Enabling Social Digital Twins in the 6G Era with Information Centric Networking

Marica Amadeo, Marco Martalò, Giuseppe Ruggeri, and Michele Nitti

The authors conceive a solution that leverages two revolutionary paradigms, namely the Social Internet of Things and Information Centric Networking to fill the gap between the deployment of a networking architecture supporting the dynamic discovery of digital twins and their instantiation at the edge.

Abstract

By creating the virtual representation of the elements and services of physical systems, Digital Twins (DTs) are expected to provide crucial support to the ambitious Sixth-generation (6G) applications. Being data-driven entities that process a variety of data in real-time, DTs can be hosted at the network edge, thus providing low latency services. However, the deployment of a networking architecture supporting the dynamic discovery of DTs and their instantiation at the edge is still an open challenge. To fill the gap, in this article we conceive a solution that leverages two revolutionary paradigms, namely the Social Internet of Things (SIoT) and Information Centric Networking (ICN). On the one hand, SIoT can provide trustworthy interactions between DTs to support effective service discovery and composition. On the other hand, ICN is a network model that includes data-centric primitives and in-network caching to facilitate service provisioning. By combining the two paradigms, we create a new framework where DT services are requested by name and discovered in a timely fashion by navigating a name-based SIoT network. Whenever possible, services already instantiated at the edge are reused to satisfy different clients. After demonstrating the benefits of the envisioned framework with a preliminary evaluation, we highlight research opportunities for this timely topic.

Introduction

To effectively support future 6G applications, the research community is promoting the Digital Twin (DT) technology [1]. A DT offers a virtual representation of the dynamics and elements of a physical system, including objects and humans, and it is able to share knowledge and experience gained through its Artificial Intelligence-empowered logic [2].

DTs can be divided into two main categories[1]: *Edge-based DTs*, which are hosted at the Edge to satisfy latency-sensitive applications, and *Cloud-based DTs*, which are hosted in the Cloud to support compute-intensive delay-tolerant applications. It is expected that 6G applications will require a variety of services from DTs that, whenever necessary, can be moved from the Cloud to the Edge to support latency-sensitive interactions.

A few works have recently targeted Edge-based DTs in 6G networks [1, 3]; however, they lack a flexible mechanism for dynamic and trusted service discovery and subsequent instantiation at the Edge [3]. To fill this gap, we present a framework that deploys latency-sensitive DT-based applications based on two revolutionary paradigms, namely Social Internet of Things (SIoT) [4] and Information Centric Networking (ICN) [5].

SIoT considers social relationships among different devices, independently from the fact that they belong to the same or different platforms owned and managed by different individuals or organizations. According to this vision, we introduce a *social dimension* in the DTs, which become collaborative entities able to create relationships with each others, as humans do. This makes the exchange of information and services among DTs easier by creating a society-based view of the trust level of each member of the community.

On the other hand, ICN is a future Internet architecture that directly names data and services at the network layer and implements ubiquitous in-network caching [6]. Routing-by-name mechanisms are implemented to deliver service requests from clients to connected nodes able to provide them. Therefore, ICN offers an effective communication model for DT-based applications, which are natively service-centric.

In our design, clients request 6G applications by name and the network first tries to solve the request at the Edge by *reusing* DT services that have been already instantiated for other clients, thus saving precious computing and bandwidth resources [7]. If service reuse cannot be applied, the request is handled in the SIoT network to discover the set of trusted DTs able to deploy the application. The new concept of *name-based social search* is designed: DT services are identified by hierarchical names and each DT maintains information about the services offered by its friends. By doing so, ICN primitives and caching can be applied in the social network to cut down the latency of the discovery procedure.

To the best of our knowledge, this is the first work exploring the synergies between SIoT and ICN to deploy DT-based applications in a Cloud-Edge network environment.

After presenting a reference use case and discussing the benefits of SIoT and ICN for deploying DT-based applications, we present the SIoT-ICN framework and evaluate its performance . We

Marica Amadeo and Giuseppe Ruggeri are with the University of Reggio Calabria, Italy; Marco Martalò and Michele Nitti (corresponding author) are with the University of Cagliari, Italy; Marica Amadeo, Marco Martalò, Giuseppe Ruggeri, and Michele Nitti are also with National Telecommunication Inter University Consortium, Italy.

ISSN: 0163-6804
Digital Object Identifier: 10.1109/MCOM.002.2300102

discuss research opportunities, before concluding the article.

Reference Use Case

The ever-increasing variety of objects and technologies will lead to an even more rapid increase in the range of possible applications, which may also have stringent requirements in terms of latency and reliability.

Among them, Metaverse applications will provide immersive experiences by means of eXtended Reality (XR) technologies. Users will have the ability not only to move inside the scene, but to interact with the elements within it (both objects and people). For instance, in a virtual museum, users may be allowed to see the artworks up close without ruining them, and to interact with them, for example, zooming in the details of a painting.

In this context, let us consider Alice, who want to have a virtual tour of the Louvre museum from her house in Cagliari. Alice will not be able to visit the entire Louvre during a single tour: the choice of which artworks to show should be based on her preferences and interests, so that the tour will be completely personalized.

This application scenario requires proper technologies that satisfy the stringent requirements in terms of low-latency and high-data rate and also provide highly personalized services.

Virtual replicas in the considered XR environment can be implemented as DTs, that is, living replicas of real assets, which are able to react autonomously, according to specific conditions [8]. However, it becomes crucial to decide which DTs best fit Alice preferences, and instantiate them close to her location (i.e., at the Edge) to offer the best experience. At the same time, other people in Cagliari could want to virtually visit the Louvre, maybe with a set of artworks partially overlapping those chosen for Alice. Therefore, to save network and computation resources and, in parallel, limit the latency, it is key to reuse, partially or entirely, already instantiated services at the Edge.

Enabling Paradigms

6G applications foresee a shift from the current interaction model based on humans looking for information provided by objects (human-object interaction), to the object-object interaction, which translates to the DT-DT interaction in the virtual world. This section describes two enabling paradigms able to support this dynamic interaction.

SIoT for DT: What and Why

An approach with the potential to efficiently manage the DTs is based on the exploitation of social networking notions into the IoT, as formalized by the SIoT concept [4]. According to this vision, DTs create social relationships among themselves as humans do, so that they are more willing to collaborate with friends with respect to strangers. The idea is to exploit social awareness as a means to turn communicating objects into autonomous decision-making entities. The new social dimension is able to mimic interactions among users and motivate a drift from egoistic behavior to altruism or reciprocity. Therefore, DTs can autonomously establish social links with each other (by adhering to rules set by their owners) so that "friend" DTs exchange data in a trustworthy manner. The SIoT benefits have been studied to show how Cloud and Edge nodes play a complementary role in information diffusion [9] and how it can provide a possible solution for collaborative Edge Computing [10]. In the following, we describe these perks, which are all desirable for DT-based applications.

Shaping the Network: The ability to quickly navigate the network and thus retrieve the needed service is assured by the existence of network hubs, that is, DTs connected to many other DTs that are able to forward the request to the most appropriate DT. In order to avoid hubs becoming congested, the SIoT structure can be shaped to guarantee the network navigability so that the DTs discovery is performed effectively. Relationships are created and updated on the basis of the objects' features (such as computational power, mobility capabilities, etc.) and activities (frequency in meeting the other objects, mainly), so that each DT can create its own set of relations with other DTs.

Trustworthiness Management: Trust is typically regarded as essential to cooperation and it has been recognized as a critical factor for the IoT [11]. It represents the degree of risk perceived by a DT during a collaboration: if the risk is too high, there is no trust between the parties, there is no incentive to enter a relationship and thus any form of cooperation is not feasible. This means that cooperation between DTs can only be effective if there is trust between them so information can efficiently circulate. Trust management systems can make use of three different contributions: static parameters regarding other DTs, such as the type of friendship or the accuracy of the connected physical object; direct trust, which is computed based on the previous experience of the DT, that is, their feedback regarding a service; and indirect trust or recommendation, which is the information obtained from friends regarding the possible DTs acting as service providers. In SIoT, each DT stores and manages the feedback locally, so that whenever a DT needs to retrieve trust information regarding a possible service provider, it is able to compute the trustworthiness level of its friends on the basis of its own experience. Even if a DT has no direct experience, it can query its own social network to ask for recommendations regarding any previous experience of its friends with that provider. The result is that each DT has its own vision of the network and can decide autonomously whether to trust or not a provider, so there is no central entity that authenticates the exchange of information.

Dynamic Discovery and Composition: The virtualization of every entity of the real world will lead to a dynamic network of billions of DTs so that the creation of smart services enabling 6G applications will foresee the composition of several DTs. By leveraging the social relationships between DTs, SIoT can dynamically discover which DTs can provide the required services. Moreover, every DT can autonomously navigate the network of social DTs so that the service composition is highly customised based on the DT that started the application, offering to the users personalized experiences.

ICN for DT: What and Why

DT-based applications will be centered around information and services, for example, *a virtual*

The ever-increasing variety of objects and technologies will lead to an even more rapid increase in the range of possible applications, which may also have stringent requirements in terms of latency and reliability.

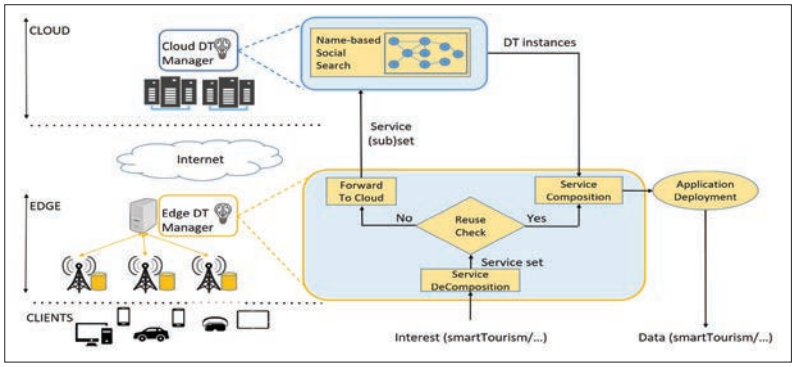

FIGURE 1. Reference scenario and main workflow: Interest packets carrying application requests are processed by the Edge DT Manager, which performs service reuse checks and, in case of failure, forwards them to the Cloud DT Manager for discovering the DTs to be deployed at the Edge.

tour of the Louvre, without a-priori known what DTs will be required to provide them and where the DTs will be located. Therefore, a service-centric communication architecture is needed to facilitate the access to DT services. In this context, we can refer to the ICN architecture, which provides *by design* service-centric primitives [12], as clarified in the following.

Service-Centric Communications: ICN communication is based on two simple, yet effective and robust, network primitives: the *Interest packet*, transmitted by the client to request services by name, and the *Data packet*, transmitted by any connected node providing the requested service and therefore acting as provider [6]. Routing-by-name mechanisms are used to forward the request. Each node maintains a Forwarding Information Base (FIB), that is, a forwarding table listing, per each name prefix, the outgoing interface(s) toward a provider. The name matching in the FIB can be complete or partial. In the presence of more FIB entries with partial matching, the longest prefix matching rule is applied.

With ICN, every DT-based application and related services reachable in the network can be mapped to a specific expressive *hierarchical name* that can be used directly by clients [13]. The name, which may resemble a Uniform Resource Identified (URI), does not need to be translated into the IP address of the device hosting the DT(s), thus facilitating the DT service discovery even in dynamic scenarios.

Service Reuse: ICN supports ubiquitous caching of named data and (reusable) service outputs [7]. Each ICN node *n* maintains a local cache. At the Interest reception, *n* looks for a name matching in the cache and, in case of a positive outcome, it immediately sends the result back to the requester. Otherwise, *n* checks for a name matching in a data structure called Pending Interest Table (PIT), which records the Interests that have been forwarded. If a matching is found, *n* discards the Interest, since an equal request is waiting to be consumed by the Data packet. Otherwise, *n* creates a new PIT entry and looks in the FIB to forward the request. PIT entries are maintained for the Interest lifetime duration, which is properly set by the client to accommodate the service provisioning and included in a dedicated packet header field [6]. If the lifetime expires before a reply is sent back, the entry is discarded.

In-network caching and PIT lookup facilitate the implementation of DT service reuse directly at the network layer. This is a departure from traditional Edge Computing platforms, based on the TCP/IP protocols, that develop additional application-layer solutions in purpose-build edge servers. When distinct clients ask for the same DT service by name, their requests could be served directly in the network, thus limiting the service provisioning latency and the network traffic.

The Envisioned SIoT-ICN Framework

In this section, we present the SIoT-ICN framework to deploy DT-based applications by providing *flexible, trustworthy and fast discovery of DT services and their deployment at the Edge with the possibility of service reuse*. The reference Edge-Cloud scenario is depicted in Fig. 1.

Main Pillars

Social Registration: Whenever a physical node registers on the social network, the related social DT is created on the Cloud and can be moved at the Edge when a client requires its latency-sensitive service(s). Following the idea of DTs as prosumers [14], a DT can play the role of both service provider, offering services when requested, and service requester, consuming services from other DTs. Either way, all the services are described by means of hierarchical names.

At the same time, friendship relations are established based on rules set by the owner of the physical device. During this process, information regarding the services produced and consumed by the DT is exchanged between friend nodes, so that every DT is able to construct an ICN social FIB with the services produced and consumed by its friends.

Name-Based Services: We assume that hierarchical namespaces are defined to identify the available applications as well as the atomic services offered by DTs. Application names are composed of a main prefix identifying the application category, for example, */smartTourism*, */smartHome*, followed by a variable set of components further specifying the related services and their possible input parameters, for example, regarding the expected quality of service and trust. For instance, the names */smartTourism/Louvre/Egyptian{720p/top10/0.6}* and */smartTourism/Louvre/Greek/{1080p/top10/0.8}* identify a virtual tour experience of the top 10 masterpieces of the Louvre Egyptian and Greek collection with, respectively, HD and full HD resolution and trust values at least equal to 0.6 and 0.8. The name *Louvre/Greek/VenusdeMilo{1080p}*, instead, identifies an atomic service that can be included in the virtual tour on Greek antiquities, that is, a video about the Venus de Milo sculpture.

Service Reuse at the Edge: As shown in Fig. 1, we consider an Edge network domain, for example, a Radio Access Network (RAN) and a backhaul network, where nodes implement the ICN architecture. An Edge DT Manager (EM) supervises all the edge resources and orchestrates the application deployment. It checks if a named application request from a client can be directly satisfied at the Edge, for example, because another client has recently requested the same application and the related DTs are locally cached. Therefore, the EM maintains two named data structures, a cache and a PIT, which record, respectively: the name of the services cached at

the Edge and the corresponding hosting node, and the received pending Interests. To improve scalability and avoid the single point of failure, a logically centralized but physically distributed EM can be implemented.

Cloud Management: Application requests that are not solved at the Edge are forwarded to the Cloud to instruct the service provisioning. There, an intelligent module, referred to as Cloud DT Manager (CM), issues the service discovery and composition in the overlay SIoT network to identify the targeted DTs. A name-based look-up in the social graph, with the possibility of reusing existing findings, is performed to speed up the search, as it will be clarified in the next sections.

Operational Steps

Request Processing at the Edge: As shown in Fig. 1, Interest packets received by the edge nodes are forwarded to the EM, which accesses the application name and performs a Service DeComposition operation, that is, it tries to identify the set of requested DT services, and checks if they are already available at the Edge with the required quality and trust parameters.

If a name matching is found for every atomic service, the application can be directly provided at the Edge. Therefore, the set of services is composed and delivered to the client. If the cache lookup fails, the EM looks in the PIT to check if an equal application request has been already received and is waiting to be solved. In case of a positive outcome, the EM simply updates the PIT entry and waits for the instructions of the CM.

Conversely, if the EM is not able to completely solve a query and reuse previous DT instances, it forwards a request including the (sub)set of missing services to the CM.

Name-Based Social Search: The CM is managed following the SIoT paradigm and makes use of objects' social connections through word of mouth among DTs to look for the requested services. The social FIB maintained by each DT contains useful information for the search, such as the case of the following tuple: *service name prefix, service type (either produced or consumed), friend ID, trustworthiness*. Whenever a friendship is created or terminated the corresponding entries related to the friend ID in the FIB are added and deleted respectively. If a DT is not able to provide a service anymore, then it informs all its friends that the service is no longer available, so that they can update their FIB and the related request is forwarded to other friends.

Indeed, whenever a DT receives a request from a friend, it will match the request's prefix with the prefixes available in its social FIB. If a complete match is found, then the DT will return to the requesting node a pointer to the friend that can solve the query. Otherwise, the DT will forward the request to other friends based on a different logic, which could include social parameters, such as trustworthiness: a DT could implement algorithms based on the longest prefix matching or directing the query to friends with similar services consumed. The definition of a social search algorithm is left for future studies.

When a node receives the requested service, it will also cache the service output so that the same data could be useful for other friends with similar interests. For instance, let us consider Fig. 2, where node 1 is interested in a service named *Reggio/LabIoT/Temperature*. The relevant DT will look in its FIB for friends that can provide the service by either having a good match with the query, for example, because they produce similar services, for example, */Reggio/Home/Humidity*, or by having similar interests, that is, similar services consumed, and then it is likely they have previously requested the same service. So node 1 forwards the request to node 2 based on its FIB, due to the similarity in the two services. Node 2 has *Reggio/LabIoT/Temperature* in its FIB, which is produced by friend ID 3. Therefore, node 2 can provide back to node 1 the pointer to node 3, which represents a complete match for the requested service. However, if node 2 has already cached the service, then it will directly provide it back to node 1.

Application Deployment: To finally deploy the requested application, the CM identifies the potential DTs that should be instantiated at the Edge based on their latency requirements. Then, the CM informs the EM about their requirements in terms of computation and storage capabilities. If the requested edge resources are available, the EM acknowledges the hosting of the DTs at the Edge and enables their placement. Vice versa, the EM transmits a negative acknowledgment to the CM which, depending on the expected latency requirements, can either: notify the client about the current inability to support the requested application, or let the Cloud provide the application without the Edge support.

Use Case in Action

To better understand the application deployment in SIoT-ICN, let us refer to the mentioned use case, where Alice client application requires a virtual tour of the Louvre in Full HD with trust set to 0.5, by sending an Interest with name *smartTourism/Louvre/{1080p/0.5}*. Alice does not a priori know what to see and let the framework discover what best suits her. Therefore, the EM forwards the request to the CM that issues the name-based search in the social network. By leveraging the social relations with friends sharing the same cultural interests, it is found that Alice should see the top Greek artworks and Italian paintings and the corresponding DT services' names are returned. After checking that there are enough resources for hosting these services at Edge, the application is deployed. In the meantime, if the EM receives a request from another client targeting the same DT services, service reuse takes place.

Preliminary Evaluation

In this section, we present a preliminary evaluation in Matlab targeting the name-based social search, and the application deployment with the placement of DTs at the Edge.

Impact of the Name-Based Social Search

To test the performance of the designed name-based social search and caching mechanism, compared to the legacy SIoT implementation [4], we consider the large dataset of a SIoT scenario in [15] consisting of a network of 16216 devices owned by 4000 users and by the municipality of Santander (Spain), which create their own relations over 11 days.

In-network caching and PIT lookup facilitate the implementation of DT service reuse directly at the network layer. This is a departure from traditional Edge Computing platforms, based on the TCP/IP protocols, that develop additional application-layer solutions in purpose-build edge servers.

Node 2

Service prefix	Service type	Friend ID	Trust
/SmartTourism/Louvre/Statue/Renaissance	Consumed	3	0.77
/SmartTourism/Colosseum	Consumed	1	0.72
/Cagliari/LabIoT/Temperature	Produced	3	0.91
/Reggio/LabIoT/Temperature	Produced	3	0.8

Node 1

Service prefix	Service type	Friend ID	Trust
/SmartTourism/Egypt/Gyza	Consumed	2	0.84
/Reggio/Home/Humidity	Produced	2	0.64

Node 3

Service prefix	Service type	Friend ID	Trust
/SmartTourism/Egypt/Gyza	Consumed	2	0.46
/Reggio/Home/Humidity	Produced	2	0.64

FIGURE 2. Example of Social FIB for three nodes involved in the name-based social search.

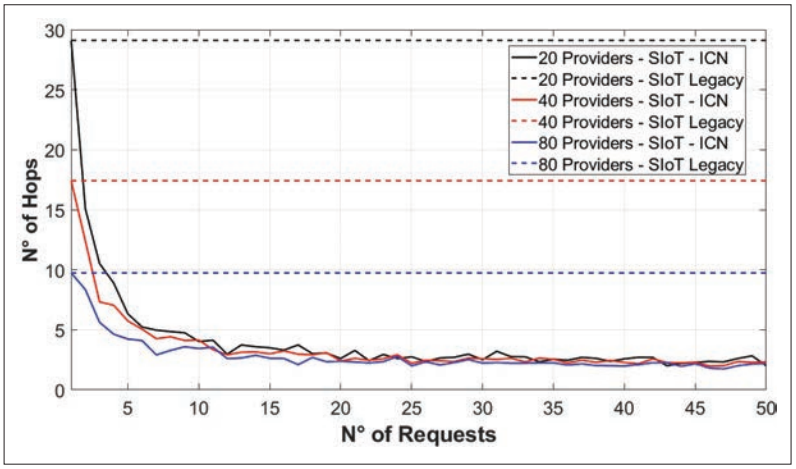

FIGURE 3. Average hop number for service discovery in the social network, when varying the number of requests and providers.

The simulations are performed considering 18 different services in the network, where each DT usually has 1 or 2 services. However, to simulate DT services with different levels of diffusion, we suppose that random DT requesters look for a service available from 20, 40 and 80 DTs. Moreover, nodes do not have full knowledge of the global social network connectivity, but they explore the network using only local knowledge. Therefore, we test two approaches with the same search mechanism, where each DT only selects a single friend as a next hop to forward its request based on the friends' degree and its distance from the requested service. Basically, a DT will contact its friends and the friends of its friends until the service is found. More advanced algorithms able to optimize the search process are left for future studies. The main difference between the two approaches is that, in the SIoT-ICN approach, every DT on the path caches the service for future use, according to the basic ICN caching strategy [12].

Figure 3 shows the average number of hops, that is, of friends, needed to find the service in SIoT-ICN and in the legacy SIoT architecture. In the latter case, nodes are able to retrieve services making only use of their social connections: even if more advanced search mechanisms could be able to achieve better results, their efficiency would still depend on the service diffusion. Instead, with SIoT-ICN, any DT obtains a hops gain after only 5 to 10 requests and finds the required service in around 3 hops, regardless of the service diffusion. This means there is no need to overload the network with the same cached content, and, even if the content is replaced, the DT can learn which friends can efficiently reach the destination to require similar data.

Impact of DT Placement and Reuse

To simulate the application deployment step of the conceived framework, we assume that DTs of two distinct sizes, namely 100 MB and 200 MB, must be instantiated at the Edge. The edge network domain consists of 16 base stations which clients are connected to, and a backhaul network organized as a three-layered fat tree topology. The edge links latency is uniformly distributed in the range [2, 5] ms. The root node connects the edge domain to the Cloud and the latency over the edge-cloud path is set to 30ms. To cover a variety of application settings, we also assume that each DT consumes input data in the range 50-500 MB. Moreover, we consider the Zipf distribution with skewness parameter $Z = 0.8$ and $Z = 1.2$ to model the popularity of the DTs. The higher the skewness, the more the service requests will be concentrated on few popular DTs. Therefore, despite the limited storage capabilities compared to the Cloud, Edge caching will be able to accommodate a large number of requests.

Figure 4 shows the performance in terms of latency of the overall deployment, which includes the time to place the DTs at the Edge, and the time to retrieve and process the input data. If the requested DTs are already installed at the Edge, SIoT-ICN enables service reuse thus cutting down the latency.

We compare SIoT-ICN against a Cloud architecture, where all requests are forwarded and managed in a remote Cloud, and an Edge architecture, where applications can be deployed at the Edge, like in SIoT-ICN, but without service reuse.

It can be observed that the latency increases with the size of the DTs that move at the Edge. However, the benefits of Edge placement get remarkable as the size of the input data increases. Being data-intensive entities that constantly need to interact with users and physical objects, it is expected that huge amount of data will be consumed by the DTs during their lifecycle. Therefore, Edge placement is crucial to limit the latency of DT-based applications.

Finally, it can be observed that, with SIoT-ICN, the latency reduction is more remarkable when $Z = 1.2$. Indeed, there is a higher probability of reusing the services of DTs already instantiated at the Edge.

Research Directions

The proposed SIoT-ICN framework can be optimized in several ways, so that major research opportunities exist in all the following areas:

- *Expressive DT representation model* definition to manage the DTs' dynamic nature. Unlike "hardcoding" a service request to specific resources (software and hardware), each model should implement the soft binding of requests to their needs in terms of functionalities and features, letting the DT Managers dynamically perform the best match. In addition, it should allow for the easy

"re-mapping" of a service request to needed features, and endow social DTs to collaborate based on interactions among DTs, people, and the environment.

- *Scalable autonomous social search* among a huge number of DTs, with intensive interactions, heterogeneous communications, and billions of services. Optimal social navigation should be faced by properly identifying relevant parameters to let the DTs rely only on local information.
- *Optimal service placement* in the presence of multiple Edge nodes with distinct computation and storage capabilities. The proposed framework calls for the definition of proper allocation algorithms based on several parameters (e.g., closeness to clients and edge overloading) and coping with a variety of requirements (e.g., application latency, energy efficiency).
- *Service reuse policy design* to understand which service caching can be highly beneficial. Investigated metrics can range from service popularity to information about the service space/temporal locality.

Besides the above-mentioned research directions, the SIoT-ICN framework can be further enhanced by tackling well-known issues, such as the data consistency when migrating DTs from Cloud to Edge and vice-versa, and digital forgetfulness of unneeded DTs.

Conclusion

In this article, we presented and preliminary evaluated the SIoT-ICN framework for the deployment of DT-based applications in Edge-Cloud environments. By adding the sociality aspect to the ICN operations, our proposal implements social-driven communications with service reuse at the Edge. Moreover, by introducing expressive name-based primitives in the social network of DTs, service discovery is improved.

These findings open new research opportunities for SIoT and ICN in the 6G era, where Edge networks will be characterized by the convergence of communication, caching and computing.

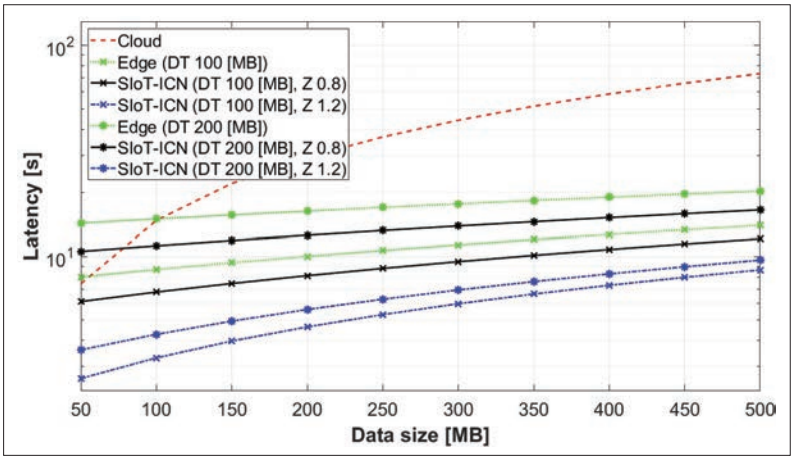

FIGURE 4. Latency performance when varying the size of the DTs and the input data size.

References

[1] L. U Khan *et al.*, "Digital-Twin-Enabled 6G: Vision, Architectural Trends, and Future Directions," *IEEE Commun. Mag.*, vol. 60, no. 1, 2022, pp. 74–80.

[2] A. Fuller *et al.*, "Digital Twin: Enabling Technologies, Challenges and Open Research," *IEEE Access*, vol. 8, 2020, pp. 108,952–71.

[3] W. Sun *et al.*, "Reducing Offloading Latency for Digital Twin Edge Networks in 6G," *IEEE Trans. Vehicular Technology*, vol. 69, no. 10, 2020, pp. 12,240–51.

[4] MS Roopa *et al.*, "Social Internet of Things (SIoT): Foundations, Thrust Areas, Systematic Review and Future Directions," *Computer Commun.*, vol. 139, 2019, pp. 32–57.

[5] S. Mastorakis *et al.*, "Icedge: When Edge Computing Meets Information-Centric Networking," *IEEE Internet of Things J.*, vol. 7, no. 5, 2020, pp. 4203–17.

[6] M. Król and I. Psaras, "NFaaS: Named Function as a Service," *Proc. 4th ACM Conf. Information-Centric Networking*, 2017, pp. 134–44.

[7] B. Nour, S. Mastorakis, and A. Mtibaa, "Compute-Less Networking: Perspectives, Challenges, and Opportunities," *IEEE Network*, vol. 34, no. 6, 2020, pp. 259–65.

[8] F. Niccolucci, A. Felicetti, and S. Hermon, "Populating the Data Space for Cultural Heritage with Heritage Digital Twins," *Data*, vol. 7, no. 8, 2022, p. 105.

[9] Y. Yi *et al.*, "Social Interaction and Information Diffusion in Social Internet of Things: Dynamics, Cloud-Edge, Traceability," *IEEE Internet of Things J.*, vol. 8, no. 4, 2020, pp. 2177–92.

[10] P. Dong *et al.*, "Collaborative Edge Computing for Social Internet of Things: Applications, Solutions, and Challenges," *IEEE Trans. Computational Social Systems*, vol. 9, no. 1, 2021, pp. 291–301.

[11] I. U. Din *et al.*, "Trust Management Techniques for the Internet of Things: A Survey," *IEEE Access*, vol. 7, 2018, pp. 29,763–87.

[12] A. Afanasyev *et al.*, "A Brief Introduction to Named Data Networking," *Proc. IEEE MILCOM*, 2018.

[13] M. Shahrul *et al.*, "Hierarchical Naming Scheme In Named Data Networking for Internet of Things: A Review and Future Security Challenges," *IEEE Access*, 2022.

[14] G. Xiong *et al.*, "A Survey on Social Manufacturing: A Paradigm Shift for Smart Prosumers," *IEEE Trans. Computational Social Systems*, 2022.

[15] C. Marche *et al.*, "How to Exploit the Social Internet of Things: Query Generation Model and Device Profiles' Dataset," *Computer Networks*, 2020, p. 107248.

Biographies

Marica Amadeo is an Assistant Professor in Telecommunications at the University Mediterranea of Reggio Calabria, Italy. Her research interests are mainly in the fields of Information Centric Networking, Vehicular Networks, Caching and Edge Computing.

Marco Martalò is currently an Associate Professor of telecommunications with the University of Cagliari, Italy. His research interests are in the design of communication and signal processing algorithms for wireless systems and networks, as well as security aspects to them.

Giuseppe Ruggeri is currently Associate Professor at the University Mediterranea of Reggio Calabria. His main research interests include Edge Computing, Internet of Things and Social Internet of Things.

Michele Nitti is an Assistant Professor at the University of Cagliari, Italy since 2015. Currently, he is a member of the editorial board for the IEEE IoT Journal, Elsevier Computer Networks and MDPI IoT. His main research interests are in architecture and services for the Internet of Things (IoT).

Open Access funding provided by Università degli Studi di Cagliari within the CRUI CARE agreement.