Trust-based Service Composition and Binding for Tactical Networks with Multiple Objectives

Yating Wang[†], Ing-Ray Chen[†], Jin-Hee Cho*, Kevin S. Chan* and Ananthram Swami*

[†]Virginia Tech
Department of Computer Science
{yatingw, irchen}@vt.edu

*U.S. Army Research Laboratory
{jinhee.cho, kevin.s.chan, ananthram.swami}.civ@mail.mil

***Abstract* —Tactical networks must select service providers to meet service requirements of an operation while facing resource constraints and high security vulnerability. In such an environment nodes provide services to support various operations and/ may request services to support the operations as well. We formulate the problem of service composition and service binding as a multi-objective optimization (MOO) problem, minimizing the service cost, while maximizing the quality of service (QoS) and quality of information (QoI). The MOO problem is essentially a node-to-service assignment problem such that by dynamically formulating service composition, and selecting the right nodes to provide requested services, the network can support concurrent operations while achieving multiple system objectives. We develop a trust-based service composition and binding protocol. We demonstrate that the trust-based scheme outperforms the counterpart non-trust-based scheme. Furthermore, our trust-based scheme can effectively penalize malicious nodes performing self-promotion attacks, thus filtering out malicious nodes and can ultimately lead to high user satisfaction.**

***Keywords*—service composition, tactical networks, trust, multi-objective optimization.**

I. INTRODUCTION

Tactical networks must support concurrent operations, such as target tracking, field surveillance, target classification, and trajectory prediction. Applying service composition techniques [9] to compose services can effectively simplify implementation details while efficiently utilizing available resources. While dynamic service composition has been extensively studied for web services, little is found for tactical networks because a tactical operation failure is often much more severe than a web service failure. Moreover, most studies focus on maximizing quality-of-service (QoS) of a single service composition request without considering resource constraints when concurrent service composition requests exist, which necessitates multiple system goals to be considered.

This paper considers dynamic service composition and service binding in a tactical network environment in which nodes provide services to support tactical mission operations, and also request services themselves. In the literature [11], the dynamic service composition and binding problem comes in two forms: (a) composition by planning in which a goal and a set of available services are given and the system completes the goal by planning and service binding using available services; and (b) composition by workflow process optimization in which the workflow with constraints is given as input and the service composition and binding problem is essentially a node-to-service assignment problem as we consider in this paper. We formulate the problem of service composition and binding as a multi-objective optimization (MOO) problem for maximizing QoS and QoI (quality-of-information) while minimizing service cost.

Our approach uses trust for making decisions on service composition and binding. In tactical networks, collaborating with sufficiently trustworthy entities is the key to successful mission execution assuming that selected trustworthy entities are sufficiently functional to execute the given mission. Malicious nodes may provide false information about service metrics; our trust-based assessment filters out these nodes leading to successful task completion.

Trust-based service composition and binding also has been studied in the web services domain [2], [6], but only a single dimension of trust was considered. In order to reflect key capabilities required for competing tactical missions, we consider two key trust dimensions, *competence* and *integrity*, as the building blocks of a "composite" trust metric.

To the best of our knowledge, no prior work has considered incorporating trust as decision criteria to solve a service composition and binding problem for tactical networks with multiple objectives. We develop an Integer Linear Programming (ILP) solution technique to solve this MOO problem.

The rest of this paper is organized as follows. Section II describes our system model including the tactical network environment, trust metric, the service model and the MOO problem definition. Section III describes the trust-based and non-trust-based service composition and service binding protocols. Section V reports comparative performance results utilizing ILP to solve the MOO problem for both trust-based and non-trust-based schemes. We also conduct sensitivity

978-0-7695-5124-1/13 $31.00 © 2013 IEEE
DOI 10.1109/MILCOM.2013.314

IEEE computer society

analyses to reveal design parameter settings under which the performance of our trust-based scheme can be further improved. Section VI concludes the paper.

II. System Model

A. Tactical Network Environments

We consider a tactical network in which nodes are heterogeneous such as sensors, robots, unmanned vehicles or other devices, which may have severe energy constraints and are mainly used for sensing and relaying data or implementing actions not suitable for human operators; nodes can be dismounted soldiers carrying sensors or handheld devices, or manned vehicles with various types of equipment, which possess intelligence for analyzing data before taking actions. In this network a node has two roles in executing operations: (1) a service provider (SP) to support an operation; and (2) a service requestor (SR) to request a service in the process of initiating (and executing) an operation. An example operation can be target tracking, field surveillance, and/or target classification. Each operation may require more than one service. For example, target tracking requires services from nodes with signal processing power and localization capability. Field surveillance can require services such as multi-modal signal processing, data aggregation, and intrusion detection. There may be cases where two services should be executed concurrently. We denote the abstract services by S_1, S_2, …, S_n.

B. Trust Metric

We use trust to assign the right nodes as SPs to the right abstract services. We consider *composite trust* consisting of *competence* and *integrity* as follows:

- **Competence**: This refers to an entity's capability to serve the received request by providing a satisfactory quality. This is often affected by (a) network conditions such as link failure or being disconnected by environmental conditions such as terrain; (b) energy level of a node (e.g., sensors vs. unmanned vehicles); or current workload and (c) inherent nature of a human entity such as willingness.
- **Integrity**: This refers to the degree to which a node complies with a given network protocol, not performing network attacks including self-promotion attacks by disseminating false information. That is, a node may lie about QoI, QoS and cost scores of its own capability and try to increase its chance to be included in mission operations in order to disrupt successful execution of the operations.

We assume the trust values are scaled in the range of [0, 1] as a real number. We denote the trust of node j evaluated by node i in trust property X (i.e., competence or integrity) as $T_{i,j}^{X}$. The overall trust is the average of these two trust values as we consider these two trust dimensions to be equally important in military missions. We adopt Bayesian inference [1], [7], [8] modeling trust by the Beta (α, β) distribution such that α/(α+β) is the estimated trust of a SP with α as the number of positive service experiences and β as the number of negative service experiences, which are accumulated by a node as it evaluates the services provided to it by an SP. This trust value is propagated to and then aggregated by other nodes in the system through a trust propagation and aggregation protocol characterized by a parameter, T_{err}, representing the trust estimation error or trust bias due to trust propagation and aggregation as each node updates other nodes' trust values in a distributed manner. For a malicious node or a bad node, the worst-case is that its trust is overestimated by T_{err} over the trust value propagated. In this paper, we test the resiliency of the trust-based service composition and binding protocol against self-promotion attacks under the worst case trust overestimation error for bad nodes.

C. Service Advertisement

A node as a SP can use two methods to advertise its availability to provide services to the network: *push method* and *pull method*.

In the *push method*, a node as a SP continuously broadcasts its availability. On the other hand, the *pull method* [9], [10] enables a node to advertise its service availability only when a peer node (i.e., a SR) shows interest. That is, the push method is proactive and consumes more resources while the pull method reactively serves upon request, trading off delay for resource consumption. We adopt the pull method for efficient resource management. A SP responds with an advertisement message only if it is capable of providing services requested by a SR. The advertisement message $\mathrm{Ad_{SP}}$ comprises four-tuple records, one for each abstract service it can provide as follows:

$$\mathrm{Ad_{SP}}: [\mathrm{k}, \mathrm{Q_k}, \mathrm{D_k}, \mathrm{C_k}] \text{ for } \mathrm{S_k} \quad (1)$$

Here we use Q, D, and C to denote the QoI, delay (for QoS) and cost metrics. The four-tuple record for S_k is: (a) k indicating the index of the servive S_k that the SP can provide ; (b) Q_k indicating the level of QoI the SP can provide for S_k; (c) D_k indicating the level of service delay (for QoS) the SP can provide for S_k; and (d) C_k indicating the service cost the SP will consume to provide S_k. If a SP can provide multiple services, then it will have a set of four-tuple records for each abstract service that it can provide. We assume that the service quality of a SP in Q, D and C is based on a priori information describing a device's capability.

D. Dynamic Service Composition

A tactical network serves multiple tactical operations, each requiring a series of abstract services. We use the notation O_m to refer to operation m and SR_m to refer to the service requester of operation m. In order to execute the given operation, the SR advertises its need for services. Then, available SPs send out advertisement messages (i.e., $\mathrm{Ad_{SP}}$ as shown in (1)). Based on the informed service availabilities, the SR composes a specification of the service requirements to execute the operation. We call the specification profile a *service composition specification*, denoted as SCS. It includes the schedule of required abstract services, S_k's, and the required Q, D and C thresholds for each abstract service. Based on the SCS, the SR proceeds with the node-to-service assignment (NSA) process to identify the best SP set to maximize performance in terms of multiple objectives defined in subsection F. In Section III, we will detail the process of NSA. An example SCS is:

$$\mathrm{SCS} = \langle [\mathrm{S}_0], [\mathrm{S}_2, \mathrm{S}_4], [\mathrm{S}_3], [\mathrm{S}_7][\mathrm{S}_4, \mathrm{S}_8], [\mathrm{S}_2] \rangle \qquad (2)$$

Here $[\mathrm{S}_2, \mathrm{S}_4]$ specifies that S_2 and S_4 are to be executed concurrently; $[\mathrm{S}_3], [\mathrm{S}_7]$ specifies that S_3 and S_7 are to be executed sequentially. Each abstract service S_k is associated with a "hard" threshold requirement $\mathrm{S}_k^{\mathrm{THRES}} = \left(\mathrm{Q}_k^{\mathrm{THRES}}, \mathrm{D}_k^{\mathrm{THRES}}, \mathrm{C}_k^{\mathrm{THRES}}\right)$ specifying the required Q, D and C thresholds where $\mathrm{Q}_k^{\mathrm{THRES}}$ is the minimum Q threshold, $\mathrm{D}_k^{\mathrm{THRES}}$is the maximum D threshold, and $\mathrm{C}_k^{\mathrm{THRES}}$ is the maximum C threshold.

E. Service Binding

A node capable of providing multiple services to an operation can be selected to execute multiple services. However, it may not be selected for executing concurrent services because this may adversely affect the Q, D and C outcomes due to heavy workload. If two operations overlap in time for service provision, a node capable of providing services to these two operations can at most choose one to execute to ensure its availability and commitment to a single operation.

F. Multiple Objective Optimization

We consider three objectives, maximizing Q while minimizing D and C. These are the key criteria to measure user satisfaction. A SR may be a human entity such as a soldier carrying devices. In order to estimate the satisfaction level for the services provided for an operation, the SR can hold two standards: *hard standard* vs. *soft standard*. The *hard standard* is the Q, D and C threshold requirement $\mathrm{S}_k^{\mathrm{THRES}}$ for S_k. This is a strict standard that each service must comply to meet user satisfaction. If there is no SP available to provide the service, or if the SP selected to provide the service fails to meet the threshold requirement, then the operation is regarded as having failed. The *soft standard* measures user satisfaction for the provided services in terms of the overall Q, D and C service levels achieved, i.e.,

$$\mathrm{Q}_m = \sum_{i \in \mathcal{S}_m} \mathrm{Q}_{m,i}\, ;\ \mathrm{D}_m = \sum_{i \in \mathcal{S}_m} \mathrm{D}_{m,i}\, ; \mathrm{C}_m = \sum_{i \in \mathcal{S}_m} \mathrm{C}_{m,i} \qquad (3)$$

$\mathcal{S}_m$ is the set of abstract services requested by O_m; Q_m, D_m and C_m are the operation-level Q, D, and C achieved, and $\mathrm{Q}_{m,i}$, $\mathrm{D}_{m,i}$ and $\mathrm{C}_{m,i}$ are the service-level Q, D, and C achieved by node i. Higher Q and lower D and C are desirable.

We capture the multiple objectives into a single scalar function to be maximized:

$$\mathrm{MOO} = \sum_{m \in \mathcal{T}} (\mathrm{Q}_m - \mathrm{D}_m - \mathrm{C}_m) = \max\ \mathrm{Q} - \mathrm{D} - \mathrm{C} \qquad (4)$$

where $\mathcal{T}$ is the set of operations.

G. Trust-based Reward and Penalty

We use trust as a reward (i.e., an incentive) or penalty to a node that has provided a service, since maintaining a sufficiently high trust level is important for tactical operation execution.

An operation can impose a minimum *user satisfaction threshold*, denoted as UST_m. This is to be compared against the *user satisfaction received* (USR_m), defined as:

$$\mathrm{USR}_m = \mathrm{Min}\left(\frac{\mathrm{Q}_m^{\mathrm{true}}}{\mathrm{Q}_m^{\mathrm{advertised}}}, \frac{\mathrm{D}_m^{\mathrm{advertised}}}{\mathrm{D}_m^{\mathrm{true}}}, \frac{\mathrm{C}_m^{\mathrm{advertised}}}{\mathrm{C}_m^{\mathrm{true}}}\right) \qquad (5)$$

Here $\mathrm{Q}_m^{\mathrm{advertised}}$, $\mathrm{D}_m^{\mathrm{advertised}}$ and $\mathrm{C}_m^{\mathrm{advertised}}$ are calculated by (3) based on advertised Q, D and C scores, while $\mathrm{Q}_m^{\mathrm{true}}$, $\mathrm{D}_m^{\mathrm{true}}$ and $\mathrm{C}_m^{\mathrm{true}}$ are calculated by (3) based on true Q, D and C scores. The rationale of defining USR_mas above is that a user (i.e., a SR) does not have knowledge of the "best" service quality, so its satisfaction level with services received is based on what has been promised to be delivered. Recall that malicious nodes may perform self-promotion attacks to advertise higher Q, and lower D and C scores to boost its chance of node-to-service assignment. Since good nodes always advertise true scores faithfully, USR_m =1 if SR_m selects only good nodes to provide services requested.

SR_m compares UST_m against USR_m to incentivize or penalize a SP by increasing or decreasing its trust. When USR_m exceeds UST_m, it is counted as a positive experience and all SPs in O_m are rewarded. As described in Section II.B, the Beta (α, β) distribution is used for trust assessment by the SRs. In the case of positive experience, α is incremented by 1 for all SPs in O_m (to both integrity and competence). On the other hand, when USR_m is less than UST_m, SR_m identifies the culprits with low performance (by comparing the advertised service quality profile with the actual received service) and considers it a negative experience against these culprits. In this case, β is increased by 1 for all identified culprits. Other SPs identified as benign will not be penalized. Every node in the system keeps track of α and β counts for all SPs in the system. The trust change toward a SP will then be propagated to make it known to other nodes in the network.

III. Service Composition and Binding Schemes

Once the SCS for O_m is dynamically formulated based on service availability (i.e., available nodes), multiple node-to-service assignment (NSA) solutions may exist to meet the service requirements and constraints specified. Which NSA solution to pick depends on if O_m is standalone. If O_m is standalone, the service requester SR_m for O_m then chooses the best NSA solution among all candidate solutions to maximize $\mathrm{MOO}_m = \mathrm{Q}_m - \mathrm{D}_m - \mathrm{C}_m$. If O_m executes concurrently with some other operations in a concurrent operation set $\boldsymbol{\mathcal{C}}$, then SR_m cooperates with other SRs in the concurrent operation set to collectively choose the best set of NSA solutions (one for each operation in set $\boldsymbol{\mathcal{C}}$) to maximize MOO in (4) with $\boldsymbol{\mathcal{T}}$=$\boldsymbol{\mathcal{C}}$. We propose two dynamic service composition and binding schemes, non-trust-based and trust-based, as follows:

- ***Non-trust-based***: SR_m selects the best NSA solution based on advertised Q, D and C scores. In this scheme, SR_mfully trusts all service advertisements even if malicious nodes may lie about their Q, D and C scores. Objective metrics, Q_m, D_m, and C_m, are computed based on (3).
- ***Trust-based***: SR_m selects the best NSA solution where each objective is computed by a trust-weighted sum. That is, Q_m,

D_m and C_m are computed the same way as in (3) except that the trust of SR_m toward node i selected for operation execution (i.e., $T_{SR_m,i}$) is taken into consideration as follows:

$$Q_m = \sum_{i \in \mathcal{S}_m} \left(T_{SR_m,i} \times Q_{m,i} \right); \tag{6}$$

$$D_m = \sum_{i \in \mathcal{S}_m} \left(D_{m,i} / T_{SR_m,i} \right); \; C_m = \sum_{i \in \mathcal{S}_m} \left(C_{m,i} / T_{SR_m,i} \right)$$

Here $T_{SR_m,i}$ is the overall trust of SR_m toward node i, calculated by $0.5T^C_{SR_m,i} + 0.5T^I_{SR_m,i}$ assuming competence and integrity are equally important. The benefit of the trust-based scheme is that the advertised Q, D and C scores are attested by the trust levels of SPs evaluated by the SR. If all nodes are trustworthy with a trust value of 1, then the trust-based solution reduces to the non-trust-based solution. For the trust-based scheme, we further introduce two trust thresholds, T^C_m and T^I_m, for competence and integrity, respectively, such that node i is qualified for the execution of operation m only if $T^C_m \leq T^C_{SR_m,i}$ and $T^I_m \leq T^I_{SR_m,i}$.

IV. Problem Definition and Protocol Description

A given mission consists of multiple operations; an operation requires a set of abstract services some of which may need to run concurrently while others must run consecutively. Services are characterized by QoI (Q), Delay (D) and Cost (C) metrics. Associated with each requested service in an operation are thresholds on minimum QoI, maximum delay and maximum cost. A node responsible for an operation sends out service requests, and available service providers (SPs) respond with the Q, D, and C metrics for the services that they can provide. Given that multiple SPs may meet the threshold criteria, the goal is to choose SPs so as to minimize aggregate D and C and maximize aggregate Q, where the aggregation is first over the set of services constituting an operation, and then over the set of operations constituting the mission. The multiple objectives are captured by a single scalar function, MOO = Q - D - C, where Q, D, and C are aggregate values. This is an assignment problem: which SP should be assigned to which abstract service in which operation. Malicious nodes may advertise false Q, D, and C metrics; detecting and filtering out such nodes is crucial. The notion of a user-satisfaction ratio is used to mark provided services as positive or negative; these are then used to update the posterior distribution of trust, modeled as a Beta distribution. The mean of the distribution is used as an estimate of trust.

In the trust-based version, to be qualified, a SP's trust scores must be above prescribed thresholds. The service-level components of Q, D, and C for a given operation are scaled by the trust score for the corresponding SP. The idea is that nodes that have previously advertised false Q, D, and C metrics are less trustworthy; hence, their metrics are discounted.

V. Numerical Results and Analysis

Our preliminary case study is for a small size problem with 15 operations ($|\mathcal{T}|$=15) and 60 nodes ($|\boldsymbol{\mathcal{N}}|$=60) as SPs for dynamic service composition and service binding. The 15 operations are divided into 9 sequential chunks, i.e., {1, 2}, {3}, {4, 5}, {6, 7, 8}, {9}, {10}, {11, 12}, {13}, and {14, 15}. For simplicity, each operation is composed of 4 distinct abstract services ($|\mathcal{S}_m|$= 4) randomly selected from S_0 to S_8. Further, each SP is assumed to be specialized to one abstract service only. Thus, a SP can be assigned to at most one service in an operation. Furthermore, for reliability reasons, no SP can service more than one operation concurrently.

We consider a tactical network environment where all nodes can communicate with each other, so mobility is not an issue for communication. We set the two trust thresholds to $T^C_m = 0.5$ and $T^I_m = 0.5$, so that every SP with trust not less than 0.5 will have a chance to provide service. The $T^I_{i,j}$ and $T^C_{i,j}$ values for integrity and competence are set to 0.5 initially for all nodes with (α=1, β=1), meaning ignorance (no knowledge). As node i accumulates positive and negative experiences for services provided by node j, node i updates its trust toward node j based on trust penalty/reward described in Section II.G.

We model the hostility of the environment by the percentage of malicious nodes, denoted as P_{bad} in the range of $[0-50]\%$. The risk taking behavior of a malicious node is modeled by a percentage of boosting parameter, P_{risk}. That is, a malicious node will boost its advertised Q, D and C scores by multiplying its true Q, D and C scores with $(1+P_{risk})$, $(1-P_{risk})$, $(1-P_{risk})$, respectively, to increase its chance of being selected.

TABLE I. Key Parameters and Default Values/Ranges

Parameter	Value	Parameter	Value	Parameter	Value
$\lvert\mathcal{S}_m\rvert$	4	$\lvert\mathcal{T}\rvert$	10	T^I_m / T^C_m	0.5
T_{err} / P_{bad}	[0-50%]	$\lvert\boldsymbol{\mathcal{N}}\rvert$	60	D_{good} / C_{good}	[1-2]
P_{risk}	[0-100%]	$T^I_{i,j}$	0.5	D_{bad} / C_{bad}	[3-5]
UST_m	[75-100%]	$T^C_{i,j}$	0.5	Q_{bad}, Q_{good}	[1-3], [4-5]

Table I lists key parameters and their default values. In particular, Q_{bad}, D_{bad} and C_{bad} are the *true* Q, D and C scores for bad nodes, which can be boosted during service advertisement. Q_{good}, D_{good} and C_{good} are the *true* Q, D and C scores for good nodes, which will be reported by good nodes faithfully during service advertisement. The Q, D, C values for a node are generated once following uniform distribution at the beginning and are not changed during system operation.

We solve the dynamic service composition and service binding MOO problem as defined in (4) with ILP techniques. The detail of the ILP formulation is given in the Appendix. We apply the ILP solution technique to both non-trust-based using Q_m, D_m and C_m defined in (3), and trust-based using Q_m, D_m and C_m defined in (6). The end product obtained by using MATLAB is expressed by a decision variable $w_{j,k,m}$ specifying if node j should be assigned to abstract service S_k of operation m so that the MOO value defined in (4) is maximized. Dynamic trust update is implemented by an *iterative ILP* solution technique in which the ILP solution is applied sequentially to "operation chunks" in time order where a chunk is defined as a set of overlapping operations.

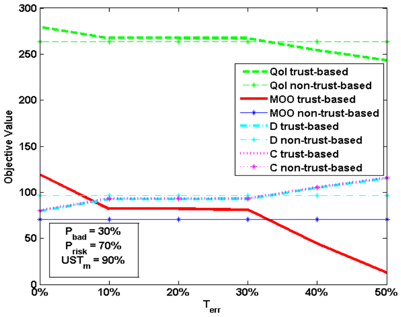

Fig.1. MOO, Q, D and C objective values vs. trust estimation error (T_{err}) under trust-based and non-trust-based schemes.

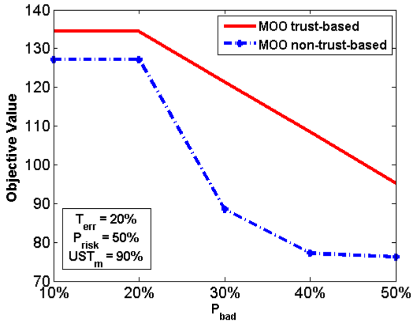

Fig. 2. MOO value vs. percentage of bad nodes (P_{bad}).

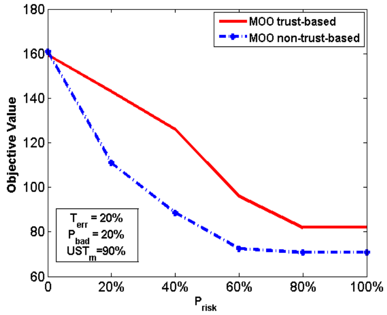

Fig. 3. Effect of risk taking by a malicious node (P_{risk}).

Depending on the outcome of the user satisfaction condition defined in (5), trust toward the nodes assigned to execute a chunk is updated and reflected as input to the next ILP execution. The 15 operations in our experiment require 9 ILP executions since these 15 operations are separated into 9 non-overlapping chunks in time order.

The underlying assumption of our iterative ILP solution technique is that trust values are static during the execution of each chunk, which is justified since trust is updated only after a tactical operation is executed. Below we report experimental results obtained from ILP solutions to examine the effect of key parameters, including P_{bad} (percentage of malicious nodes), P_{risk} (level of risk taking by a malicious node), UST_m (user satisfaction threshold), and T_{err} (trust bias). The results reported are based on the average of 100 runs with random seeds to generate *true* Q, D and C scores for good and bad nodes. The assignments of the specific abstract service that can be provided by each SP, the 4 distinct services that are required by each operation and the bad node selection given P_{bad} as input are also randomly generated but remain the same across the 100 runs.

Fig. 1 compares the two schemes in terms of Q, D, C, and MOO values for the case in which P_{bad} = 30%, P_{risk} = 70% and UST_m = 90%, with T_{err} varying in the range of 0% to 50%. The effect of T_{err} is modeled by trust overestimation for bad nodes to test the resiliency property of the trust-based scheme against self-promotion attacks. We see that trust-based scheme can still perform better than the non-trust-based scheme when the trust estimation error is bounded within 30%, that is, the MOO and Q values are higher while the D and C values are lower because of its ability to discern trustworthy SPs from untrustworthy ones. However, when the trust estimation error exceeds a threshold (30% in this case), the advantage of trust-based scheme to filter out untrustworthy nodes disappears because bad nodes (30% in this case) may have high subjective trust and can be selected to execute operations by mistake, especially if they aggressively lie about their service quality (P_{risk}=70% in this case). The trust estimation error may be introduced because of trust propagation and aggregation error. We note that many contemporary trust systems can limit the trust estimation error to 3-5% (e.g., [3], [4], [5]) which is much less than the threshold of 30%.

Fig. 2 shows that the performances of both schemes deteriorate as the percentage of bad nodes increases. However, the trust-based scheme is able to filter out bad nodes effectively (despite T_{err}=20%) and is less vulnerable to bad node population increase compared with the non-trust-based approach. Fig. 3 analyzes the effect of P_{risk} on performance. We observe a trend similar to that in Fig. 2. While the non-trust-based scheme helplessly accepts bad nodes that boost their advertised scores, the trust-based scheme is able to filter out untrustworthy nodes (T_{err}=20% in this case).

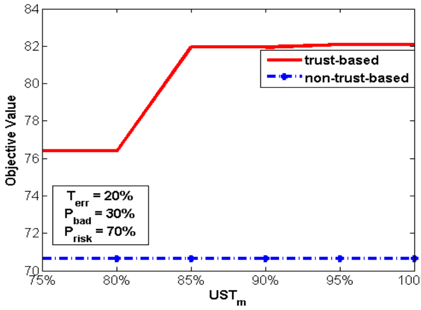

Fig. 4. Effect of user satisfaction threshold (UST_m).

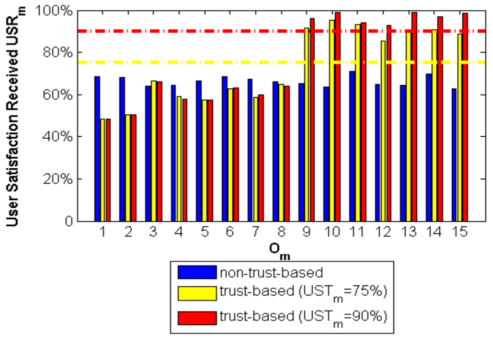

Fig. 5. User satisfaction received (USR_m) per operation.

Fig. 4 examines the impact of UST_m on protocol performance, with UST_m varying in the range of 75%-100%. Fig. 5 compares USR_m on an operation by operation basis in time order for the trust-based scheme operating at UST_m=75% and 90% against the non-trust-based scheme. We consider a combination of P_{risk}=70%, P_{bad}=30% and T_{err}=20% to reveal interesting trends. Because of high P_{risk} and T_{err}, the trust-based scheme is fooled into selecting bad nodes in the first few operations. So the first 8 operations do not pass the user satisfaction threshold. As a result, bad nodes selected to provide services in the first 8 operations are penalized with trust decrease and likely to be filtered out from later operations. This is evidenced by the result that the last 7 operations have high $\mathrm{USR_m}$ values. We see that $\mathrm{USR_m}$ values in the last 7 operations under $\mathrm{UST_m}$=90% are higher than those under $\mathrm{UST_m}$=75% because a higher user satisfaction threshold tends to cause more negative experiences and thus induces more trust penalty to be applied to bad nodes. In particular, $\mathrm{USR_m}$ is close to 100% for 3 of the last 7 operations when $\mathrm{UST_m}$=90% because only good nodes are being selected by the trust-based scheme due to dynamic trust update. On the contrary, the non-trust-based scheme consistently yields a low $\mathrm{USR_m}$ operation by operation because it has no effective way of filtering out bad nodes. Overall, the MOO function value increases (and levels off) as $\mathrm{UST_m}$ increases as demonstrated in Fig. 4. This trend supports our claim that the trust based scheme can effectively achieve high user satisfaction for MOO service quality despite the presence of bad nodes performing self-promotion attacks, especially after the system runs through the first few operations to stabilize trust update.

VI. Conclusion

We proposed a trust-based service composition and service binding protocol for a tactical network where we are concerned with multi-objective optimization. By utilizing an iterative integer linear programming solution technique for solving both trust-based and non-trust-based optimization problems, we demonstrate that our trust-based scheme outperforms the non-trust-based counterpart in user satisfaction. Furthermore, our trust-based scheme can effectively penalize malicious nodes performing self-promotion attacks, effectively filtering out malicious nodes, and can ultimately lead to high user satisfaction.

Appendix

In this appendix, we provide the implementation detail of the ILP solution technique for solving the node-to-service MOO assignment problem.

TABLE II. Variable Definitions for ILP

Variable	Definition
$\mathrm{ov_{p,q}}$	1 if operations p and q are overlapping in time; 0 otherwise
$\mathrm{s_{j,k}}$	1 if node j can provide abstract service k; 0 otherwise
$\mathrm{in_{k,m}}$	1, if operation m requires abstract service k; 0, otherwise
$\mathrm{tt_{j,m}}$	1 if $\mathrm{T_m^C} \leq \mathrm{T_{L_m,j}^C}$ and $\mathrm{T_m^I} \leq \mathrm{T_{L_m,j}^I}$; 0 otherwise
$\mathrm{to_{j,k,m}}$	$\mathrm{s_{j,k}} \times \mathrm{in_{k,m}} \times \mathrm{tt_{j,m}}$
$\mathrm{w_{j,k,m}}$	1 if node j is assigned to service k in operation m; 0 otherwise (output)

Table II defines the variables used in the ILP formulation. The binary variables $\mathrm{ov_{p,q}}, \mathrm{s_{j,k}}, \mathrm{in_{k,m}}$, and $\mathrm{tt_{j,m}}$ summarize the service composition specifications for the operations as well as status and capability of the nodes, given as input to the ILP. There is only one decision variable, namely, $\mathrm{w_{j,k,m}}$ to be determined by the ILP, specifying if node j should be assigned to abstract service k of operation m. The ILP will search for an optimal solution of $\mathrm{w_{j,k,m}}$ for all j's, k's and m's to maximize MOO in both trust-based and non-trust-based schemes. The objective function MOO= $\sum_{\mathrm{m}\in\mathcal{T}}(\mathrm{Q_m} - \mathrm{D_m} - \mathrm{C_m})$ as defined by (3), (4) and (6) can be computed as a linear function of $\mathrm{w_{j,k,m}}$ (the only decision variable to be decided by the ILP). The service-to-node assignment MOO problem is formulated as follows:

$$\underline{\text{Given}}\text{: } \mathcal{T}, \mathcal{S}_m, \mathcal{N}, \mathrm{ov_{p,q}}, \mathrm{s_{j,k}}, \mathrm{in_{k,m}}, \mathrm{tt_{j,m}}$$

$$\underline{\text{Find}}\text{: } \mathrm{w_{j,k,m}}$$

$$\underline{\text{Maximize}}\text{: } \sum\nolimits_{\mathrm{m}\in\mathcal{T}}(\mathrm{Q_m} - \mathrm{D_m} - \mathrm{C_m})$$

$$\underline{\text{Subject to}}\text{: } \forall \mathrm{j}\ \forall\{\mathrm{p},\mathrm{q}\}\ \mathrm{ov_{p,q}} \times (\mathrm{w_{j,k,p}} + \mathrm{w_{j,k,q}}) \leq 1;$$

$$\sum\nolimits_{\mathrm{j}} \mathrm{w_{j,k,m}} = \mathrm{in_{k,m}};\ \mathrm{w_{j,k,m}} \leq \mathrm{to_{j,k,m}}$$

Acknowledgment

This work was supported in part by the U. S. Army Research Laboratory and the U. S. Army Research Office under contract number W911NF-12-1-0445. This research was also partially supported by the Department of Defense (DoD) through the office of the Assistant Secretary of Defense for Research and Engineering (ASD (R&E)). The views and opinions of the authors do not reflect those of the DoD or ASD (R&E).

References

[1] K. Akkarajitsakul, E.E. Hossain, and D. Niyato, "Coalition-based cooperative packet delivery under uncertainty: A dynamic Bayesian coalitional game," *IEEE Trans. on Mobile Computing*, vol. 12, no. 2, pp. 371-385, 2013.

[2] S. Bansal, A. Bansal, and M.B. Blake, "Trust-based dynamic web service composition using social network analysis", *IEEE Workshop on Business Applications for Social Network Analysis*, Aug. 2010.

[3] F. Bao, I.R. Chen, M. Chang, and J.H. Cho, "Hierarchical Trust Management for Wireless Sensor Networks and Its Application to Trust-Based Routing and Intrusion Detection," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, 2012, pp. 169-183.

[4] I.R. Chen, F. Bao, M. Chang, and J.H. Cho, "Dynamic trust management for delay tolerant networks and its application to secure routing," *IEEE Trans. on Parallel and Distributed Systems*, Preprint, May 2013.

[5] J.H. Cho, A. Swami, and I.R. Chen, "A survey of trust management in mobile ad hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 4, pp. 562-583, 2011.

[6] G. Dai and Y. Wang, "Trust-aware component service selection algorithm in service composition," *4th Conf. on Frontier of Computer Science and Technology*, pp. 613-618, Shanghai, China, Dec. 2009.

[7] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Trans. on Sensor Networks,* vol. 4, no. 3, , pp. 1-37, May 2008.

[8] A. Jøsang, and R. Ismail, "The Beta reputation system," *Bled Electronic Commerce Conf.*, Bled, Slovenia, pp. 1-14, June 2002.

[9] S. Kalasapur, M. Kumar, and B. A. Shirazi, "Dynamic service composition in pervasive computing," *IEEE Trans. on Parallel and Distributed Systems*, vol. 18, no. 7, pp. 907-918, July 2007.

[10] E. Karmouch and A. Nayak, "A distributed constraint satisfaction problem approach to virtual device composition," *IEEE Trans. on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 1997-2009, 2012.

[11] C. Sandionigi, D. Ardagna, G. Cugola and C. Ghezzi, "Optimizing service selection and allocation in situational computing applications," *IEEE Trans. on Services Computing*, 2013.