A Context-based Strategy for SLA Negotiation in the IoT Environment

Fan Li, and Siobhán Clarke
Distributed Systems Group, SCSS, Trinity College Dublin, Dublin, Ireland
Email: {fali, Siobhan.Clarke}@scss.tcd.ie

***Abstract*—In the Internet of Things (IoT), billions of physical devices are connected and provide a near real-time state of the world. By adopting a service-oriented computing paradigm, the capabilities of these devices (whether mobile or static) can be abstracted as IoT services and delivered to users in a demand-driven way. Service providers can have a comprehensive competitive edge by tailoring their services to match users' requests through a negotiation process, with the particular service provisioning specified in a service level agreement (SLA), which can be further used to monitor and guarantee the quality of service (QoS). The challenges for SLA negotiation in the IoT include the scale and the dynamics of the environment. Existing SLA negotiation approaches are focused on cloud computing, which do not consider the domain-specific properties of IoT services. In this paper, we designed a context-based negotiation strategy to evaluate offers and generate counteroffers, and integrate it with the WS-Agreement Negotiation standard. The evaluation results demonstrate that our proposal can produce a higher utility and maintain a good success rate compared to other negotiation strategies.**

***Index Terms*—Internet of Things, SLA negotiation, WS-Agreement, negotiation strategy**

I. Introduction

The Internet of Things (IoT) envisions a large number of physical objects, connecting over the Internet to provide a near real-time state of the world. These real-world data and device capabilities can be abstracted as services [1], and provided to application tasks on demand to create scalable, adaptable, and flexible IoT solutions [2]. Based on a pay-as-you-go model, the same services can be delivered to different users with different service properties or QoS, by reconfiguring the services or devices [3]. For mission-critical IoT applications, "best effort" services are not sufficient [4]. To deliver requested services with pre-negotiated quality, Service Level Agreements (SLAs) are widely used as a contract-like concept to assure obligations and guarantees of involved parties in the context of a particular service provisioning [5]. Before a service is delivered, an SLA is created, which in turn requires a dynamic negotiation process where both parties express their own demands and preferences to resolve possible conflicts and arrive at a consensus [6].

SLA negotiation has been widely used in cloud computing [7], but has not been fully considered in IoT middlewares [8], even though a potentially huge number of IoT devices are likely to engage in service provisioning [9]. Challenges for SLA negotiation emerge from IoT characteristics such as its large-scale and highly dynamic nature. The IoT will be an ultra large-scale network containing billions of nodes that offer the same or similar functionalities. Human intervention for SLA negotiation may be infeasible. Unlike cloud services, IoT services are likely to have more negotiable attributes, which are not limited to QoS parameters. For example, the service coverage can be affected by the sleep schedule of sensor nodes [10]. Combined with the existence of mobile sensors, it is possible to tailor a service's location based on a user's expectation by adjusting a resource management mechanism. Also, to reduce energy consumption and maximize profit, the pricing mechanism may be dynamically set based on devices' sleeping schedule or power consumption mode, and providers may be willing to change devices' status in line with the users' demands [11]. IoT services exhibit dynamic behavior in terms of availability and mobility of resources, unpredictable workload, and unstable wireless network condition. Thus, the negotiation strategy should guarantee a good success rate and utility when the negotiation time is limited.

Since IoT devices are geographically distributed in different locations, we assume a middleware is deployed on fog nodes (e.g., base stations, routers, switches, etc.) to tailor service properties with candidate service providers based on the user's preference. We refer to the deployed nodes as gateways. In this paper, we design a context-based negotiation strategy for gateways that maps a user's negotiation preference and available resources to the decision-making function's parameters, and integrate the strategy to WS-Agreement Negotiation protocol.

The remainder of this paper is organized as follows. Section II summarizes related work. Section III describes the context-based negotiation strategy for IoT SLA negotiation. Section IV details the experimental setup and evaluation results and Section V concludes the paper with a discussion about future research directions.

II. Related Work

Compared to cloud computing, SLA negotiation in the IoT environment is still in the preliminary stage. For example, Gaillard *et. al.* proposed a centralized SLA management component [12]. But this framework relies on human intervention to finish the negotiation process. Mingozzi *et. al.* presented a cross-layer SLA negotiation framework for M2M applications [13], which only supports QoS negotiation in a single request-reply interaction, and the negotiation strategy is not specified.

Faratin *et. al.* proposed three types of negotiation tactics: time-dependent tactic, resource-dependent tactic, and

978-1-5386-9151-9/19/$31.00 ©2019 IEEE

behavior-dependent tactic [14]. They concluded that there is a tradeoff between the number of deals and the utility gained, and suggests that the use of different combinations of tactics may outperform the use of a single tactic. Yao *et. al.* employed fuzzy truth propositions and adopted a concession strategy that takes time and resources into consideration [15], but lacks a detailed description of the constraint model. Fharna *et. al.* added an adaptive mechanism to the negotiation strategy and proposed a policy-based approach [16] based on the performance observations of three decision functions, but this may not be practical when counterpart is adopting various unknown strategies. Misura *et. al.* proposed a cloud-based mediator platform where automatic negotiation is performed between device owners and application owners [17]. The mathematical model they adopted is a linear combination of previous offers, which is similar to the behavior-dependent tactic. However, this approach does not consider IoT service properties and context information. In addition, it assumes that the strategy will be specified when a device is registered in the platform, which may not be suitable for a dynamic environment where workload, supply, and demand are continuously changing. Also, device owners may not be willing to expose their strategies to the market. For the negotiation with incomplete information, Zheng *et. al.* [18] proposed a game-theory based strategy that combines the concession and tradeoff tactics to resolve possible conflicts, which demonstrates a good balance between utility and success rate under Monte Carlo simulations. However, this approach does not guarantee a solution will be found when one exists. Again, the strategy does not consider the impact of the negotiation context such as time and resources.

Metaheuristic algorithms and machine learning can also be applied to negotiation, except for the decision function. For example, Sim *et. al.* combined Bayesian learning with genetic algorithms to search for the optimal strategy when negotiating with incomplete information [19]. Narayanan *et. al.* use the Markov chain to model bilateral negotiations among agents, and Bayesian learning is adopted by agents to learn an optimal strategy [20]. Alkayal *et. al.* proposed a negotiation model based on particle swarm optimization to reduce the negotiation time and increase the throughput [21]. However, the common disadvantage shared by these approaches are the long negotiation time, which may increase significantly with the number of tasks or the number of negotiable parameters.

III. Negotiation Strategy

Although currently there is no standard SLA negotiation language for the IoT environment, it is possible to apply the web service SLA negotiation models to IoT services by extending existing approaches. WS-Agreement Negotiation (WSAG-Negotiation) is a standard negotiation specification for web services, which supports negotiating and creating SLAs by exchanging offers between a service provider and consumer [22]. In WSAG-Negotiation, The SLA-supported services are published by a service provider in the form of agreement templates (SLAT). An SLAT is a partially completed agreement filling default values of negotiable SLA parameters that the providers are expecting to offer, which can be regarded as the blueprint to create offers and the final SLA. A negotiation session begins when a provider's offering, outlined in an SLAT, cannot satisfy the user's request. However, the negotiation strategy is not defined in WSAG-Negotiation. In this section, we describe the negotiation strategy using the following denotations.

A *negotiation session* between a gateway g and a service provider p is defined as a finite sequence of offer collections $(X^{t_1}_{g\to p}, X^{t_2}_{p\to g}, X^{t_3}_{g\to p}, ...)$ with $t_1 < ... < t_n$ ($t_i \in [t_0, t_{deadline}]$). $X^{t_1}_{g\to p}$ is a vector of negotiation offers that proposed by g to p at time t_1. A *negotiation offer*, proposed by p to g at time t is defined as $x^t_{p\to g}$, consisting of a collection of negotiable terms J and the offer state s. Each negotiable term j ($j \in J$) has a *negotiable space* noted by Ω^g_j, which is the collection of possible values of term j. The value of term j offered in $x^t_{p\to g}$ is noted by $x^t_{p\to g}[j]$. The offer state derives from the state model of WSAG-Negotiation [22], which controls the interactions between negotiation parties and indicates the rules to take action after receiving a new offer. WSAG-Negotiation defines four possible states for an offer: $\{$*Advisory*, *Solicited*, *Acceptable*, *Rejected*$\}$. An *Advisory* offer indicates multiple back-and-forth interactions, which means the counterpart can either accept, reject or propose a new offer. A *Solicited* offer indicates a single request-reply interaction, which means the counterpart can only accept or refuse the offer; An *Acceptable* offer indicates all the negotiated items are acceptable, no further negotiation is required; and a *Rejected* offer indicates that the offer is rejected and the negotiation is terminated. The negotiation space may be expanded with a tolerance value τ ($0 \le \tau < 1$), within which the value may go beyond the negotiation space and is still acceptable. This is a *soft negotiation space* Ω'^g_j. The negotiable terms of IoT services may consist of both functional and non-functional features, we classified them into four types: QoS parameters, data rate, temporality and service coverage.

QoS parameters include service non-functional properties such as latency, availability, etc. In general, consumers and providers may have conflicting interests, e.g., a consumer hopes to obtain service with lower price but higher availability, whereas the provider attempts to offer the service with a higher price but lower availability. This means for consumers, the availability is a *"higher-is-better"* parameter, and the price is *"lower-is-better"* parameter. The negotiation space of QoS parameters is defined as Ω^g_{qos}=$[min_{qos}, max_{qos}]$, presenting the ranges of preferred values and reserved values. The soft negotiation space Ω'^g_{qos} of *higher-is-better* and *lower-is-better* attributes are $[min_{qos} - \tau, max_{qos}]$ and $[min_{qos}, max_{qos} + \tau]$ respectively. In a competitive market, providers may regard Ω^p_{qos} as business sensitive information and may not be willing to disclose to the public. This means that a QoS negotiation could take place with incomplete information, which may take multiple rounds to come to an agreement.

Data Rate includes both the data reporting rate and data sampling rate, which are important for emergency response applications such as fire detection. Negotiation of a data rate may occur under an assumption of incomplete information. As a higher data rate indicates a more timely and reliable measurement, it is a *higher-is-better* parameter for consumers. The negotiation space is defined as Ω^{g}_{rate}=$[min_{rate}, max_{rate}]$ and Ω'^{g}_{rate}=$[min_{rate} - \tau, max_{rate}]$.

Temporality relates to the time a service is available. We assume that service providers would specify their negotiation space of temporality Ω^{p}_{AT} in the initial offer in the form of negotiation constraints when the request cannot be fully satisfied. Ω^{p}_{AT} lists all the available time ranges around the requested time slot for gateways to compromise and select the most preferred one. We define Ω^{p}_{AT}=$\{AT^{p}_{1}, AT^{p}_{2}, ..., AT^{p}_{n}\}$ ($n \geq 1$) where AT^{p}_{i} represents the available time range $[t_{i0}, t_{ie}]$ offered by p.

Service Coverage specifies the spatial feature of an IoT service. We assume providers specify their negotiation space of service location Ω^{p}_{loc} in negotiation constraints if the request cannot be fully satisfied. For simplicity, we model the coverage with a circle. For service provider p, the negotiation space of coverage is defined as Ω^{p}_{loc}=$\{loc^{p}_{1}, loc^{p}_{2}, ..., loc^{p}_{n}\}$ ($n \geq 1$) where loc^{p}_{i} represents the available service area $[loc_{center}, loc_{radius}]$. For gateway g, the negotiation space is defined as $\Omega_{loc^{g}}$=$\{loc^{p}_{r}, d\}$ where loc^{p}_{r} represents the requested location, and d represents the acceptable distance in meters.

A. Scoring Function

To select the best offer $x^{t}_{p\to g}$ from $X^{t}_{p\to g}$, gateways need to evaluate service properties using a scoring function V^{g}_{j}. We list three types of scoring functions targeting each individual feature. For QoS parameters and data rate, the scoring functions of higher-is-better attributes and lower-is-better attributes are defined in Equation 1 and Equation 2 respectively [14]:

$$V^{g}_{j}(x^{t}_{p\to g}[j]) = \frac{x^{t}_{p\to g}[j] - min^{g}_{j}}{max^{g}_{j} - min^{g}_{j}} \tag{1}$$

$$V^{g}_{j}(x^{t}_{p\to g}[j]) = 1 - \frac{x^{t}_{p\to g}[j] - min^{g}_{j}}{max^{g}_{j} - min^{g}_{j}} \tag{2}$$

For temporality, the evaluation is based on matching the requested time and the available time proposed by a service provider. This matching degree depends either on the time or the duration. The time-dependent matching is inspired by the rating function proposed in [23]:

$$V^{g}_{AT}(AT_i) = \frac{AT_i \cap R_{time}}{R_{time}} \tag{3}$$

where R_{time} is the request time slot $[t_{r0}, t_{re}]$. We also define a duration-dependent scoring function for each time range AT_i:

$$V^{g}_{AT}(AT_i) = \begin{cases} 1, & \text{if } R_{dur} > AT_{i_{dur}} \\ AT_{i_{dur}}/R_{dur}, & \text{otherwise.} \end{cases} \tag{4}$$

where R_{dur} and $AT_{i_{dur}}$ denote the request duration and duration of available time respectively.

Service coverage is evaluated by calculating the distance between the requested area and the service area. We refer to the coordinates that have the same value before the third, fourth, fifth decimal places as the same location ($d = 11$m), close location ($d = 110$m) and nearby location ($d = 1100$m) respectively. The distance-based scoring function is defined as [23]:

$$V^{g}_{loc}(loc_i) = \begin{cases} 1, & dist(r_{eq}, i) < r \\ 1 - \frac{dist(r_{eq}, i)}{d+r}, & r \leq dist(r_{eq}, i) < d + r \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where r represents the service coverage radius, and $dist(r_{eq}, i)$ represents the distance between the requested location and the centre location of service coverage.

B. Bilateral Negotiation Process

In each round, provider p may prepare more than one counter offer as a response, with each offer satisfying parts of the requirements. When a gateway g receives counteroffers $X^{t}_{p\to g}$ from p, the decision-making model it adopted is illustrated in Algorithm 1. Firstly, g validates $X^{t}_{p\to g}$ based on constraints specified in SLAT (Line 2). Then, g evaluates $X^{t}_{p\to g}$ using the scoring functions, and select the most preferred offer $x^{t}_{p\to g}$ that has the maximum cost performance (Line 3). The cost performance is defined as:

$$C_p(x^{t}_{p\to g}) = \frac{V^{g}_{j}(x^{t}_{p\to g})}{p_{rice}} \tag{6}$$

where $V^{g}_{j}(x^{t}_{p\to g}) = \sum_{j=1}^{k} w_j V^{g}_{j}(x^{t}_{p\to g}[j])$ is the total score computed with the weight of attribute j ($\sum_{j=1}^{k} w_j = 1$). Next, g takes actions at time t' based on the state of $x^{t}_{p\to g}$. If $x^{t}_{p\to g}$ is an advisory offer, a pre-defined negotiation tactic is performed to update the current expectations of negotiable terms by making a compromise on conflicted items (Lines 18). The updated expectations are further compared to $x^{t}_{p\to g}$ to get the most optimized solution (Lines 21-37). If the current round is approaching the deadline, to increase success rate, a ultimatum is composed by replacing the expectations of conflicted terms with reserved values (Line 23). If $x^{t}_{p\to g}$ is acceptable in the current round, the negotiation is successfully finished, and a pending SLA is created for the consumer to authorize (Lines 7-9). However, if the negotiation tactic can not produce valid results that satisfy the constraints, a solicited offer is created (Line 36). Lines 10-16 and 22-24 indicate that the strategy can guarantee to find a solution when negotiation parties have intersected negotiation space.

The negotiation goal of a gateway is to find an acceptable deal that satisfies the negotiation constraints of a consumer but which, on the other hand, maximizes the utility. A game-theory based negotiation strategy that combines the concession and tradeoff tactic demonstrates a good balance in utility and success rate [18]. However, this approach only supports the negotiation attributes whose value varies between 0 and 1. Besides, it uses a static concession rate without considering negotiation context information, such as time, resources and user's preference. Thus, we modified the approach in five aspects: (i) defining a context-based utility function to make

Algorithm 1 Perform Negotiation Strategy

Input: received offers $X^t_{p\to g}$, negotiable terms in received offer $x^t_{p\to g}[J]$, last proposal $x^{t-1}_{g\to p}$, consumer identifier id_c, provider identifier id_p, soft negotiation spaces of terms Ω'^g

while received offers $X^t_{p\to g}$ **and** $t \in [t_0, t_{deadline}]$ **do**
  $X^t_{p\to g} \leftarrow$ validatedOffers($X^t_{p\to g}$)
  $x^t_{p\to g} \leftarrow$ getOptimizedOffer($X^t_{p\to g}$)
  switch $x^t_{p\to g}$.getState() **do**
    case Rejected
      sendResponseMsg(id_c)
    case Acceptable
      sla← createSLA(slat, $x^t_{p\to g}$)
      sendResponseMsg(id_c, sla)
    case Solicited
      if $\exists j \in J, x^t_{p\to g}[j] \notin \Omega'^g_j$ **then**
        $x^{t+1}_{g\to p} \leftarrow$ newOffer($x^t_{p\to g}$,Reject)
      else
        $x^{t+1}_{g\to p} \leftarrow$ newOffer($x^t_{p\to g}$,Accept)
      end if
      sendNegotiationMsg(id_p, $x^{t+1}_{g\to p}$)
    case Advisory
      suc← performNegotiationTactic($x^t_{p\to g}$,$x^{t-1}_{g\to p}$)
      if suc **then**
        $V' \leftarrow$ getTacticOutput()
        if $\exists j \in J, x^t_{p\to g}[j] \notin \Omega'^g_j$ **then**
          if due **then**
            $V' \leftarrow$ setUltimatum($x^t_{p\to g}$,Ω'^g_j)
            $x^{t+1}_{g\to p} \leftarrow$ newOffer(V',Solicited)
          else
            $x^{t+1}_{g\to p} \leftarrow$ newOffer(V',Advisory)
          end if
        else
          if due **or** $C_p(x^t_{p\to g}) > C_p(V')$ **then**
            $x^{t+1}_{g\to p} \leftarrow$ newOffer($x^t_{p\to g}$,Accept)
          else
            $x^{t+1}_{g\to p} \leftarrow$ newOffer(V',Advisory)
          end if
        end if
      else
        $x^{t+1}_{g\to p} \leftarrow$ newOffer($x^{t-1}_{g\to p}$,Solicited)
      end if
      sendNegotiationMsg(id_p, $x^{t+1}_{g\to p}$)
end while

concessions and tradeoffs (Section III-C); (ii) using the scoring value as the inputs of utility function; (iii) dynamically computing the probability of playing a tradeoff tactic in each round based on the amount of resources (i.e., $P_t = 1 - 1/N$, N is the number of candidate services providers available in the environment); (iv) adopting a greedy concession tactic where the degree of concession depends on the comparison of the current expectation and the received offer; (v) switching to concession tactic if tradeoff tactic cannot produce valid results that satisfy the negotiation constraints. We refer to this modified mixed negotiation strategy as UMI, the flowchart of concession and tradeoff tactic is shown in Figure 1.

C. Utility Function

The utility function is designed to measure the level of satisfaction on each negotiable terms. The utility function is

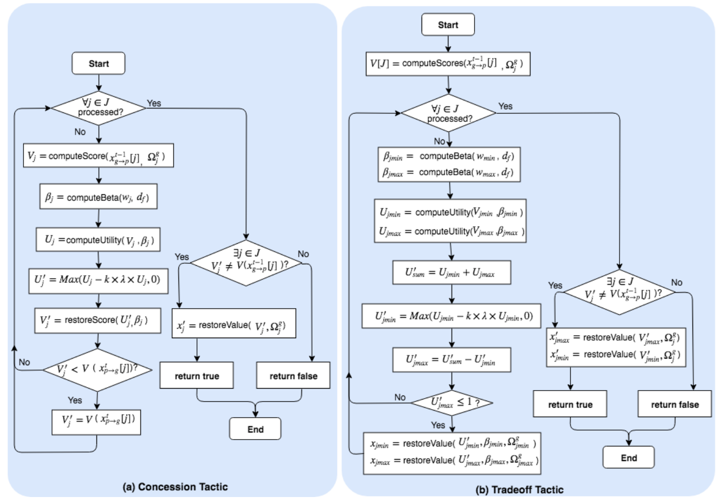

Fig. 1: Concession and tradeoff tactic in UMI

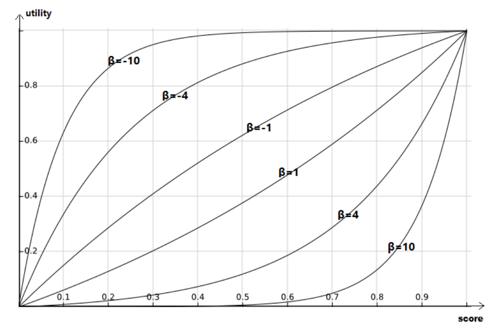

Fig. 2: Exponential utility function when $\tau = 0$

defined as following:

$$U^g_j(x^t_{p\to g}[j]) = \frac{ke^{\beta V^g_j(x^t_{p\to g}[j])} - 1}{e^\beta - 1} + 1 - k \qquad (7)$$

where β ($\beta \in \mathbb{R}$ and $\beta \neq 0$) determines the convexity degree of the curve, V^g_j is scoring value of term j computed by corresponding scoring function ($V^g_j \in [0, 1]$), k is the initial constant, which can be computed using the formula:

$$k = \frac{1 - e^\beta}{e^{-\beta\tau/\Delta j} - e^\beta} \qquad (8)$$

where τ is the tolerance value, and Δj is the length of negotiation space Ω^g_j (i.e., $\Delta j = max^g_j - min^g_j$).

In the utility function, β controls the concession rate. When $\beta < 0$, the concession rate is high in the beginning, but the concession rate decreases as the number of negotiation rounds increases; when $\beta > 0$, the concession rate is small in the beginning and increases as the number of negotiation rounds increases. Since a bigger concession rate accelerates the negotiation convergence, we compute β based on the negotiation desirability factor (DF, $0 < DF \leq 1$) and the attribute weights w_j specified by the user:

$$\beta = \begin{cases} 0.7e^{5(0.5-DF)} + 0.3e^{3w_j}, & DF \leq 0.5 \\ -0.7e^{5(0.5-DF)} - 0.3e^{3(1-w_j)}, & DF > 0.5 \end{cases} \qquad (9)$$

IV. Evaluation

A. Experimental Setup

To test our algorithm, we designed two types of providers: static providers and mobile providers. Under each type, we classified the providers based on the service level they can provide: high-performance (HP), moderate-performance (MP), and low-performance (LP). We define θ as the degree of intersection (DoI) between the negotiation spaces of providers and consumers, which is equal to 0.7, 0.4 and 0.2 for HP, MP and LP providers respectively. For mobile providers, we define θ_{loc} as the probability to accept the requested location, which is equal to 0.9, 0.5, 0.2 for HP, MP and LP providers respectively, and the price is linearly dependent on the standard Euclidean distance between the current offering and the requested properties. For static providers, the HP, MP, and LP providers have six, four and two service instances uniformly distributed in the simulated area respectively, and the price is defined with a range if it is negotiable (PIN), or a static value if it is non-negotiable (PNN). For each consumer, the requested location is randomly assigned within the simulated area.

We created an SLAT for a hazardous gas detection service based on concepts proposed in IoT literature [24][25], which specifies the service type, operation parameters, service coverage, sampling parameters, price, and QoS attributes including accuracy, availability, and response time. Two test cases are defined to target different negotiation scale. Test case one has two candidate providers for each request: one mobile MP provider and one static MP provider. Test case two has twelve candidate providers for each request: six mobile providers (LP, MP, and HP) and six static providers (LP-PIN, LP-PNN, MP-PIN, MP-PNN, HP-PIN, HP-PNN).

Three metrics are used to evaluate the performance: negotiation efficiency, success rate, and the average response time. The efficiency is defined as the ratio of the negotiated cost performance and the initially expected cost performance. We compare the performance against four other tactics: the mixed strategy (UMC) that inspired our work [18], behaviour-dependent relative tit for tat (BDR), time-dependent linear tactic (TDL) and resource-dependent patient tactic (RDP) [14]. BDR, TDL, and RDP are chosen because they are pure tactics which shows a good performance in terms of utility and successful deals. Considering providers may be resource-constraint devices, providers only play TDL while consumers play the five different tactics respectively. To reduce the chance variation, each test case is repeated for 100 times under two different values of desirability ($DF = 0.3$ and $DF = 0.9$).

B. Results

Results in Figure 3 shows the efficiency when the maximum negotiation round r equals to 20. UMI has no obvious advantage when the number of candidate service providers is small. As the number of candidate providers increases the efficiency in UMI improves compared to other tactics. We perform the single-factor ANOVA and Tukey's Method [26]

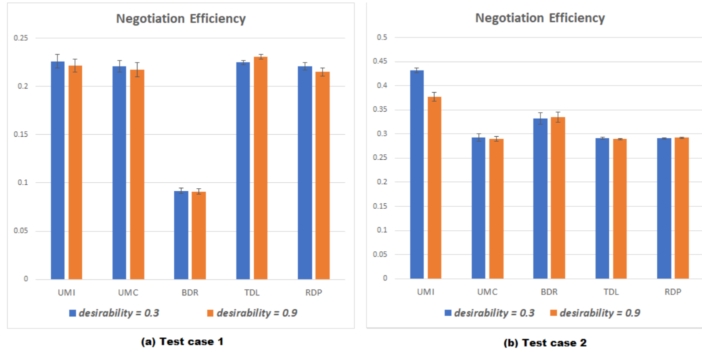

Fig. 3: Negotiation efficiency using different tactics ($r = 20$)

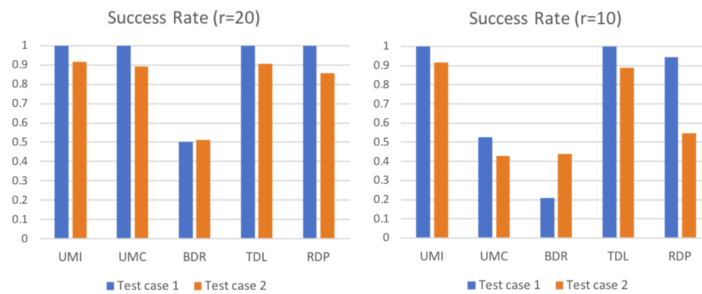

Fig. 4: Success rate ($DF = 0.3$)

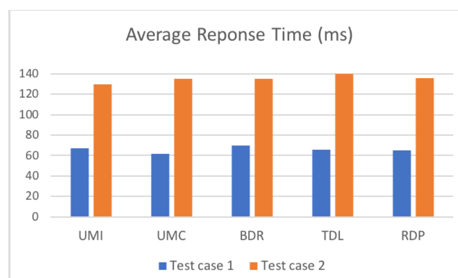

Fig. 5: Average response time ($DF = 0.3$, $r = 20$)

to determine if there was a significant difference between these five tactics. For test case 1, $F = 380.41$ and $p < 0.0001$ when $DF = 0.3$, and $F = 399.08$ and $p < 0.0001$ when $DF = 0.9$. From the 95% Tukey simultaneous confidence interval (CI), BDR is significantly different from other tactics, while the rest tactics have no significant statistical difference. For test case two, $F = 61$ and $p < 0.0001$ when $DF = 0.3$, the 95% simultaneous CI shows that UMI is significantly different from other tactics, while the rest tactics have no significant statistical difference. When $DF = 0.9$, $F = 30.87$ and $p < 0.0001$, the 95% simultaneous CI shows that UMI and BDP have no significant statistical difference, but they are significantly different from UMC, TDL, and RDP. Figure 4 shows the success rate when r decreases from 20 to 10 ($DF = 0.3$). UMI and TDL demonstrate a good and stable performance compared to other tactics regardless of the maximum negotiation rounds, while UMC and BDR are greatly impacted by the parameter, which makes them unsuitable for IoT service negotiation considering the massive message payload. Figure 5 shows that the five approaches have a similar performance in terms of responsiveness under different scenarios. The negotiation time increases significantly as more candidates providers existing in the environment. It suggests that predicting the negotiation result based on historical data and selecting the candidate providers that are most likely to make an agreement before actual negotiation may be useful to reduce the response time while keeping a good utility and success rate.

V. Conclusion and Future Work

This paper tackles automatic SLA negotiation for IoT services when consumers and service providers have a conflicting preference on service properties. Previous work on SLA negotiation focuses on cloud computing and web services, which does not consider the domain-specific properties of IoT services. In this paper, we introduced a negotiation strategy that considers the functional and non-functional service properties. In each negotiation round, we select the best offers by using different scoring functions that target different service properties and propose counteroffers based on these offers by performing a context-based negotiation tactic. To increase the success rate and utility, we designed an exponential utility function and integrated a mixed negotiation strategy into the WS-Agreement Negotiation protocol to guarantee a successful negotiation when providers and consumers have intersected negotiation space. The evaluation result shows that the proposed tactic produces a higher utility compared to other existing approaches, and maintains a high success rate when the number of interactions is limited.

As IoT is a large-scale environment with energy-constrained devices, time and overhead can be a key requirement for SLA negotiation. In future work, the intelligent opponent learning-based strategy may be useful to reduce the response time and negotiation rounds in multi-attribute SLA negotiation.

Acknowledgment

This work is supported by Science Foundation Ireland (SFI) under the project SURF - grant 13/IA/1885.

References

[1] S. Meyer, A. Ruppen, and C. Magerkurth, "Internet of things-aware process modeling: integrating iot devices as business process resources," in *International conference on advanced information systems engineering*. Springer, 2013, pp. 84–98.

[2] B. Kantarci and H. T. Mouftah, "Sensing services in cloud-centric internet of things: A survey, taxonomy and challenges," in *Communication Workshop (ICCW), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1865–1870.

[3] A. Elfatatry and P. Layzell, "Negotiating in service-oriented environments," *Communications of the ACM*, vol. 47, no. 8, pp. 103–108, 2004.

[4] P. Swiatek and A. Rucinski, "Iot as a service system for ehealth," in *e-Health Networking, Applications & Services (Healthcom), 2013 IEEE 15th International Conference on*. IEEE, 2013, pp. 81–84.

[5] H. Ludwig, A. Keller, A. Dan, R. P. King, and R. Franck, "Web service level agreement (wsla) language specification," *IBM Corporation*, pp. 815–824, 2003.

[6] K. Saravanan and M. Rajaram, "An exploratory study of cloud service level agreements-state of the art review." *KSII Transactions on Internet & Information Systems*, vol. 9, no. 3, 2015.

[7] D. Kyriazis, "Cloud computing service level agreements–exploitation of research results," *European Commission Directorate General Communications Networks Content and Technology Unit, Tech. Rep*, vol. 5, p. 29, 2013.

[8] A. Palade, C. Cabrera, F. Li, G. White, M. Razzaque, and S. Clarke, "Middleware for internet of things: an evaluation in a small-scale iot environment," *Journal of Reliable Intelligent Environments*, pp. 1–21, 2018.

[9] F. Marino, C. Moiso, and M. Petracca, "Automatic contract negotiation, service discovery and mutual authentication solutions: a survey on the enabling technologies of the forthcoming iot ecosystems," *Computer Networks*, 2018.

[10] N. Gupta, N. Kumar, and S. Jain, "Coverage problem in wireless sensor networks: A survey," in *Signal Processing, Communication, Power and Embedded System (SCOPES), 2016 International Conference on*. IEEE, 2016, pp. 1742–1749.

[11] K. Mišura and M. Žagar, "Negotiation in internet of things," *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije*, vol. 57, no. 2, pp. 304–318, 2017.

[12] G. Gaillard, D. Barthel, F. Theoleyre, and F. Valois, "Service level agreements for wireless sensor networks: A wsn operator's point of view," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*. IEEE, 2014, pp. 1–8.

[13] E. Mingozzi, G. Tanganelli, and C. Vallati, "A framework for qos negotiation in things-as-a-service oriented architectures," in *Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2014 4th International Conference on*. IEEE, 2014, pp. 1–5.

[14] P. Faratin, C. Sierra, and N. R. Jennings, "Negotiation decision functions for autonomous agents," *Robotics and Autonomous Systems*, vol. 24, no. 3-4, pp. 159–182, 1998.

[15] Y. Yao and L. Ma, "Automated negotiation for web services," in *Communication Systems, 2008. ICCS 2008. 11th IEEE Singapore International Conference on*. IEEE, 2008, pp. 1436–1440.

[16] F. H. Zulkernine and P. Martin, "An adaptive and intelligent sla negotiation system for web services," *IEEE Transactions on Services Computing*, vol. 4, no. 1, pp. 31–43, 2011.

[17] K. Misura and M. Zagar, "Internet of things cloud mediator platform," in *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*. IEEE, 2014, pp. 1052–1056.

[18] X. Zheng, P. Martin, K. Brohman, and L. Da Xu, "Cloud service negotiation in internet of things environment: A mixed approach," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1506–1515, 2014.

[19] K. M. Sim, Y. Guo, and B. Shi, "Blgan: Bayesian learning and genetic algorithm for supporting negotiation with incomplete information," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 198–211, 2009.

[20] V. Narayanan and N. R. Jennings, "Learning to negotiate optimally in non-stationary environments," in *International Workshop on Cooperative Information Agents*. Springer, 2006, pp. 288–300.

[21] M. F. Abulkhair, E. S. Alkayal, and N. R. Jennings, "Automated negotiation using parallel particle swarm optimization for cloud computing applications," in *Computer and Applications (ICCA), 2017 International Conference on*. IEEE, 2017, pp. 26–35.

[22] O. Waeldrich, D. Battré, F. F. Brazier, K. Clark, M. Oey, A. Papaspyrou, P. Wieder, and W. Ziegler, "WS-Agreement Negotiation Version 1.0," p. 64, 2011.

[23] X. Jin, S. Chun, J. Jung, and K.-H. Lee, "Iot service selection based on physical service model and absolute dominance relationship," in *Service-Oriented Computing and Applications (SOCA), 2014 IEEE 7th International Conference on*. IEEE, 2014, pp. 65–72.

[24] V. Tsiatsis, A. Gluhak, T. Bauge, F. Montagut, J. Bernat, M. Bauer, C. Villalonga, P. Barnaghi, and S. Krco, "The sensei real world internet architecture," 2010.

[25] C. Villalonga, M. Bauer, F. L. Aguilar, V. A. Huang, and M. Strohbach, "A resource model for the real world internet," in *European Conference on Smart Sensing and Context*. Springer, 2010, pp. 163–176.

[26] G. Snedecor and W. Cochran, "Statistical methods, ed. 6, ames, iowa, 1967," *Iowa State University Press, Section*, vol. 12, pp. 349–352.