A Multi-tiered Social IoT Architecture for Scalable and Trusted Service Provisioning

Giancarlo Sciddurlo*†, Ingrid Huso*, Domenico Striccoli*†,
Giuseppe Piro*†, Gennaro Boggia*†
*Dept. of Electrical and Information Engineering - Politecnico di Bari, Bari, Italy,
†CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni
Email: {name.surname}@poliba.it

***Abstract*—In the Social Internet of Things paradigm, the Trust Management System computes trust values of involved social objects, identifies trusted relationships, and selects the most suitable object able to provide a target service. State-of-the-art mechanisms conceived to address these tasks generally avoid considering the actual availability of social objects and demand the implementation of complex algorithms to constrained nodes. This work presents a novel multi-tiered and fog-based Social Internet of Things architecture to solve these open issues, ensuring fast service provisioning, high scalability, fault tolerance, and security. On the one hand, the Trust Management System hosted at the first fog layer of the architecture jointly addresses the trustworthiness of service providers and monitors the resource availability exposed by social objects, thus simplifying the forwarding of service requests to trusted and unloaded nodes. From another hand, to securely implement advanced services at a large scale, a second fog layer exploits a Blockchain-based storage for sharing services, relationships, and trust values across organizations and service domains. Computer simulations demonstrate the effectiveness of the proposed architecture in a realistic Social Internet of Things while showing the performance gain obtained against a baseline approach.**

***Index Terms*—Social Internet of Things, Trust Management System, multi-tiered architecture, resource management**

I. Introduction

The Social Internet of Things (SIoT) paradigm recently born thanks to the promising integration of Social Network capabilities in the Internet of Things (IoT) domain [1]. By autonomously generating social relationships, smart objects can improve resource visibility, object reputation assessment, and service discovery [2] [3]. In a typical SIoT deployment, the Trust Management System (TMS) is the logical entity that evaluates the behavior of social objects and dynamically assigns them trust values through automatic mechanisms. Then, identifying trusted relationships supports the selection of the most suitable object able to supply a given request [4]. This latter task is referred to as *service provider selection*.

The scientific literature already formulated different methodologies addressing these key functionalities. Most of the solutions, including those presented in [5]–[10], implements the service provider selection without considering the availability of the actual resources. Consequently, requests may be frequently directed to social objects with higher trust values, favoring network congestion episodes and increasing latencies. Furthermore, some other valuable contributions expect to implement trust computation and service provider selection directly in the SIoT nodes [5]–[7], [10], [11]. Nevertheless, as explicitly highlighted [12], this represents an evident drawback for SIoT devices with limited storage and computation capabilities. Indeed, to the best of the authors' knowledge, the design of a more effective SIoT architecture able to jointly address these issues still represents an uncovered research goal.

To bridge this gap, the work presented herein conceives a novel multi-tiered SIoT architecture, where key functionalities are properly implemented to guarantee low latency, high scalability, fault tolerance, and security. Specifically, the lower level of the architecture embraces physical objects and their logical abstractions, exposing resources and services. The TMS entity, hosted at the first fog layer of the architecture, jointly addresses the trustworthiness of service providers under its control and the monitoring of the availability of resources exposed by related social objects. In this way, it can support an effective service provider selection without burdening on the constrained capabilities of social objects and preventing the network from blocks and slowdowns. Blockchain shares available services, relationships, and trust values across organizations and service domains at the second fog layer of the architecture. This allows to securely extend the boundaries of offered novel applications also at a large scale. The effectiveness of the proposed approach is investigated through computer simulations in a realistic SIoT scenario. The performance gain with respect to the baseline solution that does not leverage the conceived enhanced functionalities for the TMS is evaluated as well. Obtained results demonstrate that the proposed approach can serve incoming requests faster while guaranteeing a trusted and scalable service provisioning.

The rest of the paper is organized as follows. Section II reviews the state-of-the-art on TMS mechanisms for the SIoT. Section III illustrates the designed architecture. Section IV presents the conceived TMS functionalities. Section V investigates the performance of the proposed approach. Finally, Section VI concludes the paper and draws future research activities.

II. Related works

Social relationships are at the basis of the SIoT. The contributions [1] and [13] classified them through different

categories to promote trustworthy interactions in a service-oriented environment:

- Ownership Object Relationship (OOR): established among objects belonging to the same owner;
- Parental Object Relationship (POR): established among objects that are part of the same family and generally produced by the same manufacturer;
- Co-Work Object Relationship (C-WOR): established among objects working together for a common goal or in the same application;
- Co-Location Object Relationship (C-LOR): established among objects always located in the same place;
- Social Object Relationship (SOR): established among objects without common attributes or characteristics coming into contact because their owners come in contact or have a social relationship.

As a result, to generate any form of relationship, each social object must verify some conditions, such as the examination of the owner profile (OOR and POR), the geographical position (C-LOR), and the operational context (C-WOR and SOR).

A social TMS in charge of evaluating and managing the trustworthiness of social objects was introduced, for the first time, in [5]. That study investigated a centralized architecture and identified its main deployment issues (e.g., single point of failure, low scalability). From now on, the scientific literature proposed many other SIoT system architectures, discussing the design of recommendation schemes based on the trust evaluation and defining different strategies aiming to offer an appropriate service provider selection through the TMS.

For example, the paper [6] faces the service provider search among nodes in a distributed manner with a new approach in a fast and autonomous way. The proposed strategy allows reaching the suitable provider, considering the energy constraints of nodes to increase the network lifetime. However, it neglects the aspects of load balancing, storage-saving, and the management of service requests that offer high scalability to the network. The work proposed in [7] defines functions and parameters to compute competence and willingness to quantify a trust value in a SIoT environment. Nevertheless, the entire algorithm computation for trust value is in charge of the IoT devices, not optimizing the computational loads. The contribution in [11] provides a scheme of access service recommendation for the SIoT, addressing both load balancing and network stability aspects. Here, within a distributed architecture, each node stores the profiles of the other nodes involved in the network. However, the nodes involved may not have a sufficient storage capacity to keep track of the whole set of information needed. The authors in [8] propose a service-based grouping decentralized architecture for SIoT network as an approach to reduce the service discovery time. They exploit fog computing technology to boost the computational system capability. Nonetheless, the proposal does not provide any secure distributed storage technology for the management of social relationships.

The studies [10] and [9] present a blockchain-based trustful architecture for information spreading in SIoT environments. The described models provide a secure and transparent mechanism for trust evaluation. However, the first proposes efficient interactions to find the most suitable service provider in the network without considering any factor related to the employment of device resources. The latter, instead, presents an algorithm that exploits the information entropy to increase the system security but turns out to be effective only for specific time intervals.

Unfortunately, none of the studies discussed so far presents a well-defined paradigm that jointly embraces all the aspects of efficient resource management, scalability, and reliability of the Social Network, as well as trustworthiness and resource availability of service providers in SIoT environments.

III. The conceived SIoT architecture

Fig.1 depicts the novel SIoT architecture proposed in this work. It leverages a multi-layered decentralized configuration based on fog computing technology. Such a configuration allows improving efficiency, increasing responsiveness, and reducing the computational loads of the network nodes by exploiting the higher computational capability of the fog nodes.

The lower layer of the architecture, namely SIoT layer, menages objects virtualization. Social objects reproduce the digital counterparts of physical IoT devices while also collecting social skills not explicitly supported in the real world. The attributes that identify a social object and characterize its profile are:

- Device ID, which represents a device unique identifier;
- Owner ID, an identifier of the owner of the device;
- Manufacturer ID, useful to define the device manufacturer;
- Context, which indicates the type of task or service that the device can perform. A device can have more context-related identifier values, depending on the number of tasks/services it can accomplish.
- Resource capabilities, that indicates the resources a device can employ to provide services;
- Master node list, which indicates the set of master nodes responsible for managing all information related to the device;
- Friend list, which stores all the relationships identified by a social object within the Social Network.

Master nodes constitute the first sub-layer of the fog layer, namely fog sub-layer 1. The primary role of the fog sub-layer 1 is to perform the TMS for the management of service requests. A social object can act either as a service requester and a service provider. Moreover, to encourage service discovery, they are grouped into service communities according to the service they can provide. It is supposed that they can provide more than one type of service, thus belonging to more than one service community at the same time. In turn, each community is managed by a master node, which handles the Social Network of providers for the specific service, potentially generating a virtual topology for each service community. After establishing social relationships, a social object communicates

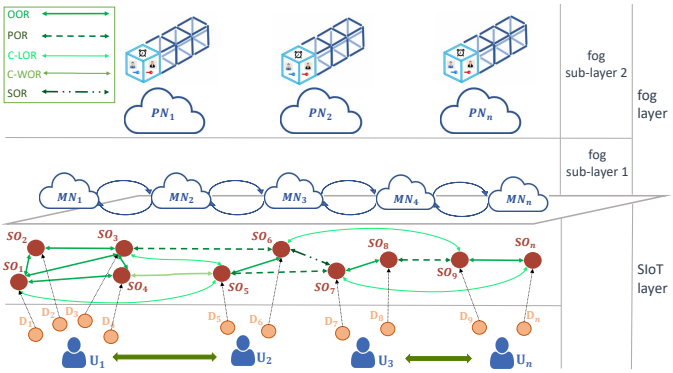


Fig. 1. The proposed SIoT architecture.

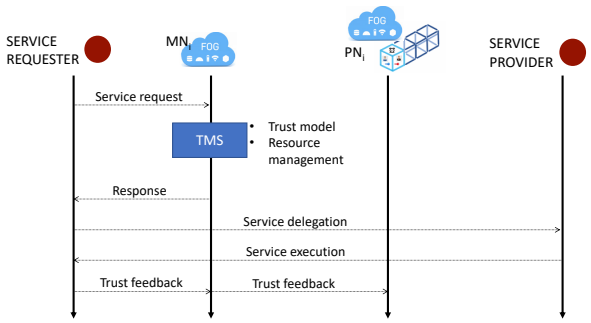


Fig. 2. The service provisioning procedure.

to the network its availability to perform services. Accordingly, it searches an existing community characterized by the services that it can provide in the master nodes. If it cannot join any of the existing service communities, it creates a new one. Since the SIoT encompasses several services, a service-based grouping approach strongly minimize the latencies in the service discovery procedure [8].

The fog sub-layer 2 interacts with the fog sub-layer 1 below. It deploys the primary nodes, characterized by high storage capacities, storing all the information related to social objects profiles, social relationships, and reputations on a distributed database. Such information pool is hosted on a Blockchain, enabling privacy and security for the stored information defining the SIoT environment. Adopting a Blockchain in this type of framework ensures a strong and secure traceability of the nodes, supporting the identification process of the most suitable social object to provide a service with an high degree of trustworthiness. Furthermore, such a hierarchical, distributed, and decentralized approach turns out to be of fundamental importance for the TMS execution since it allows increasing scalability and efficiency. Indeed, supposing that the information related to the reputation of a service requester is not available at the master node, it can be anyhow retrieved from the Blockchain on the primary node.

IV. Details on the Conceived Service Provisioning Procedure

The fundamental objective of the proposed architecture is to improve network navigability and boost the service search process. This is done by carefully considering service communities and social relationships settling the Social Network of objects. Most scientific works in this context perform this task by focusing on service providers' trustworthiness, mainly associated with evaluating the users' behavior. Differently, the strategy proposed herein goes one step further by jointly investigating the service trustworthiness and resource consumption assessment, thus promoting the service discovery beyond reliability and security.

Figure 2 shows the overall service provisioning procedure. A social object issues a service request and sends it to the closest master node. If the master node does not manage the related service community, the request is forwarded to the master node able to process the request. Through this procedure, the suitable service provider identification would not be restricted to the limited knowledge of the requester or the fog node directly connected to it, allowing a global view of each service provider's trustworthiness. Through the trust model and the resource management functionality (further described below), the TMS determines a trust ranking of providers among the social objects which can potentially provide the requested service. Then, the most suitable provider in the ranking is selected for the service execution.

Finally, the service requester provides to the system its degree of satisfaction for the service received. The feedback is expressed with a value equal to 1 for service accomplished. A value equal to 0, instead, is given for a service not correctly completed. Both the fog sub-layers store the feedbacks for subsequent evaluations of the trust level.

TABLE I
Direct Social Factor Rate

Type of relationship	POR	OOR	C-LOR	C-WOR	SOR
SO_{ij}	0.9	0.8	0.7	0.6	0.5

A. Trust model

Given the i-th object requesting a service s_k and the j-th object exposing a service, the TMS calculates the trust value $T_{s_k}(i,j)$. In summary, $T_{s_k}(i,j)$ is defined through two main factors, which are the sociality factor and reputation.

The sociality factor, $S_f(i,j)$, rates the relationship established between the considered social objects by describing the degree of confidence in the case of both direct and indirect friendship (e.g., a friend of friends). In the case of direct friendship, it is set as $S_f(i,j) = SO_{ij}$, according to the type of social relationship (see Table I). In indirect friendship, instead, it is evaluated by considering the social objects' common friends. Precisely, assuming that the number of i and j common friends is equal to C, $S_f(i,j)$ is computed as: $S_f(i,j) = \frac{\sum_{c=1}^{C} SO_{jc}}{C}$, where SO_{jc} represents the direct social factor rate between j and its common friends with i.

On the other hand, the reputation, $R_{s_k}(i,j)$, represents the opinion on the trustworthiness of a service provider for the service s_k, based on past experiences through feedback values assigned to previous interactions among social objects. It is calculated as a linear combination of three different contributions:

- the direct feedback $\Delta_{s_k}(i,j)$ describes how the i-th requester evaluated the j-th provider for the service s_k in the past;
- the indirect feedback $\Theta_{s_k}(i,j)$ describes how the friends of the i-th requester evaluated the j-th provider for the service s_k in the past. Assuming that the considered requester has F friends, $\Theta_{s_k}(i,j)$ is computed as:

$$\Theta_{s_k}(i,j) = \frac{1}{F}\sum_{f=1}^{F}\Delta_{s_k}(f,j), \tag{1}$$

  where $\Delta_{s_k}(f,j)$ is the feedback given by the f-th friend of the i-th requester.
- the indirect non-friend feedback $\Pi_{s_k}(i,j)$ specifies how the other non-friend social objects evaluated the j-th provider for the service s_k. Assuming that the total number of non-friends that have previously evaluated the provider j is equal to P, $\Pi_{s_k}(i,j)$ is computed as:

$$\Pi_{s_k}(i,j) = \frac{1}{P}\sum_{\pi=1}^{P}\Delta_{s_k}(\pi,j), \tag{2}$$

Finally, the reputation factor is obtained as:

$$R_{s_k}(i,j) = \alpha\Delta_{s_k}(i,j) + \beta\Theta_{s_k}(i,j) + \gamma\Pi_{s_k}(i,j), \tag{3}$$

where α, β, and γ are the weights ($0 < \alpha,\beta,\gamma < 1$ and $\alpha+\beta+\gamma = 1$) that determine the relevance for each factor considered in the evaluation of the reputation.

To conclude, the trust value associated to the i-th object requesting a service s_k and the j-th object exposing a service, is obtained as:

$$T_{s_k}(i,j) = S_f(i,j)\cdot R_{s_k}(i,j). \tag{4}$$

B. Resource management

Leveraging the trust model, it is possible to recognize trusted social objects, discarding all providers below a configured threshold from the service provider selection. Moreover, by determining the trust value for each service provider, the master node obtains a ranking based on social object reliability.

Besides, the proposed TMS considers a further investigation addressing the resource capability of social objects. This contribution is restrictive for the trust evaluation, especially in an environment constituted by nodes with limited resources. In fact, in the case of several service requests assigned to the same social object, which entertains low resource capability, it would increase the risk of skipping the execution of the service due to a lack of available resources. As a matter of fact, its opportunity to provide the requested service decreases, causing possible congestion in the network. Hence, the master nodes monitor the status of the social objects and the resources required for the service execution. Precisely, after the ranking computation, the resource capacity of the candidate provider is monitored to verify the social object availability for the service execution. If this check fails, the candidate provider is temporarily dropped from the list. The master node updates the ranking and performs the same investigation on the new candidate until the service provider that meets the required resources consumption to execute the service is found.

TABLE II
RESOURCE CAPABILITY CLASSES [11].

Social object class	Resource Capability
Smartphone	0.8
Smart gateway	0.6
Smart camera	0.4
Sensor	0.2

TABLE III
SERVICES CHARACTERISTICS.

Service ID	1	2	3	4	5	6	7
Resource Consumption	0.3	0.2	0.2	0.1	0.1	0.2	0.1
Execution Time [s]	2	7	3	7	2	8	5

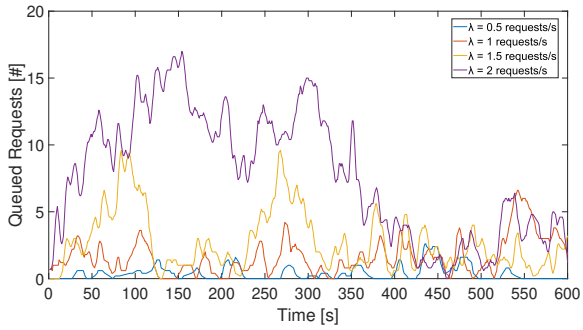
V. PERFORMANCE EVALUATION

The performance of the proposed SIoT is investigated herein through computer simulations. To this end, a MATLAB script is used to model a Social Network with heterogeneous objects, various traffic loads, together with all the procedures described in Section IV.

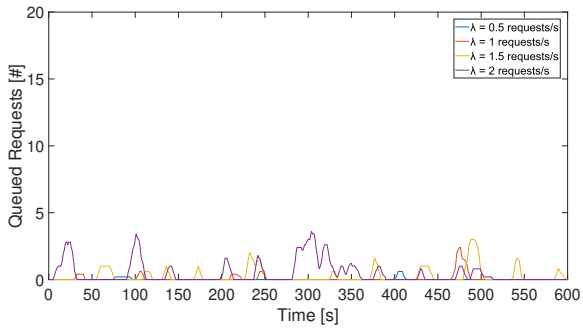
The proposed scenario considers a fog layer composed of five master nodes coordinated by a primary node for the service request management. The number of social objects ranges from 50 to 300. They are uniformly distributed among four computing classes (smartphones, smart cameras, sensors, and smart gateways). A social object randomly generates a service request according to a Poisson distribution with different λ values (from 0.5 to 2 requests/s), to simulate different traffic loads. According to what described in Section III, each social object is characterized by an ID, an owner ID, a manufacturer ID, the geographical position, a list of services it can provide, and its resource capability. In line with [11], resource capabilities are set as summarized in Table II. The Social Network is created by considering POR, OOR, and C-LOR relationships, based on the knowledge of owner, manufacturer, and geographical position attributes, respectively.

Seven different types of service communities are configured and distributed among all the master nodes. Each social object joins the proper service community handled by a master node, following the procedure explained in Section III. As reported in Table III, each service is identified by an ID, the resource consumption needed to be accomplished (spanning from 0.1 to 0.3), and the execution time (spanning from 2 s to 8 s).

Finally, the performance of the proposed approach is compared with the baseline architecture, where the TMS calculates the trustworthiness of social objects by taking into account

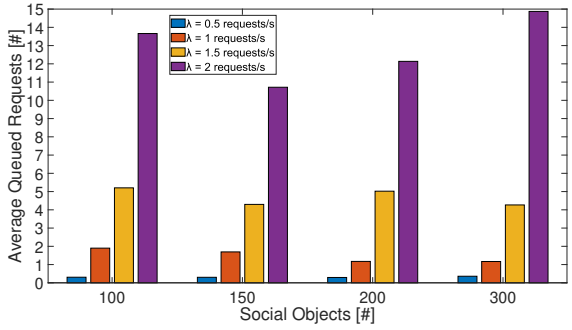


(a) Baseline approach

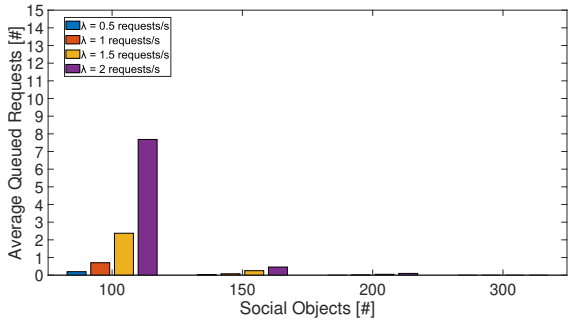


(b) Proposed approach

Fig. 3. Queued Requests Evaluation.



(a) Baseline approach



(b) Proposed approach

Fig. 4. Queued Request increasing traffic load.

relationships and reputation parameters without any resource control.

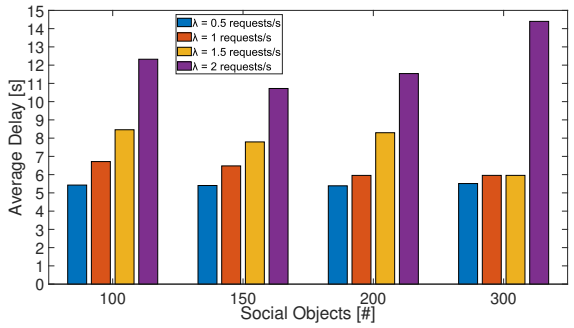
A. Simulation results

Figure 3 shows the number of queued requests during the time for different traffic loads. The results are obtained for a single scenario conducted on a social network with 150 social objects. For each λ, results are obtained over ten simulation runs to account for different network topologies and service distributions, and are averaged on a five seconds time window sliding by one second. Reported curves highlight the ability of the proposed approach to handle most of the requests in real-time: service requests are distributed to available trusted objects while preventing unpleasant queuing phenomena. On the contrary, the baseline approach distributes the requests without considering resource availability. As a consequence, most requests are relegated to a small set of trusted social objects, which monopolize scheduler assignments and overload their available resources in a short time. Therefore, a service provider selected to run a service by the TMS may not have sufficient resources, contributing to the formation of a queue of pending requests and, in turn, in a latency increase.

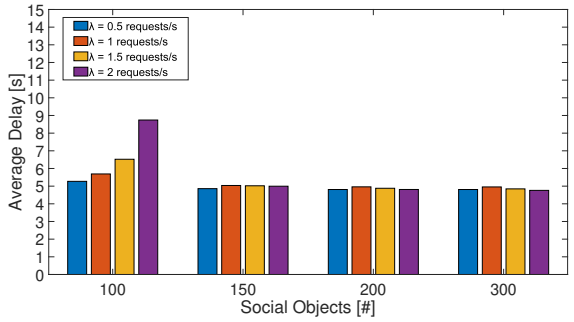
In order to generalize the afore discussed findings, Figure 4 shows the average queued requests for different traffic loads. Resource management allows minimizing the number of pending requests, thus improving network scalability. On the contrary, in the baseline approach, the network fails to handle large amounts of traffic, testifying the increase of queued requests and, consequently, the average delay in accomplishing them all. Moreover, the average queued requests scheduled in the proposed TMS on the data plane decrease when the number of nodes increases. It demonstrates the explicit scalability improvement differently from the baseline approach. Indeed, this allows the network to react effectively to a substantial traffic increase (i.e., from 0.5 requests/s to 2 requests/s) without overloading the resources of social objects.

Figure 5 depicts the average delay experienced for a request in the proposed and the baseline approach. Delay does not consider the time needed to exchange control messages or interactions between master nodes, since it is negligible if compared to the service execution time. Thanks to the intelligent management of the available resources of the social objects, the delay performance of the proposed approach are almost the same for high numbers of social objects, also outperforming the baseline approach, especially for high traffic loads. Unlike the baseline approach, the proposed test does not reveal any performance decay in terms of average delay in fulfilling requests. In fact, by increasing the number of nodes and the request rate λ, the average delay is still constant. The variations between the two approaches are more visible for most populated configurations. In fact, considering 300 social objects and an average request rate equal to 2 requests/s, the average delay experienced reduces up to 60%.

Finally, to provide further insight, Figure 6 shows the evolution over time of the feedback aggregation for six service providers. Three of them have been forced to act as malicious nodes, not accomplishing a service after a request and receiving negative feedback accordingly. Indeed, this



(a) Baseline approach



(b) Proposed approach

Fig. 5. Average Delay increasing traffic load.

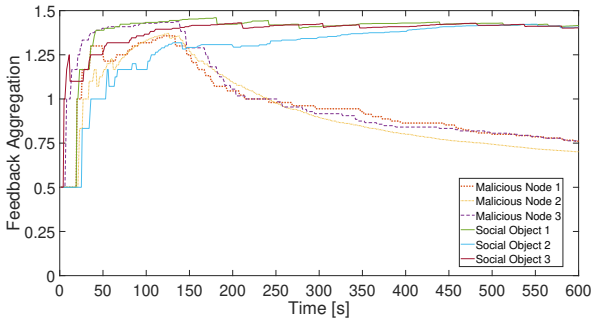


Fig. 6. Malicious social objects detection.

result testifies the capability of the conceived TMS to detect potential malicious nodes. In fact, after a warm-up period of about 100 s, the identification of malicious nodes appears unmistakable. Since the proposed model is reputation-based, a trustworthiness reduction due to negative feedback brings the system to choose only the trustworthy social objects for the scheduled requests in the service provider selection.

VI. Conclusions

This work presented a multi-tiered architecture to increase the responsiveness in service provisioning of social objects in SIoT environments. It leverages fog computing to achieve navigability, resource management, scalability, and reliability of the network. An efficient strategy for the trust evaluation of service providers has also been proposed, based on social factors, reputation, and resource availability.

Results show that the service requests handled by the conceived TMS allow reducing the average delay and the queued requests if compared to the baseline solution implemented in literature, at the same time excluding potentially malicious nodes from the network. Future research activities will explore a further investigation of the conceived approach, considering other quality-of-service parameters and network lifetime due to battery consumption. A future extension of this work will investigate how the proposed architecture could deal with the well-known attacker models present in the scientific literature.

Acknowledgements

This work was supported by the PRIN project no. 2017NS9FEY entitled “Realtime Control of 5G Wireless Networks: Taming the Complexity of Future Transmission and Computation Challenges” funded by the Italian MIUR, and by the GUARD project funded by the EU H2020 research and innovation programme under grant agreement 833456. It has been also partially supported by the Italian MIUR PON projects Pico&Pro (ARS01_01061), AGREED (ARS01_00254), FURTHER (ARS01_01283), RAFAEL (ARS01_00305).

References

[1] L. Atzori, A. Iera, G. Morabito, and M. Nitti, “The social internet of things (siot) – when social networks meet the internet of things: Concept, architecture and network characterization,” *Computer Networks*, vol. 56, 11 2012.

[2] L. Atzori, A. Iera, and G. Morabito, “From ”smart objects” to ”social objects”: The next evolutionary step of the internet of things,” *IEEE Communications Magazine*, vol. 52, no. 1, pp. 97–105, 2014.

[3] W. Z. Khan, M. Y. Aalsalem, M. K. Khan, and Q. Arshad, “When social objects collaborate: Concepts, processing elements, attacks and challenges,” *Computers and Electrical Engineering*, vol. 58, pp. 397–411, 2017.

[4] R. Faqihi, D. Ramakrishnan, and D. Mavaluru, “An evolutionary study on the threats, trust, security, and challenges in siot (social internet of things),” *Materials today: proceedings*, 11 2020.

[5] M. Nitti, R. Girau, and L. Atzori, “Trustworthiness management in the social internet of things,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1253–1266, 2014.

[6] A. Zannou, A. Boulaalam, and E. H. Nfaoui, “Siot: A new strategy to improve the network lifetime with an efficient search process,” *Future Internet*, vol. 13, no. 1, 2021.

[7] L. Wei, J. Wu, C. Long, and B. Li, “On designing context-aware trust model and service delegation for social internet of things,” *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4775–4787, 2021.

[8] B. Farahbakhsh, A. Fanian, and M. H. Manshaei, “TGSM: Towards trustworthy group-based service management for social IoT,” *Internet of Things*, vol. 13, p. 100312, 2021.

[9] Y. Yi, Z. Zhang, L. T. Yang, X. Deng, L. Yi, and X. Wang, “Social interaction and information diffusion in social internet of things: Dynamics, cloud-edge, traceability,” *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2177–2192, 2021.

[10] M. Amiri-Zarandi and R. A. Dara, “Blockchain-based trust management in social internet of things,” in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing*, 2020, pp. 49–54.

[11] Z. Chen, R. Ling, C.-M. Huang, and X. Zhu, “A scheme of access service recommendation for the social internet of things,” *International Journal of Communication Systems*, vol. 29, no. 4, pp. 694–706, 2016.

[12] R. K. Chahal, N. Kumar, and S. Batra, “Trust management in social internet of things: A taxonomy, open issues, and challenges,” *Computer Communications*, vol. 150, pp. 13–46, 2020.

[13] W. Z. Khan, Q. u. A. Arshad, S. Hakak, M. K. Khan, and Saeed-Ur-Rehman, “Trust management in social internet of things: Architectures, recent advancements and future challenges,” *IEEE Internet of Things Journal*, pp. 1–1, 2020.