17th ICCRTS

"Operating Agility"

Title of Paper:
Trust-based Task Assignment in Military Tactical Networks

Topic(s):
8-Networks and Networking

Name of Author(s):

MoonJeong Chang, Ph.D.
Virginia Tech– Department of Computer Science
mjjang@vt.edu

Jin-Hee Cho, Ph.D.
Army Research Laboratory – CISD
jinhee.cho@us.army.mil

Ing-Ray Chen, Ph.D.
Virginia Tech – Department of Computer Science
irchen@vt.edu

Kevin S. Chan, Ph.D.
Army Research Laboratory – CISD
kevin.s.chan@us.army.mil

Ananthram Swami, Ph.D.
Army Research Laboratory – CISD
ananthram. swami@us.army.mil

Point of Contact:
MoonJeong Chang, Ph.D.
Virginia Tech– Department of Computer Science
mjjang@vt.edu

# Trust-based Task Assignment in Military Tactical Networks


## Abstract

Resource or task assignment problems have been studied extensively in military network environments as efficient and effective resource allocation is the key to successful mission completion. However, existing works on asset-task assignment problems are based only on the best match between functionalities of a node and requirements of a task. We propose a composite trust based task assignment protocol where a task requires all members to have a specific trust level for successful mission completion while allowing each member to maximize its utility. The proposed protocol explores the tradeoff between trust and risk, and selects a best set of members for each task based on task specific trust requirements. We show that the proposed composite trust based task assignment protocol outperforms the non-trust based counterpart in mission success ratio. Further, there exists an optimal acceptable risk level that can best balance meeting the condition of selecting a sufficient number of members for task execution while preserving qualified trustworthy nodes as task members to maximize the mission completion ratio.



**Key words –** trust, risk, composite trust metric, task assignment, tactical network, mission completion, task


## I. Introduction

A military tactical network is often assigned a mission consisting of multiple tasks in which effective and efficient asset-task assignment is critical to successful mission completion. We consider a mission group composed of heterogeneous entities executing multiple tasks under a common mission in a wireless tactical network. Example missions may include rescuing personnel in battlefield situations, constructing military facilities, conducting surveillance or monitoring, destroying certain targets, or managing disasters. The mechanism to assign these entities, the so called "assets," to tasks under the common mission significantly affects mission performance.

We take a soft security approach by introducing the concept of "trust" to solve this task assignment problem in a hostile environment. The concept of trust originally derives from social sciences and is defined as the degree of subjective belief about the behaviors of a particular entity [11]. According to *Merriam Webster*'s Dictionary [26], trust is defined as "assured reliance on the character, ability, strength, or truth of someone or something." Blaze et al. [2] first introduced the term trust management and identified it as a separate component of security services in networks. Since its inception, trust management has received considerable attention due to its crucial necessity and diverse applicability in decision making processes. We interpret our "task assignment" problem as a decision making process based on trust.

Trust is defined differently based on the application domain [3]. However, we find a common definition of trust applicable across domains: willingness to take a risk. Cho et al. [6] discussed the key characteristics of a desirable trust metric in tactical networks as : (1) trust should be measured based on potential risks; (2) trust should be context-dependent; (3) trust should be based on each party's own interest; (4) trust is learned (i.e., a cognitive process); and (5) trust may represent system reliability. Our trust metric and task assignment protocol fully addresses these features with the emphasis on "the tradeoff between trust and risk" and "context-dependent trust."

Josang et al. [15] and Solhaug et al. [23] have discussed the relationship between trust and risk. Risk tends to be low under high trust. However, the risk level should be determined based on the event probability because high risk exists even in the case of extremely high trust. They also discussed the important aspects of opportunity and prospect (i.e., a positive consequence caused by the opportunity). That is, if we do not take a risk, there would also be no gain. However, if the gain obtained by taking a risk is trivial, we do not take the risk.

We propose a composite trust based task assignment mechanism that assigns the right entities to the right task to maximize performance while maintaining an acceptable risk. Depending on the capability of each node, a node may perform multiple tasks during its lifetime to maximize its utilization. Each node is incentivized to maintain its trust level while participating in the execution of tasks more actively during its lifetime.

Task assignment problems have been studied extensively in various domains. Chang et al. [4] and Jiang et al. [12] studied dynamic task or resource allocation problems in computing systems. Chang et al. [4] examined a dynamic task allocation problem in a large distributed computing system for maximizing system throughput based on the tradeoff between resource utilization and communication overhead. Jiang et al. [12] proposed a contextual resource-based task allocation method with load balancing in software systems where the software has different computable functions and requires different resources. These studies [4], [12] dealt with task assignment problems in a computing system, so are not directly applicable in wireless networks. Choi et al. [9] and Choudhury et al. [10] investigated market-based task allocation algorithms for multi-robot systems. Choi et al. [9] proposed a consensus-based bundle algorithm for rapid conflict-free matching between tasks and robots. Choudhury et al. [10] conducted an empirical study on a robot-task assignment based on a robot's capability and tasks' similarity to maximize task performance.

Task assignment problems have been studied in wireless sensor networks (WSNs) [1], [17], [22], [14]. AbdelSalam et al. [1] proposed an energy-aware task management protocol to maximize network lifetime by optimizing task load among team members in WSNs. Le et al. [17] proposed a multi-round Knapsack algorithm for a sensor-task assignment problem. Rowaihy et al. [22] proposed centralized and distributed mission assignment algorithms based on heuristic strategies to maximize profits of missions under the contention between multiple simultaneous missions. However, the above task assignment studies in WSNs [1], [17], [22] only dealt with static tasks where a node executes one task during its lifetime. Johnson et al. [14] proposed sensor-mission assignment algorithms to conserve energy where missions can be static (i.e., all missions start and end at the same time) or dynamic (i.e., missions arrive and end at different times). However, this work [14] also assumes that a node performs only one task during its lifetime.

Task assignment problems have been studied in other network environments. Jin et al. [13] used genetic algorithms to solve a task allocation and scheduling problem based on the tradeoff between energy consumption and task completion in multi-hop wireless networks. Similarly, Kulkarni et al. [16] studied a task allocation problem to maximize mission completion probability while maximizing energy efficiency in autonomous underwater vehicle networks, given a stringent mission deadline. Again, these works [13], [16] only dealt with static missions with one entity-one task assignment.

Our recent works [7], [8] investigated a mission assignment protocol in mobile ad hoc networks (MANETs). In [7], we proposed a mission-dependent trust management protocol based on context-dependent trust characteristics to maximize mission success probability in military MANETs. We examined how to select group members for a mission that requires the participation of a subset of the entire group. However, the issues of multiple dynamic missions and risk analysis were not addressed. In [8], we employed a combinatorial auction algorithm to solve a multiple task mission assignment problem in MANETs and analyzed the advantage of the proposed auction algorithm in communication overhead and mission completion probability.

Unlike the existing works discussed above, we introduce the concept of trust in the task assignment problem. We model dynamic missions and assume that entities including task leaders and members make decisions to achieve their own goals. In addition, based on the inherent rationale that "trust is the willingness to take a risk," we analyze how much risk can be tolerated to maximize mission performance in the presence of attacks in a heterogeneous wireless network. Two factors affect mission effectiveness: (1) how many tasks are assigned and executed by a sufficient number of members; and (2) how well the selected members perform to complete the task. A task fails when the task is not assigned with a sufficient number of members during assignment, or the task is not executed to completion after assignment.

The contributions of this work are as follows. First, we employ the concept of trust and propose a task assignment protocol based on the tradeoff between trust and risk. By accepting a certain level of risk, we facilitate assignments of more tasks to entities; on the other hand, by best matching trust vs. risk between entities and tasks, we control the overall risk level. Second, our approach reflects context-dependent trust characteristics by assigning the right entity to the right task. We introduce the concept of "task-dependent" trust, i.e., an entity is evaluated based on the requirements of the task. Third, we use the so called "composite trust metric" to assess trust of entities by considering diverse dimensions of trust derived from communication and social networks. Fourth, unlike most existing works that only consider one task performed by one entity, our task assignment protocol enables an entity to be assigned dynamically to multiple tasks during its lifetime as would be the case in military tactical networks.

The rest of this paper is organized as follows. Section 2 discusses the system model including the mobility model, node behavior model, and task model. Section 3 explains our composite trust metric. Section 4 describes our proposed task assignment protocol. Section 5 discusses the metrics used in this study and reports simulation results. Section 6 concludes the paper and suggests future work directions.

## II. System Model

We consider a tactical military wireless network where nodes communicate through multiple hops. The group comprises heterogeneous entities[1] with vastly different functionalities (e.g., sensing, destroying, monitoring) and natures (i.e., humans or machines). For example, the entities may be sensors, robots, unmanned vehicles or other devices, humans (e.g., dismounted soldiers) carrying sensors or handheld devices, and manned vehicles with various types of equipments. A mission consists of multiple tasks where each task is considered to be the fundamental unit. Each task has its own start and end times. Thus, some tasks may be executed concurrently while others may not.

Leveraging the existing hierarchy of military structures, a commander node (CN) governs the mission group. Under the CN, multiple task leaders (TLs) lead task teams. The CN selects TLs based on trustworthiness[2] and the TLs recruit members for task execution. When a TL leaves or is disconnected from the network voluntarily or involuntarily, and cannot continue to lead the task team, the CN selects a new TL among available members based on trustworthiness. A symmetric key, as a group key, is used for communications among members to prevent outside attackers. The CN, acting as a trusted authority (TA), takes care of key management, including key generation, distribution, revocation, and update.

Upon a node's disconnection from the mission group, the TA generates and redistributes a new key so that non-member nodes will not be able to access a valid group key. Nevertheless, each group member keeps old trust information even for non-member nodes so that the information can be reused for future interactions. This is useful to cope with potential newcomer attackers who wash out their low trust levels by frequently rejoining the group. In addition, even if a node leaves the group, as it rejoins the group, it keeps old trust information of other nodes to update their trust values.

In the initial network deployment, we assume that there is no predefined trust. Without prior interactions, the initial bootstrapping will establish a shallow level of trust based only on limited direct observations, indirect information through third parties, and authentication by a challenge/response process. Over time, participating nodes will establish a stronger trust level with more confidence based on direct or indirect interactions. Our trust management protocol allows each node to evaluate the trust levels of other nodes as well as to be evaluated by other nodes. Trust decays over time without further updates or interactions between entities. Node mobility also may hinder continuous interactions with other group members, lowering the chances of evaluations of each other in the group. This includes cases such as a node moving to other areas causing its

[1] We use the terms node, entity, and group member interchangeably. Note that a group member is a member of a mission group while a task member is a member of a task team where the mission group consists of multiple task teams.

[2] We use the terms trustworthiness and trust interchangeably because the proposed trust scheme is based on the evaluation of evidences only, not beliefs, feelings, or cognition.

disconnection from the current group, leaving a group, voluntary disconnection for saving power or involuntary disconnection due to physical location or terrain. On the other hand, node mobility could enhance trust evaluation of distant nodes.

## 2.1 Mobility Model and Node Deployment

This work considers both stationary entities such as sensors and mobile entities such as humans, robots, or vehicles. Initially nodes are randomly distributed over the operational area. Before an entity is assigned to a particular task, it will follow a random mobility pattern. However, after it is assigned to a particular task, it will move towards the TL's location for easy communications among members. When a node switches from one task to another, it moves towards the location of the next TL. While a node is executing a task, it will move around within the area in which the TL resides. In addition, a node may voluntarily leave and join the group with rates $\lambda$ and $\mu$. Each node is assumed to have capability to monitor its neighboring nodes, with the inaccuracy modeled by false positive and false negative probabilities. Nodes are heterogeneous with different speed, monitoring capability, and cooperation probabilities (i.e., packet dropping). A node may be compromised with a certain rate. The various assumptions regarding distributions, made below, are not required by the protocol. These will be invoked only in the numerical evaluation section so as to obtain insights on how heterogeneity affects performance. In summary, we model a node with the following parameters:

- **Speed ($\mathbf{v_i}$)**: A node is assigned a speed for its lifetime.
- **Detection error ($\mathbf{P_i^{fp}/P_i^{fn}}$)**: A node has inaccuracy in its monitoring capability, characterized by a false positive probability and a false negative probability.
- **Group join and leave ($\boldsymbol{\lambda}$ / $\boldsymbol{\mu}$):** A node may leave or join a group where the inter-arrival time of the events is exponentially distributed.
- **Cooperativeness ($\mathbf{P_i^{coop}}$):** A node may drop a packet based on its inherent characteristics of cooperativeness.
- **Node compromise ($\mathbf{1/T_i^{comp}}$):** A node may be compromised with a certain rate, $1/\mathrm{T_i^{comp}}$ where $\mathrm{T_i^{comp}}$ is selected from $[\mathrm{MIN\_T_i^{comp}}, \mathrm{MAX\_T_i^{comp}}]$. After other peers detect that a node has been compromised based on the agreement of more than a certain number of peer nodes, the node's trust is revoked, causing a sudden drop of all trust property values to zero.

## 2.2 Node Types

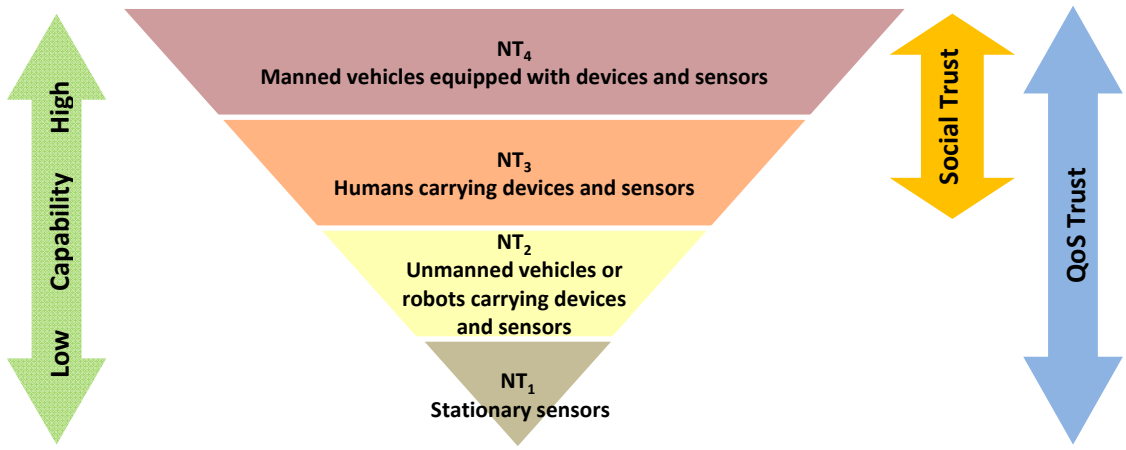


**Figure 1: Description of Node Types.**

We consider 4 types of nodes where higher type nodes have higher and more versatile capabilities than lower type nodes. For example, node type 4, $NT_4$, has higher capability than node type 1, 2, or 3, denoted as $NT_1$, $NT_2$,

and $NT_3$. $NT_3$ has higher capability than $NT_1$ and $NT_2$, but not $NT_4$. We model that two node types ($NT_3$, $NT_4$) possess capabilities humans have only such as social trust while the other two node types ($NT_1$, $NT_2$) possess capabilities both humans and machines have in common such as quality-of-service (QoS) trust. Figure 1 summarizes the conceptual description of node type modeled in this work.

We consider both uncooperative and malicious nodes. An uncooperative node acts for its own interest. So it may drop packets arbitrarily just to save energy. A malicious node may seek to disrupt task execution, so it can drop packets, jam the wireless channel, perform good or bad mouthing attacks (i.e., provide negative recommendations against good nodes or positive recommendations for other colluding malicious nodes) and even forge packets. The selfish and malicious behaviors of a node are evaluated based on trust values in availability and integrity respectively.

### 2.3 Task Modeling

We consider the scenario that a mission is given to a mission group. A mission consists of multiple tasks. Each "dynamic" task may start and end at different times, with the task duration of task m denoted as $\mathrm{DT_m}$. Each TL has criteria to recruit the right members for the task execution in terms of required functionality (i.e., a minimum node type such as minimum $NT_2$ meaning that a node with a node type equal to or above $NT_2$ is eligible) and trust level in each trust property X. We will discuss our proposed composite trust metric in Section 3.

More specifically, each task has unique and common task properties. The ***unique task properties*** include (1) a minimum required node type; and (2) a minimum trust threshold for each trust property X of task m denoted as $\mathrm{T_m^{X-th}}$. In addition, all tasks have common characteristics. We model the ***common task properties*** as mutually independent. The common task properties considered are: importance, urgency, and difficulty.

- **Importance ($\mathbf{I_m}$)**: This property refers to the impact of a task failure on mission completion. Some tasks are more important than other tasks to the completion of a given mission. The importance value of task m, $\mathrm{I_m}$, is represented as an integer in five categories (i.e., 5-1) where a higher number indicates higher importance: very high, high, medium, low, and very low.
- **Urgency ($\mathbf{U_m}$)**: This property indicates how urgently a given task should be completed. Due to the dynamics of the network, some tasks may need to adjust their completion time as long as there is no schedule conflict upon the availability of existing members. However, depending on the degree of task urgency, a task only can be extended with limited flexibility. Similar to importance values, urgency is modeled with five categories (i.e., 5-1) where a higher number indicates higher urgency. The allowed extension time is applied as 20%, 40%, 60%, 80%, and 100% of the original completion time corresponding to 5, 4, 3, 2, and 1 in the urgency value.
- **Difficulty ($\mathbf{D_m}$)**: This property represents the degree of difficulty of task execution. We associate this property with the amount of required workload. This affects the minimum number of members in the worst case and correspondingly the maximum possible workload per time unit to be assigned to each member. The difficulty value of task m, $\mathrm{D_m}$, is an integer representing the five levels of difficulty (i.e., 5-1) similar to importance and urgency. The amount of workload assigned to task m is computed as $\mathrm{D_m W}$ where W represents the workload per unit of difficulty.

In the next section, we employ urgency and difficulty values in computing the risk of each task, and importance in computing the mission completion ratio. Since risk affects mission completion, the three task property values above eventually affect mission performance.

## III. Composite Trust Metric

Our proposed trust metric spans two types of trust: social trust and QoS trust. **Social trust** can be derived from human relationships such as friendship, familiarity, intimacy, honesty, or centrality (betweenness). Castelfranchi and Falcone [3] discussed that social relationships can enable high productivity. In our scenario,

we consider a task requiring social trust aspect of a human entity in terms of relationships with other human entities. We use social connectedness and reciprocity as property of social trust. They are defined as:

- **Social Connectedness (SC)**: This describes the number of social connections in our social circle [27]. We associate this with the number of other nodes each entity encounters during a certain period of time. Two factors may affect this trust property: mobility pattern and inherent sociability.
- **Reciprocity (R)**: This is the degree of mutual giving and receiving [24]. That is, when a favor is received, an entity tends to return something for the past favor. The extent of the reciprocity can be interpreted as how long an entity tends to return or how much the entity can return the past favor in relation with the net gain the entity may have. The inherent emotional status of a node (e.g., willingness to reciprocate) and the amount of expected future net gain by returning the past favor may affect this trust.

**QoS trust** can be evaluated in both humans and machines. The examples are competence, availability, dependability, or reliability that mainly accounts for capability of an entity. We use competence and integrity to measure QoS trust. We define them by:

- **Competence (C)**: This refers to an entity's capability to serve the received request, leading to service availability. Competence may be affected by: (1) unintentional availability due to network or node conditions (e.g., failure and involuntary disconnections); and (2) intentional nature of an entity such as cooperativeness or willingness to serve.
- **Integrity (I)**: This is the honesty of an entity in terms of attack behaviors. We categorize integrity as QoS trust because we consider network attack behaviors such as selfishness or maliciousness which can be observed in both humans and machines.

### 3.1 Objective Trust

We model the ground truth trust values of each node, the so called "objective trust," based on inherent behavior seeds. The objective trust values are used to evaluate the accuracy of measured trust. Note that trust value scales between 0 and 1 as a real number in this work. Objective ***social connectedness*** **trust** of node j is based on node j's inherent sociability ($\mathrm{P_j^{SC}}$) in the range of [0, 1], and the relative largeness of the number of encountered nodes by:

$$\text{if (node j is a member)}, \mathrm{T_j^{SC}(t) = P_j^{SC} N_j^{enc} c;\ \text{else}\ T_j^{SC}(t) = 0} \tag{1}$$

$\mathrm{N_j^{enc}}$ is the number of nodes node j encounters during a trust update interval and c is a constant parameter to normalize $\mathrm{N_j^{enc}}$.

Objective ***reciprocity trust*** of node j is based on a given initial seed behavioral relationship ($\mathrm{P_j^R}$) in the range of [0, 1]. We assume that reciprocity of node i evaluated by node j can be evaluated based on reciprocity of node j evaluated by node i because reciprocity is based on mutual interactions. That is, it is assumed that if node i receives favors from node j, node i is more likely to return favors. The objective reciprocity trust of node j is given by:

$$\text{if (node j is a member)}, \mathrm{T_j^R(t) = P_j^R;\ \text{else}\ T_j^R(t) = 0} \tag{2}$$

Objective ***competence trust*** of node j is computed based on node j's inherent cooperativeness ($\mathrm{P_j^{coop}}$) in the range of $[\mathrm{GB_{min}},\ 1]$ and the link reliability based on network or node conditions ($\mathrm{P_{link}}$) as:

$$\text{if (node j is a member)}, \mathrm{T_j^C(t) = P_j^{coop} P_{link}; \text{else}\ T_j^C(t) = 0;} \tag{3}$$

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

$$\text{if (node j's trust is not revoked)}, T_j^I(t) = 1; \text{ else } T_j^I(t) = 0; \quad (4)$$

As discussed in Section 1, we define that node j is compromised when a certain number of member nodes declare that node j is compromised.

### 3.2 Subjective Trust

Trust measured by each node towards other nodes, the so called "subjective trust," is based on peer-to-peer trust evaluation [5]. The peer-to-peer trust evaluation is periodically updated based on either direct observations or indirect information. When two entities are neighbors within wireless radio range (R), they evaluate each other based on direct observations using pre-installed monitoring mechanisms. The peer-to-peer evaluation is performed between nodes except the CN. Only the CN receives trust evaluation information from all TLs and uses the average trust value to evaluate each TL. The CN will use them for the selection of a new TL when the current TL is detected as untrustworthy, or abandons the task.

The trust value that node i evaluates towards node j in trust property X at time t, $T_{i,j}^X(t)$, is represented as a real number in the range of [0, 1] where 1 indicates complete trust, 0.5 ignorance, and 0 distrust. 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:

$$T_{i,j}^X(t) = \alpha T_{i,j}^{D-X}(t) + (1-\alpha) T_{i,j}^{ID-X}(t) \text{ where } 0 < \alpha < 1 \quad (5)$$

$T_{i,j}^X(t)$ is based on both direct trust evidences, $T_{i,j}^{D-X}(t)$, (i.e., node i's direct observations or experiences) and indirect trust evidences, $T_{i,j}^{ID-X}(t)$, such as recommendations from third parties. The recommendations will be received from node i's 1-hop neighbors. A parameter α is used here to weigh these two trust evidences. A larger α means that trust evaluation will rely more on direct observations. We follow our prior work [5] to determine the optimal α under which subjective trust values generated are the most accurate compared to ground truth trust values. This work uses an optimal α to study the proposed task assignment protocol.

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

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

$HD(i,j)$ is the number of hop distances between nodes i and j. Thus, when nodes i and j are encountered as 1-hop neighbors during the time period $\Delta t$, node i can collect direct trust evidences based on its own observations or experiences. $\Delta t$ represents the periodic trust update interval. When nodes i and j are more than 1 hop away, node i relies on its past trust experience with node j to update $T_{i,j}^{D-X}(t)$ with the decay factor γ.

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

$$T_{i,j}^{ID-X}(t) = \begin{cases} \dfrac{\sum_{k \in R_i^{trw}} T_{k,j}^X(t)}{|R_i^{trw}|} & \text{if } |R_i^{trw}| > 0 \\ \gamma T_{i,j}^X(t-\Delta t) & \text{otherwise} \end{cases} \quad (7)$$

$R_i^{trw}$ is the set of 1-hop neighbors of node i providing trustworthy recommendations towards node j. That is, when node i receives recommendations from its 1-hop neighbors, it only accepts those recommenders whose trust is not revoked. It calculates $T_{i,j}^{ID-X}(t)$ with as the average of trustworthy recommendations. If $R_i^{trw}$ is an empty set, node i will use its past experience $\gamma T_{i,j}^X(t-\Delta t)$ with a decay factor, γ, considered due to no

recommendations available.

## IV. Task Assignment Protocol

The proposed task assignment protocol is basically based on a single item auction with multiple preferences [21] by which each bidder can bid on multiple items and select one in the end. TLs are auctioneers and members are bidders. TLs advertise the specifications of the items dynamically upon the availability of tasks (items) and group members (bidders) will bid on any interested item based on the valuation of the item and the price they should pay. TLs will process the bids received and select the winners that give the most promising benefit to the mission. We examine how auctioneers and bidders can make decisions based on trust and identified risk. Since we consider dynamic tasks where tasks arrive and end at different times, the auction process of a task will be dynamically conducted upon the receipt of the task assigned by the CN to each TL. The CN assigns a task to one TL and each TL looks for its members based on certain criteria. Now we describe the specifics of task assignment.

### 4.1 Advertisement of Task Specification

Each TL advertises its task specification including a set of requirements for task execution to all members in the network. The specification includes:

$$[\mathrm{ID_m}, \mathrm{L_m}, \mathrm{NT_{exc}^{min}}, \mathrm{T_m^{exc}}, \mathrm{W_m}, \mathrm{T_{m,i}^{reward/penalty}}] \tag{8}$$

$\mathrm{ID_m}$ is the identification (ID) of task m, $\mathrm{L_m}$ is the location for the task m (i.e., location of the task leader), $\mathrm{NT_{exc}^{min}}$ is the minimum required node type, $\mathrm{T_m^{exc}}$ indicates the start and end time of task m, and $\mathrm{W_m}$ is the maximum required workload per time unit per member to execute task m (e.g., the amount of packets to send/process). For the estimation of $\mathrm{W_m}$, each TL computes the maximum possible workload per time unit based on the minimum required number of members ($\mathrm{N_m^{min}}$) to complete task m. Thus, a TL may want to issue more winner notifications than $\mathrm{N_m^{min}}$ because it does not burden members with the maximum workload and some members may have their contract terminated due to their misbehaviors or unavailability. $\mathrm{T_{m,i}^{reward/penalty}}$ is reward or penalty depending on whether a node completes a task successfully or not. TL of task m will apply the reward or penalty in each trust component's trust value of node i (i.e., competence, integrity, reciprocity, and social connectedness) and is computed by:

$$\mathrm{T_{m,i}^{reward}} = \mathrm{T_{m,i}^{penalty}} = \varphi \times \frac{\mathrm{I_m}}{\mathrm{I_m^{max}}} \tag{9}$$

$\varphi$ is a constant to range the reward or penalty.

### 4.2 Bidding

After a node receives the task specifications from TLs, a node bids on some tasks that fit its availability and qualification (i.e., node type). Recall that a node may perform multiple tasks during its lifetime, so it should not have any schedule conflict that may affect the performance of another task execution. Each node makes a decision on a task to bid based on the net gain from the possible contract. Node i's net gain by performing task m ($\mathrm{s_{i,m}}$), the so called "score," can be computed as:

$$\mathrm{s_{i,m}} = \mathrm{v_m} - \mathrm{p_{i,m}} \text{ where } \mathrm{v_m} = \frac{\mathrm{DT_m}}{\mathrm{DT_{max}}} \text{ and } \mathrm{p_{i,m}} = \frac{\mathrm{W_m}}{\mathrm{w_i}} \tag{10}$$

$\mathrm{v_{i,m}}$ is the valuation of executing task m by node i and $\mathrm{p_{i,m}}$ is the "price" node i should pay to execute task m. $\mathrm{DT_m}$ is the duration of task m, and $\mathrm{DT_{max}}$ is the maximum duration among all tasks. $\mathrm{v_{i,m}}$ is computed based on the relative degree of task duration. We assume that a node prefers tasks with longer duration because it may

have privileges to access resources and chances to maintain a high trust level by continuous active interactions with other nodes. $p_{i,m}$ is estimated 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 is beyond the capability node i can handle per time unit. Note that a node only bids on positive net gains and may apply for bids on multiple tasks (i.e., multiple preferences) to have the chance to be assigned to a certain task.

### 4.3 Winner Determination based on Risk Analysis

A TL may receive multiple bids from multiple entities. The TL determines winners based on certain criteria. The criteria are designed to screen right entities while meeting an acceptable risk level. Each task requires a selected entity to maintain a certain level of trust per trust property X during task execution while not causing the task to fall below an acceptable risk level. The risk probability $r^X_{m,j}(t)$ when node j is selected to execute task m or is currently executing task m at time t is calculated by:

$$r^X_{m,j}(t) = e^{-\rho_1 \frac{T^X_{i(m),j}(t)}{T^{X-th}_m}} \frac{U_m}{U^{max}_m} \frac{D_m}{D^{max}_m} \tag{11}$$

$T^{X-th}_m$ is the minimum trust threshold for an entity to execute task m without increasing the risk level above task m'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" of task m (discussed in Section 2.3). $\rho_1$ is a constant design parameter that can be determined based on $P^{risk}_m$ to guarantee that the acceptable risk level is less than $P^{risk}_m$ if node j's trust evaluated by TL i(m) (i.e., node i as the TL for task m), $T^X_{i(m),j}$, is at least $T^{X-th}_m$. $U^{max}_m$ is the maximum task urgency among all tasks and $D^{max}_m$ is the maximum difficulty among all tasks. The acceptable risk threshold for task m, $P^{risk}_m$, is obtained by:

$$P^{risk}_m = e^{-\rho_2 I_m} \tag{12}$$

$I_m$ is the task importance of task m, and $\rho_2$ is a constant parameter to normalize $P^{risk}_m$ [18], [25].

Based on $r^X_{m,j}(t)$ for each trust property X, the TL of task m is able to obtain the risk probability when node j is selected as its member, calculated by the average risk probability among all trust properties as follows:

$$r_{m,j}(t) = \frac{\sum_{X \in T} r^X_{m,j}(t)}{|T|} \tag{13}$$

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

A TL aims to maximize task completion ratio while meeting an acceptable risk level to the task. Hence, the objective function of a TL for task m is formulated as:

$$\text{maximize } P^{completion}_m(t), \qquad \text{given} \sum_{j \in M} r_{m,j}(t) \le P^{risk}_m \tag{14}$$

$P^{completion}_m(t)$ is the completion probability of task m at time t. M is the set of task members assigned to task m. $P^{completion}_m(t)$ can be binary, 0 or 1, based on if task m is completed within the mission time.

### 4.4 Winner Notification and Node Commitment

After reviewing the qualifications of bidding nodes and analyzing the potential risk level, TLs determine winners and notify them of the acceptance as task members. When each node receives an acceptance notification, it chooses the task with the highest score based on Equation 10.

When the node decides to commit itself to a certain task, it notifies the TL of the chosen task. Note that each node can choose one task so as not to cause any schedule conflict. If there are multiple advertisement messages from multiple TLs, this means that these tasks tend to execute concurrently. After a TL receives the commitment notice from a member, a contract is made between the TL and the member node. If the contract is terminated due to the node's misbehavior or unavailability, the node will be eliminated from the task membership and the TL needs to trigger dynamic task reassignment to recover the shortage of the resource (discussed below on "Dynamic Task Assignment Protocol"). As a penalty for the contract termination, the TL will decrease the evicted node's trust value, which will be propagated in the network.

### 4.5 Dynamic Task Reassignment Protocol

***Lack of members***: During the first round of task assignment, some TLs may not be able to recruit a sufficient number of nodes for task execution. Such a TL will run the so called "reassignment protocol" as follows: (1) the TL will check if the completion time can be extended in terms of the urgency of the task and availability of the existing members; if the completion time is adjustable, the TL requests the extension and notifies all existing members; (2) if the completion time is not extensible, the TL looks for other available entities to fill the deficiency; if the TL finds a sufficient number of nodes from the available pool, the task will be performed with the new task members; and (3) if the TL could not find a sufficient number of nodes from the pool, the task will be labeled as "incomplete."

***Termination of contract***: The contract between a TL and a member can be terminated for any of the following reasons: (1) a member's unavailability due to network or node conditions; and (2) a member's low trust level due to its misbehaviors. The risk probability of the member will be checked based on Equation 13 periodically. When the sum of risk levels exposed by all members for task m exceeds the given acceptable risk level, $\mathrm{P}_{\mathrm{m}}^{\mathrm{risk}}$, the TL identifies a member with the maximum risk level and terminates the contract with that member until the sum of risk levels of all members is less than $\mathrm{P}_{\mathrm{m}}^{\mathrm{risk}}$. Then, the TL again runs the reassignment protocol described above.

### 4.6 Task Failure

A task may fail due to the following reasons: (1) when a TL cannot find a sufficient number of members for the given task in the initial task assignment period; (2) when a TL cannot find an available node to replace a member leaving the group; and (3) when a certain fraction of current members are performing with trust values below the task's minimum trust thresholds. For condition (3), we define this state as failure where the system is not operable due to a lack of qualified nodes to execute task m. We define the failure condition as:

$$\sum_{\mathrm{j}\in\mathrm{M}} \mathrm{F_j(t)} > \mathrm{TH_{mem}} \text{ where } \mathrm{F_j(t)} = \begin{cases} 1 \text{ if } \mathrm{T_j^X(t)} < T_{\mathrm{m}}^{\mathrm{X-th}} \text{ for any X} \\ 0 \end{cases} \quad (15)$$

Here M is the set of members j's for task m, $\mathrm{F_j(t)}$ is 1 when any objective trust value on trust property X of node j does not meet the threshold for trust property X; 0 otherwise. $\mathrm{TH_{mem}}$ is the maximum number of unqualified nodes tolerable to execute task m. That is, this failure occurs when a sufficient number of qualified members to execute the task m is not available based on ground truth trust values.

### 4.7 Metrics

We use the following two metrics: trust bias, and mission completion ratio.

- **Trust Bias**: This is the time-averaged difference between measured trust, $T_{i,j}(t)$, and objective trust, $OT_j(t)$, based on ground truth. This metric measures the degree of trust accuracy in terms of how much the measured trust deviates from the actual trust. Given the entire mission lifetime LT, $B_{i,j}$ is obtained by:

$$B_{i,j} = \frac{\int_0^{LT} B_{i,j}(t)}{LT} \quad \text{where } B_{i,j}(t) = |T_{i,j}(t) - OT_j(t)|/OT_j(t) \tag{16}$$

- **Mission Completion Ratio ($P^{MC}$)**: This is the mission completion ratio during the entire mission time. This metric is computed based on the sum of the task completion probabilities, $P_m^{completion}$, weighted by the relative importance of each task during the entire mission time as follows:

$$P^{MC} = \sum_{m \in L} P_m^{completion} \frac{I_m}{\sum_{all} I_m} \tag{17}$$

## V. Numerical Results and Analysis

In order to validate the performance of our proposed task assignment protocol, we simulate the protocol using SMPL (Simulation Modeling Programming Language) [20], an event-driven simulator. Table 1 explains the model parameters and their meanings, as well as their default values. Table 2 gives characteristics of nodes of various types in terms of their default speed and workload values. Table 3 shows the competence threshold ($T_m^{C-th}$) and task workload ($W_m$) based on the difficulty level of a task ($D_m$). We consider a group mission comprising an equal number of tasks in each difficulty level, e.g., there are 4 tasks in each distinct task difficulty level (1-5) for a total of 20 tasks in a group mission. Table 4 shows the resulting acceptable risk level ($P_m^{risk}$) of task m, given task m's importance and $\rho_2$ values. Each row in Table 4 represents a mission formation, with the average $P_m^{risk}$ (in the first column) specifying the average task acceptable risk level for the mission (over all tasks in the mission). Note that the mission completion ratio is significantly affected by the success or failure of a task with high importance, as given in Equation 17. Hence, we do not allow more than 5% risk for a task with an importance value above 3. All simulation data are reported based on 100 simulation runs.

We compare our task assignment protocol (Trust-based TA) with a non-trust based task assignment protocol (non-trust based TA) in terms of mission completion ratio. For non-trust based TA, we strictly follow all procedures of the proposed auction protocol except that the winner selection process is based on trust-risk analysis by task leaders for fair comparison.

**Table 1: Protocol parameters and default values.**

| Parameter | Meaning | Default value |
|---|---|---|
| $\lvert M \rvert$ | Total number of tasks given to a mission group | 20 |
| $\alpha$ | Weight of direct evidences for trust evaluation where $0 < \alpha < 1$ | 0.2 |
| $1/\lambda$ | Inter-arrival time for a node's group join event | Once per 1 hour |
| $1/\mu$ | Inter-arrival time for a node's group leave event | Once per 4 hours |
| $P_{link}$ | Link reliability | 0.99 |
| c | Constant parameter to normalize $N_j^{enc}$ in Equation 1 | 10 |
| $T_m^{I-th}$ | Trust threshold given to mission m for integrity | 0.9 |
| $T_m^{C-th}$ | Trust threshold given to mission m for competence | Refer to Table 3 |
| $T_m^{R-th}$ | Trust threshold given to mission m for reciprocity uniformly selected from the given range | [0.5, 0.9] |
| $T_m^{SC-th}$ | Trust threshold given to mission m for social connectedness uniformly selected from the given range | [0.5, 0.9] |
| $\rho_1$ | Constant design parameter to normalize $r_{m,j}^{X}(t)$ in Equation 11 | 0.4 |
| $\rho_2$ | Constant parameter to normalize $P_m^{risk}$ in Equation 12 | 0.976 |
| LT | Total mission time | 24 hrs. |
| $T_{update}$ | Trust update interval | 20 min. |
| $\gamma$ | Trust decay factor in Equation 6 | 0.95 |
| $P_i^{fp}, P_i^{fn}$ | False positive and negative probabilities of detection error of node i uniformly selected from the given range | (0, 0.05] |
| $T_i^{comp}$ | Time interval node i becomes compromised uniformly selected from the given range | [3, 24] hrs. |
| $T_i^{SC}$ | Initial trust value given for social connectedness of node i uniformly selected from the given range | [0.7, 1.0] |
| $T_i^{R}$ | Initial trust value given for reciprocity of node i uniformly selected from the given range | [0.7, 1.0] |
| $P_i^{coop}$ | Initial trust value given for cooperativeness of node i uniformly selected from the given range | [0.7, 1.0] |
| $P_{comp}$ | Fraction of the number of nodes becoming compromised over time over all nodes in the network | 0.25 |
| R | Wireless radio range | 250 m |
| N | Total number of nodes in the network; each of the four types has N/4 nodes | 100 |
| $TH_{mem}$ | Maximum number of untrustworthy nodes tolerable for mission execution | ceil(N/3) |
| $\varphi$ | Constant parameter to normalize reward/penalty | 0.05 |

**Table 2: Workload capacity ($w_i$) and speed ($v_i$) of node i.**

| Node type | Number | $w_i$ | $v_i$ |
|---|---|---|---|
| NT1 | 25 | 1 packet / (10, 30] sec. | 0 m/s |
| NT2 | 25 | 1 packet / (5, 10] sec. | [0.1, 1] m/s |
| NT3 | 25 | 1 packet / (1, 5] sec. | [1, 2] m/s |
| NT4 | 25 | 1 packet / (0.1, 1] sec. | [2, 10] m/s |

**Table 3: Competence threshold ($\mathbf{T_m^{C-th}}$) and task workload ($\mathbf{W_m}$) based on the difficulty level of a task ($\mathbf{D_m}$).**

| $D_m$ | $W_m$ | $T_m^{C-th}$ |
|---|---|---|
| 1 | 1 packet / (1.5, 2.0] sec. | 0.4 |
| 2 | 1 packet / (1.0, 1.5] sec. | 0.5 |
| 3 | 1 packet / (0.5, 1.0] sec. | 0.6 |
| 4 | 1 packet / (0.1, 0.5] sec. | 0.7 |
| 5 | 1 packet / (0, 0.1] sec. | 0.8 |

**Table 4: Acceptable risk level ($\mathbf{P_m^{risk}}$) of task m with $\rho_2$, with each row representing a mission formation of 20 tasks.**

| Average $P_m^{risk}$ | $I_m$ = 1 | $I_m$ = 2 | $I_m$ = 3 | $I_m$ = 4 | $I_m$= 5 | $\rho_2$ |
|---|---|---|---|---|---|---|
| 8% | 0.286 | 0.082 | 0.023 | 0.007 | 0.002 | 1.25 |
| 10% | 0.335 | 0.112 | 0.038 | 0.013 | 0.004 | 1.09 |
| 12% | 0.377 | 0.142 | 0.054 | 0.020 | 0.006 | 0.97 |
| 14% | 0.415 | 0.172 | 0.071 | 0.030 | 0.012 | 0.88 |
| 16% | 0.449 | 0.202 | 0.090 | 0.040 | 0.018 | 0.8 |

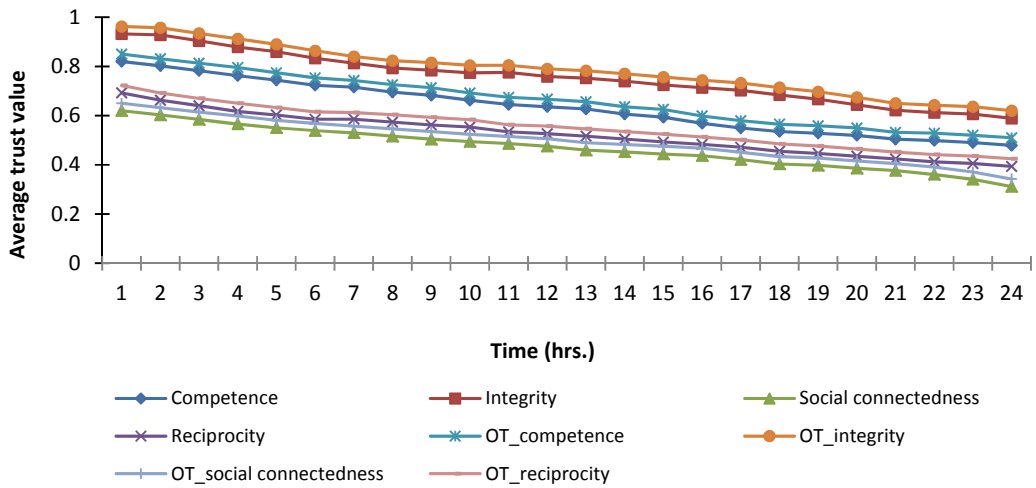


**Figure 1: Subjective vs. objective trust values for trust property X of type 4 nodes.**

Figure 1 shows the average trust values of type 4 nodes, as evaluated by all other nodes, for each trust property X over time, given α=0.2 and γ=0.95 and the default values specified in Table 1. The label "OT_(trust property X)" stands for objective trust of trust property X. We observe that the average trust bias is less than 3% for all trust properties. The reason of decreasing trust over time is because nodes are compromised over time.

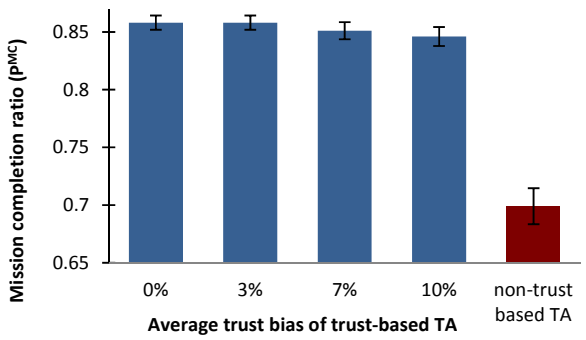


**Figure 2: Mission completion ratio ($\mathbf{P^{MC}}$) vs. average trust bias when $\mathbf{P_m^{risk}}$=0.12.**

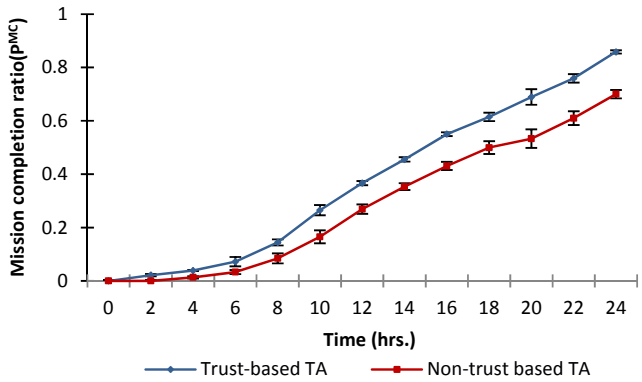


**Figure 3: Mission completion ratio ($\mathbf{P^{MC}}$) vs. mission time with average trust bias of 3% and $\mathbf{P_m^{risk}}$=0.12.**

Figure 2 compares the mission completion ratio ($\mathrm{P^{MC}}$) of trust-based TA vs. non-trust based TA. For trust-based TA, it also shows $\mathrm{P^{MC}}$ with respect to the time-averaged trust bias (i.e., lower trust bias means higher trust accuracy). We observe that trust-based TA outperforms non-trust based TA. The main reason is that trust-based TA can identify qualified nodes based on specific trust criteria that the task requires. Like trust-based TA, non-trust based TA also considers if a node is of the right type, if a non-conflicting schedule is produced, and if a node maximizes its own utility (i.e., a longer task duration with less workload as given in Equation 10). However, non-trust based TA does not consider task failure cases due to mismatched member nodes being selected for task execution, leading to a higher risk and consequently a lower mission completion ratio.

In Figure 2, 0% trust bias is obtained when we use objective trust values (ground truth) while 3%-10% trust bias is obtained when we use subjective trust values generated from the trust-based TA. We use 0% trust bias as an upper bound ideal performance case against which our trust-based TA is compared. We observe that as the trust bias decreases, $\mathrm{P^{MC}}$ is close or equal to the ideal case based on objective trust. When the trust bias is higher, $\mathrm{P^{MC}}$ decreases because trust is mostly underestimated, leading to an insufficient number of nodes being assigned to tasks in the initial task assignment period. That is, when trust values are underestimated, task leaders are more likely to miss chances to select qualified nodes due to the trust bias. However, compared to non-trust TA, trust-based TA performs well even with a trust bias as high as 10%. The reason is that nodes whose trustworthiness is underestimated are perfectly capable of performing tasks with low risk once they are selected for task execution.

Figure 3 shows $\mathrm{P^{MC}}$ vs. the mission time based on Equation 17, comparing trust-based TA with non-trust based TA. We use 3% trust bias and 12% average acceptable risk level ($\mathrm{P_m^{risk}}$) (defined in Table 4) in this experiment. We observe that as the mission time increases, the mission completion ratio increases linearly because of the better chance of all the tasks being completed with a longer mission time.

In Figures 2 and 3, we also report the standard deviation (SD) of the simulation results. For trust-based TA, the SD is less than 1%, i.e., the mean percentage difference (MPD) from the mean is less than 1%. For non-trust based TA, the SD is around 2%. This implies that trust-based TA results in fewer membership changes than non-trust based TA since membership changes significantly affect the mission completion ratio and, consequently, result in a higher variance of the mission completion ratio for non-trust based TA.

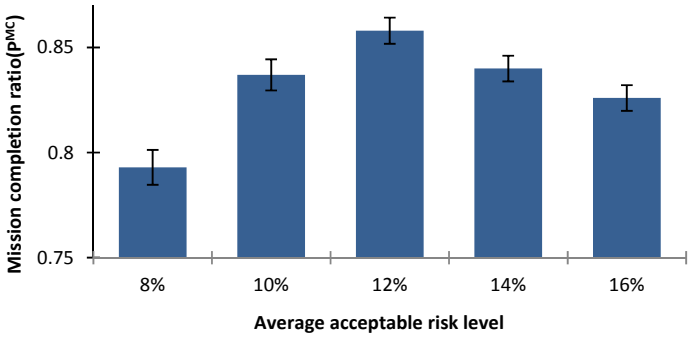


**Figure 4: Mission completion ratio ($P^{MC}$) vs. average acceptable risk level with 3% trust bias.**

Figure 4 shows $P^{MC}$ vs. the average acceptable risk level ($P_m^{risk}$) of a mission (defined in Table 4) when trust bias is less than 3%. We observe that an optimal $P_m^{risk}$ exists in the given experimental environment. When $P_m^{risk}$ is stringent (i.e., less than 12%), task leaders are more likely to fail to obtain a sufficient number of members for task execution in the initial task assignment period. On the other hand, when $P_m^{risk}$ is loose (i.e., more than 12%), task leaders may recruit a sufficient number of members for their tasks, but the task may still fail due to a lack of qualified members as described in Equation 15. Therefore, we observe that an optimal $P_m^{risk}$ exists, and $P_m^{risk}$=12% for mission formation is identified as the optimal value for maximizing $P^{MC}$. This optimal $P_m^{risk}$=12% strikes the best balance between meeting the condition of selecting a sufficient number of members while preserving qualified trustworthy nodes as task members. Again the data reported in Figure 4 also have SD less than 1%.

## VI. Conclusions

In this paper, we proposed a trust-based task assignment protocol that uses composite trust to select qualified nodes to maximize mission completion ratio while meeting an acceptable risk level. We measured composite trust derived from a multi-genre network including communication, information, and social networks. Unlike existing work in the literature, we solved the task assignment problem for assigning nodes to multiple tasks where tasks dynamically arrive and depart. We demonstrated that our trust-based task assignment protocol outperforms the non-trust based counterpart. Further, because our trust-based task assignment protocol considers the intrinsic tradeoff between trust and risk, we are able to identify an optimal acceptable risk level for mission formation to best balance the need of selecting a sufficient number of members for task execution vs. the high risk leading to task failure due to a lack of trustworthy nodes being selected as task members.

**Acknowledgement**

Dr. MoonJeong Chang was supported in part by the Army Research Office under Grant W911NF-12-1-0016.

## References

[1] H. S. AdbelSalam and S. Olariu, “Toward Efficient Task Management in Wireless Sensor Networks,” *IEEE Trans. on Computers*, vol. 60, no. 11, pp. 1638 – 1651, Nov. 2011.

[2] M. Blaze, J. Feigenbaum, and J. Lacy, “Decentralized Trust Management,” *Proc. IEEE Symposium on Security and Privacy*, pp. 164 – 173, Oakland, CA, May 1996.

[3] C. Castelfranchi and R. Falcone, *Trust Theory: A Socio-Cognitive and Computational Model*, Wiley Series in Agent Technology (Editor: M. Wooldridge), 2010.

[4] H. Chang and W. Oldham, “Dynamic Task Allocation Models for Large Distributed Computing Systems,” *IEEE Trans. on Parallel and Distributed System*, vol. 6, no. 12, pp. 1301-1315, Dec. 1995.

[5] I. R. Chen, F. Bao, M. Chang, J. H. Cho, “Integrated Social and QoS Trust-based Routing in Delay Tolerant Networks,” *Wireless Personal Communications*, pp. 1-17, June 2011 (online version).

[6] J.H. Cho, A. Swami, and I.R. Chen, "A Survey on Trust Management for Mobile Ad Hoc Networks," *IEEE*

*Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 562-583, 4th Quarter 2011.

[7] J. H. Cho, A. Swami, and I. R. Chen, "Mission-Dependent Trust Management in Heterogeneous Military Mobile Ad Hoc Networks," *Int'l Command and Control Research and Technology Symposium,* Santa Monica, CA, June 2010.

[8] J. H. Cho, A. Swami, and T. Cook, "Combinatorial Auction-based Multiple Dynamic Mission Assignment," *Proc. IEEE Military Communications*, Baltimore, MD, Nov. 2011.

[9] M. L. Choi, L. Brunet, and J. P. How, "Consensus-based Decentralized Auctions for Robust Task Allocation," *IEEE Trans. on Robotics*, vol. 25, no. 4, pp. 912-926, Aug. 2009.

[10] B.B. Choudhury, B.B. Biswal, and D. Mishra, "Development of Optimal Strategies for Task Assignment in Multi-robot Systems," *Proc. IEEE Int'l Advance Computing*, pp. 1130-1135, Pariala, India, March 2009.

[11] K. S. Cook (editor), *Trust in Society*, vol. 2, Feb. 2003, Russell Sage Foundation Series on Trust, New York.

[12] Y. Jiang and J. Jiang, "Contextual Resource Negotiation-Based Task Allocation and Load Balancing in Complex Software Systems," *IEEE Trans. on Parallel and Distributed Systems*, vol. 20, no. 5, pp. 641-653, May 2009.

[13] Y. Jin, J. Jin, A. Gluhak, K. Moessaner and M. Palaniswami, "An Intelligent Task Allocation Scheme for Multiple Wireless Networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 23, no.3, pp. 444-451, March 2012.

[14] 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. on Parallel and Distributed Systems*, vol. 21, no. 11, pp. 1692-1705, Nov. 2010.

[15] A. Josang and S. LoPresti, "Analyzing the Relationship between Risk and Trust," *Proc. 2nd Int'l Conf. Trust Management*, pp. 135-145, 29 March – 1 April, 2004, Oxford, UK.

[16] I. S. Kulkarni and D. Pomili, "Task Allocation for Networked Autonomous Underwater Vehicles in Critical Mission", *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 5, pp. 716-727, June 2010.

[17] T. Le, T. J. Narman and W. Vasconcelos, "Agent-based Sensor-Mission Assignment for Tasks Sharing Assets," *Int'l Workshop on Agent Technology for Sensor Networks*, Budapest, Hungary, May 2009.

[18] M. Li, J. Li, H. Song, and D. Wu, "Risk Management in the Trustworthy Software Process: A Novel Risk and Trustworthiness Measurement Model Framework," *5th Int'l Joint Conf. on INC, IMS and IDC*, Aug. 2009, Seoul, Korea.

[19] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, "URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks*," IEEE/ACM Trans. on Networking*, vol. 12, no. 6, pp. 1049-1063, Dec. 2004.

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

[21] N. Nisan, T. Roughgarden, E. Targos, and V. V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, Sept. 2007.

[22] H. Rowaihy, M. P. Johnson, O. Liu, A. Bar-Noy, and T. Brown, "Sensor-Mission Assignment in Wireless Sensor Networks," *ACM Trans. on Sensor Networks*, vol. 6, no. 4, article 36, July 2010.

[23] B. Solhaug, D. Elgesem, and K. Stolen, "Why Trust is not Proportional to Risk?" *Proc. 2nd Int'l Conf. on Availability, Reliability, and Security*, pp. 11-18, Vienna, Austria, April 2007.

[24] R. Trivers, "The Evolution of Reciprocal Altruism," *The Quarterly Review of Biology*, vol. 46, no. 1, pp. 35-57, March 1971.

[25] H. Tsai and Y. Huang, "An Analytic Hierarchy Process-Based Risk Assessment Method for Wireless Networks," *IEEE Trans. on Reliability*, vol. 60, no. 4, pp. 801-816, Dec. 2011.

[26] Merriam Webster's Dictionary [Online]: http://www.merriamwebster.com/dictionary/trust%5B1%5D.

[27] Social Report 2010: Social Connectedness [Online]: http://socialreport.msd.govt.nz/documents/social-connectedness-social-report-2010.pdf