



Article

Internet of Conscious Things: Ontology-Based Social Capabilities for Smart Objects

Michele Ruta ¹, Floriano Scioscia ^{1,*}, Giuseppe Loseto ², Agnese Pinto ¹, Corrado Fasciano ¹,
Giovanna Capurso ¹ and Eugenio Di Sciascio ¹

- ¹ Department of Electrical and Information Engineering, Polytechnic University of Bari, Via E. Orabona 4, I-70125 Bari, Italy; michele.ruta@poliba.it (M.R.); agnese.pinto@poliba.it (A.P.); corrado.fasciano@poliba.it (C.F.); giovanna.capurso@poliba.it (G.C.); eugenio.disciascio@poliba.it (E.D.S.)
- ² Department of Engineering, LUM “Giuseppe Degennaro” University, Strada Statale 100 km 18, I-70010 Casamassima, Italy; loseto@lum.it
- * Correspondence: floriano.scioscia@poliba.it; Tel.: +39-080-596-3054

Abstract: Emerging distributed intelligence paradigms for the Internet of Things (IoT) call for flexible and dynamic reconfiguration of elementary services, resources and devices. In order to achieve such capability, this paper faces complex interoperability and autonomous decision problems by proposing a thorough framework based on the integration of the Semantic Web of Things (SWoT) and Social Internet of Things (SIoT) paradigms. SWoT enables low-power knowledge representation and autonomous reasoning at the edge of the network through carefully optimized inference services and engines. This layer provides service/resource management and discovery primitives for a decentralized collaborative social protocol in the IoT, based on the *Linked Data Notifications* (LDN) over *Linked Data Platform on Constrained Application Protocol* (LDP-CoAP). The creation and evolution of *friend* and *follower* relationships between pairs of devices is regulated by means of novel dynamic models assessing trust as a usefulness reputation score. The close SWoT-SIoT integration overcomes the functional limitations of existing proposals, which focus on either social device or semantic resource management only. A smart mobility case study on Plug-in Electric Vehicles (PEVs) illustrates the benefits of the proposal in pervasive collaborative scenarios, while experiments show the computational sustainability of the dynamic relationship management approach.

Keywords: social internet of things; semantic web of things; edge intelligence; service discovery; smart mobility



Citation: Ruta, M.; Scioscia, F.; Loseto, G.; Pinto, A.; Fasciano, C.; Capurso, G.; Di Sciascio, E. Internet of Conscious Things: Ontology-Based Social Capabilities for Smart Objects. *Future Internet* **2024**, *16*, 327. <https://doi.org/10.3390/fi16090327>

Academic Editors: Domenico Ursino, Francesco Cauteruccio and Luca Virgili

Received: 16 July 2024

Revised: 2 September 2024

Accepted: 5 September 2024

Published: 8 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Smart cities and digital societies call for increasingly intelligent, distributed and flexible service and resource management. As cities evolve toward a living organism model on both physical and digital planes, ref. [1] pervasive computing infrastructures based on Internet of Things (IoT) sensors and smart devices act as its nervous system. The Edge Computing (EC) paradigm is increasingly adopted instead of first-generation architectures, strongly relying on Cloud Computing infrastructures: with the adoption of capable devices, several key data management and processing tasks can be performed at the edge of the network, thus achieving lower response latencies, more efficient bandwidth usage and higher data privacy and security. Advanced IoT scenarios require highly flexible composition and dynamic reconfiguration of services, resources and devices, each endowed with well-defined roles and responsibilities. Due to the high heterogeneity of vendors, device classes, as well as application areas, interoperability and decentralized autonomous decisions, are still challenging issues. In the latest years, a few interesting novel paradigms have been proposed, which may be applied at the EC level in IoT architectures. They typically exploit Artificial Intelligence to emulate aspects of human decision-making behaviors [2] in networked multi-agent systems. The Semantic Web of Things (SWoT) [3] vision merges

the Semantic Web (SW) and the IoT. Basically, a subset of useful knowledge representation and autonomous reasoning capabilities mutated from the Semantic Web are adapted and optimized for EC contexts. Its mission is to support user goals and provide innovative services by improving the intelligence of embedded objects and autonomic information management in pervasive contexts. Orthogonally, the Social Internet of Things (SIoT) paradigm [4] introduces a layer of social networking principles into the IoT ecosystem to enhance the efficiency, usability, and intelligence of interconnected devices. The benefits of SIoT architectures include:

- *Dynamic interactions*: smart objects are designed to interact not only based on pre-defined technical protocols but also through social relationships, akin to human social networks. Relationships mirror human social connections, such as co-location, co-work, friendship, and ownership, and facilitate more intuitive interactions independent from user engagement and from human Social Networking Services (SNSs);
- *Context-aware collaboration*: IoT devices act as autonomous agents building networks of social relationships and exploiting them to share information and services, and perform tasks collaboratively [5]. The social layer adds context and relational data to the interactions, enabling devices to make more informed and context-aware decisions;
- *Resource optimization*: by leveraging social relationships, smart devices can optimize the allocation and negotiate access to shared resources, like network bandwidth or energy power, based on their social standing or priority within the network.

SIoT approaches have found significant applications in real-world scenarios, including smart mobility [6], smart building [7], and smart manufacturing [8]. Nevertheless, ensuring interoperability among diverse devices from different manufacturers remains a critical challenge for the seamless functioning of SIoT networks. Monitoring and maintaining social relationships among a large number of devices is a challenging task requiring sophisticated algorithms and frameworks. The analysis in Section 3 highlights the benefits and drawbacks of existing SIoT proposals. Specifically, to overcome these limitations, this paper proposes a comprehensive framework, aiming to combine the best of SWoT and SIoT paradigms for EC-oriented scenarios. SWoT enables knowledge representation and autonomous reasoning at the edge of the network through carefully optimized inference services and engines. Specifically, non-standard inferences for semantic matchmaking are adapted for service/resource discovery, by defining a decentralized collaborative social protocol, through which smart objects can interact with their *friends* and *followers* along well-defined patterns in order to search for needed services, based on articulated request descriptions with well-defined semantics. The outcomes of discovery, and particularly the ability (respectively, inability) of an object to provide useful information to another one, contribute to strengthening (resp. weaken) their social links over the course of time, such that low-utility relationships can be dropped and new ones can be formed, based on a fully dynamic model. This allows coping with the volatile nature of IoT networks, where devices and active services can appear or disappear without notice.

The main contributions of the paper are as follows:

- A novel framework combining a SWoT-based distributed collaborative service discovery protocol with a SIoT dynamic relationship and trust management model. This work extends the early proposal in [9] defining a completely renewed approach for social interactions among smart objects; to the best of our knowledge, existing works have focused on either one or the other aspect, but not on both together or on their mutual influence, incurring in functional limitations with respect to the proposed approach, as discussed in Section 3;
- a decentralized collaborative service discovery protocol [9] based on *Linked Data Notifications (LDN)* over *LDP-CoAP* [10], which combines the *Linked Data Platform (LDP)* [11] for semantically annotated resource management and the *Constrained Application Protocol (CoAP)* [12] for Machine-to-Machine (M2M) interactions; this explicit and

standard-based formalization of interaction primitives improves on existing research proposals for SIoT architectures based on custom interaction models;

- A completely novel dynamic model for SIoT relationship management, measuring trust with a reputation system based on the usefulness of service suggestions in the decentralized collaborative social discovery protocol with additional features to prevent device isolation due to a “cold start” situation (i.e., when a device joins a new network where it has no friends) or to network partitioning after a link removal; this zero-configuration approach is an improvement with reference to most existing SIoT proposals;
- A case study on a Plug-in Electric Vehicle (PEV) charging service marketplace, to illustrate the features and benefits of the proposed approach for Mobility-as-a-Service (MaaS) in a smart city scenario;
- An experimental evaluation assessing the efficiency and effectiveness of both the dynamic relationship management and the collaborative service discovery in device networks of various sizes and configurations.

The remainder of the paper is as follows. Section 2 recalls basics of SWoT knowledge representation and reasoning, whereas Section 3 discusses related work. Section 4 describes the overall framework, followed by the case study presentation in Section 5. Experiments are presented in Section 6, before the conclusions.

2. Fundamentals of the Semantic Web of Things

The convergence of SIoT and SWoT technologies marks a significant advancement in the development of intelligent and context-aware IoT ecosystems. Combining the social relationships and interactions of SIoT with the rich, machine-interpretable metadata annotated by means of knowledge representation technologies significantly boosts the interoperability of IoT systems, allowing devices from different manufacturers and with varying protocols to interact with each other seamlessly. In the SWoT, knowledge representation involves the formalization of information in order to enhance the ability of interconnected devices to understand, reason, and act based on the rich contextual and relational information inherent in distributed networks. Semantic Web languages and technologies grounded on Description Logics (DLs) [13] are adopted to facilitate sharing and understanding of unambiguous knowledge across heterogeneous ubiquitous micro-devices. When SIoT leverages the SWoT, it benefits from enhanced interoperability through the use of common semantic frameworks and vocabularies. Specifically, this work relies on a general framework for the SWoT [14] based on an evolution of the classical Knowledge Base (KB) model, named a *ubiquitous Knowledge Base* (u-KB). A u-KB represents a distributed knowledge base whose individuals (i.e., assertional knowledge) are physically tied to micro-devices on the objects disseminated in a reference environment. As shown in Figure 1, the following key elements are exploited in the proposed approach for implementing the reference SWoT framework in a social IoT context:

- Pervasive sensing and IoT identification technologies exploited at the field layer, interconnecting embedded micro-devices dipped in the environment. Interaction among social devices is performed by means of CoAP protocol ensuring lightweight communication and information dissemination;
- Data acquired by sensing devices exposed at the upper layers in a uniform fashion, according to LDP and LDN guidelines described in Section 4.1. Data are annotated with respect to well-known ontologies and taxonomies, providing a shared understanding of domain knowledge. By integrating these semantic structures, SIoT devices can communicate more effectively, using a common language to describe their functions, states, and interactions;
- u-KB modeling in a *Web Ontology Language (OWL) 2* [15] subset corresponding to the *Attributive Language with unqualified Number restrictions (ALN)* DL. It provides moderate expressiveness while granting polynomial complexity to both standard and non-standard inference tasks;

- Dynamic service/resource discovery and composition processes, detailed in Section 4.2, exploiting semantic matchmaking between ontology-based annotations describing requests and available resources.

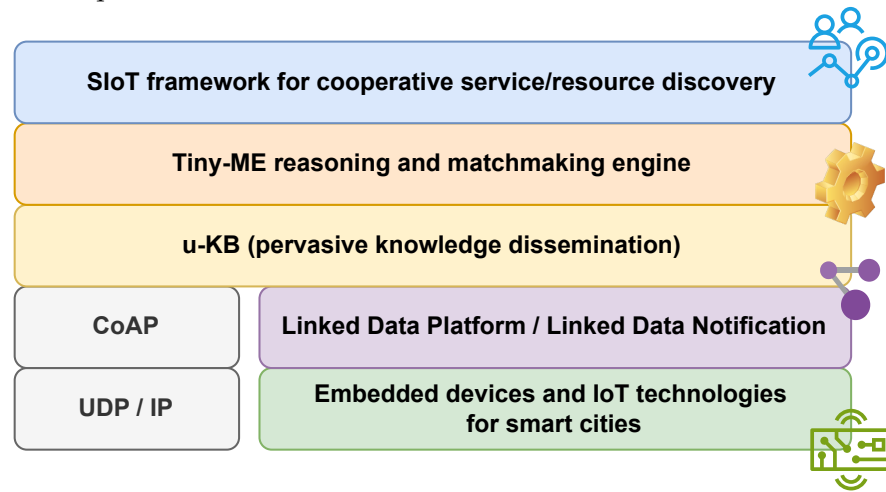


Figure 1. Semantic Web of Things architecture for SIoT.

The following non-standard inference services [16], implemented in *Tiny-ME (the Tiny Matchmaking Engine)* [17], have been adopted to support approximate matches, resource ranking and aggregation for covering complex requests:

- *Concept Contraction*: if a request \mathcal{R} and a supplied resource \mathcal{S} are not compatible with each other, Contraction determines which part of \mathcal{R} is conflicting with \mathcal{S} . By giving up only conflicting requirements G (for *Give up*) in \mathcal{R} , an expression K (for *Keep*) remains, which is a contracted version of the original request. The solution G to Contraction represents “why” \mathcal{R} and \mathcal{S} are not compatible;
- *Concept Abduction*: if request and resource are compatible, but \mathcal{S} does not satisfy \mathcal{R} completely, Abduction determines what should be hypothesized in \mathcal{S} in order to obtain a full match, i.e., to make the subsumption relation true. The solution H (for *Hypothesis*) to Abduction can be interpreted as what is requested in \mathcal{R} and not specified in \mathcal{S} ;
- *Concept Covering*: in advanced distributed systems it is often useful to aggregate multiple low-complexity resources $\mathbf{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_n\}$ in order to satisfy an articulate request \mathcal{R} . The (Abduction-based) Covering service finds a pair $\langle \mathcal{S}_c, H \rangle$ where $\mathcal{S}_c \subseteq \mathbf{S}$ includes concepts compatible with \mathcal{R} whose conjunction covers \mathcal{R} as much as possible; H is the residual part of \mathcal{R} possibly not covered by concepts in \mathcal{S}_c .

Finally, Concept Contraction and Abduction also induce logic-based *penalty* functions, as metrics of semantic distance between \mathcal{R} and \mathcal{S} [16].

3. Related Work

The SIoT aims at extended device collaboration. *Service Oriented Architecture (SOA)* is one of the dominant paradigms, where intelligent service/resource discovery is a core capability [5]. Most SIoT approaches for ranking services/resources with reference to a request/profile specification are based on (i) object social connectivity, (ii) object mobility patterns and/or (iii) preference similarity. This work exploits (i) and (iii), while (ii) is taken into account implicitly through dynamic relationship set-up and tear-down. In [18] a cosine similarity was computed combining metrics (ii) and (iii), improving results with reference to each single metric, although preferences and resources were described by means of a simplistic set of keywords, lacking proper semantics. Similarly, ref. [19] presented an automated service discovery process for mobile crowdsourcing applications. A large-scale IoT network was split into multiple virtual communities, grouping IoT devices that share common characteristics modeled using social relations. A Natural Language Processing

(NLP) approach handled crowdsourcing textual requests and retrieved the list of IoT devices capable of accomplishing the requested tasks, although semantic processing was completely missing.

Semantics-based approaches improve context-awareness as well as discovery capabilities in multi-agent systems [20]. In [21], a distributed OWL Knowledge Base (KB) management system has been exploited to enhance M2M social interactions in control networks. Unfortunately, supported inferences were very basic, limiting the applicability of the proposal. In [22], semantics-based situation detection and goal retrieval were exploited for matchmaking with task profiles for recommending activities to users, based on their current context. Nevertheless, social interactions occurred only between devices and users; furthermore, adopted rule-based reasoning could not retrieve approximate matches when exact ones did not exist. Likewise, ref. [23] focused on activities of daily living in smart homes, generating task-oriented recommendations for user's situational goals. A coastal monitoring system has been proposed in [24] leveraging the *Lysis* [25] SIoT platform to virtualize physical IoT devices, supporting social relationships as well as semantic resource descriptions and queries in order to establish automatic device cooperation. A similar approach was adopted in [26] for Social Internet of Industrial Things scenarios. In both works, an ontology-based model describes social objects, but no formal framework is provided to exploit it for resource/service discovery. In [27], Near-Field Communication (NFC) technology-mediated social interactions between everyday objects and agents running on mobile devices. Ontology-based representations enabled both reactive and proactive behaviors by means of rule-based reasoning for planning toward situation-aware goals. Though interesting, the approach does not appear general enough to support a wide range of SIoT scenarios. A cognitive reasoning-based framework for SIoT services discovery in smart spaces was proposed in [28]. Unfortunately, the dynamic evolution of social relationships between smart object pairs was not exploited for service recommendation in the reasoning phase. Similarly, the LDP-CoAP framework in [9] provides a social collaborative protocol for resource discovery based on standard Semantic Web technologies, but relationships among smart objects are essentially static.

Social connectivity management leverages the properties of things relationships. Several classifications exist in the literature. In [29] and subsequent works, *parental* (same manufacturer), *co-location* (same environment), *co-work* (cooperation), *co-ownership* (same owner) and *social* (sporadic or frequent contact) were identified as basic object relationships. Similarly, ref. [26] defined four types of relationships between industrial device assets: *parental* (component-of), same functionality, replacement, and cooperation; friend discovery, however, was based on a purely syntactic match of attributes. The analysis in [30] was inspired by the literature on human SNS, instead, and resulted in an ontology allowing objects to manage their policies, friends and reputation. The ontology model developed in [31] included social relationships among events, people and objects in IoT environments. The *SIoTpredict* framework [32] predicted future relationships among devices in a dynamic and heterogeneous SIoT network with a Bayesian nonparametric learning model by exploiting the trajectories of IoT objects and their corresponding location.

Trust management is one of the most relevant issues in the SIoT [33,34]. In this work, the focus is on application-level trust, particularly in social object relationships. Trust metrics can be retrieved from friends (recommendation), reliable intermediaries (reputation) or direct knowledge [5,35]. Ref. [36] introduced a *subjective* and an *objective* (i.e., shared) model, exploiting techniques mutated from peer-to-peer networks. The collaborative cognitive radio framework in [37] leveraged the SIoT relationship types described above in [29] to improve both the reach of channel status queries and the trustworthiness of information through weights depending on the degree of friendship. A refinement was proposed by [38], where trust was defined as a function of reputation on past transactions, relationship type and energy status. Reputation combined direct and friend recommendation scores, while the relationship type between the two devices was derived from a correlation of trustworthiness with reference to mutual friends. Similarly, ref. [39] composed a *honesty*

model out of spatial (object proximity), temporal (frequency and duration of interactions), relationship and credibility (cooperativeness and penalties) metrics. The credibility aspect is the closest to the proposal in this work. Ref. [40] adopted a similar approach, albeit in that case honesty referred to the absence of malicious behaviors. For this purpose, a trust propagation and aggregation scheme was introduced, conceptually similar to the objective trust model in [36]. The above works, however, aimed only at improving transaction success and did not consider dynamic social relationships. That issue was tackled by [41], exploiting Bayesian belief propagation networks. Similarly, ref. [42] proposed a hybrid trust management framework for SIoT networks. The *Probabilistic Neighbourhood Overlap* method estimated tie strengths between nodes in a social graph generated by combining the social networks of both devices and their owners. In a SIoT network, malicious connections should be kept to a minimum. A node behaviors analysis module based on machine learning was exploited in the trust assessment process by [43], enabling not only the discovery of malicious node connections but also the identification of the kind of performed attack in order to prevent them in the future. The *ConTrust* [44] method proposed a context-dependent trust management technique for selecting a trustworthy service provider and allocating jobs in a SIoT environment. The trust model fused trust theory with social networks.

It can be noted that all the above works focus either on service discovery and composition or on dynamic relationship and trust management, disregarding or trivially simulating the other aspect. Table 1 summarizes relevant features of some of the previously mentioned frameworks with respect to the approach proposed here. In particular, this work is the first proposal of a complete stack, including (i) semantics-based cooperative service discovery and composition, (ii) full standard-based implementation, (iii) device trust assessment and (iv) dynamic social relationships; the work enhances a previous proposal [9] with reference to dimensions (ii) and (iii), and it introduces a novel formal framework for (i) and (iv).

Table 1. Comparative analysis of SIoT frameworks (✓: supported, ✗: not supported, *: partial support).

Features	[22]	[25]	[26]	[32]	[44]	[9]	This Work
Friend/follower relationships	✗	✗	*	✗	✗	✓	✓
Co-location/co-work relationships	✗	✓	✗	✓	✗	✗	*
Dynamic relationships	✓	✓	✓	✓	✗	✗	✓
Context-awareness	✓	✓	✓	✓	✓	✓	✓
Collaborative service discovery	✗	✗	✓	✗	✗	✓	✓
Autonomous device interactions	✗	✗	✗	✗	✗	✓	✓
Trust management	✗	✓	✗	✗	✓	✗	✓
Real-time data processing	✓	✓	✓	✗	✗	✓	✓
Semantic Web integration	✗	*	✓	✗	✗	✓	✓

4. Social Awareness in the Semantic Web of Things

This section outlines the proposal for social things intelligence and knowledge-based cooperation. A high-level SIoT architecture is defined, where every device can expose and/or request *services* (i.e., functionalities). When a device detects a context change that requires a reconfiguration, it issues a request on the social network. Then, a collaborative discovery process starts, exploiting semantic matchmaking to retrieve the best available services from the network for covering the request. Technologies grounding the framework, collaborative service discovery and dynamic relationship management are detailed in the next subsections.

4.1. LDP-CoAP Social Framework

The reference framework, illustrated in Figure 2, is implemented through the Linked Data Notifications protocol [45], on top of the Linked Data Platform [11] mapping [10] for the Constrained Application Protocol, ref. [12] a lightweight HTTP-like protocol for the Web of Things. LDP defines standard rules for accessing and managing Linked Data. LDN is a specialized use of LDP following an interaction model where servers (also called

receivers) advertise a receiving endpoint (*inbox*) for messages (*notifications*) and RDF-based data pushed by generic applications (*senders*). Then, the receiver exposes the notification data (according to appropriate access control) for use by *consumers*. Messages can be retrieved and used according to consumer needs. According to the proposed architecture, a pair of devices can establish two basic kinds of social relationships:

- **Friend**, a bidirectional relationship where nodes N_i and N_j can exchange both information and services. In particular, a device N_j sends a *friendship* request; since the receiver N_i accepts it, they become able to (i) read and write on each other’s wall, acting as an LDN receiver and an LDN sender (to write) or consumer (to read), respectively; (ii) request the friend’s service descriptions; (iii) activate or deactivate the friend’s services.
- **Follower**: a unidirectional relationship where a node N_k is interested only in receiving the updates published by N_i on its wall. In other words, if N_k sends a *follower* request to N_i , N_k becomes an LDN consumer for N_i .

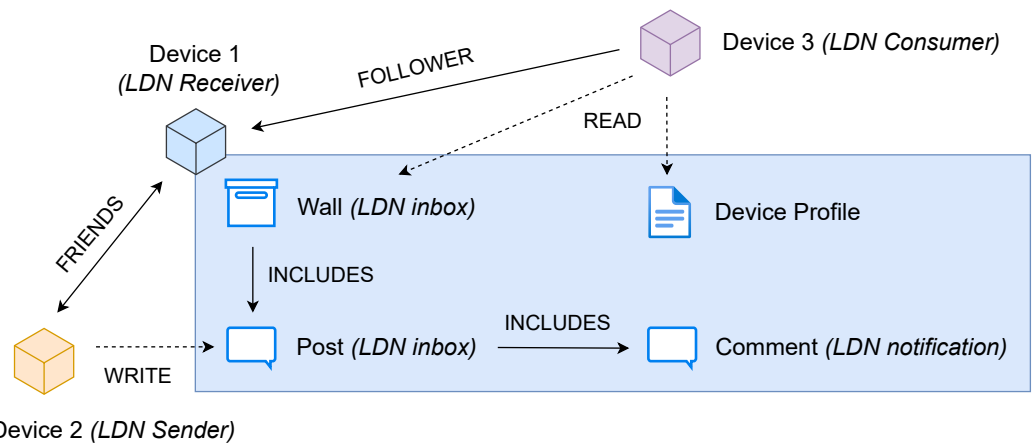


Figure 2. Social IoT framework and interaction model.

As depicted in Figure 3, social devices expose an LDP interface composed of a set of resources described exploiting both several well-known RDFS vocabularies (e.g., *Dublin Core* [46] and *Semantically Interlinked Online Communities* (SIOC) [47]) and an additional ontology, named *Semantic Web of Social Things* (SWST) (<http://swot.sisinfab.poliba.it/onto/swst/>, accessed on 28 June 2024), proposed in [9] to model basic elements of a social network. The inference services described in Section 2 are used to regulate the interactions between nodes. Nodes (i.e., things) are divided into two types: *smart* devices, able to execute inference tasks; *basic* devices, endowed with low memory and processing capabilities, only providing sensing/acting services without performing autonomous reasoning. Regardless of type, each object is modeled as an LDN receiver and owns a *wall*. It is an LDN inbox containing one or more *posts*, defined as nested resources, including *comments*. Posts and comments are used to enable the proposed interaction model. For example, sensor devices can observe the surrounding environment and use posts to share observations with actuator agents on the network, which usually present limited or no sensing capabilities. By reading the posts, these devices can become aware of current conditions and activate or deactivate specific services. Similarly, smart devices, equipped with both sensing and actuating capabilities, can use posts to initiate a discovery process. If a smart agent cannot independently address the perceived changes, it can find other social agents that provide useful services. Moreover, a basic node, when becoming a friend of a smart node, can also select it as *semantic facilitator*, i.e., reasoning supporter. As detailed in [9], post descriptions have the following attributes:

- Creation date (*dcterms:created*);
- Sender device (*swst:postedBy*);

- Content of the post (*sioc:about*), as Internationalized Resource Identifier (IRI) of the individual representing the received OWL annotation;
- *Like* value (*swst:likeValue*).

Instead, comment annotations associated with a post include the following:

- Creation date;
- Sender device;
- *Tagged* (i.e., activated) services, selected through the covering process described in Section 4.2 (*sioc:topic*);
- Content of the comment, representing the part of the original post not covered by tagged services provided by the friend device.

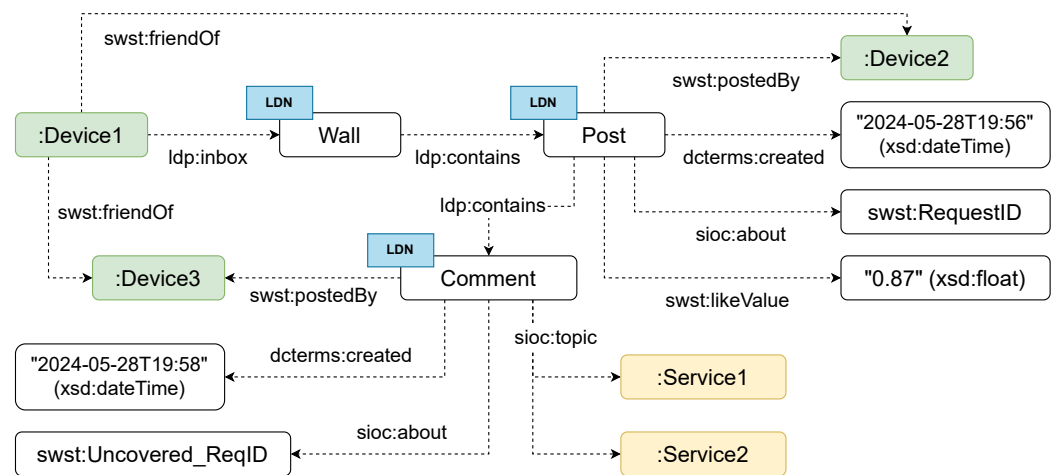


Figure 3. Reference ontology-based data modeling.

Each social thing also exposes a semantic-based *profile describing its own* main properties: type of device, device name, supported ontologies used as reference vocabularies to define the OWL-based annotations of the services exposed by the device; location of the device (e.g., in an area, building, department, apartment), (possible) friend and followed devices. Finally, all functionalities exposed by a device are contained in the *services* resource and characterized by name, IRI of the OWL individual modeling the service within the reference KB, timestamp of the last modification of the individual description, service current state and specific value to be used to activate the functionality.

4.2. Cooperative Service/Resource Discovery

The social network of things exists to support cooperative service discovery and configuration. Exploiting the non-standard inferences recalled in Section 2, whenever a device requires new services or a change to its current settings, its request is formalized as a DL annotation *R*. The proposed framework does not constrain *when* a device should issue a request, neither *how* to derive a proper request annotation. As outlined in [7], smart objects are expected to execute data mining and inference on perceptions collected from on-board or nearby sensors in order to characterize the environment and detect relevant events and changes (details on the data mining aspect are out of the scope of this paper). When the request is formulated, a cooperative multi-hop discovery process starts, using *Concept Covering* to satisfy it as much as possible through aggregation of available services/resources.

Figure 4 shows the sequence diagram for a canonical discovery example. Three devices are considered: *D*₁ is the requester and *D*₂ is a friend to both *D*₁ and *D*₃. When *D*₁ generates its request *R*₁, it will post it on its wall (post *P*₁). A local covering process starts, where *D*₁ exploits its embedded matchmaking engine to cover *R*₁ as much as possible with the services it can access: this set is formed by *D*₁'s own services and the ones belonging to *D*₁'s friends, which were read via LDP-CoAP at friendship establishment: the CoAP

Observe option [48] is adopted to notify the device whenever a friend’s service set changes. The covering result consists of a set of selected services and possibly the uncovered part U_1 of the request. This information is appended to P_1 as comment C_1 on D_1 ’s wall, along with the covering degree associated with P_1 as the like value L_1 . If there is some uncovered part U_1 , then D_1 picks some friends to help find further services. Selection criteria include the number of friend’s services referring to the same ontology of the request and historical performance data concerning the number and relevance of retrieved services, as well as communication latency and bandwidth.

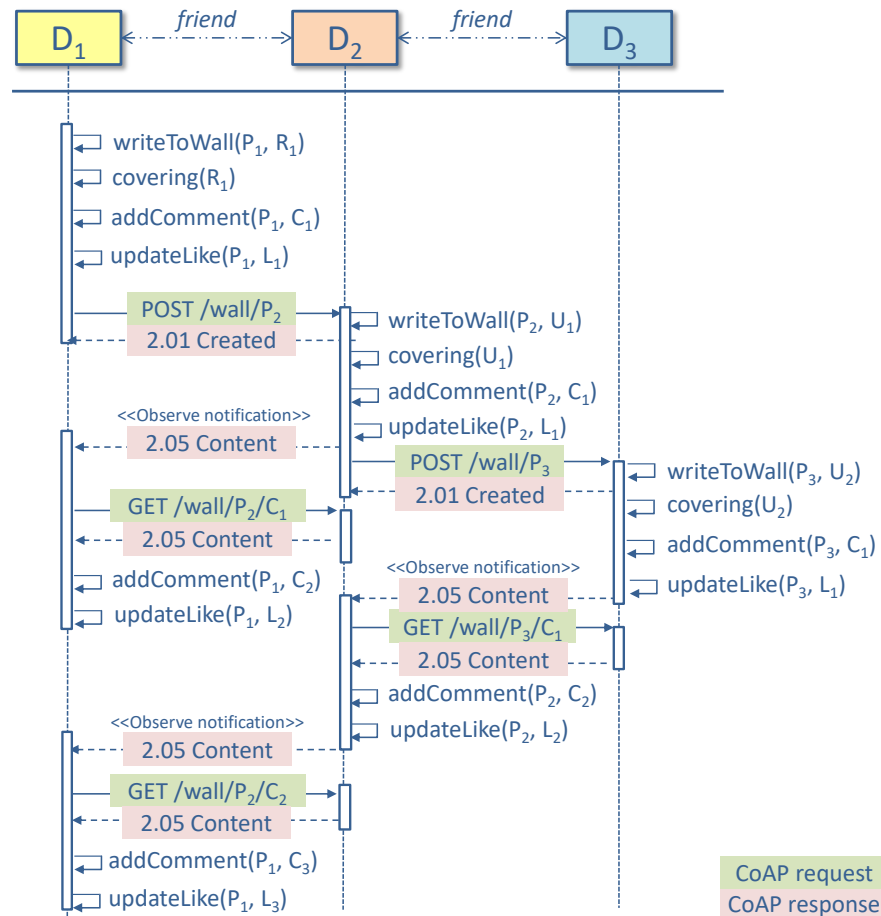


Figure 4. Distributed service/resource discovery.

In the reference figure, D_1 picks D_2 and POSTs a new post P_2 on D_2 ’s wall, attaching U_1 as the new request. D_2 computes semantic matchmaking using its friends’ service descriptions while excluding its own services as they have already been considered by D_1 . The new covering process yields (possibly) further selected services, a new (smaller) uncovered part U_2 and an updated (higher) like value. D_2 adds this result as a comment to P_2 . Complying with LDN, this update is notified to D_1 through the Observe CoAP pattern (actually, a node writing a post on a friend’s wall is automatically added to the list of CoAP observers for the post, so that it is notified when a comment is added; this is an extension of the standard CoAP Observe pattern, which is normally available for GET requests only; otherwise, after a POST on a friend’s wall, a further GET request would be needed to start observing). Then, D_1 GETs the content of the comment and updates the like value associated with its own original request on its own wall. Supposing the request has not been fully covered yet, D_2 repeats the process recursively, writing a new post P_3 on its friend D_3 ’s wall (which is not a friend of D_1). D_3 executes matchmaking with cached service descriptions of its friends (except D_2). The outcome leads to a comment and updated like value on P_3 : they are notified to D_2 , which GETs the content of the comment

and reports it as a new comment C_2 on post P_2 into its own wall. This in turn activates notification for D_1 , which GETs the comment. Finally, it updates its original post P_1 with the updated set of selected services, uncovered parts and like values. The process goes on until at least one of the following conditions becomes true:

- The request is fully covered or the covering procedure reaches a minimum threshold l_{min} of like (i.e., satisfaction) value. When a device reaches this limit value, the covering procedure can be stopped, and there is no need to forward the potentially uncovered part of the request to other friends. In this way, each device can prevent the network from being flooded with unnecessary communication which can degrade performance and lead to delays or lost messages;
- A maximum distance d_{max} from the request source is reached in the social graph. It represents the maximum number of hops a message can make from its original sender during the discovery process. In particular, a direct friend has a depth of 1, whereas a friend of a friend has a depth of 2. This parameter ensures that only nodes within a relevant proximity receive the message, improving the overall efficiency of the network by focusing communication on the most pertinent areas. Basically, the proposed protocol exploits a classic expanding ring search, with the peculiarity that it is not based on topological distance, but on social distance.

The proposed service/resource discovery is a distributed process, as outlined in Algorithm 1. When a device holds a request, after performing the local covering process, updating the comment (with the uncovered part of the request) and the like value (with the coverage ratio), (lines 1–4) is finally ready to exploit its social capabilities to identify friends who can advantageously help it in fulfilling the request (lines 5–7). Every friend device receiving the updated request verifies that the distance constraint in the social graph from the original sender is satisfied and starts a new decentralized covering. Then, the uncovered part of the request and the like value are updated. The *forward_to_friends* procedure (Algorithm 2) goes on recursively to the next level of friends until the request is fully covered, or the like threshold l_{min} is met, or the social distance limit is reached.

Algorithm 1: Procedure *Distributed_Discovery*(D, R, l_{min}, d_{max})

Require: Device D , Request R , minimum threshold of like l_{min} , maximum distance d_{max}

Ensure: Distributed Service/Resource Discovery

- 1: $P \leftarrow \text{create_post}(D, R)$ {Create a post for the request R }
 - 2: $\text{cover_request}(D, P)$ {Cover the request R with D 's available services}
 - 3: $\text{latest_uncovered_part} \leftarrow P.\text{comments}[-1].\text{uncovered_part}$ {Initialize the latest uncovered part}
 - 4: $\text{latest_like_value} \leftarrow P.\text{like_value}$ {Initialize the latest like value}
 - 5: **if** ($\text{latest_like_value} < l_{min}$ **and** $\text{latest_uncovered_part} \neq \text{null}$) **then**
 - 6: $\text{forward_to_friends}(D, P, \text{latest_uncovered_part}, \text{latest_like_value}, l_{min}, d_{max})$ {Forward the new request to friends}
 - 7: **end if**
-

Algorithm 2: Procedure *forward_to_friends*($D, P, U, like, l_{min}, d_{max}$)

Require: Device D , Post P , Uncovered Part U , Like value $like$, minimum threshold of like l_{min} , maximum distance d_{max}

Ensure: Forward request to friends

- 1: **if** $\text{calculate_social_distance}(D, F) \leq d_{max}$ **then**
 - 2: **for** F **in** $\text{select_friends}(D)$ **do**
 - 3: $\text{forward_post}(F, P, U)$ {Forward the post to the friend F }
 - 4: $\text{cover_request}(F, U)$ {Cover the request U with F 's available services}
 - 5: $U \leftarrow P.\text{comments}[-1].\text{uncovered_part}$ {Update the latest uncovered part}
 - 6: $like \leftarrow P.\text{like_value}$ {Update the latest like value}
 - 7: **if** ($like \geq l_{min}$ **or** $U = \text{null}$) **then**
 - 8: **break** {Check if the request is fully covered or meets the like threshold}
 - 9: **end if**
 - 10: $\text{forward_to_friends}(F, P, U, like, l_{min}, d_{max})$ {Recursively forward the request to the next level of friends}
 - 11: **end for**
 - 12: **end if**
-

Both l_{min} and d_{max} are tunable parameters. Furthermore, it is important to remark some points:

- The natural hierarchic structure of social network items (comment \in post \in wall) is neatly mapped to LDP containers and resources, where each item can be uniquely addressed and explored;
- The LDN notification framework accommodates the needs of social network applications; at a lower level, the proposed machine-to-machine SIoT approach implements LDN with the CoAP Observe pattern;
- Service discovery is collaborative and recursive, exploiting service descriptions shared between friends and the possibility to write posts and comments on each other's wall. Semantic matchmaking underpins logic-based service relevance ranking and request covering;
- Although not shown in Figure 4 for the sake of readability, selected services are activated at the end of each discovery step by *tagging* them through LDP-CoAP PATCH methods [9];
- Becoming a follower of a device D grants automatic registration as observer for every post published on D 's wall from that moment on, through the CoAP Observe pattern;
- A discovery session produces, for each participating device D , exactly one wall post, with a number of comments equal to the devices D has directly or indirectly involved in the matchmaking process. As described in the next subsection, devices can exploit them as a log to adapt their relationships dynamically, calculating metrics of usefulness for each friend.

4.3. Managing Dynamic Social Relationships

The proposal aims to extend the framework described in [9] to endow social objects with proactive adaptivity to environmental modifications in order to improve self-organization through social relationship refinement. Based on past experience (i.e., previous interactions over the social network), each object is able to adapt its relationships dynamically according to context changes, in order to improve not only the processing on the single node but also the overall network performance. Removing unnecessary relationships aims to reduce network traffic, e.g., fewer messages will be forwarded during a service discovery process. Conversely, new meaningful interactions can increase the network effectiveness and reduce the relative computational load per node: a request could be satisfied with lower processing time and activating the minimum set of services. The following models are proposed to suggest and remove friendship relationships based on the experience of each social object.

Suggestion. After every time period T , each smart device queries its wall to retrieve the list of devices that sent a comment to or were tagged in a post. For each such device D , the *suggest* function $S(D)$ is computed as follows:

$$S(D) = [w \cdot C(D) + (1 - w) \cdot U(D)] \cdot \frac{X - 1}{X} \quad (1)$$

with

$$C(D) = \sum_i^{N_c(D)} X^{(t_i - T)} \quad (2)$$

$$U(D) = \sum_j^{N_u(D)} X^{(t_j - T)} \quad (3)$$

The following symbols are used in the above formulae:

- $N_c(D)$: number of comments created by D ;
- $N_u(D)$: number of useful services provided by D , i.e., D 's services tagged as result of Concept Covering;

- w (between 0 and 1): used to weigh the contribution of $U(D)$, considering a device as direct provider of useful services, and $C(D)$, representing the ability of D to contribute to service discovery by acting as a “bridge” toward other useful devices;
- T : timestamp at which all functions are evaluated;
- $t_{i/j} \leq T$: timestamp at which a comment/post was created;
- $X > 1$: decay rate coefficient, weighing the past history of a device: the higher the value, the lower the relevance of older contributions.

Division by $\frac{X}{X-1}$ normalizes $S(D)$ with reference to the theoretical convergence value of the geometric series with ratio $\frac{1}{X}$, implying that $0 \leq S(D) \leq 1$. According to the proposed model, a device will be suggested as a good candidate friend if it provides services frequently and/or it is a friend of many useful devices. New friendship requests will be sent to devices with an S score higher than a reference threshold ($TH_{suggest}$). Considering the example in Figure 5a, if device D_1 frequently activates services of D_5 through intermediate friends D_2 and D_4 , it will learn and ask for D_5 's direct friendship as in Figure 5b.

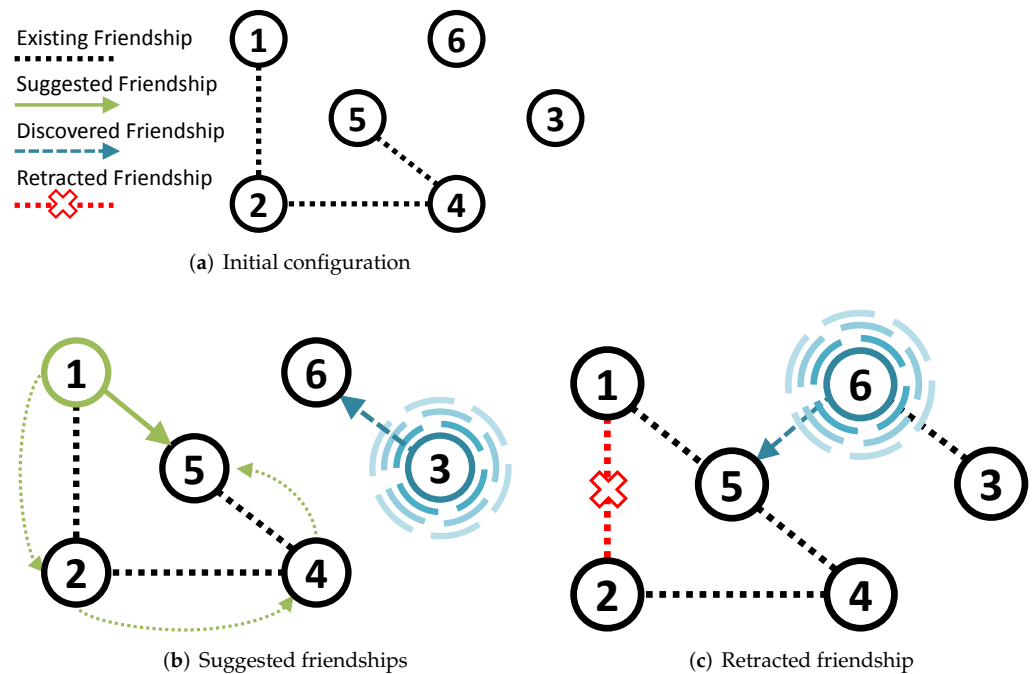


Figure 5. Sample network with loosely connected nodes.

Retraction. Along with the suggestion of new friends, a *retraction* score $R(D)$ is calculated for each friend device D to identify worthless relationships:

$$R(D) = S(D) - P(D) \tag{4}$$

with

$$P(D) = \left[\sum_k^{N_i(D)} X^{(t_k - T)} \cdot P_k(D) \right] \cdot \frac{X - 1}{X} \tag{5}$$

where $P(D)$ exploits past results of local Concept Covering processes. In detail:

- $N_i(D)$ is the number of D 's services, detected as incompatible with reference to the requests received in the period T ;
- $t_k \leq T$ is the timestamp at which the covering process was performed and the incompatibility was identified;
- $P_k(D) \in [0, 1]$ is the penalty score measured by Concept Contraction.

For each friend device, if $R(D)$ (which is always in $[-1, 1]$) is less than a TH -retract threshold, the friendship will be removed: the meaning is that D often provides conflicting services (wasting processing and communication resources) and/or it does not have a useful impact in satisfying received requests. The example in Figure 5c shows device D_1 breaking its friendship with D_2 .

Satisfaction. In loosely connected networks, social devices could be without any friends, e.g., due to a “cold start” configuration with isolated objects (as shown in Figure 5a) or frequent friendships retractions. In the protocol described in Section 4.2 isolated objects are unable to both establish new friendship relations and cooperate. In order to overcome this problem, a *satisfaction* indicator, based on likes, identifies and solves object deadlocks. Periodically, each device D evaluates its satisfaction degree with reference to received or generated requests by means of the formula:

$$Sat(D) = \left[\sum_h^{N_p(D)} X^{(t_h - T)} \cdot like_h(D) \right] \cdot \frac{X - 1}{X} \tag{6}$$

where $N_p(D)$ is the number of posts (i.e., requests) on D 's wall and $t_h \leq T$ represents the timestamp at which the $like_h(D)$ value related to the post was calculated. Also in this case, $Sat(D)$ is compared with a reference threshold (TH -satisfaction) and devices with a lower score will perform a further device discovery process, as during the device initialization. In Figure 5b, isolated device D_3 finds D_6 and they establish friendship; then in Figure 5c D_6 , still having a $Sat(D_6) < TH$ -satisfaction, links to D_5 too.

5. Case study: Social Smart Mobility for Electric Vehicles

A *smart mobility* case study referring to power management of Plug-in Electric Vehicles is discussed to highlight the capabilities and benefits of the proposed framework in dynamic environments. Figure 6 depicts the envisioned scenario. A 2017 BMW i3 plug-in electric taxi (T_1) usually serves “district A” of a large city. It has friendship relations with vehicles of its PEV taxi company and some of the charging stations in the city.

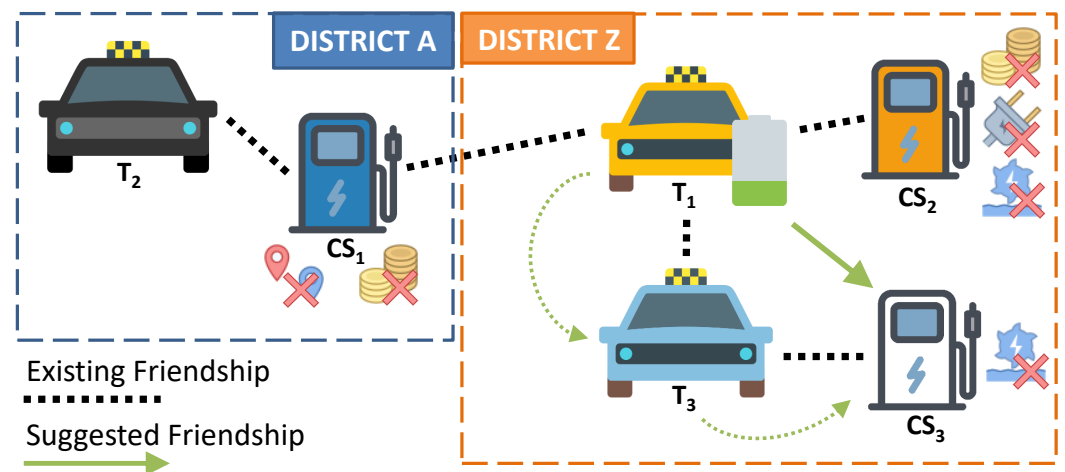


Figure 6. Social smart mobility scenario.

A possible \mathcal{ALN} formalization of electric vehicle power management aspects (reported in OWL 2 Manchester syntax [49] with reference to the reference domain ontology) is in Figure 7. For the sake of conciseness, figures in this section show only some relevant classes of the ontology.

```

T1 SubClassOf: Electric_Vehicle and
(has_AC-Compatible_Plug only VDE-AR-E_2623-2-2Type2_Plug) and
(has_DC-Compatible_Plug only CCSCOMBO2_Plug) and
(has_EVSE_Profile only (
  (has_Output_Rate only (daW max 74)) and
  (has_Current only (Ampere max 32)) and
  (has_Voltage only (Volt max 230)) )

```

Figure 7. Electric taxi profile semantic description

Notice T_1 electric car model adopts a standard VDE-AR-E 2623-2-2 (Type 2) *a.k.a.*, Mennekes connector and receptacle for AC charging, incompatible with SAEJ1772 (Type 1). Also, Fast DC charging currently has many standard connectors: CHAdeMO, CCS Combo1 or CCS Combo2 receptacles can be found on different chargers and are incompatible with each other; i3 supports CCS Combo2 plug only.

A customer requests to be taken to “district Z”, seldom visited by the taxi. After leaving the customer, the PEV taxi informs the driver that the battery level is below 50% and recharge is needed. The taxi formulates the semantic-based recharge request depicted in Figure 8, corresponding to a charging rate of at least 25 km per charging hour, minimum available energy of 20 kWh at a charging station and a maximum distance of 5 km between taxi and charging station and 30 km between charging station and dispatcher, and maximum price of 50 cents per kWh. This is used in a Concept Covering task (Section 2) on the locally cached set of functionalities exposed by all direct friends of the taxi. Unfortunately, those service instances present some conflicting features and cannot be selected. As detailed in Figure 8, the charging station CS_1 is too far from the current taxi position (*has_Distance*), because it is located in “district A”, and it is too expensive (*has_Fee_Per_kWh*). At the charging station CS_2 located in the “district Z”, DC charger equipment is CHAdeMO (*has_DC-Compatible_Plug*) which is not suitable to charge taxi T_1 . Furthermore, the price per kWh and the distance between the charging station and the dispatcher exceed the desired one. According to Section 4.3, the taxi periodically computes *Retract R(D)* scores for each friend. If CS_2 service of device D still remains incompatible with taxi needs, D ’s score will progressively decrease until friendship breaks up when it becomes less than the given *TH-retract* threshold.

```

TaxiRechargeRequest EquivalentTo: T1 and
(has_Charging_Rate_Per_Hour only (km min 25)) and
(has_Available_Power only (kWh min 20)) and
(has_Dispatcher_Distance only (km max 30)) and
(has_Distance only (km max 5)) and
(has_Fee_Per_kWh only (cents max 50))

CS1 EquivalentTo: ChargingStation and
(has_AC-Compatible_Plug only VDE-AR-E_2623-2-2Type2_Plug) and
(has_Charging_Rate_Per_Hour only (km exactly 45)) and
(has_Available_Power only (kWh exactly 500)) and
(has_Dispatcher_Distance only (km exactly 25)) and
(has_Distance only (km exactly 12)) and
(has_EVSE_Profile only AC_Level_2) and
(has_Fee_Per_kWh only (cents exactly 55))

CS2 EquivalentTo: ChargingStation and
(has_DC-Compatible_Plug only CHAdeMo_Plug) and
(has_Charging_Rate_Per_Hour only (km exactly 200)) and
(has_Available_Power only (kWh exactly 100)) and
(has_Dispatcher_Distance only (km exactly 45)) and
(has_Distance only (km exactly 3)) and
(has_EVSE_Profile only DC_Level_1) and
(has_Fee_Per_kWh only (cents exactly 60))

```

Figure 8. Semantic annotations of taxi request and friends’ services.

The request cannot be completely satisfied. So, taxi fleet agents autonomously cooperate in order to share useful information about electricity suppliers among their friends and suggest the services they offer. A new charging station service CS_3 in “district Z”, summarized in Figure 9, is selected and tagged (i.e., suggested) by the friend T_3 . Service description has a dispatching distance higher than the desired maximum but a non-exact match is tackled by means of non-standard reasoning services for semantic matchmaking recalled in Section 2. Furthermore, the experience factor *suggest* $S(D)$ described in Section 4.3 is exploited to refine social relationships. Let us suppose the taxi becomes a frequent visitor of “district Z”, and where it often needs charging. If CS_3 service of device D_{new} is frequently tagged, $S(D_{new})$ value will increase during the social object’s lifetime and a new friendship relation will be suggested to directly retrieve and use the services of the charging station.

CS_3 **EquivalentTo**: ChargingStation and
 (has_DC-Compatible_Plug **only** CCSCCombo2_Plug) and
 (has_Charging_Rate_Per_Hour **only** (km max 150)) and
 (has_Available_Power **only** (kWh **exactly** 300)) and
 (has_Dispatcher_Distance **only** (km **exactly** 40)) and
 (has_Distance **only** (km **exactly** 4)) and
 (has_EVSE_Profile **only** DC_Level_1) and
 (has_Fee_Per_kWh **only** (cents **exactly** 45))

Figure 9. Semantic description of selected service.

This is just a small example to highlight the benefits of the proposed semantic-based SIoT framework. In real smart mobility scenarios, more complex requests and service descriptions are expected, and non-standard inferences will be even more useful for service ranking and composition to satisfy detailed requests.

6. Experiments

In order to implement the proposed social intelligence framework, the LDP-CoAP proposal (<http://github.com/sisinflab-swot/ldp-coap-framework>, accessed on 28 June 2024) described in [10] has been extended to support the LDN specification. Novel functionalities have been evaluated through the LDN Test Suite (<http://github.com/csarven/ldn-tests>, accessed on 28 June 2024). By default, the suite directly queries LDN resources by means of HTTP messages; therefore for LDP-CoAP tests, requests are sent and received through an enhanced version of the *Californium* [50] CoAP proxy (<http://github.com/sisinflab-swot/californium-ldp>, accessed on 28 June 2024). The suite consists of three sets of tests referred to LDN sender, receiver and consumer. LDP-CoAP fully supports all LDN requirements as reported on the official test summary web page (<http://linkedresearch.org/ldn/tests/summary>, accessed on 28 June 2024).

An evaluation of the efficiency and effectiveness of the collaborative social protocol, as well as the impact of the dynamic model proposed for SIoT relationship management, has been carried out by testing social objects with 18 different network configurations, generated by combining the parameters reported in Table 2. A preliminary phase has been performed to empirically validate values (the same for all devices) of parameters used by the experience functions described in Section 4.3: *decay rate* $X = 2$, ensuring an adequate and not excessive decay for older elements of the time series; $w = 0.3$, to favor devices exposing tagged (i.e., selected) services over objects only sending comments; $TH-suggest = 0.5$, $TH-retract = -0.1$ and $TH-satisfaction = 0.2$, aiming to remove friendships only if really needed (in volatile contexts devices should be able to exploit availability opportunistically) and to suggest new relationships appropriately, while limiting the network overhead. *Maximum hop count* of the cooperative service discovery process—detailed in Section 4.2—has been set to 2 for all devices.

Table 2. Network configuration parameters.

Parameter	Value	Description
Generation algorithm	RND	<i>Randomly</i> generated nodes with mean number of friends set to 1. Loosely connected network with frequent isolated objects.
	BA	Nodes generated exploiting the preferential attachment rule of the <i>Barabási–Albert</i> model [51]. Each node has 2 friends on average. Hierarchical topology similar to a sensor network with several edge nodes and one or more sinks.
	DM	Nodes generated using the <i>Dorogovtsev–Mendes</i> algorithm [52]. Strongly connected network with nodes having 4 friends on average.
Nodes	10	Small-size network.
	100	Medium-size network.
	1000	Large-size network.
Request rate	20	20% of nodes identifies a new event and posts a message in each period. <i>Few</i> device requests.
	70	70% of nodes identifies a new event and posts a message in each period. <i>Frequent</i> device requests.

For each configuration, tests have been repeated 20 times (corresponding to 20 random initial topologies) monitoring the network evolution during 10 periods of 60 s. Experiments have been conducted exploiting three sets of 10, 100 and 1000 device descriptions, respectively. Each device exposes 5 services, each associated with an OWL annotation generated as the logical conjunction of 10 OWL classes (also including negated classes) randomly selected from a total of 168 classes in the reference ontology. After each period, the following output values have been computed to identify specific behaviors characterizing the social network and in particular to evaluate the capability to adapt social relationships dynamically according to context changes: (i) number of friends per node; (ii) like value received by a post; (iii) number of forwarded messages (i.e., how many devices are contacted to satisfy a request); (iv) number of activated services per request; (v) size of payload data exchanged on the network to satisfy a request (including uncovered part of requests forwarded to friends and comments retrieved from the wall of contacted objects). Obtained results have been averaged in order to filter out the bias deriving from conditions of single runs. Devices have been initialized at the beginning of each test whereas requests have been generated at random instants, with uniform probability distribution within the simulation time.

Small-size networks. Figure 10 shows test results for scenarios with 10 nodes. For loosely connected networks (RND), like values are very low (below 0.45) in the first period, as reported in Figure 10b, due to the few friendship relationships (Figure 10a). For each request, a device starts a local covering process with few available services, so it is only partially satisfied through a minimal set of useful functionalities (Figure 10d) and a large amount of data (corresponding to the uncovered part of the request) is forwarded to friends (Figure 10f). Thanks to the *Satisfaction* and *Suggestion* indexes described in Section 4.3, *unsatisfied* devices (e.g., isolated nodes) promptly look for useful friendships. In the second period, it is possible to notice both a considerable increase in the like value and a reduction in the data exchanged on the network. Having more friends, each device is able to locally satisfy a request in a better way and to forward a smaller uncovered part. A similar trend can be also identified for BA and DM network configurations; in these cases, however, the network starts with a higher degree of connectivity in the first period, generating higher like values—as shown in Figure 10c—which increase more slowly with reference to the previous scenario. Nevertheless, semantic-based experience factors allow satisfactory like values, reducing at the same time the network traffic. In fact, introducing more useful friendships and removing useless ones allows social things to select the most suitable services at each hop of the cooperative covering process, converging faster to the final result. Moreover, the request rate mainly affects the behavior of social networks with few connections: device relationships are modified more frequently, leading to lower like values and less efficient cooperative covering. Few messages are forwarded to friends, as reported in Figure 10e, due to the more variable social network configuration. On the

contrary, social devices are able to establish stable relationships after about 4 periods for all other configurations.

Medium-size networks. Also in scenarios with 100 nodes (results detailed in Figure 11) loosely connected networks quickly take advantage of dynamic relationship adaptivity. Compared with small networks, mean like values are increased (Figure 11b) by virtue of a wider choice of possible friends to exploit as service providers. This also implies decreased transmitted data (Figure 11f), since the distributed discovery converges in fewer hops. In small-world (BA) and strongly connected (DM) network scenarios, mean like values (Figure 11c) improve their performance over the periods, while results on exchanged data are very similar to small networks. The presence of many devices facilitates the discovery of additional useful friends, as reported in Figure 11a. As a consequence, higher like scores have been obtained: all network parameters tend to steady-state values in about three (friendships and exchanged data) to five (activated services and forwarded messages) periods.

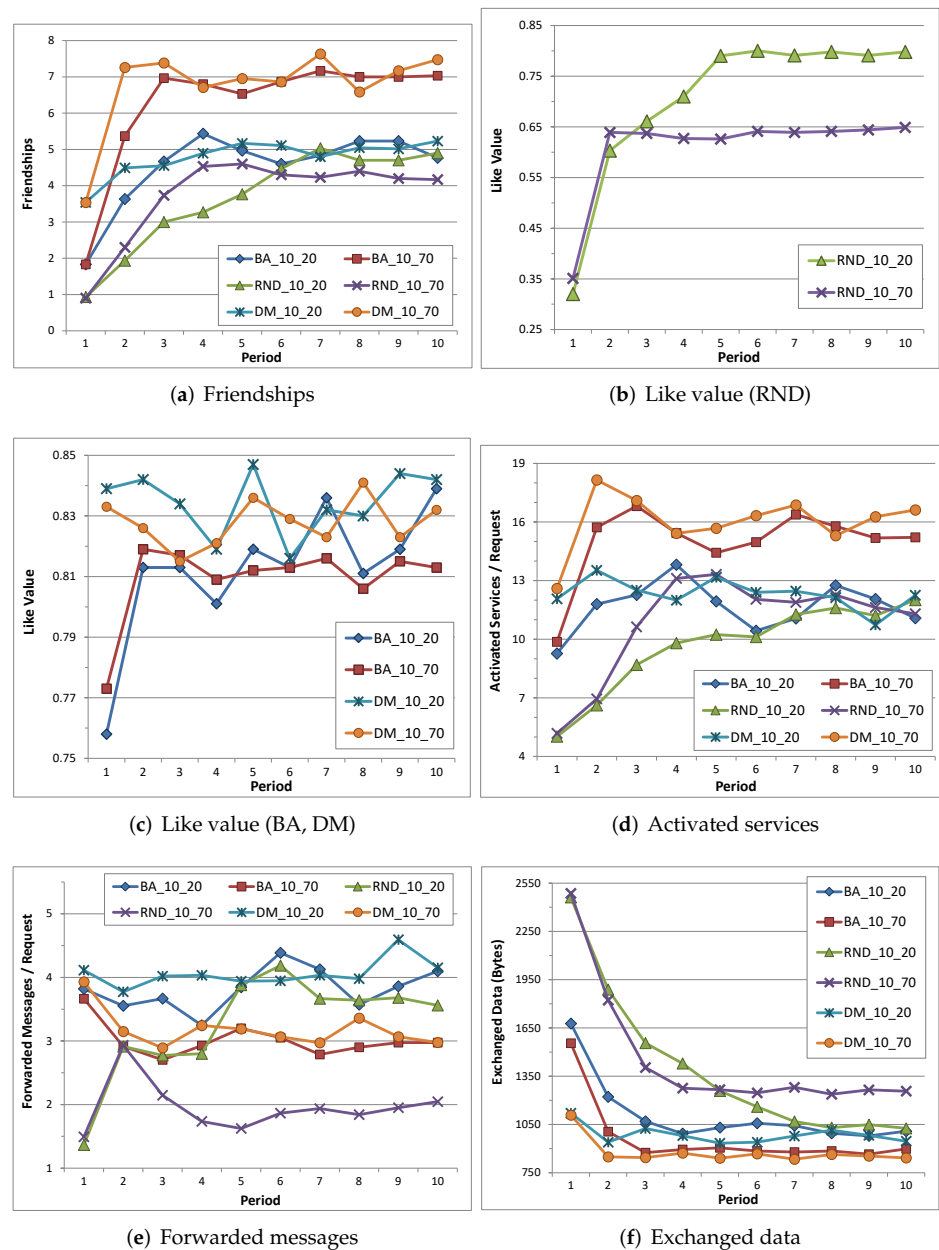


Figure 10. Test results for small-size networks. Legend denotes values of parameters for each configuration (<generation algorithm>_<number of nodes>_<request rate>).

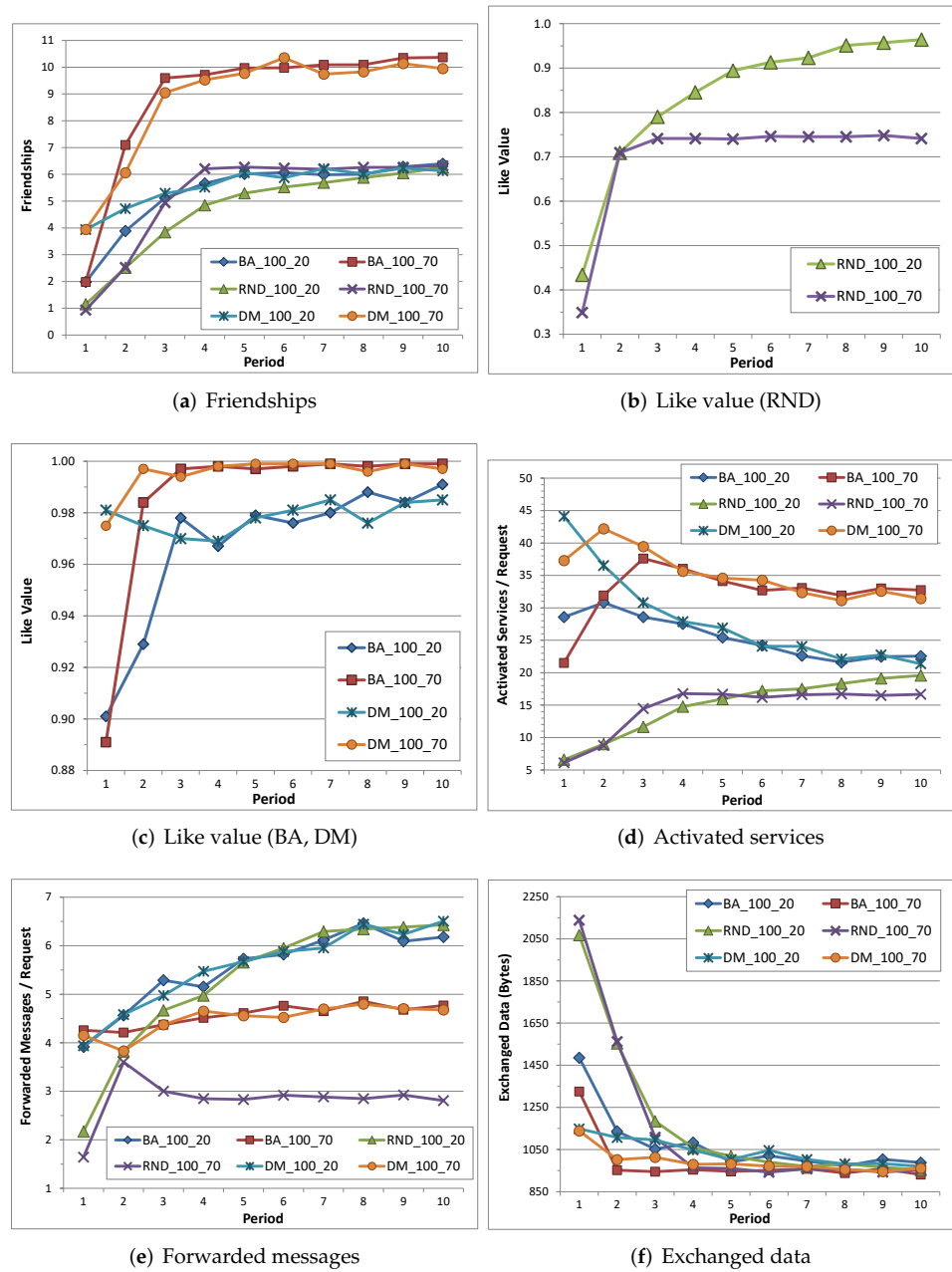


Figure 11. Test results for medium-size networks. Legend denotes values of parameters for each configuration (<generation algorithm>_<number of nodes>_<request rate>).

Large-size networks. Finally, results for scenarios with 1000 nodes are reported in Figure 12. Large networks follow similar trends highlighted for small and medium configurations. Main differences compared to medium scenarios concern: (i) fewer friends per device (Figure 12a), as friendships are removed more frequently in large networks, so the number of friends grows more slowly; (ii) fewer activated services per request (Figure 12d) because, when nodes reach the steady-state configuration, the covering process is able to select few services which largely satisfy a request, rather than many services each covering a small part. Thanks to these optimized relationships, mean like values in large loosely connected (RND) networks (Figure 12b) increase with reference to small and medium ones; on the other hand, for BA and DM network models the is not a significant difference in like values between large (Figure 12c and medium networks, meaning in the latter case, devices are already able to find a sufficient number of friends to satisfy their requests. Moreover, devices save storage space (activated functionalities are tagged and

saved on the device wall) and keep the network overhead required to contact and activate remote services as low (Figure 12f) as in small (Figure 10f) and medium (Figure 11f) ones. This is a clear benefit of the approach, envisioning good scalability: even on large networks, nodes establish a balanced number of high-quality friends and share their resources without overloading the network.

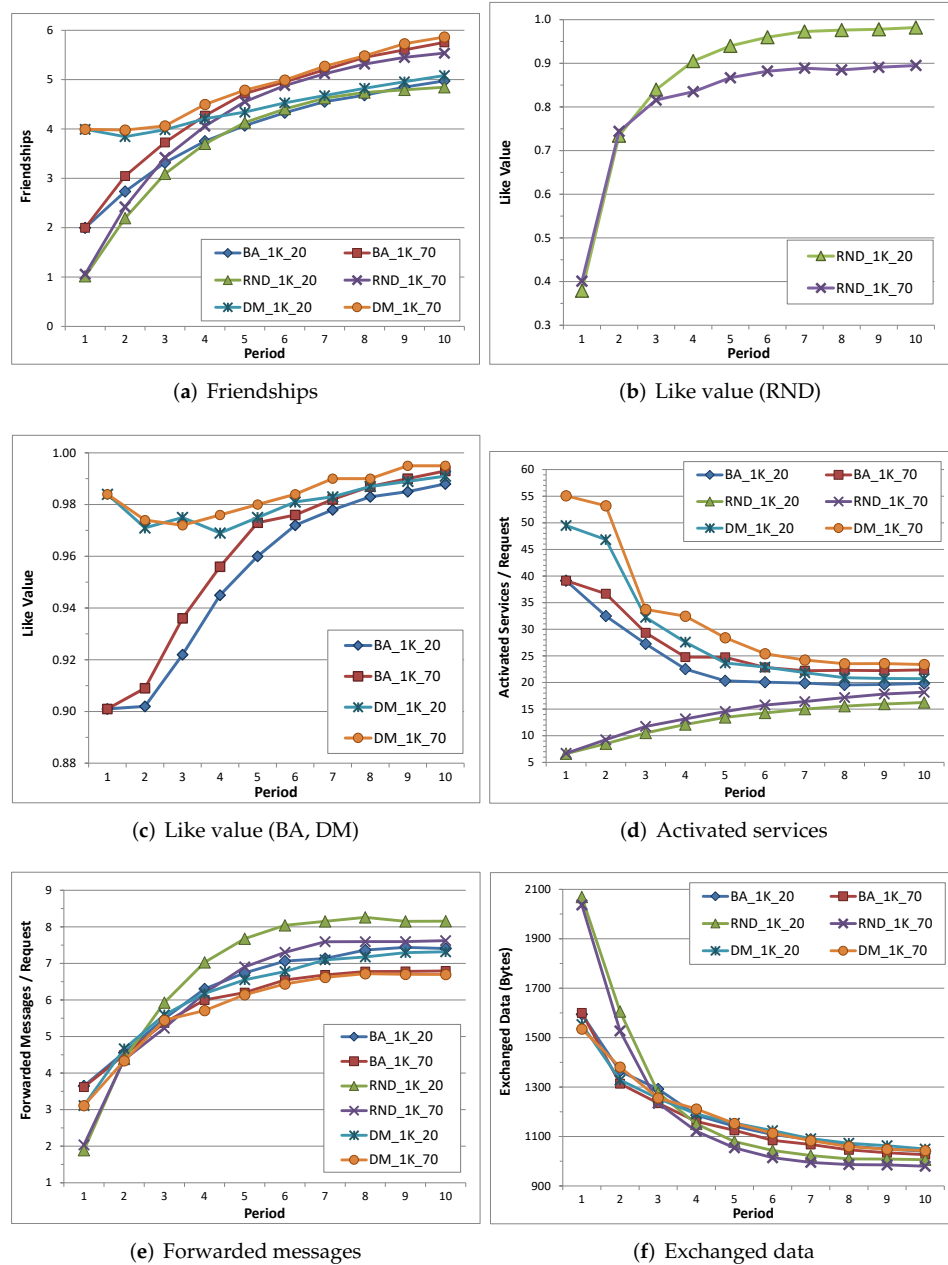


Figure 12. Test results for large-size networks. Legend denotes values of parameters for each configuration (<generation algorithm>_<number of nodes>_<request rate>).

The same test suite has been exploited to compare results of the proposed dynamic approach with the static SIoT relationship management model in [9]. A direct experimental comparison has been possible only with that framework, as the other state-of-the-art works reported in Section 3 have not published their code. Tests have been run using the same initial configurations, comparing the steady-state results after $N = 10$ time periods. The framework in [9] does not allow for dynamic relationship configuration; therefore, performance critically depends on the initial overlay network topology. Figure 13 highlights

the deltas related to the different parameters for the 18 network configurations tested in the evaluation. The main discussion remarks are as follows:

- Medium-size networks exhibit higher variations in the average number of friendships per node (Figure 13a). With 100 nodes, the network is large enough to facilitate multiple connections without becoming overwhelmingly complex. Medium-sized networks often see the formation of dynamic tightly knit communities or clusters, which naturally enhances the number of friendships as nodes within these clusters are more likely to be interconnected. In particular, novel friendships are created for small and medium networks with frequent device requests (i.e., RND_70, BA_70 and DM_70) where several interactions foster more connections and higher levels of friendships. In large fully connected networks, the number of friendships tends to remain stable due to a dynamic equilibrium in the formation and retraction of connections. As the network grows, new relationships are established, but existing ones may also be retracted if they are deemed less useful or redundant;
- With the proposed approach like values in Figure 13b become higher in all configurations, particularly in loosely connected networks with frequently isolated objects (i.e., RND configurations). In such networks, dynamic relationships allow nodes to continuously form and adjust connections based on evolving interactions and needs, and the like values tend to rise more noticeably due to the enhanced potential for discovering novel and useful connections. As nodes interact with a more diverse set of peers, they can uncover and leverage previously inaccessible resources and services. On the contrary, in fully connected networks (i.e., DM configurations), new connections are relatively saturated. Many nodes are already connected and aware of existing services, leaving less room for discovering and activating new functionalities. As a consequence, the number of activated services tends to increase in small and loosely connected networks (cells with light colors in Figure 13c), whereas it even decreases in large and fully connected networks (blue cells). In these cases, the network tends to reach an optimal configuration by activating only the most suitable subset of services, as testified by high like values; this means that network efficiency is maximized and unnecessary complexity is avoided;
- In small and medium networks, the number of forwarded messages remains stable or slightly increases due to the simplicity and directness of social interaction routes (Figure 13d). The limited network size allows for efficient message routing with fewer forwarding steps. In contrast, large networks exhibit an increase in the number of forwarded messages due to their complexity and the need for more indirect communication paths also in case of dynamic and re-configurable relationships. Nevertheless, across all network configurations (Figure 13e), there is a noticeable decrease in the amount of exchanged data related to service discovery. This trend reflects the improvements introduced by the proposed approach in the efficiency of the service discovery process, regardless of the network size or connectivity pattern. However, the decrease is particularly significant in loosely connected networks where the dynamic relationship management model significantly enhances the overall behavior of the social network. Devices are capable of quickly contacting new nodes and active services while minimizing unnecessary data exchanges. As a result, the network operates more efficiently, with improved resource utilization and enhanced communication pathways.

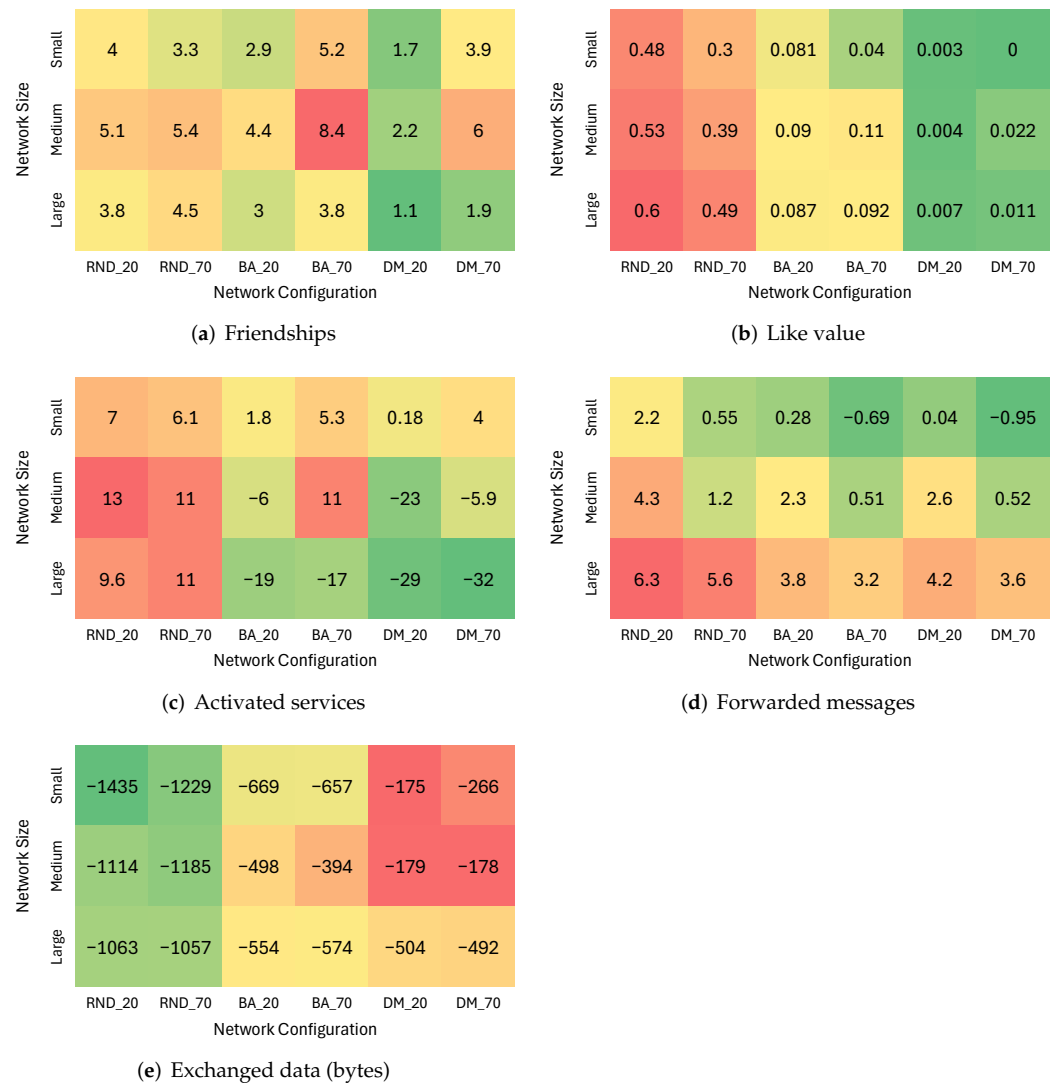


Figure 13. Comparison of dynamic (this paper) vs. static [9] relationship management.

7. Conclusions

This paper has introduced a novel comprehensive framework combining the Semantic Web of Things and the Social Internet of Things to enable dynamic autonomous management of smart object networks in Edge Computing. While basically general-purpose, the approach has been illustrated by means of a smart mobility case study.

Key features of the proposal include the following: a decentralized collaborative service discovery protocol, grounded on Linked Data Notifications on top of LDP-CoAP and on non-standard inference services in Description Logics over service descriptions in Web Ontology Language; a dynamic social relationship management protocol, based on the friends’ usefulness as a service and device recommender. Experimental evaluation has validated the sustainability and efficiency of the proposal for object networks of various sizes and topologies, extending the preliminary results obtained in previous works about processing time and memory consumption of the service discovery procedure [7,9] and non-standard reasoning algorithms [17] on resource-constrained IoT devices. The dynamic relationship management based on a close SWoT-SIoT integration allows devices to increase their number of friends even in small or poorly connected networks, leading to higher request satisfaction rates and reduced network load, since the most effective service/resource providers are queried directly. As the network scale grows, the framework remains able to reach steady-state conditions fairly quickly, with a balanced number of friends for each de-

vice, leading to consistently high request satisfaction and low network load. Experimental comparisons have shown improvements with reference to static relationship management in [9]: in random networks, friendships are 50% to 250% higher, request satisfaction is 100% to 150% higher, and the number of activated services is 100% to 150% higher, all with a 100% lower amount of exchanged data; in Barabási–Albert networks, established friendships are 100% to 400% higher, like values are 10% higher, while the number of activated services is 10% to 50% higher in small networks and decreases up to –50% in large networks, implying that the discovery of the best services is more effective, as demonstrated also by –60% exchanged data; in strongly connected (Dorogotsev–Mendes) networks, friendships are 50% to 150% higher, like values are 2% higher on average, the number of activated services increases up to 30% in small networks (implying more matches are found) and decreases up to –55% in medium- and large-size networks (implying requests are satisfied equally well or better with fewer services, as the high number of available services allows a more effective selection), with a 15% to 30% decrease in exchanged data.

Future work includes expanding on the type of social relationships in order to diversify supported interaction patterns. Autonomic management of weights in the model described in Section 4.3 is also under early investigation, based on discovery performance as well as computational and network load. As the collaborative discovery protocol represents a form of dialogue among participating social objects, ref. [53] emphasizes that this aspect of the proposal is under scrutiny by means of a more explicit formalization exploiting computational argumentation frameworks. From an infrastructure standpoint, integration into the cloud-to-thing continuum can enable leveraging a large-scale distributed architecture for collaborative tasks such as environmental monitoring and federated learning [54]. Finally, increasing the Technology Readiness Level (TRL) of the proposal will be achieved through practical implementations in real-world use cases spanning from smart mobility to building management systems, smart grid, precision agriculture, and last-mile logistics.

Author Contributions: Conceptualization, M.R., E.D.S. and G.L.; methodology, F.S. and G.L.; software, G.L. and G.C.; validation, C.F. and A.P.; formal analysis, F.S., G.L. and G.C.; investigation, F.S., A.P. and C.F.; resources, G.L., C.F. and G.C.; data curation, G.L. and A.P.; writing—original draft preparation, all authors; writing—review and editing, M.R. and E.D.S.; visualization, G.L., A.P. and G.C.; supervision, E.D.S.; project administration, A.P.; funding acquisition, M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research and APC were supported by the MOST National Research Centre for Sustainable Mobility (code CN00000023), funded by Italian Ministry of University and Research program under the National Recovery and Resilience Plan funded by NextGenerationEU, and by the *Digital Enterprise* grant (code NIL6S28), co-funded by Lutech Advanced Solutions S.p.A. and European Regional Development Fund for Apulia Region 2014/2020 Operating Program.

Data Availability Statement: Data available on request due to restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Stufano Melone, M.R.; Borgo, S.; Camarda, D. Digital Twins of Cities vs. Digital Twins for Cities. In *Proceedings of the International Conference on Innovation in Urban and Regional Planning*; Springer: Cham, Switzerland, 2023; pp. 192–203. [\[CrossRef\]](#)
2. Gharajeh, M.S. Behavior-based decision making: A tutorial. *Int. J. Dyn. Control* **2018**, *6*, 1816–1840. [\[CrossRef\]](#)
3. Scioscia, F.; Ruta, M. Building a Semantic Web of Things: Issues and perspectives in information compression. In *Proceedings of the 2009 IEEE International Conference on Semantic Computing*, Berkeley, CA, USA, 14–16 September 2009; IEEE Computer Society: Washington, DC, USA, 2009; pp. 589–594. [\[CrossRef\]](#)
4. Atzori, L.; Iera, A.; Morabito, G. From “Smart Objects” to “Social Objects”: The next evolutionary step of the Internet of Things. *Commun. Mag. IEEE* **2014**, *52*, 97–105. [\[CrossRef\]](#)
5. Roopa, M.; Pattar, S.; Buyya, R.; Venugopal, K.R.; Iyengar, S.; Patnaik, L. Social Internet of Things (SIoT): Foundations, thrust areas, systematic review and future directions. *Comput. Commun.* **2019**, *139*, 32–57. [\[CrossRef\]](#)
6. Fadda, M.; Anedda, M.; Girau, R.; Pau, G.; Giusto, D.D. A social internet of things smart city solution for traffic and pollution monitoring in cagliari. *IEEE Internet Things J.* **2022**, *10*, 2373–2390. [\[CrossRef\]](#)

7. Ruta, M.; Scioscia, F.; Loseto, G.; Sciascio, E.D. A Semantic-Enabled Social Network of Devices for Building Automation. *IEEE Trans. Ind. Inform.* **2017**, *13*, 3379–3388. [CrossRef]
8. Patel, P.; Ali, M.I.; Sheth, A. From raw data to smart manufacturing: AI and Semantic Web of Things for Industry 4.0. *IEEE Intell. Syst.* **2018**, *33*, 79–86. [CrossRef]
9. Ruta, M.; Scioscia, F.; Loseto, G.; Gramegna, F.; Ieva, S.; Pinto, A.; Di Sciascio, E. Social Internet of Things for domotics: A knowledge-based approach over LDP-CoAP. *Semant. Web* **2018**, *9*, 781–802. [CrossRef]
10. Loseto, G.; Ieva, S.; Gramegna, F.; Ruta, M.; Scioscia, F.; Di Sciascio, E. Linked Data (in low-resource) Platforms: A mapping for Constrained Application Protocol. In *The Semantic Web—ISWC 2016. ISWC 2016*; Groth, P., Simperl, E., Gray, A., Sabou, M., Krötzsch, M., Lecue, F., Flöck, F., Gil, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2016; Volume 9982, pp. 131–139. [CrossRef]
11. Malhotra, A.; Arwe, J.; Speicher, S. Linked Data Platform 1.0. W3C Recommendation, W3C. 2015. Available online: <http://www.w3.org/TR/ldp/> (accessed on 4 September 2024).
12. Shelby, Z.; Hartke, K.; Bormann, C. The Constrained Application Protocol (CoAP). RFC 7252. 2014. Available online: <http://www.ietf.org/rfc/rfc7252> (accessed on 4 September 2024).
13. Baader, F.; Calvanese, D.; McGuinness, D.L.; Nardi, D.; Patel-Schneider, P. *The Description Logic Handbook*; Cambridge University Press: Cambridge, UK, 2002. [CrossRef]
14. Ruta, M.; Scioscia, F.; Di Sciascio, E. Enabling the Semantic Web of Things: Framework and architecture. In Proceedings of the 2012 IEEE Sixth International Conference on Semantic Computing, Palermo, Italy, 19–21 September 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 345–347. [CrossRef]
15. W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview (Second Edition). W3C Recommendation. 2012. Available online: <https://www.w3.org/TR/owl2-overview/> (accessed on 4 September 2024).
16. Ruta, M.; Di Sciascio, E.; Scioscia, F. Concept Abduction and Contraction in Semantic-based P2P Environments. *Web Intell. Agent Syst.* **2011**, *9*, 179–207. [CrossRef]
17. Ruta, M.; Scioscia, F.; Bilenchi, I.; Gramegna, F.; Loseto, G.; Ieva, S.; Pinto, A. A multiplatform reasoning engine for the Semantic Web of Everything. *J. Web Semant.* **2022**, *73*, 100709. [CrossRef]
18. Li, Z.; Chen, R.; Liu, L.; Min, G. Dynamic resource discovery based on preference and movement pattern similarity for large-scale social Internet of Things. *IEEE Internet Things J.* **2016**, *3*, 581–589. [CrossRef]
19. Khanfor, A.; Ghazzai, H.; Yang, Y.; Haider, M.R.; Massoud, Y. Automated service discovery for social internet-of-things systems. In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 12–14 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5. [CrossRef]
20. Amin, F.; Majeed, A.; Mateen, A.; Abbasi, R.; Hwang, S.O. A systematic survey on the recent advancements in the Social Internet of Things. *IEEE Access* **2022**, *10*, 63867–63884. [CrossRef]
21. Ramis Ferrer, B.; Iarovyi, S.; Gonzalez, L.; Lobov, A.; Martinez Lastra, J.L. Management of distributed knowledge encapsulated in embedded devices. *Int. J. Prod. Res.* **2015**, *54*, 5434–5451. [CrossRef]
22. Hussein, D.; Park, S.; Han, S.N.; Crespi, N. Dynamic social structure of things: A contextual approach in CPSS. *IEEE Internet Comput.* **2015**, *19*, 12–20. [CrossRef]
23. Hussein, D.; Han, S.N.; Lee, G.M.; Crespi, N. Social cloud-based cognitive reasoning for task-oriented recommendation. *IEEE Cloud Comput.* **2015**, *2*, 10–19. [CrossRef]
24. Girau, R.; Anedda, M.; Fadda, M.; Farina, M.; Floris, A.; Sole, M.; Giusto, D. Coastal monitoring system based on social Internet of Things platform. *IEEE Internet Things J.* **2019**, *7*, 1260–1272. [CrossRef]
25. Girau, R.; Martis, S.; Atzori, L. Lysis: A platform for IoT distributed applications over socially connected objects. *IEEE Internet Things J.* **2017**, *4*, 40–51. [CrossRef]
26. Gulati, N.; Kaur, P.D. Towards socially enabled internet of industrial things: Architecture, semantic model and relationship management. *Ad Hoc Netw.* **2019**, *91*, 101869. [CrossRef]
27. Lin, C.H.; Ho, P.H.; Lin, H.C. Framework for NFC-based intelligent agents: A context-awareness enabler for Social Internet of Things. Available online: <https://journals.sagepub.com/doi/10.1155/2014/978951?icid=int.sj-full-text.similar-articles.2> (accessed on 4 September 2024).
28. Hussein, D.; Han, S.N.; Lee, G.M.; Crespi, N.; Bertin, E. Towards a dynamic discovery of smart services in the social internet of things. *Comput. Electr. Eng.* **2017**, *58*, 429–443. [CrossRef]
29. Atzori, L.; Iera, A.; Morabito, G.; Nitti, M. The Social Internet of Things (SIoT) – When Social Networks Meet the Internet of Things: Concept, Architecture and Network Characterization. *Comput. Netw.* **2012**, *56*, 3594–3608. [CrossRef]
30. Voutyras, O.; Bourellos, P.; Gogouvitis, S.; Kyriazis, D.; Varvarigou, T. Social monitoring and social analysis in Internet of Things virtual networks. In Proceedings of the 2015 18th International Conference on Intelligence in Next Generation Networks, Paris, France, 17–19 February 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 244–251. [CrossRef]
31. Li, A.; Ye, X.; Ning, H. Thing Relation Modeling in the Internet of Things. *IEEE Access* **2017**, *5*, 17117–17125. [CrossRef]
32. Aljubairy, A.; Zhang, W.E.; Sheng, Q.Z.; Alhazmi, A. Siotpredict: A framework for predicting relationships in the social internet of things. In *Proceedings of the Advanced Information Systems Engineering: 32nd International Conference, CAiSE 2020, Grenoble, France, 8–12 June 2020, Proceedings 32*; Springer: Cham, Switzerland, 2020; pp. 101–116. [CrossRef]

33. Yan, Z.; Zhang, P.; Vasilakos, A.V. A survey on trust management for Internet of Things. *J. Netw. Comput. Appl.* **2014**, *42*, 120–134. [CrossRef]
34. Chahal, R.K.; Kumar, N.; Batra, S. Trust management in social Internet of Things: A taxonomy, open issues, and challenges. *Comput. Commun.* **2020**, *150*, 13–46. [CrossRef]
35. Daly, E.M.; Haahr, M. Social network analysis for information flow in disconnected delay-tolerant MANETs. *IEEE Trans. Mob. Comput.* **2009**, *8*, 606–621. [CrossRef]
36. Nitti, M.; Girau, R.; Atzori, L. Trustworthiness Management in the Social Internet of Things. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 1253–1266. [CrossRef]
37. Nitti, M.; Murrioni, M.; Fadda, M.; Atzori, L. Exploiting Social Internet of Things Features in Cognitive Radio. *IEEE Access* **2016**, *4*, 9204–9212. [CrossRef]
38. Chen, Z.; Ling, R.; Huang, C.M.; Zhu, X. A scheme of access service recommendation for the Social Internet of Things. *Int. J. Commun. Syst.* **2016**, *29*, 694–706. [CrossRef]
39. Jayasinghe, U.; Lee, H.W.; Lee, G.M. A Computational Model to Evaluate Honesty in Social Internet of Things. In Proceedings of the Symposium on Applied Computing, Marrakech Morocco, 3–7 April 2017; ACM: New York, NY, USA, 2017; pp. 1830–1835. [CrossRef]
40. Chen, R.; Bao, F.; Guo, J. Trust-based Service Management for Social Internet of Things Systems. *IEEE Trans. Dependable Secur. Comput.* **2016**, *13*, 684–696. [CrossRef]
41. Panda, M.; Abraham, A. Development of a reliable trust management model in social internet of things. *Int. J. Trust Manag. Comput. Commun.* **2014**, *2*, 229–258. [CrossRef]
42. Narang, N.; Kar, S. A hybrid trust management framework for a multi-service social IoT network. *Comput. Commun.* **2021**, *171*, 61–79. [CrossRef]
43. Magdich, R.; Jemal, H.; Ayed, M.B. A resilient Trust Management framework towards trust related attacks in the Social Internet of Things. *Comput. Commun.* **2022**, *191*, 92–107. [CrossRef]
44. Latif, R. ConTrust: A novel context-dependent trust management model in social Internet of Things. *IEEE Access* **2022**, *10*, 46526–46537. [CrossRef]
45. Capadislis, S.; Guy, A.; Lange, C.; Auer, S.; Sambra, A.; Berners-Lee, T. Linked Data Notifications: A resource-centric communication protocol. In *Proceedings of the European Semantic Web Conference*; Springer: Cham, Switzerland, 2017; pp. 537–553. [CrossRef]
46. Dublin Core Metadata Initiative. Dublin Core Metadata Element Set, Version 1.1. 2012. Available online: <https://www.dublincore.org/specifications/dublin-core/dces/> (accessed on 4 September 2024).
47. Bojars, U.; Breslin, J.G.; Decker, S. Porting social media contributions with SIOC. In *Recent Trends and Developments in Social Software*; Springer: Cham, Switzerland, 2010; pp. 116–122. [CrossRef]
48. Hartke, K. Observing Resources in the Constrained Application Protocol (CoAP). RFC 7641. 2015. Available online: <https://tools.ietf.org/html/rfc7641> (accessed on 4 September 2024).
49. Horridge, M.; Patel-Schneider, P.F. OWL 2 Web Ontology Language Manchester Syntax (Second Edition). W3C Working Group Note, W3C. 2012. Available online: <https://www.w3.org/TR/owl2-manchester-syntax/> (accessed on 4 September 2024).
50. Kovatsch, M.; Lanter, M.; Shelby, Z. Californium: Scalable cloud services for the internet of things with coap. In Proceedings of the 2014 International Conference on the Internet of Things (IOT), Cambridge, MA, USA, 6–8 October 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–6. [CrossRef]
51. Barabási, A.L.; Albert, R. Emergence of scaling in random networks. *Science* **1999**, *286*, 509–512. [CrossRef] [PubMed]
52. Dorogovtsev, S.N.; Mendes, J.F. Evolution of networks. *Adv. Phys.* **2002**, *51*, 1079–1187. [CrossRef]
53. Lippi, M.; Mamei, M.; Mariani, S.; Zambonelli, F. An argumentation-based perspective over the Social IoT. *IEEE Internet Things J.* **2017**, *5*, 2537–2547. [CrossRef]
54. Loconte, D.; Ieva, S.; Pinto, A.; Loseto, G.; Scioscia, F.; Ruta, M. Expanding the cloud-to-edge continuum to the IoT in serverless federated learning. *Future Gener. Comput. Syst.* **2024**, *55*, 447–462. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.