIGURE 2. The two layers of the auction process.

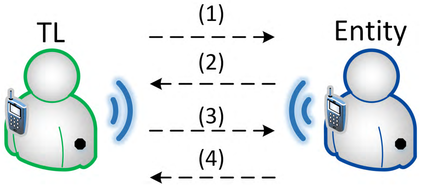


FIGURE 3. TL-Entities auction resulting in entity task assignment: (1) advertisement of a task specification; (2) an interested entity bidding on a task; (3) winner determination and notification by the TL; and (4) node commitment to the task assigned with the notification to the TL.

A. TASK LEADER SELECTION

In the auction process between the HL and TLs, the HL, acting as an auctioneer advertises the specification (discussed in Section V-B) of all tasks to all nodes acting as bidders. Each node bids on tasks for which it meets the task requirements. A node can participate only in one task at a particular time t (no concurrent task execution at time t, but can execute multiple tasks during its entire lifetime) but can apply for multiple tasks; this can ultimately reduce communication overhead in the process of task assignment. The HL selects winners based on the required node type and the degree of trustworthiness. The HL sends out winner notifications to all qualified candidates. Each candidate if receiving multiple winner notifications will select one based on its risk behavior type. If a task is not assigned, it will be auctioned in the next round until all tasks are assigned. Upon success or failure of a task, the TL will receive a reward or penalty proportional to the importance level of the task. A risk seeking node selects a task with high importance while a risk-averse node selects a task with low importance. We will give details on the computation of reward/penalty in Section V-F.

B. ADVERTISEMENT OF TASK SPECIFICATION

The task specification disseminated during the auction process includes a set of requirements for task execution by:

$$\left[ID_m, L_m, I_m, NT_m^{min}, N_m^{min}, N_m^{max}, ET_m, W_m \right] \quad (12)$$

ID_m is the identifier (ID) of task m, L_m is the location of the task leader, I_m is the importance level, NT_m^{min} is the minimum required node type, N_m^{min} and N_m^{max} are the minimum and maximum numbers of member nodes, ET_m refers to the start and end time of task m, and W_m is a maximum workload required per time unit for each member to perform task m (e.g., a number of packets to process). To obtain W_m, each TL estimates the maximum workload possible per time unit based on N_m^{min} to complete task m. Thus, a TL may want to issue more winner notifications than N_m^{min} so it will not burden members with the maximum workload and some members may have the contract terminated due to their misbehavior or unavailability.

C. BIDDING

Upon each node receiving TLs' task specifications, it can bid on multiple tasks relevant for its availability (i.e., schedule), qualification (i.e., node type), and preference (i.e., score described below). Since each node can perform multiple tasks during its lifetime, it needs to resolve any schedule conflict impacting the performance of another task execution. After the node finds right task(s), it decides a task to bid based on its score in task m ($s_{i,m}$), which is obtained by:

$$s_{i,m} = v_{i,m} - p_{i,m} \tag{13}$$

where

$$v_m = \frac{DT_m}{DT_{max}} \text{ and } p_{i,m} = \frac{W_m}{w_i}$$

where $v_{i,m}$ is the "valuation" node i will gain from being selected to execute task m; $p_{i,m}$ is the "price" node i will pay to execute task m; DT_m is the task duration; and DT_{max} is the maximum duration among all tasks. Here $v_{i,m}$ is estimated by the relative degree of task duration. A node is more likely to choose tasks with longer duration due to its high benefit of having privileges to access resources and chances to obtain a high trust level by continuous active interactions with other nodes. $p_{i,m}$ is based on node i's maximum capability to handle workload per time unit w_i vs. the required workload per time unit by task m (W_m). w_i is affected by the inherent capability and cooperative attitude of node i. Thus, $s_{i,m}$ may be negative when the workload exceeds the node's capability. Recall that a node only bids on positive net gains in score $s_{j,m}$ and may apply for bids on multiple tasks (i.e., multiple preferences). A bidder's message to a TL is:

$$[ID_n, NT_n, C_n] \tag{14}$$

where ID_n is the identifier of bidder n, NT_n is its node type, and C_n is the workload capacity of the bidder.

D. WINNER DETERMINATION

Since a TL can receive bids from multiple entities, it needs to determine winners based on qualification criteria to select right entities while meeting an acceptable risk level. Each TL needs to check a selected entity to keep a certain level of trust per trust property X during task execution and makes sure that the exposed risk level with current members selected for task execution does not exceed a given acceptable risk level.

The risk level $r^X_{m,j}(t)$ perceived by the TL (i.e., trustor) of task m when node j is selected to execute task m at time t, is calculated by:

$$r^X_{m,j}(t) = e^{-\rho_1 \frac{T^X_{i(m),j}(t)}{T^{X-th}_m}} \frac{U_m}{U^{max}_m} \frac{D_m}{D^{max}_m} \tag{15}$$

where T^{X-th}_m is the minimum trust threshold in trust property X without increasing the risk level above the task's acceptable risk threshold P^{risk}_m (discussed below). Each trust property X may have a different trust threshold T^{X-th}_m to reflect the nature of the unique task property. ρ_1 is a constant parameter chosen based on P^{risk}_m to guarantee that the acceptable risk level is less than P^{risk}_m. U^{max}_m is the maximum task urgency among all tasks and D^{max}_m is the maximum difficulty among all tasks. Equation (15) indicates that while the risk of selecting a member to join a task increases only linearly with the task's urgency and difficulty, it increases exponentially with the member's distrust expressed as the ratio $\frac{T^X_{i(m),j}(t)}{T^{X-th}_m}$ which critically endangers successful task execution. Lund *et al.* [40] suggested that the computation of the risk level $\mathcal{R}$ can be expressed as a function of the consequential loss $\mathcal{L}$ of a harmful event and the probability $\mathcal{P}$ of its occurrence, i.e., $\mathcal{R} = \mathcal{L} \times \mathcal{P}$ where $\mathcal{P}$ corresponds to the distrust level and $\mathcal{L}$ represents an impact upon failure. We adopt the exponential form in Equation (15) to reflect the risk of distrust. Based on $r^X_{m,j}(t)$ for each trust property X, the TL of task m computes the risk level when node j is selected as its member, as the average risk level among all trust properties:

$$r_{m,j}(t) = \sum_{X \in T} \frac{r^X_{m,j}(t)}{|T|} \tag{16}$$

where T is the set of trust properties X's. Since the weight of each risk per trust property X is implicitly based on T^{X-th}_m, we simply use the average risk.

A TL has a goal to maximize task completion ratio while meeting an acceptable risk level to the task. The TL's objective function on task m is formulated by:

$$\text{Maximize } P_m(t), \quad \text{given} \sum_{j \in M} r_{m,j}(t) \leq P^{risk}_m \tag{17}$$

where $P_m(t)$ is the completion ratio of task m at time t, M is the set of task members assigned to task m, and P^{risk}_m is the acceptable risk threshold for task m modeled as:

$$P^{risk}_m = e^{-\rho_2 I_m} \tag{18}$$

I_m is the task importance of task m, and ρ_2 is a constant parameter to normalize P^{risk}_m. Equation (18) indicates that a task's acceptable risk threshold is exponentially related to the task importance to reflect the consequential loss of an important task [40]. A more stringent risk threshold allows less vulnerability for a task with high importance [41]. Here we note that $P_m(t)$ can be either 0 or 1 based on if task m is completed within the mission time.

TABLE 2. Acceptable risk level per risk behavior type.

Behavior Type	Risk-Seeking	Risk-Neutral	Risk-Averse
P^{risk}_m	$e^{-\rho_2 I_m}(1+\epsilon)$	$e^{-\rho_2 I_m}$	$e^{-\rho_2 I_m}(1-\epsilon)$

Table 2 lists the would-be "adjusted" acceptable risk level based on the TL's risk behavior type. Here ϵ is a design parameter specifying the adjustment increment. A risk-seeking TL takes a high risk by relaxing the acceptable risk level threshold for task m, P^{risk}_m, by ϵ while a risk-averse TL takes a low risk by tightening P^{risk}_m by ϵ. In the winner selection process, a TL checks if an applicant node is qualified for the required node type, and then estimates the risk level exposed by the applicant. When both conditions are met, the TL gives preference to an applicant with the minimum eligible node type. This may give other TLs better chances to recruit qualified members if they require members with a high node type.

E. WINNER NOTIFICATION AND NODE COMMITMENT

After reviewing the qualifications of bidding nodes and analyzing the potential risk level, a TL determines winners and notifies them of the acceptance as task members (step 3 in Fig. 3). If a node receives multiple winner notifications, it chooses the task based on its risk behavior type and the task's importance (I_m) as follows:

- **Risk-seeking**: A node chooses a task with the highest importance among all winner notifications received.
- **Risk-neutral**: A node chooses a task with the medium importance among all winner notifications received.
- **Risk-averse**: A node chooses a task with the lowest importance among all winner notifications received.

After a node decides to commit to a task, it notifies the tasks issuer (TL) of the commitment (step 4 in Fig. 3), after which the TL issues a contract between itself and the committed node. In the case where multiple TLs issued multiple advertisements at the same time, then these tasks would have been checked a priori for their ability to be run concurrently. Furthermore, each node can choose only one task so as not to cause scheduling conflict.

F. COMPUTATION OF REWARD OR PENALTY

The reward or penalty received depends on whether or not a node as a member completes a task successfully. When a member successfully completes a given task, it will gain trust based on the reward. Similarly, when a member fails the task, it will lose trust based on the penalty.

1) TASK LEADERS

The HL gives a reward or penalty to a TL based on the completion or failure of task m assigned to the TL:

$$TL^{I_m}_{reward} = TL^{I_m}_{penalty} = \tau I_m \tag{19}$$

where I_m is the importance of task m and τ is a constant to normalize $TL^{I_m}_{reward}$ or $TL^{I_m}_{penalty}$. The reward or penalty is based on the importance level of the task because a TL makes a decision based on its risk behavior type, which is related to the importance level of the task.

2) MEMBERS

A TL also gives a reward or penalty to a member node depending on whether the member node successfully completes the given task or not. A member node's decision on which task to choose depends on its behavior type. Thus, a TL gives a reward or penalty based on member j's risk behavior type. Specifically, the TL gives a higher reward (payoff) or penalty to risk-seeking members while giving a smaller reward or penalty to risk-averse members. The reward or penalty given to member j is computed by:

$$M^{m,j}_{reward} = M^{m,j}_{penalty} = \tau r^{dec}_j \tag{20}$$

where r^{dec}_j is the reward/penalty factor with 1 for risk averse, 2 for risk neutral, and 3 for risk seeking members, and τ is a normalization constant to normalize $M^{m,j}_{reward}$ and $M^{m,j}_{penalty}$.

G. DYNAMIC TASK REASSIGNMENT

1) LACK OF MEMBERS

In the case where the available members to execute a task are insufficient (at time of task advertisement) a reassignment protocol will be run where the TL first attempts to extend the task completion time of the task based on its knowledge of the tasks urgency and member availability. If it decides that an extension can be made, it notifies all members regarding the extension request. If it decides that an extension cannot be made, then it looks for other members to fill the deficiency, and if acquired, they take on the responsibility of task execution. If these options fail, the TL simply marks the task as incomplete.

2) TERMINATION OF CONTRACT

The TL-member task contract can be terminated in the case when the member is disconnected or unreachable. A decrease in a member's trust level could further trigger an overall increase in the tasks risk, which results in the TL terminating the contract with the member having the maximum risk. Termination of high risk nodes continues until the sum of risk levels is below the threshold, P^{risk}_m, after which the TL again runs the reassignment protocol, as described above.

H. TASK FAILURE

The main reasons of failing a task are: (a) lack of members in the initial task assignment period (i.e., a TL cannot find a sufficient number of members); (b) lack of qualified members successfully leading to task completion (i.e., a TL cannot find a qualified member when a member leaves the group); and (c) some of current members have their trust level below the minimum trust threshold. The third failure condition in (c) is defined as:

$$\sum_{j \in M} F_j(t) > N_{th} \tag{21}$$

where

$$F_j(t) = \begin{cases} 1 & \text{if } T^X_j(t) \leq T^{X-th}_m \text{ for any } X \\ 0 & \text{otherwise} \end{cases}$$

M refers to a set of members for task m, $F_j(t)$ is 1 when any objective trust value on property X of node j does not satisfy the threshold for X; 0 otherwise. Since it is impossible to reach a consensus when there are more than 1/3 untrustworthy/compromised nodes, we set N_{th} to 1/3, representing the maximum tolerable threshold after which the task can no longer be executed reliably.

TABLE 3. Binary variable definitions for ILP.

Variable	Definition
$a_{j,m}$	1 if node j is available (is a member); 0 otherwise
$b_{j,m}$	1 if $(a_{j,m} \times t_{j,m} \times nt_{j,m} \times v_{j,m}) > 0$
$bt_{j,m}$	1 if node j's behavior type matches the importance level of task m; 0 otherwise
$C_{p,q}$	1 if tasks p and q ask for members (task assignment) concurrently; 0 otherwise
$d_{j,m}$	1 if $(w_{j,m} \times bt_{j,m}) > 0$ (node j selects task m to commit); 0 otherwise
$nt_{j,m}$	1 if node j's node type satisfies the minimum node type of task m; 0 otherwise
$O_{p,q}$	1 if at the time of task assignment to task p, task q is still in execution; 0 otherwise
S_m	The set of m tasks in a mission
S_m^C	A set holding concurrent tasks for which $C_{p,q} = 1$ for any two tasks p, q in the set
S_m^O	A subset of S_m holding two tasks p, q for which $O_{p,q} = 1$
S_n	The set of n nodes for task assignment
TA_m	1 if $\sum_{j\in m} d_{j,m} \geq N_m^{min}$ (task m recruits sufficient members during task assignment); 0 otherwise
TE_m	1 if $\sum_{j\in m}(1 - t^*_{j,m}) \leq TH$; 0 otherwise
$t_{j,m}$	1 if $T_j^X \geq T_m^X$ for any trust property X; 0 otherwise 0
$t^*_{j,m}$	1 if $T_j^X \geq T_m^X$ for any trust property X over the task execution period; 0 otherwise
$v_{j,m}$	1 if $s_{j,m} > 0$; 0 otherwise
$w_{j,m}$	1 if $b_{j,m} \times (\sum_{j\in m} r_{j,m} < P_m^{risk}) \times (N_m^{min} \leq \sum_{j\in m} 1 \leq N_m^{max}) > 0$ (node j is a winner for task m); 0 otherwise

VI. ILP-BASED OPTIMAL TASK ASSIGNMENT

In this section, we formulate the task assignment optimization problem with trust and risk management as an ILP problem [4]. The reformulated ILP optimization problem is known to be NP-complete [4], [42] and can only be used to yield optimal solutions for networks of a moderate size. However, it provides a theoretical basis to evaluate the performance of our trust-based task assignment protocol.

We note that ILP is not to be used at runtime to solve the task assignment problem. It is a solution technique applied at design time to find the optimal solution, given knowledge of task properties and node trust/risk behaviors as input. Unlike COSTA which is to be executed by every node at runtime, ILP is to be performed at static time to generate an optimal solution against which our COSTA is compared for performance evaluation.

In Table 3, we summarize knowledge of tasks and nodes in the system in the form of ILP binary variables as input to ILP. For example, $O_{p,q}$ is 1 if p and q are concurrent tasks; 0 otherwise. The only decision variables are $w_{j,m}$ which decides if node j is selected to execute task m, and $d_{j,m}$ which decides if node j commits to task m.

The objective of our trust-based task assignment problem is to find the best bidding (i.e., which node should bid on which task), winner selection, and task selection (i.e., which task is selected when multiple winner announcements are received by a node) to maximize mission completion ratio P_{MC}. The task assignment optimization problem thus is formulated as an ILP problem as follows:

$$\begin{aligned}
&\text{Given:} \quad S_m, S_n, O_{p,q}, S_m^O, C_{p,q},\\
&\qquad\qquad S_m^C, t_{j,m}, t^*_{j,m}, nt_{j,m}, v_{j,m}, a_{j,m}\\
&\text{Find:} \quad w_{j,m}, d_{j,m}\\
&\text{Maximize:} \quad \sum_{m\in S_M} TA_m \times TE_m \times \frac{I_m}{\sum_{allm} I_m}\\
&\text{Subject to:} \quad \sum_{m\in S_m^C, w_{j,m}=1} d_{j,m} = 1; \sum_{m\in S_m^O} d_{j,m} = 1 \qquad (22)
\end{aligned}$$

The objective function (under *Maximize*) is the mission completion ratio as defined in Equation (24). The first constraint (under *Subject to*) specifies that a node can only select one task among concurrent tasks (in a set S_m^C) to join at a time. The second constraint (under *Subject to*) specifies that a node can only execute one task at a time. We note that this ILP formulation optimally assigns nodes to tasks once without considering task reassignment.

VII. RESULTS AND ANALYSIS

In this section, we first describe the performance metrics and experimental settings used for performance evaluation of COSTA. Then, we report comparative performance analysis results of COSTA against the baseline counterparts.

A. PERFORMANCE METRICS

We consider three performance metrics: trust bias, mission completion ratio, and communication overhead.

1) TRUST BIAS

$(B_{i,j})$ is the time-averaged difference between measured trust, $T_{i,j}(t)$, and objective trust, $OT_j(t)$. Given a mission lifetime LT, $B_{i,j}$ is obtained by:

$$B_{i,j} = \int_0^{LT} \frac{B_{i,j}(t)}{LT} dt \qquad (23)$$

where

$$B_{i,j}(t) = \frac{|T_{i,j}(t) - OT_j(t)|}{OT_j(t)}$$

2) MISSION COMPLETION RATIO

(P_{MC}) refers to the ratio of a mission being successfully completed during a given entire mission time. This metric is estimated by summing the task completion ratio, P_m, up where each task completion ratio is weighted by its relative importance during the mission time. P_{MC} is estimated by:

$$P_{MC} = \sum_{m \in L} P_m \frac{I_m}{\sum_{all} I_m} \tag{24}$$

L is a set of tasks belonging to the mission.

3) COMMUNICATION OVERHEAD

(C_{total}) is the number of hop messages per time unit for a node to perform trust evaluation ($C_{TE}(t)$) and run the task assignment protocol during the entire mission lifetime (LT). It is computed by:

$$C_{total} = \frac{\int_0^{LT} C_{task} + C_{TE}(t)dt}{LT} \tag{25}$$

where C_{task} consists of $C_{adv}(t)$, $C_{bid}(t)$, $C_w(t)$, $C_m(t)$, and $C_{ra}(t)$ corresponding to the costs of advertisement of tasks by auctioneers, bidding by members, winner notifications by auctioneers, commitment by members, and task reassignment by auctioneers upon the failure of task assignment, respectively.

Below we provide a detailed description of C_{total} computation. We define G as the set of current group members in the mission group, and L as the set of auctioneers. That is, L is a single-member set containing the HL only for the auction process for TL selection between the HL and TLs, and is the set of TLs for the auction process for member selection between TLs and members.

$C_{adv}(t)$ is the cost for an auctioneer (i.e., HN or TL) to disseminate their advertisement messages on available tasks at time t given by:

$$C_{adv}(t) = \sum_{l \in L} \sum_{k \in G} N^{adv}_{l,k}(t) \tag{26}$$

$N^{adv}_{l,k}(t)$ is the number of hops that an advertisement message travels from auctioneer l to entity k at time t.

$C_{bid}(t)$ is the cost for group members to send bidding messages to auctioneers (i.e., HN or TL) of the bidding tasks at time t obtained by:

$$C_{bid}(t) = \sum_{k \in G} \sum_{l \in L} B^l_k N^{bid}_{l,k}(t) \tag{27}$$

$N^{bid}_{l,k}(t)$ is the number of hops a bidding message travels from entity k to auctioneer l at time t, B^l_k is 1 when entity k bids on the task led by auctioneer l; 0 otherwise.

$C_w(t)$ is the cost for auctioneers (HN or TLs) to notify winners at time t estimated by:

$$C_w(t) = \sum_{l \in L} \sum_{k \in G} W^l_k N^w_{l,k}(t) \tag{28}$$

$N^w_{l,k}(t)$ is the number of hops a winner notification message travels from auctioneer l to entity k at time t and W^l_k is 1 when auctioneer l selects entity k as a winner; 0 otherwise.

$C_m(t)$ is the cost for members to send commitment messages to auctioneers at time t given by:

$$C_m(t) = \sum_{k \in G} \sum_{l \in L} C^l_k N^m_{k,l}(t) \tag{29}$$

$N^m_{k,l}(t)$ is the number of hops that a commitment message travels from entity k to auctioneer l at time t and C^l_k is 1 when entity k decided to commit itself to the task led by auctioneer l; 0 otherwise.

$C_{ra}(t)$ is the cost for running a dynamic reassignment protocol at time t, obtained by:

$$C_{ra}(t) = \sum_{l \in L} \sum_{k \in G} F_k C^{ra}_{k,l}(t) \tag{30}$$

F_k is 1 when entity k is not able to execute a given task; 0 otherwise. The cost for auctioneer l to run a dynamic reassignment protocol to replace entity k at time t, $C^{ra}_{k,l}(t)$, can be obtained by considering the following two cases. First, if l is a HN, $C^{ra}_{k,l}(t)$ is:

$$C^{ra}_{k,l}(t) = C^{adv}_{k,l}(t) + C^{bid}_{k,l}(t) + C^w_{k,l}(t) + C^m_{k,l}(t) \tag{31}$$

If the auctioneer is HN, $C^{adv}_{k,l}(t)$ is the cost for HN to advertise the task in order to replace TL k at time t. $C^{bid}_{k,l}(t)$ is the cost for available members to bid on the task led by HN in order to replace TL k. $C^w_{k,l}(t)$ is the cost for HN to disseminate a winner notification in order to replace TL k. $C^m_{k,l}(t)$ is the cost for a member to send a commitment message to HN in order to replace TL k. If l is a TL, $C^{ra}_{k,l}(t)$ is:

$$C^{ra}_{k,l}(t) = C^l_D(t) + E_l \Big[C^{adv}_{k,l}(t) + C^{bid}_{k,l}(t) + C^w_{k,l}(t) + C^m_{k,l}(t) \Big] \tag{32}$$

If the auctioneer is TL l, $C^l_D(t)$ is the cost for TL l to adjust the deadline of its task at time t. E_l is 1 when the deadline of the task led by TL l is not extensible; 0 otherwise. $C^{adv}_{k,l}(t)$ is the cost for TL l to advertise the available task in order to replace entity k. $C^{bid}_{k,l}(t)$ is the cost for available members to bid on the task led by TL l in order to replace entity k. $C^w_{k,l}(t)$ is the cost for TL l to disseminate a winner notification in order to replace entity k. $C^m_{k,l}(t)$ is the cost for a member to send a commitment message to TL l in order to replace entity k. $C^l_D(t)$ is obtained by:

$$C^l_D(t) = \sum_{j \in S} \left(N^E_{l,j}(t) + N^E_{j,l}(t) \right) \tag{33}$$

where S is a set of members belonging to a task led by l, $N^E_{l,j}$ is the number of hops that the deadline extension request message travels from TL l to entity j at time t. $N^E_{j,l}(t)$ is the number of hops that the deadline extension reply message travels from entity j to TL l.

$C^{adv}_{k,l}(t)$ is computed as:

$$C^{adv}_{k,l}(t) = \sum_{a \in GI} N^{adv}_{l,k,a}(t) \tag{34}$$

where $N^{adv}_{l,k,a}(t)$ is the number of hops that the advertisement message to replace entity k travels from auctioneer (i.e., HN or TLs) l to entity a at time t; GI is the set of members in the network but idle (available) at the time of request.

$C^{bid}_{k,l}(t)$ is obtained by:

$$C^{bid}_{k,l}(t) = \sum_{a \in GI} B^l_a N^{bid}_{l,k,a}(t) \tag{35}$$

where $N^{bid}_{l,k,a}(t)$ is the number of hops the bidding message to replace entity k travels from entity a to auctioneer l. B^l_a is 1 when entity a bids on the task led by auctioneer l; 0 otherwise. $C^w_{k,l}(t)$ is measured by:

$$C^w_{k,l}(t) = \sum_{a \in GI} W^l_a N^w_{l,k,a}(t) \tag{36}$$

where $N^w_{l,k,a}(t)$ is the number of hops the winner notification message to replace entity k travels from auctioneer l to entity a at time t. W^l_a is 1 when entity a is a winner of the task led by auctioneer l; 0 otherwise.

$C^m_{k,l}(t)$ is calculated as:

$$C^m_{k,l}(t) = \sum_{a \in GI} C^l_a N^m_{l,k,a}(t) \tag{37}$$

where $N^m_{l,k,a}(t)$ is the number of hops the commitment message to replace entity k travels from entity a to auctioneer l at time t, and C^l_a is 1 when entity a decided to commit itself to the task led by auctioneer l; 0 otherwise.

$C_{TE}(t)$ is the cost for evaluating trust value at time t, obtained by:

$$C^j_{i,TE_{ID}}(t) = \sum_{i \in N} \sum_{j \in R(i)} \left(N^R_{i,j}(t) + N^R_{j,i}(t) \right) \tag{38}$$

where $N^R_{i,j}(t)$ and $N^R_{j,i}(t)$ are the number of hops that the recommendation request and reply messages travels from entity i (or j) to entity j (or i) at time t. Note that we consider the evaluation of the indirect trust value because direct trust evaluation can be evaluated by monitoring or piggybacking with other communication messages.

B. EXPERIMENTAL SETUP

The ILP formulation in Section VI is implemented and solved with MS Office Excel Solver. Our proposed COSTA protocol described in Section V is simulated using an event-driven simulator in C, SMPL [43] with which we simulate task arrival, TL selection, node bidding, node selection, task execution, task abort, and trust update events. We report both analytical solutions obtained from solving the ILP problem and numerical results from simulation.

Table 4 lists model parameters used in the performance analysis. We have three parameter sets: input, derived, and design. Input parameters characterize the operational and MANET environments. Unless otherwise specified, input parameters will take on their default values shown in the 4th column for the experiment results as the default values are shown in Table 4. Derived parameters are calculated from input parameters. Design parameters are protocol parameters for which we aim to identify their optimal settings to maximize the task assignment performance. Input parameters characterize the operational and MANET environments.

All results reported are based on 100 simulation runs with the standard deviation (SD) less than 5%. We allow 2 hours of warm-up time for the network to establish acceptable trust levels among participating nodes. We use $(\alpha, \gamma) = (0:9, 0{:}95)$ to obtain 3% average trust bias at each trust update.

We conduct a comparative performance analysis in terms of mission completion ratio and communication overhead for the following three trust assignment schemes:

- *COSTA-Risk* is the scheme described in Section V. The original COSTA protocol considers risk behavior as described in Section III-F.
- *COSTA-No-Risk* is the scheme that is exactly as the COSTA in Section V except that it does not consider decision making based on risk behavior during bidding, winner selection, and commitment.
- *NT* is a non-trust based task assignment protocol. It strictly follows the procedure of the proposed auction protocol except that there is no trust-risk analysis in member selection, so a TL just randomly picks nodes with matching node types for the task.

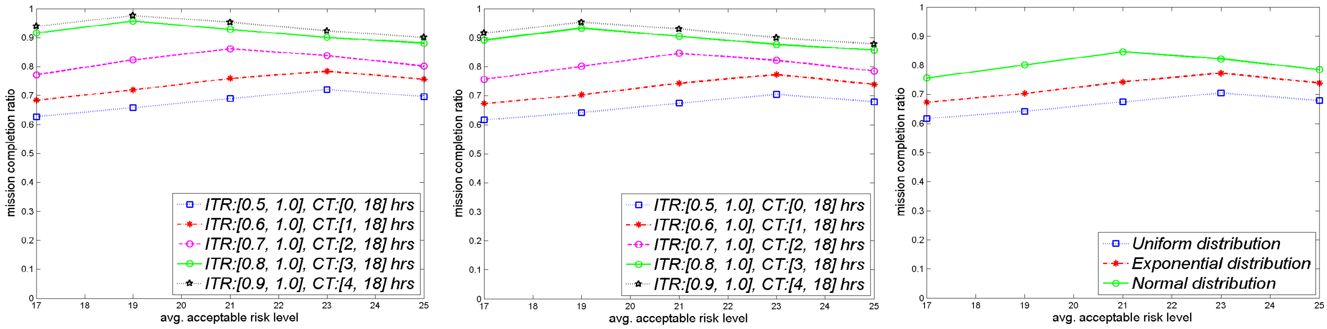
C. EFFECT OF NODE TRUST AND HOSTILITY

Figure 4 shows the effects of node initial trust range (ITR) and node compromise time (CT) on the mission completion ratio (P_{MC}) with RBR = (30%, 30%, 40%) and P^{risk}_m varying over the range of [17, 25]. Figure 4 (a) shows analytical solutions generated from ILP, while Figure 4 (b) shows COSTA-Risk solutions generated from simulation. We observe that ILP results are remarkably similar to simulation results in terms of the effect of P^{risk}_m on P_{MC} under a wide range of ITR and CT values. Furthermore, there exists an optimal P^{risk}_m under which P_{MC} is maximized and both solutions identify the same optimal P^{risk}_m for maximizing P_{MC} with varying ITR and CT values. We observe that ILP solutions consistently generate a slightly higher P_{MC} value than those by COSTA-Risk solutions, with less than 3% discrepancy between them. The reason is that task information, including arrival sequence, importance, etc. are given as input to ILP, so ILP in searching for an optimal solution will tend to assign nodes to more important tasks as well as to pick the optimal member combination for each task while satisfying the constraints.

The optimal solutions from ILP are not achievable in practice as task information is not known a priori and the system must do dynamic task assignment as tasks arrive. Nevertheless, by comparing optimal solutions obtained from ILP with our COSTA-Risk solutions, we gain confidence in the accuracy and the ability of COSTA-Risk in approaching characteristics of optimal task assignment. The most striking result is that the same optimal acceptable risk level is identified in both ILP and simulation experiments. Henceforth, we report results based on simulation.

TABLE 4. Parameters used in the performance analysis.

Parameter	Meaning	Type	Default Value
$\|M\|$	Total number of tasks given to a mission group	Input	20
$1/\lambda$, $1/\mu$	Mean inter-arrival time for a node's group join/leave event	Input	1 hr, 4 hrs
LT	Total mission time	Input	18 hrs
DT_m	Duration of task m	Input	$[1, 6]$ hrs
I_m	Importance of task m	Input	1-5
DF_m	Difficulty level of task m	Input	1-3
U_m	Urgency of task m	Input	1-3
P_i^{fp}, P_i^{fn}	False positive and negative probabilities of detection error of node i uniformly selected from the given range	Input	$(0, 0, 05]$
T_i^{SC}, T_i^{R}	Initial trust value given for trust property X of node i uniformly selected from the given range where X = social connectedness or reciprocity	Input	$[0.5, 0.9]$
P_i^C	Initial trust value given for cooperativeness of node i uniformly selected from the given range	Input	$[0.8, 1.0]$
P_{cp}	Percentage of the number of nodes becoming compromised over time over all nodes	Input	25%
N	Total number of nodes in the network; each of the four types has $N/4$ nodes	Input	120
N_{th}	Maximum number of untrustworthy nodes tolerable for mission execution	Input	$\lceil N/3 \rceil$
ITR	Initial trust value range	Input	$[0.5, 1.0]$
CT	Compromise time	Input	$[0, 18]$ hrs
RBR	Risk behavior ratio, i.e., percentage breakup of risk-averse, risk-neutral, and risk-seeking nodes in the mission group	Input	(30%, 30%, 40%)
T_u	Trust update interval	Input	20 min.
T_m^{R-th}, T_m^{SC-th}	Trust threshold of task m for trust property $X = R, SC$	Input	$[0.5, 0.9]$
T_m^{I-th}	Trust threshold of task m for trust property $X = I$	Input	0.9
$s_{i,m}$	Score of a received bid	Derived	
$T_{i,j}^X(t)$	Subjective trust of node j evaluated by node i for trust property X at time t	Derived	
$T_j^X(t)$	Objective trust of node j for trust property X at time t	Derived	
$r_{m,j}(t)$	Average exposed risk level by employing node j for task m in terms of trust property X at time t	Derived	
P_m^{risk}	Acceptable risk level of task m, $e^{-\rho_2 I_m}$	Design	
α	Weight of direct evidence for trust evaluation where $0 < \alpha < 1$	Design	
γ	Trust decay factor in Equations (6) and 11	Design	
ϵ	Risk adjustment increment based on TL's risk behavior type	Design	



(a) Optimal ILP solutions

(b) COSTA-Risk simulation results

(c) Effect of probability distribution on performance with ITR [0.8, 1.0] and CT [3, 18] hrs

FIGURE 4. Effect of node initial trust range (ITR) and node compromising time (CT) on mission completion ratio with respect to average acceptable risk level.

Figures 4 (a)-(b) are the cases when trust values are selected based on uniform distribution for ITR. In Figure 4 (c), we analyze the sensitivity of COSTA-Risk results to the probability distribution. The results support the hypothesis that the trend identified is insensitive to the probability distribution function used. The reason that an optimal P_m^{risk} exists is due to the inherent trade-off between trust and risk. As we see from Figures 4 (a)-(c), a higher optimal P_m^{risk} is identified with more untrustworthy nodes while a lower optimal P_m^{risk} is identified with more trustworthy nodes. With a stringent P_m^{risk}, a task is more likely to fail due to not being able to recruit sufficient members for task execution in the initial task assignment period. On the other hand, with a relaxed P_m^{risk}, task leaders may be able to recruit sufficient members for task execution, but the task may fail due to recruiting more untrustworthy nodes.

D. EFFECT OF NODE RISK BEHAVIOR RATIO (RBR)

Risk Behavior Ratio (RBR) is the percentage breakup of risk-averse, risk-neutral, and risk-seeking nodes in the mission group. In Table 4, RBR = (30%, 30%, 40%) refers to 30% of the nodes are risk-averse, 30% of the nodes are risk-neutral, and 40% of the nodes are risk-seeking. Figure 5 shows the

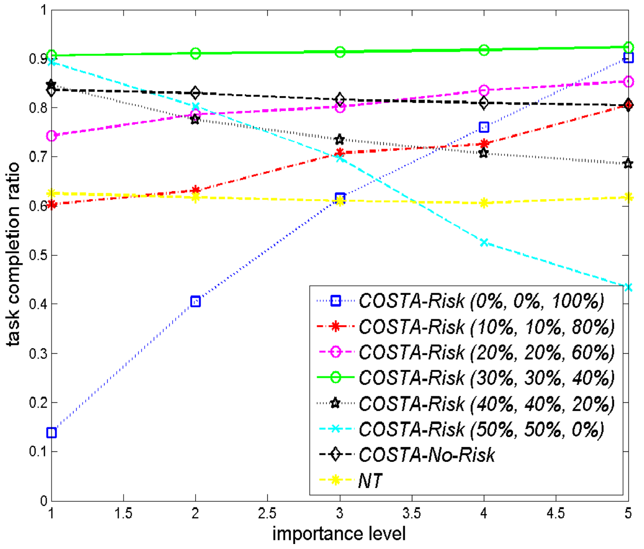


FIGURE 5. Effect of RBR on task completion ratio.

effect of RBR on the average task completion ratio in y-axis over all tasks having a particular importance level in x-axis.

Figure 5 indicates that COSTA performance is sensitive to RBR. When the node risk behavior is evenly distributed, e.g., RBR = (30%, 30%, 40%), COSTA-Risk performs the best in terms of the average task completion ratio for all task importance levels. This is so because COSTA-Risk is able to leverage risk behavior information to assign risk-averse nodes to low-importance tasks, risk-neutral nodes to medium-importance tasks, and risk-seeking nodes to high-importance tasks. Since the node population is evenly distributed among these three risk behavior types, every task regardless of its importance level will recruit enough nodes for task execution. We attribute the superior performance of COSTA-Risk (30%, 30%, 40%) over COSTA-No-Risk and NT to its ability to exploit the trade-off between trust and risk to maximize the average task completion ratio for all tasks in distinct importance levels.

However, when the node risk behavior population distribution is extremely skewed, e.g., RBR = (10%, 10%, 80%), COSTA-Risk does not necessarily perform better than COSTA-No-Risk. We see from Figure 5 when RBR = (10%, 10%, 80%) only high-importance tasks will have a high task completion ratio, whereas low-importance tasks will have a low task completion ratio because of a lack of risk-averse nodes (only 10% population) to execute low-importance tasks. We also see from Figure 5 that when RBR = (10%, 10%, 80%), even NT has a higher task completion ratio than that of COSTA-Risk for low-importance tasks with importance level equal to 1. Consequently, when RBR = (10%, 10%, 80%), COSTA-Risk may perform worse than COSTA-No-Risk or even NT in terms of the mission completion ratio P_{MC} since low-importance tasks do not have a high task completion ratio. A similar argument can be applied to other skewed risk behavior population distributions such as RBR = (0%, 0%, 100%) for which only high-importance tasks will have a high task completion ratio, or RBR = (50%, 50%, 0%) for which only low- and medium-importance tasks will have a high task completion ratio.

Figure 5 reveals that when given knowledge of RBR, one can decide the best COSTA protocol to maximize protocol performance in terms of the task completion ratio. Then given knowledge of mission composition (how many tasks and their importance levels) one can deduce the best COSTA protocol to maximize the mission completion ratio P_{MC}.

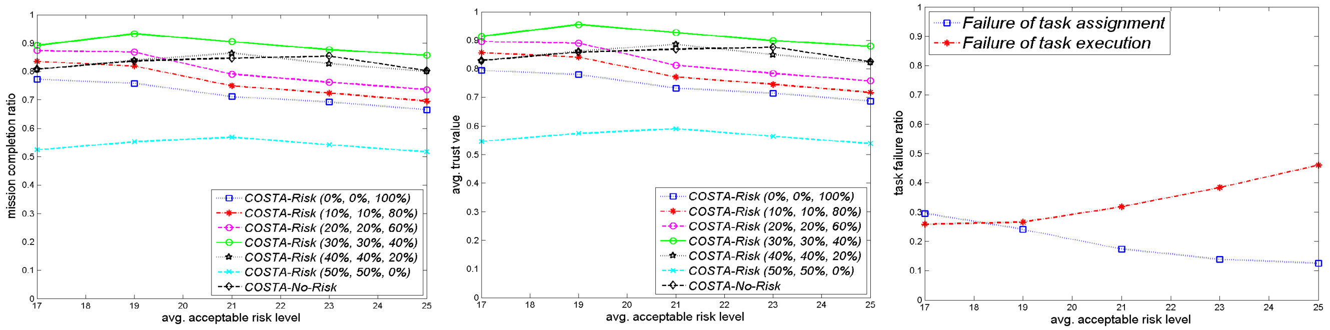
E. EFFECT OF ACCEPTABLE TASK RISK LEVEL (P_M^{RISK})

Figures 6 (a)-(c) analyze the sensitivity of the results with respect to the acceptable task risk level P_m^{risk}. Figure 6 (a) shows that there exists an optimal P_m^{risk} under which the mission completion ratio P_{MC} is maximized. This optimal P_m^{risk} value increases as there are fewer risk-seeking nodes. Specifically, the optimal P_m^{risk} values are 17, 19 and 21 when the percentages of risk-seeking nodes are 60-100%, 20-40%, and 0%, respectively. When there are many risk-seeking nodes accounting for 60-100% of node population, these risk-seeking nodes tend to select important tasks and this risk-seeking behavior leaves medium and low-importance tasks unfulfilled. The system is better off by allowing a smaller task risk level to discourage nodes to select only high-importance tasks. On the other hand, when risk-seeking nodes accounting for only 0-10% node population, high-importance tasks will be unfulfilled. In this case it is better off to allow a higher task risk level to encourage nodes to select high-importance tasks. Figure 6 (a) clearly indicates that the mission completion ratio P_{MC} is sensitive to the task risk level P_m^{risk}. Figure 6 (b) shows the average trust value of legitimate members in the network as the task acceptable risk threshold varies. We observe that the trend matches well with that in Figure 6 (a). That is, there exists an optimal P_m^{risk} that can maximize the mission completion ratio and, as a result, also maximize the average trust value of all legitimate nodes in the system.

Figure 6 (c) illustrates the inherent trade-off leading to the existence of an optimal acceptable risk level based on the two main failure types: task failure caused by the lack of members in task assignment (denoted as ''failure of task assignment'') vs. task failure caused by low trust levels of selected members (denoted as ''failure of task execution''). We observe that a task tends to fail due to a lack of members for task assignment under a more stringent (lower) P_m^{risk} because a stringent P_m^{risk} decreases the chance of recruiting sufficient members for task execution. However, a more stringent P_m^{risk} is less likely to cause the failure of task execution because selected members tend to be more trustworthy (qualified). On the other hand, a higher P_m^{risk} relaxes the member selection criteria, so a task may fail due to low trustworthiness of selected members.

F. COMPARISON OF COMMUNICATION OVERHEAD

Figure 7 breaks up the communication overhead incurred per event and also shows the total communication overhead



(a) Mission completion ratio (P_{MC}) vs. acceptable task risk level (P_m^{risk}) under varying RBR

(b) Average trust value for group members vs. acceptable task risk level (P_m^{risk}) under varying RBR

(c) Task failure ratio in COSTA-Risk vs. acceptable task risk level (P_m^{risk}) under ITR = [0.7,1] and trust bias = 3%

FIGURE 6. Effect of acceptable task risk level (P_m^{risk}).

(C_{total}) in the four schemes. Note that NT with intrusion detection system (IDS), denoted as NT-IDS, refers to NT that is capable of detecting compromised nodes using a distributed IDS [44] installed at each node. Thus, NT-IDS does not send any messages to compromised nodes. We denote NT without IDS as NT-No-IDS.

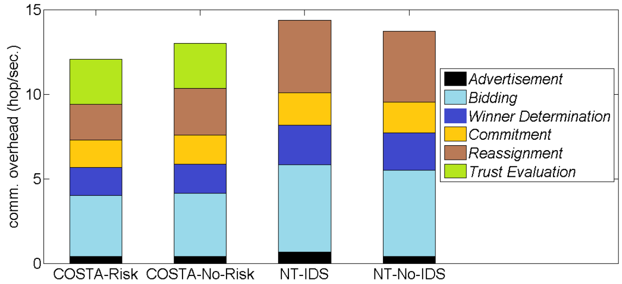


FIGURE 7. Comparison of communication overhead (C_{total}).

Figure 7 shows that non-trust based schemes, NT-IDS and NT-No-IDS, incur a higher overhead because they need to rerun the task assignment protocol more frequently than trust-based schemes (i.e., COSTA-Risk and COSTA-No-Risk). Notice that the reassignment cost in NT is high because unqualified members often need to be replaced in the middle of task execution. Trust-based task assignment schemes (i.e., COSTA-Risk and COSTA-No-Risk), on the other hand, incur a high overhead for running trust evaluation periodically. In Figure 7, COSTA-Risk and COSTA-No-Risk compare favorably in C_{total} with NT-IDS and NT-No-IDS because they reduce the reassignment cost by selecting well qualified members in the initial task assignment. COSTA-Risk has the lowest C_{total} among all because of its ability to exploit the trust-risk trade-off in member selection to maximize the mission completion ratio and minimize the reassignment cost.

VIII. CONCLUSIONS

In this work, we proposed task assignment protocol using the concept of multidimensional trust in choosing qualified member nodes that can maximize mission completion ratio while meeting an acceptable risk level. The composite trust metric takes into account the attributes of communication, information, and social networks. We considered a node's risk behavior and investigate its effect on the task completion ratio, and the optimal acceptable task risk level. Our simulation results validated with ILP solutions demonstrated that the COSTA-Risk (i.e., our proposed trust-based task assignment protocol with risk behavior) outperforms the non-trust based schemes (NT) as well as the counterpart trust-based protocol that does not consider risk behavior (COSTA-No-Risk), while incurring relatively low communication overhead. We identified the optimal acceptable risk level to best balance trust and risk to maximize mission completion ratio. Given knowledge of node risk behavior and node/task characteristics as input, a system designer can apply the optimal task risk level identified to maximize the mission completion ratio and the payoff in terms of trust to all legitimate nodes in the system.

In the future, we plan to investigate more sophisticated risk and payoff models for human based tasks where human psychology and crowd behaviors play very important roles. We also plan to investigate solution techniques to multiple-objective optimization problems based on the trade-off between trust and risk for task assignment in MANETs in which a system may have multiple conflicting objectives while nodes may have different objectives to maximize their own payoffs.

REFERENCES

[1] K. S. Cook, *Trust in Society*. Ed., Russell Sage Found. Ser. Trust,New York: NY USA: Feb. 2003.

[2] C. Castelfranchi and R. Falcone, *Trust Theory: A Socio-Cognitive and Computational Model*, Ed. Hoboken, NJ, USA: Wiley, 2010.

[3] J. Cho, K. Chan, and S. Adali, "A survey on trust modeling," *ACM Comput. Surv.*, vol. 48, no. 2, p. 28, Nov. 2015.

[4] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Feb. 1993.

[5] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *Proc. IEEE Symp. Secur. Privacy*, May 1996, pp. 164–173.

[6] J.-H. Cho, A. Swami, and I.-R. Chen, "Modeling and analysis of trust management with trust chain optimization in mobile ad Hoc networks," *J. Netw. Comput. Appl.*, vol. 35, no. 3, pp. 1001–1012, May 2011.

[7] Z. Movahedi, M. Nogueira, and G. Pujolle, "An autonomic knowledge monitoring scheme for trust management on mobile ad Hoc networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2012, pp. 1898–1903.

[8] Y. Wu, Y. Zhao, M. Riguidel, G. Wang, and P. Yi, "Security and trust management in opportunistic networks: A survey," *Secur. Commun. Netw.*, vol. 8, no. 9, pp. 1812–1827, Jun. 2015.

[9] H. Xia, Z. Jia, L. Ju, X. Li, and Y. Zhu, "A subjective trust management model with multiple decision factors for MANET based on AHP and fuzzy logic rules," in *Proc. IEEE/ACM Int. Conf. Green Comput. Commun.*, Apr. 2011, pp. 124–130.

[10] W. Li, A. Joshi, and T. Finin, "Coping with node misbehaviors in ad Hoc networks: A multi-dimensional trust management approach," in *Proc. 11th Int. Conf. Mobile Data Manage.*, May 2010, pp. 85–94.

[11] Z. M. H. Islam, and A. A. Khan, "Detection of dishonest trust recommendations in mobile ad Hoc networks," in *Proc. 15th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2014, pp. 1–7.

[12] P. B. Velloso, R. P. Laufer, D. D. O. O. Cunha, O. C. M. B. Duarte, and G. Pujolle, "Trust management in mobile ad Hoc networks using a scalable maturity-based model," *IEEE Trans. Netw. Service Manag.*, vol. 7, no. 3, pp. 172–185, Sep. 2010.

[13] H. Li and M. Singhal, "Trust management in distributed systems," *Computer*, vol. 40, no. 2, pp. 45–53, Feb. 2007.

[14] A. Jøsang and S. L. Presti, "Analyzing the relationship between risk and trust," in *Proc. Int. Conf. Trust Manage.*, Apr. 2004, pp. 135–145.

[15] B. Solhaug, D. Elgesem, and K. Stolen, "Why trust is not proportional to risk?" in *Proc. 2nd Int. Conf. Availability, Rel. Secur.*, Vienna, Austria, Apr. 2007, pp. 11–18.

[16] J. Cho, A. Swami, and I. Chen, "Mission-dependent trust management in heterogeneous military mobile ad Hoc networks," in *Proc. Int. Command Control Res. Technol. Symp.*, Jun. 2010, pp. 1–18.

[17] J.-H. Cho, A. Swami, and T. Cook, "Combinatorial auction-based multiple dynamic mission assignment," in *Proc. Mil. Commun. Conf. (MILCOM)*, Nov. 2011, pp. 1327–1332.

[18] Y. Wang, I. R. Chen, J. H. Cho, and J. J. P. Tsai, "Trust-based task assignment with multi-objective optimization in service-oriented ad Hoc networks," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 1, pp. 217–232, Mar. 2017.

[19] Y. Wang, I.-R. Chen, J.-H. Cho, A. Swami, and K. S. Chan, "Trust-based service composition and binding with multiple objective optimization in service-oriented mobile ad Hoc networks," *IEEE Trans. Services Comput.*, vol. 10, no. 4, pp. 660–672, Aug. 2017.

[20] D. H. Lee, "Resource-based task allocation for multi-robot systems," *Robot. Auto. Syst.*, vol. 103, pp. 151–161, May 2018.

[21] J. Schwarzrock, I. Zacarias, A. L. Bazzan, R. Q. D. A. Fernandes, L. H. Moreira, and E. P. de Freitas, "Solving task allocation problem in multi unmanned aerial vehicles systems using swarm intelligence," *Eng. Appl. Artif. Intell.*, vol. 72, pp. 10–20, Jun. 2018.

[22] A. T. Tolmidis and L. Petrou, "Multi-objective optimization for dynamic task allocation in a multi-robot system," *Eng. Appl. Artif. Intell.*, vol. 26, nos. 5–6, pp. 1458–1468, May 2013.

[23] J. Du, E. Gelenbe, C. Jiang, Z. Han, Y. Ren, and M. Guizani, "Cognitive data allocation for auction-based data transaction in mobile networks," in *Proc. IEEE Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Limassol, Cyprus, Jun. 2018, pp. 207–212.

[24] J. Du, C. Jiang, Z. Han, H. Zhang, S. Mumtaz, and Y. Ren, "Contract mechanism and performance analysis for data transaction in mobile social networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 6, no. 2, pp. 103–115, Apr. 2019.

[25] M. Asghari and C. Shahabi, "On on-line task assignment in spatial crowdsourcing," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 395–404.

[26] A. Whitbrook, Q. Meng, and P. W. H. Chung, "Reliable, distributed scheduling and rescheduling for time-critical, multiagent systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 732–747, Apr. 2018.

[27] T. Li, T. Jung, H. Li, L. Cao, W. Wang, X.-Y. Li, and Y. Wang, "Scalable privacy-preserving participant selection in mobile crowd sensing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2017, pp. 59–68.

[28] H. W. D. Chang and W. J. B. Oldham, "Dynamic task allocation models for large distributed computing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 6, no. 12, pp. 1301–1315, Dec. 1995.

[29] Y. Jiang and J. Jiang, "Contextual resource negotiation-based task allocation and load balancing in complex software systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 5, pp. 641–653, May 2009.

[30] M. P. Johnson, H. Rowaihy, D. Pizzocar, A. Bar-Noy, S. Chalmers, T. L. Porta, and A. Preece, "Sensor-mission assignment in constrained environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 11, pp. 1692–1705, Nov. 2010.

[31] Y. Jin, J. Jin, A. Gluhak, K. Moessner, and M. Palaniswami, "An intelligent task allocation scheme for multiple wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 3, pp. 444–451, Mar. 2012.

[32] J. Berg, J. Dickhaut, and K. McCabe, "Trust, Reciprocity, And social history," *Games Econ. Behav.*, vol. 10, no. 1, pp. 122–142, Jul. 1995.

[33] W. Haobo, L. Chunlin, Y. Chunxiang, L. Qingqing, and L. Jun, "Ad Hoc grid task scheduling algorithm considering trust-demand," in *Proc. 2nd Int. Conf. Future Comput. Commun.*, May 2010, pp. 109–113.

[34] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman key distribution extended to group communication," in *Proc. 3rd ACM Conf. Comput. Commun. Secur.*, New Delhi, India, Jan. 1996, pp. 31–37.

[35] J. Chavas, *Risk Analysis in Theory and Practice*. Amsterdam, The Netherlands: Elsevier, 2004.

[36] Ministry of Social Development. *The Social Report 2016*. [Online]. Available: http://socialreport.msd.govt.nz/documents/2016/msd-the-social-report-2016.pdf

[37] R. L. Trivers, "The evolution of reciprocal altruism," *Quart. Rev. Biol.*, vol. 46, no. 1, pp. 35–57, Mar. 1971.

[38] I. R. Chen, F. Bao, M. Chang, and J.-H. Cho, "Integrated social and QoS trust-based routing in delay tolerant networks," *Wireless Pers. Commun.*, vol. 66, no. 2, pp. 443–459, Sep. 2012.

[39] N. Nisan, T. Roughgarden, E. Targos, and V. V. Vazirani, *Algorithmic Game Theory*. Cambridge, U.K.: University Press, Sep. 2007.

[40] M. S. Lund, B. Solhaug, and K. Stølen, "Evolution in relation to risk and trust management," *Computer*, vol. 43, no. 5, pp. 49–50, May 2010.

[41] H.-Y. Tsai and Y.-L. Huang, "An analytic hierarchy process-based risk assessment method for wireless networks," *IEEE Trans. Rel.*, vol. 60, no. 4, pp. 801–816, Dec. 2011.

[42] M. R. Garey and D. Johnson, *Comput. Intractability: A Guide to Theory NP-Completeness*. W.H. San Francisco, CA, USA: Freeman Company 1979.

[43] M. H. MacDougall, *Simulating Computer Systems, Computer Systems Series*. The MIT Press, 1987.

[44] R. Mitchell and I.-R. Chen, "A survey of intrusion detection in wireless network applications," *Comput. Commun.*, vol. 42, no. 3, pp. 1–23, Apr. 2014.

JIN-HEE CHO received the M.S. and Ph.D. degrees in computer science from Virginia Tech, in 2004 and 2008, respectively. Prior to joining Virginia Tech, she has been a Computer Scientist with the U.S. Army Research Laboratory (USARL), Adelphi, MD, USA, since 2009. She has been an Associate Professor with the Department of Computer Science, Virginia Tech, since 2018. She has published over 120 peer-reviewed technical papers in leading journals and conferences in the areas of trust management, cybersecurity, metrics and measurements, network performance analysis, resource allocation, agent-based modeling, uncertainty reasoning and analysis, information fusion/credibility, and social network analysis. She is a member of the ACM. She received the Best Paper Awards from the IEEE TrustCom 2009, BRIMS 2013, the IEEE GLOBECOM 2017, and the IEEE CogSima 2018 and the 2017 ARL's Publication Award. She was a recipient of the 2015 IEEE Communications Society William R. Bennett Prize in the field of communications networking. She was selected for the 2013 Presidential Early Career Award for Scientists and Engineers (PECASE), in 2016.

HAMID AL-HAMADI received the B.S. degree in information technology from Griffith University, Brisbane, QLD, Australia, in 2003, the M.S. degree in information technology from the Queensland University of Technology, Brisbane, in 2005, and the Ph.D. degree in computer science from the Virginia Polytechnic Institute and State University, VA, USA, in 2014. He has worked for Universe Computers as an on-site Network Engineer at Kuwait National Petroleum Company. He was a Core Network Engineer with Tawasul Telecom, Kuwait. He is currently an Assistant Professor with the Department of Computer Science, Kuwait University, Kuwait. His current research interests include the Internet of Things, security, mobile cloud, trust management, and reliability and performance analysis.

ING-RAY CHEN received the B.S. degree from National Taiwan University, Taiwan, and the M.S. and Ph.D. degrees in computer science from the University of Houston, USA. He is currently a Professor with the Department of Computer Science, Virginia Tech. His research interests include mobile computing, wireless systems, security, trust management, and reliability and performance analysis. He was a recipient of the IEEE Communications Society William R. Bennett Prize in the field of communications networking. He serves as an Editor for the IEEE Transactions on Services Computing, the IEEE Transactions on Network and Service Management, and *The Computer Journal*.