INVITED PAPER

Scalable Personalized IoT Networks

In the context of the Internet of Things, this article surveys the readiness and interplay of various distinct technologies, including scalable sensing, information-centric networks, and AI, that are likely to be necessary in order to enable personalized IoT networks.

By Amr El-Mougy, *Member IEEE*, Ismael Al-Shiab, *Student Member IEEE*, and Mohamed Ibnkahla, *Senior Member IEEE*

ABSTRACT | The Internet of Things (IoT) has enabled unprecedented interactions with our physical world, with the aim to deliver a wide range of customizable services in many domains. With recent advancements in IoT technology, users are increasingly expecting these services to be intelligent and context aware. Nevertheless, there is still no framework capable of delivering personalized IoT services on a large scale. For such a framework to be conceived, it is likely that technologies from many domains have to be utilized. This paper examines the readiness of the leading state-of-the-art technologies in several key fields for realizing the goal of a truly scalable and personalized IoT experience. We discuss the important requirements and challenges for realizing this goal. Then, we identify the major approaches that can contribute to this goal and categorize them into: technologies for adaptive personalized sensing, scalable solutions for user-centric networking, and intelligence techniques that leverage context awareness and adaptability at the application and system levels. In the first category, our discussion centers around virtualization and reprogrammability at the sensing layer. In the second category, we investigate the readiness of Fog computing and information-centric networking to develop scalable personalized IoT infrastructures. These approaches were chosen for their combined ability to match dynamic user requirements with available system resources, while guaranteeing overall efficient utilization. Finally, in the third category, we examine context awareness, reasoning, and machine learning techniques as well as semantic technologies for realizing proactive and adaptive intelligent IoT systems and applications. This paper offers a focused discussion of the key topics that drive the research in the important and timely topic of scalable and personalized IoT networks.

KEYWORDS | Artificial Intelligence (AI); context awareness; fog computing; information-centric networking (ICN); intelligent networks; personalized Internet of Things (IoT); ubiquitous sensing.

I. INTRODUCTION

The Internet of Things (IoT) has revolutionized the way we interact with our physical world. Everyday, new devices are being connected at a rate that may reach tens of billions by 2025 [1]. These devices provide tremendous volumes of data and services in limitless domains on a scale never before witnessed in the world. This has been made possible by significant advances in many fields such as ubiquitous sensing, dynamic and programmable networking, communication technologies, and Fog computing, which have allowed for the development of numerous smart systems and applications [1]–[4].

With these advancements, users are increasingly demanding context aware, intelligent, and often proactive IoT experiences. This leads us to ask the following question: How smart are smart systems really? Modern systems often support many customizable features and adaptations. Nevertheless, these smart features usually come in generic one-size-fits-all implementations, preventing users from being able to fine-tune how their applications serve them. More importantly, most adaptations are done at the application level. Underlying layers still largely support applications in a generic way, with customizations applying to aggregated applications or classes of traffic.

Manuscript received July 31, 2018; revised December 1, 2018; accepted January 11, 2019. Date of publication February 20, 2019; date of current version March 25, 2019. This work was supported in part by the Cisco Research Chair in Sensor Technology for the Internet of Things, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC)/Cisco Industrial Research Chair in Sensor Networks for the Internet of Things, in part by the Carleton University Start-Up Grant, in part by the NSERC Discovery Grants Program, and in part by the Canadian Foundation for Innovation. *(Corresponding author: Amr El-Mougy.)*
A. El-Mougy is with the Faculty of Media Engineering and Technology, German University, Cairo 11432, Egypt (e-mail: amr.elmougy@guc.edu.eg).
I. Al-Shiab and **M. Ibnkahla** are with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S5B6, Canada (e-mail: mohamed.ibnkahla@carleton.ca).

Digital Object Identifier 10.1109/JPROC.2019.2894515

The challenges to support personalized IoT services become compounded in scalable networks. Even though there are already many proposals for intelligent IoT networks on the scale of a smart home [5] or building [6], there are completely different challenges to host services on the scale of a smart city. Here, a large number of devices need to be interconnected and large volumes of data need to be transported. In personalized networks, this has to be done while matching resource management tasks with dynamic user requirements. Users also expect their intelligent services to be seamless, even across heterogeneous network boundaries that may be under different administrative or management domains.

Accordingly, in this paper, we provide a thorough and comprehensive study of scalable personalized IoT networks. We focus on personalization below the application layer, which we define as techniques, systems, and protocols for enabling the network to cater to individual user requests and requirements. In IoT environments, these personalization techniques can be implemented in many contexts such as direct interaction between users and sensors, where personalization here can imply adapting sensor operation (duty cycles for example) to serve each individual user; between users and edge devices, where adaptations may include predictive resource provisioning; and between users and the core network, where personalization can include programmable and fine-grained resource management per individual user request.

We utilize a layered approach to this paper, as shown in Fig. 1. In the first layer, we discuss recent advances in the sensing layer. In scalable IoT networks, hardware reuse is critical for cost efficiency. Thus, we study how personalized adaptations can be ensured with hardware reuse through technologies such as virtualization and programmability at the sensing layer. Particularly, we investigate virtualization on several levels (sensor, gateway, and network), and how these solutions enable personalized sensing.

The second layer focuses on Fog computing, which can leverage local context-awareness and provide scalable resources close to the edge. Thus, we investigate the ability of state-of-the-art Fog computing systems to support individual user needs. In the third layer, we investigate one of the most prominent core networking technologies for the IoT, which is information-centric networks (ICNs). This is selected for its information dissemination capabilities, particularly based publish/subscribe communications. Our goal is not to survey ICN architectures, as many comprehensive surveys are already available for them [7], [8]. However, our goal is to examine their readiness and suitability for scalable personalized IoT networks. Thus, we focus here on particular related aspects such as mobility, adaptability, and the ability to support dynamic requirements on the scale.

The first three layers comprise the infrastructure necessary to support the personalized applications and services at the third layer. The fourth layer investigates how to make these networks intelligent. Thus, we study Artificial Intelligence (AI) techniques such as reasoning and machine learning [9], as well as context awareness, and examine how they can be used to achieve the goal of scalable personalized IoT networks. We analyze the readiness of state-of-the-art techniques for the requirements of large scale predictive and personalized services and identify the methods with the highest potential. It is important to note that intelligence techniques have been proposed across the other three layers of the architecture in Fig. 1. This will be discussed in the remaining sections of this paper.

II. CHALLENGES AND REQUIREMENTS

IoT environments have specific challenges that are now well known and have been clearly established in [3], [10], and [11]. These challenges include energy efficiency, limited device capabilities, and dynamic application requirements. However, this section focuses only on the challenges associated with scalable and personalized IoT networks.

In particular, we discuss the challenge of personalization itself: how can IoT services on a large scale be customized to the dynamic needs of every user, while maintaining overall system efficiency and low cost? This includes techniques that directly involve users such as predicting mobility patterns, and also networking techniques that have an impact on user experience such as resource management. We investigate the interplay between system components and the tradeoffs that arise from within. We also discuss architectural imperatives of a scalable IoT network, and how services can be deployed quickly and efficiently.

Another important challenge is the dynamic and diverse service requirements that are to be expected. These requirements may include latency, energy efficiency, reliability, and so on. IoT services are expected to share resources of the underlying infrastructure. This means that the infrastructure may be required to support conflicting requirements at the same time. Prioritization, resource management, and scheduling are among the solutions to this problem.

A. Personalization Aspects

One of the most important defining attributes of human users is their unpredictability. People want services on-demand without time constraints. This can be highly challenging in a network that has to switch off its devices periodically to save energy. One possible solution to this challenge is to implement a degree of redundancy in the network, which is an expensive solution. Another solution would be to anticipate demand and try to adapt network operations accordingly.

To address the challenge in deployment cost, it is highly recommended that devices be reused across different services and applications. This can result in faster service deployment and significant financial savings. However, many challenges arise in this case. First, how can device operation (including duty cycling) be managed across the demands of multiple services. Second, how can service

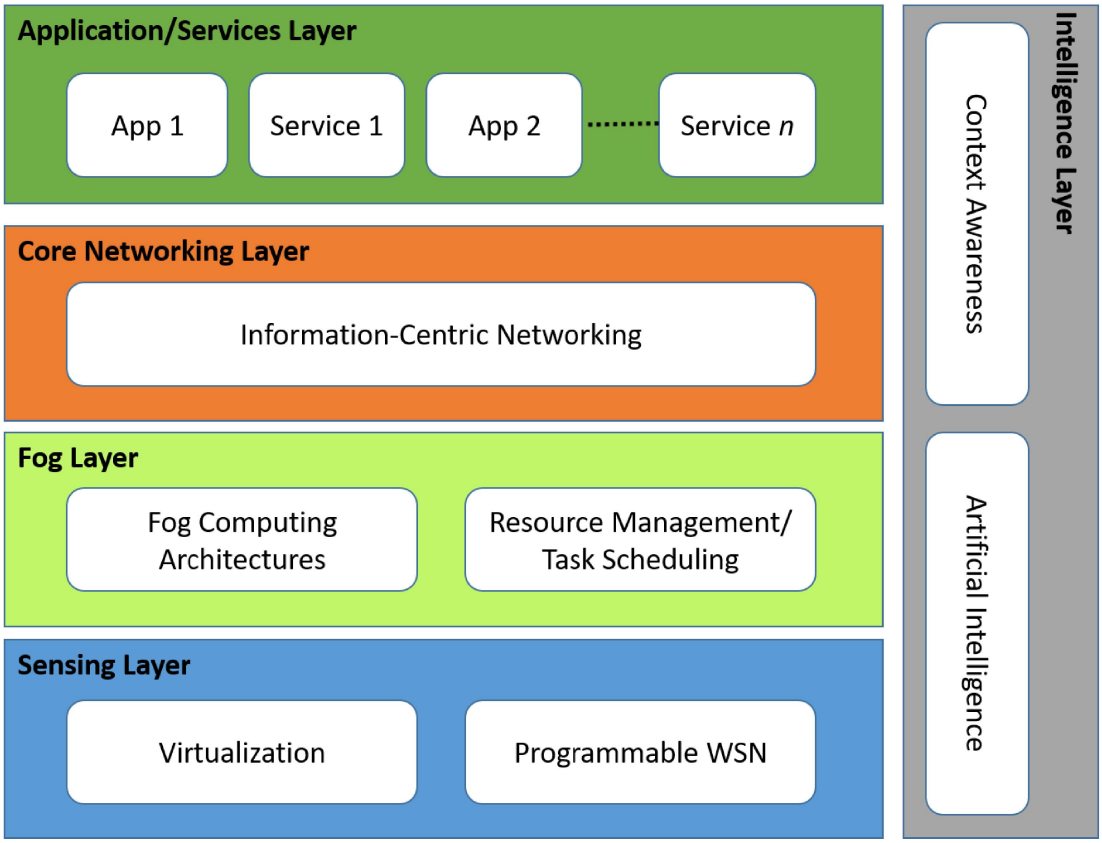

Fig. 1. *Layered approach to the study of personalized scalable IoT.*

composition be formulated using shared resources. Also, how can devices accommodate the service heterogeneity and ensure compatibility for the largest possible percentage of users. These challenges may be addressed through standardization efforts or through intelligent frameworks that ensure flexible deployment.

Another important issue is how users can discover available resources according to their current needs. Current technology can probe its immediate surroundings to discover resources [12]. However, application and service requirements are likely to go beyond what the immediate neighborhood of a user can offer. Hence, service discovery in IoT has become a thriving research direction [13]–[15]. There is a critical need for IoT systems to be able to predict the services that users may require and discover their presence ahead of time.

It is also important to note that users are increasingly becoming reliant on context aware and intelligent services, as mentioned before. The fields of context awareness and AI include a very large number of techniques [9] used to design systems of ubiquitous computing, pervasive computing, or ambient intelligence. Even though these terms are closely related, they have core differences. Ubiquitous computing refers to computing everywhere, where everyday objects are transformed into "smart" objects, capable of sensing and adaptation activities. Examples include a smart traffic light that counts vehicles to determine congestion. Ubiquitous computing is often mixed with pervasive computing. However, it is generally acceptable in the computer science community to refer to pervasive computing as the paradigm that allows a user to control their environment using smart devices. These devices might also be context aware, providing recommendations and more intelligent adaptations to the user. Finally, ambient intelligence refers to embedding invisible and intelligent services in our surroundings in order to perform sensing, computing, and adaptation tasks for the users.

There is a limited effort that studies how these intelligent computing paradigms interact with the underlying infrastructure. Ultimately, these paradigms require that information is extracted from and distributed in the network in a timely manner to the interested parties. How these parties can be defined, and how their interests change with time, is not easy to determine. Also, in a large scale network, it may be important to prioritize the needs of certain computing systems over others. This may not only depend on the criticality of the underlying application itself but also on the complexity of the algorithms used in these computing paradigms.

B. Architectural Imperatives

To design an efficient scalable IoT network, the optimum arrangement of physical and logical elements is crucial. There are many decisions to be made in this case. First, the overall hierarchy of the network needs to be designed.

Second, geographical placement of the network elements needs to be determined, as well as the position of each element in the hierarchy. There are many technologies available in virtually every aspect of network design (communication technology, sensing technology, networking technology, etc.). It is vital to choose the optimum mix that can guarantee a sufficient degree of compatibility while hosting the desired network capabilities.

In order to make these choices, the first step would be to clearly specify the system requirements. These may include energy efficiency, reliability and robustness, multitenancy, mobility support, modularity, scalability, and openness, to name a few. Addressing all these requirements is highly challenging, even if the network is under a single administrative control. Most likely, such a large scale system will be under multiparty control, making things even more challenging. Thus, a possible solution would be to open the stage completely for organizations and individuals to deploy their own services and network platforms, and leave it up to governments to regulate deployments and provide incentives to follow good practices.

C. Connectivity Issues

Modern ubiquitous technologies demand seamless connectivity at all times [16]. This can be highly challenging in IoT networks for many reasons. First, communication technologies for the IoT are mostly low power and low range (e.g., Bluetooth and ZigBee) and utilize duty cycling to extend the lifetime of their batteries. Recently, technologies such as LoRA and Sigfox offer long-range connectivity but at significantly low rates (often under 1 kB/h). Thus, the challenge for personalized IoT is to offer uninterrupted services in the presence of intermittent connectivity. Some research efforts have suggested using cellular networks as an IoT umbrella [17]. This solution ensures connectivity at the expense of the high energy consumption associated with cellular networks. It is worth mentioning that 5G networks offer special support for the IoT. These standards are surveyed in detail in [16].

Device heterogeneity is also a significant challenge in IoT networks. There is a wide range of incompatible communication standards in the IoT market and they satisfy the requirements of different applications. Thus, the challenge is how to be able to connect through all these technologies. Some research efforts have addressed this challenge through the use of ubiquitous gateways [18]. However, in this case, these gateways become a single point of failure for the devices they serve and an attractive target for attackers. In Section III, we will discuss how these gateways can be virtualized to allow adaptive service deployment.

Mobility is also an important challenge that impacts connectivity. Users expect services on the go, even at high speeds (such as on the train). On the other hand, IoT communication technologies typically have limited ranges and are not designed to support high mobility. Up to this day, a cellular umbrella seems to be the only solution to this problem. Mobility prediction can also be used to provide an illusion of seamless connectivity by caching content in advance of outages.

D. Latency and Dynamic User Requirements

The future smart city shall host a large set of services and applications. For this environment to be personalized, it has to cater to different user and application requirements that change over time. Flexible and programmable networking is an important solution. This is offered by technologies such as software-defined networks [19] and network function virtualization (NFV) [20]. The philosophy behind these technologies is often extended beyond the scope of the routers and switches, to the sensing layer devices themselves, as will be discussed later in this paper. This is a promising direction as it enables a high degree of automation in the network. However, service orchestration and resource management across the sensing layer and network core are still open issues.

Some applications also demand strict latency guarantees. Traditional networking paradigms may not be at all suitable for these applications. Thus, paradigms such as Fog computing are proposed specifically to address this challenge [21]. Fog computing reduces the round trip delay and brings computation resources close to the user. However, migrating services to the Fog layer is not straightforward, as it brings forth deployment challenges and additional infrastructure costs [22]. Mobility also complicates the support for dynamic user requirements. Shared network resources may not be able to provide support for users, especially under unpredictable mobility paths. In this case, resource reservation and planning are not possible, and users may end up experiencing changing quality.

III. SENSING LAYER

In IoT applications, users request services through the cloud that may require data collected from sensing networks (SNs) under one or more administration domains. In scalable and personalized IoT, user requirements are expected to be dynamic. Moreover, users may change the service scale in terms of covered geographical area and the number and type of sensors used. In this environment, there is a need to form composite sensing services on the fly to satisfy the changing user needs. These sensing resources should be released when they are no longer needed by the user. Here, virtualization and programmability are important tools for addressing this critical challenge, which is why they are chosen as the focus of our discussion in the first layer of the reference architecture shown in Fig. 1. Virtualization decouples the offered services from the hardware, which allows hardware reuse among different applications, service automation through initiation, deletion, and scaling, and support for on-demand service deployment. Programmability enables network reconfiguration based on the needed services and thus better network management.

A. Scalable and Personalized IoT Requirements in the Sensing Layer

Users request services by launching their applications. These services are then decomposed and translated into functional requirements that could be satisfied by the sensing resources, as shown in Fig. 2. Afterward, core networks represented in Fog, cloud, and Internet service providers transfer the generated sensed data in an efficient and secure way to the appropriate applications. Any changes in the requested services or their scale should be satisfied within acceptable time and cost while assuring security and privacy, while achieving overall utilization of system resources. Examples of sensing service requirements translated from the user applications might include covered geographical area, sensing accuracy, and the maximum accepted latency. These requirements need optimized resource allocation of sensing resources to application requirements. Moreover, it requires a quantitative calculation of propagation delay, middle nodes processing delay, buffering/queuing delay, and throughput. All of the mentioned service requirements are expected to be dynamic in personalized and scalable IoT.

In addition, applications might need SNs that expand into different administration domains and geographical locations. In this case, two additional requirements of high importance to the user are the total cost and the application time-to-deploy. The total cost depends on the scale and Quality-of-Service (QoS) requirements of the requested service, while the time-to-deploy depends on the ability to form an SN that satisfies the requirements. Moreover, an overall system objective is the utilization of existing sensing resources, which is not of direct importance to the user but is related to the time to deploy and is of great importance to the system as a whole.

B. Scalable and Personalized IoT Using Virtualized Sensing Networks

To satisfy personalization and scalability requirements, virtualization at the different levels of the sensing layer including sensor, gateway, and network offers a great opportunity. In this case, virtualization could be implemented by mapping virtualized SNs to different applications. This mapping satisfies initial requirements of the applications and creates an adaptation layer that reacts to changing user requirements. Furthermore, the virtualization reduces the time and cost to scale services through hardware reuse and achieves better overall utilization of sensing resources since virtual SNs (VSNs) could be quickly formed as needed on the top of already existing physical SNs/devices. Moreover, virtualization assures users' privacy and security through isolating the different VSNs and having complete control over the interactions between them. On the other hand, VSNs might incur overhead that results from their formation and maintenance. However, virtualization approaches such as SenShare [23] show that VSNs can be designed and deployed

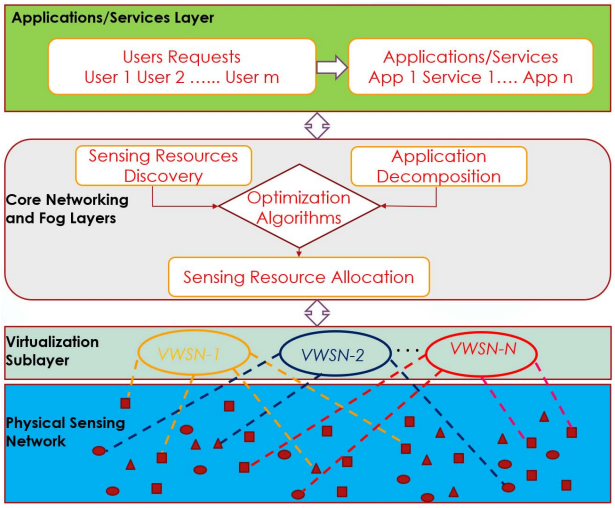

Fig. 2. *VSNs for personalized and scalable IoT.*

while still having light computing and virtualization management overhead.

SN virtualization comes in three levels: sensor level, gateway level, and SN level. Sensor-level virtualization can be achieved using sensor virtual machine (VM)-based virtualization approaches such as Mate [24], VM* [25], and Squawk [26]. In such solutions, complete VMs are deployed on the sensor motes and provide sensed data to different applications. Sensor nodes connect to the outside world through gateways and communicate directly or through multihop.

User security requires securing devices at the sensing level, which could be achieved by using IoT gateways. IoT gateways play a significant role in isolating real sensors from the outside world along with achieving classical gateway functions like protocol conversion, sensed data delivery to upper layers, and support for sensor control and management. However, they might form network bottlenecks and target for security attacks. Such problem could be resolved using virtualized IoT gateways and deploying it on-demand in locations and numbers based on changing requirements and with redundancy and ability for migration. There are few interesting efforts in the literature for virtualizing gateway functionalities in NFVs and moving part of it to the service providers/cloud to support scalability, elasticity, and mobility as in [27]–[30]. Also, container-based technologies such as LEGIoT [31] build on the latest enhancements in containers and support resources utilization through on-demand services and multitenancy.

SN-level virtualization allows forming scalable number of VSNs using overlay based [23], [32]–[34], or cluster based [35]–[38]. In overlay-based approaches, an overlay network is built on top of the existing SNs, using middlewares for example. In cluster-based approach, clusters of related sensor nodes are formed to serve the different applications where clustering could be optimized to better satisfy personalization and scalability needs. Sensor-level programmability is assumed at the sensing layer to be able to form the VSNs and could be realized using

modular and efficient sensor operating system (OS) like Riot [39] or Contiki [40]. Modular sensor OS consumes less resource and offers continuous and efficient reprogrammability on the sensor level.

Fig. 2 shows the cross-layer process from applications to sensing resources allocation and VSNs creation. Users specify preferences at the application layer without worrying about the underlying details. Moreover, they can change their requirements and scale as needed during the application lifetime. The applications and support/maintenance services are then decomposed and translated into functional requirements at the Fog and Core networking layers. In parallel, the sensing resources discovery and logging continuously update the pool of existing resources and their functional capabilities. Afterward, optimization algorithms and techniques evaluate the best resource allocation scheme while considering cost and time-to-deploy. The resultant allocated resources for every application are then used to form the needed VSNs at the virtualization sublayer and the optimization and allocation continue with every change in the user's requirements.

In the virtualization sublayer, the different VSNs are assumed to interact with sensors through the IoT gateways, where the latter takes the responsibility of communicating disseminated functions/configurations to the sensing nodes and supports other classical gateway and data collection tasks.

C. Sensing Resources Allocation

In personalized and scalable IoT, sensing resources allocation is a continuous and challenging process. First, the applications compete for limited shared resources and conflicting QoS requirements. Second, the physical sensing resources might fail at anytime and affect the provided services. Third, the requirements are very dynamic and continuously change during application's lifetime. Fourth, the user's application scale might expand into multiple SNs under different administration domains.

In virtualized SNs, sensing resources are differentiated from nodes/motes and abstracted into a set of attributes [41]. These attributes are directly related to the functional requirements requested by the applications. For example, a sensing node could be abstracted into the following functionalities: temperature and humidity sensing, camera recording, and gyroscope sensing. As a result, the SN is abstracted to the overall set of attributes that exist in the sensing nodes under its administration. In parallel, applications are decomposed into a set of functional requirements that could be satisfied by the existing resources, knowing their attributes in advance. The mapping of applications to sensing resources then becomes straightforward. Let us define R as the set of all available resources in the SN, N as the total number of sensing nodes, A as the total number of applications, F as the total set of decomposed functional requirements, and S as the set of all applications' serviceability represented in the satisfied decomposed functional requirements to the requested ones. In this case, resources and functional requirements could be presented in the following equations:

$$R_{n,r,i} = \begin{cases} 1, & \text{if } R_{n,r} \text{ is used for application } a_i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$F = \cup_{s=1}^{A} F_s \quad (2)$$

where $R_{n,r}$ is the rth resource in node n and F_s is the set of all functional requirements for an application.

For dynamic and changing requirements in personalized and scalable IoT, optimization techniques are important to find the best applications for sensing resources allocation. In such optimization, proper selection of constraints assures that all applications and functional requirements are satisfied without exceeding physical resources capabilities. Afterward, the generated assignments are used to form the needed virtual SNs.

As an example in smart city context, the transportation agency needs to run App1 to collect sensed data from smart vehicles, traffic lights, city bridges, and road status and analyze it to help in reducing traffic jams and collisions. On the other hand, the city government needs to run App2 for environment monitoring that collects temperature, humidity, freezing rain, and air quality sensor readings and communicate them with necessary entities to alarm inhabitants. Here, App1 needs sensing nodes with camera recording, traffic lights status, GPS, and accelerometer functionalities. Also, it requires small latency and hard deadlines needed for real-time operation. On the other hand, App2 requires sensing nodes with temperature, humidity, freezing rain, and air quality functionalities. App2 is latency tolerant as readings could be aggregated from multiple sensing nodes and partially predicted but normally requires large scale coverage. The requirements of both applications may change at any time. For example, in case of high traffic jams at some parts of the city, App1 needs more sensed data collected from the video recording cameras, traffic lights, and city's bridges in the affected areas. Similarly, App2 may need more accurate sensed data or larger coverage, in case of sudden severe weather conditions. In both cases, these changes are translated into an updated set of functional requirements that need to be allocated optimally and dynamically to the available sensing resources in the region.

An example of linear optimization that maximizes the overall SN utilization (SenNU) while satisfying all applications requirements is presented in the following equation [42]:

$$\begin{aligned} &\max_{s} \text{SenNU}(s) = \sum_{i=1}^{A} a_i s_i \\ &\text{s.t.} \sum_{v \in fid(P_{i,cp})} R_{v,r,i} \geq 1 \\ &\forall a_i \in A \sum_{n \in N} R_{n,r,i} \geq m_i \\ &\forall n \in N \sum_{i=1}^{A} d_i \left(\sum_{\mu=1}^{\sigma_n} (c_{i,\mu} \times R_{n,\mu,i}) \right) \leq e_n \end{aligned} \quad (3)$$

where a_i represents application i, s_i represents application i serviceability, $P_{i,cp}$ is a functional requirement decomposed from application i, $fid(P_{i,cp})$ is the set of all nodes that could satisfy functional requirement $P_{i,cp}$, $m_i \in M$ is the set of minimum resources to sustain application i, $d_i \in D$ represents the duration of an application i, and en is the remaining energy in node n. σ_n is the number of resources in node n, and $c_{i,u}$ is the power consumption when application i uses resource u.

In this optimization problem, the set of functional requirements for every application F_s are assumed to change over time and reassignment happens with every change detected. This suits the nature of personalized and scalable IoT networks. Nevertheless, a quantitative evaluation study is needed to show the satisfaction of all functional requirements against fast change in user requirements. Moreover, the strict nature of the constraints results in declining many applications/needed changes in case of lack of resources that fully satisfy functional requirements. On the other hand, utilizing transient resources was explored and formulated in [43] with dynamic optimal mapping that maximizes the number of functional requirements satisfied and use the transient resources during its sojourn time with cost and utility function.

IV. FOG LAYER

The main idea of Fog computing is to bring computing resources closer to the edge [21], thus bridging an important gap between the cloud and the users. This makes it a valuable tool for scalability, as it leverages distributed computing on a large scale. It is also a platform with great potential for resource sharing, allowing many services to coexist. In addition, it is an important solution for scalable real-time support, as it can respond to user queries with significantly lower latency than the centralized cloud. In this section, we first discuss the potential of Fog computing in personalized IoT applications, then we focus on the challenges in migrating personalized services to the Fog.

A. Potential of Fog Computing in Personalized IoT

One of the most powerful features of Fog computing is that it provides local context awareness for users and allows the coupling of computing resources with IoT components, which leverages a multitude of scalable services [44]. Fog computing also enables latency-sensitive and resource-intensive services such as real-time analytics, image processing, computer vision, or heavy encryption techniques. This makes Fog computing a promising enabler for personalized services and also augments the resources of the constrained sensing layer.

Much of the research efforts in the area of Fog computing have been application specific, with healthcare and connected vehicles being the leading domains. This is because applications in these two domains typically require strict latency guarantees. In healthcare, examples include the architecture in [45], which targets patients with chronic obstructive pulmonary disease. The proposed architecture utilizes a sensing layer that measures the necessary attributes within the user's living environment, as well as their vital signs, and a Fog layer that is responsible for the collection and real-time analysis of this raw data. In connected vehicles, several models have been investigated in [46], either based on roadside units or direct connections between vehicles. The authors also suggested using vehicles as Fog nodes, including moving or parked vehicles. The work showed a significant reduction in latency using the proposed models.

Fog computing has also found applications in the area of tactile Internet (TI) [47], where interactions and exchanges are allowed across the globe. Advances in haptic technologies and AI are now allowing applications that could not have been foreseen before. These include remote surgery, industrial control and management, or gaming. However, these applications require near real-time response, which can be challenging and varying across the Internet. In addition, many of these applications require image processing or computer vision, which are computationally intensive. Accordingly, Fog computing has been proposed to address some of these challenges, by offloading some of the processing tasks to the Fog. The work in [48] discusses these issues in Robotics. Virtualized SNs can also play an important role in TI, by offering sensing services on demand that can support context awareness (for example through crowd sensing). This can have a big impact on the immersiveness and the overall quality of the experience of the TI application.

There have also been Fog computing systems that do not target a specific application but target a particular challenge in Fog computing. For example, the work in [49] leverages fast deployment of IoT applications based on Fog computing. An application management module ensures coordination between these processes. Other architectures focus on managing the functions of a Fog computing system in general. For example, the architecture in [50] specifies modules for task scheduling and distribution, as well as offloading and management between the Fog and the cloud.

B. Migrating Personalized Applications to the Fog

From the discussion in Section IV-A, the case for Fog computing to personalize IoT services is clear, even if Internet-based resources will forever remain superior. As the number of IoT devices continues to grow, especially wearables, the case for deploying Fog computing services will become more pressing. Nevertheless, there are challenges associated with migrating IoT services to the Fog. One of the key challenges is heterogeneity, which is a core challenge in virtually all IoT components. The diversity of technologies and applications poses significant hurdles to the deployment of generic Fog computing services. In this context, defining requirements and constraints become very difficult. Here, semantic technologies have been pursued as a candidate solution [21], due to their ability

to standardize application requirements and resource capabilities, making it easy to match these two together in a user-centric approach.

Another key challenge is related to the spatial constraints of Fog computing [22]. Fog nodes are limited in coverage, which means that a network of nodes will be needed to cover larger areas. This introduces interoperability challenges and coordination difficulties, especially in the presence of duty cycling, which limits the presence of some devices [22]. In personalized IoT scenarios, where users expect seamless services, this may be undesirable. Solutions to this challenge generally fall under the umbrella of resource management and task scheduling. Promising solutions anticipate user mobility and initialize migration and handover procedures ahead of time [51]. Coordination between Fog nodes has been addressed in [52], where the authors study how resources can be clustered to enable resource sharing while considering latency.

Task scheduling also has an impact on seamless operation. If not managed appropriately, users may encounter service blocking at handovers. Koponen *et al.* [53] compare between different approaches to assigning tasks to Fog nodes. The first approach simply chooses a Fog node at random, the second chooses the node that will lead to the minimum latency, while the third chooses the node with maximum available resources. The performance evaluations show that the second approach achieves lowest blocking probability.

Thus, we can say that service placement, in general, is a challenge in personalized applications. On the one hand, placing services directly on sensor networks is likely going to have minimum latency. However, duty cycling and resource constraints over these networks imply a possible lack of continuous presence. On the other hand, placing services on the cloud means that they can be always present, but latency may be high. The Fog is somewhere in between, however, interoperability and coordination challenges may hinder large scale deployment.

The localized nature of Fog computing services also means that these services remain largely memoryless, unless user data can be transferred to a new Fog node upon migration. As mentioned before, there are heterogeneity challenges against this issue. The memoryless operation challenge is more amplified when decisions need to be made that may affect users in other coverage zones, which may lead to conflicts and reduced efficiency. This remains an open challenge that has received limited attention from the research community.

Since Fog computing is a key enabling technology for localized context-aware services, many research efforts have focused on a metric known as Quality of Context (QoC) to evaluate their services. QoC management encompasses all aspects of contextual information, including acquisition, modeling, exchange, and evaluation. Thus, QoC-centric solutions for Fog computing often address the varying nature of context information in their proposed mechanisms. The work in [54] presents such a QoC-centric solution, where the authors propose indicators to measure the quality of information with respect to parameters such as precision and freshness, and propose mechanisms for aggregation, fusion, and self-configuration that operate based on these indicators. These QoC metrics are of high importance in smart city scenarios where scalability is important, as mobility imposes a high degree of variability in context information.

V. CORE NETWORKING LAYER: INFORMATION

Data collected from the sensing layer typically needs to be delivered directly to an end user or to a storage/processing location. The latter may either be a Fog or cloud layer. In this section, we study content delivery based on ICNs for these scenarios. ICN is a promising networking paradigm that tailors nicely to IoT requirements due to its ability to support event-based communications using the publish/subscribe pattern. We examine the readiness of ICNs to support personalization in large scale IoT networks.

ICNs were conceived to address the gradual shift in Internet communications that have taken place over the years from host-centric to information-centric communications. Ultimately, it is clear that people generally care about the information they get rather than where it came from. Thus, ICNs decouple resources from their hosts by introducing two key concepts: content naming and in-network caching [8]. Using these two concepts, content can be identified and requested without being associated with a static server-related URL and can be stored and retrieved from anywhere in the network (including cache-equipped routers). This has significant potential to reduce delay and communication overhead.

These features make ICNs attractive for IoT networks. First, decoupling resources from locations provides a boost to scalability. This is because the number of information sources becomes of less relevance, but rather the amount of information they produce. Second, in-network caching is important for latency-sensitive applications, by strategically choosing caching locations close to information destinations. Furthermore, ICNs leverage event-based communication patterns such as publish/subscribe, which is of great relevance to IoT applications and can significantly reduce overhead. Publish/subscribe also leverages personalization, as it allows users to specify exactly which information should be pushed to them. It also leverages context-awareness, by allowing applications to subscribe without human intervention to information sources that are relevant at any given time to any application.

To illustrate, consider for example a driver using a traffic management app in their vehicle. This driver would like real-time information about road conditions or traffic before and during a trip. Using ICNs, the app can subscribe to all relevant information sources. These may include weather updates and traffic sensors over a particular route. Accordingly, the network would forward relevant information to the app as they become available. Predictive and

intelligent caching can also optimize where the data are stored by estimating the user's mobility. Thus, data can be available at cache stores even before the user needs them, thereby significantly reducing latency.

This section discusses how ICNs addresses important IoT challenges such as mobility, scalability, and dynamic user requirements. Focus is given on user-centric caching solutions as the prominent tool in ICNs for addressing these challenges. It is important to note that this section assumes the reader has a basic familiarity with the concepts of ICN. For a review of these concepts, the reader can consult [8] and [55]–[60].

A. User-Centric Caching in ICNs

On-path caching is one of the core features of ICNs. It can reduce latency and overhead in content retrieval and lead to reductions in traffic congestion. For this reason, caching in ICNs is a highly active research area, and there are several excellent survey papers on the topic [61]–[63]. Thus, in this section, we will not duplicate their efforts but rather focus on the challenges pertaining to personalized IoT networks, which is a topic that has received much less attention from the research community. User-centric caching is about how techniques serve individual user interests. Thus, we discuss the readiness of existing caching metrics to support user-related challenges such as mobility and dynamic content requests, and how these challenges can be addressed on a large scale.

In a large scale distributed IoT networks, supporting user requirements in the presence of mobility can be quite challenging. Traditional networking involves host-to-host connections. In networks where information sources utilize duty cycling, this may not always be possible. ICNs address this issue by decoupling information sources from user requests. Thus, sensors can publish their data without waiting for requests, and users can be on the move and request data from the network without worrying about the state of the sensors. The challenge here is to design caching mechanisms that optimize information storage according to mobility patterns.

In addition, supporting dynamic user requests and scalability involves content management at cache stores. Challenges such as staleness detection, duplication and redundancy detection, and popularity. First, staleness detection is to identify already cached content that is unlikely to be requested in the future. In IoT networks, it is likely that content may have a short lifetime (for example, temperature readings may be valuable only for a short time), and accordingly may become stale faster. Fortunately, expired IoT content is not difficult to detect. For example, Katsaros *et al.* [64] suggest that data/application instructions can also be cached, which can be used to identify the lifetime of each item. The second challenge is to detect duplicate or redundant content. Duplicate content is storing the same object more than once, while redundant content is storing several objects that are equivalent in one way or the other (for example, storing several temperature readings from sensors within close geographical proximity). These challenges may be easy to address within one cache store, but in ICNs, there is a network of caches that collectively contribute to the overall network performance.

The third challenge, popularity, has to do with identifying which content to store. Ideally, we want to store content that is likely to be requested in the future. This is traditionally done in Internet caches using a metric such as cache hit ratio, which identifies the number of requests that an object receives. However, in personalized IoT networks, short content lifetime implies that hit ratios may never be high enough to be used as a reliable popularity metric. To address this issue, Vural *et al.* [65] proposed data item lifetime and freshness as alternative metrics to cache content.

Another challenge has to do with the nature of traffic in IoT networks. Generally, ICN caching techniques are designed for the traditional pull-based traffic, where objects have to be explicitly requested before they are delivered. In IoT networks, push-based traffic is expected to be more dominant [63]. Here, content may be pushed to users without receiving any requests. Thus, caching techniques that count requests to determine popularity may not be suitable.

1) Qualitative Evaluation of Caching Techniques for Personalized IoT Requirements: In this section, we evaluate qualitatively the readiness of some prominent caching strategies to support personalization in the IoT. We consider caching techniques that were proposed specifically for the IoT or for ICNs in general. Earlier, in this section, we have identified certain challenges as important for personalized IoT applications: content management, mobility, and traffic characteristics.

The work in [62] has provided a taxonomy for caching solutions in non-IoT ICNs, categorized into: probabilistic, graph-based, label-based, and popularity-based caching. These solutions all relate to content management. Thus, we shall add two more categories to this taxonomy: IoT-specific and mobility-supporting caching solutions. In each of these categories, we choose prominent solutions and discuss their design and potential for personalized IoT applications.

Probabilistic caching techniques store content based on fixed or dynamic probabilities. These mechanisms can diversify content, which may be advantageous in applications where traffic demand is unpredictable. An example of a dynamic probability mechanism is ProbCache [66], which considers the routing path and stores content close to the user. According to ProbCache, each router calculates the probability to store content according to the following equations:

$$\text{ProbCache} = \text{TimesIn} \times \text{CacheWeight} \tag{4}$$

$$\text{TimesIn} = \sum_{n=1}^{x-y+1} \frac{N_n}{T_{tw} \times N_x} \tag{5}$$

$$\text{CacheWeight} = \frac{y}{x} \tag{6}$$

where TimesIn is a factor that gives priority to content that came from far away, and CacheWeight is a counter that favors storing content close to the subscriber. The variable x is the length of the path between publisher and subscriber, and y is the number of hops between the publisher and the current router (where ProbCache is being calculated). Finally, T_{tw} is a timer for how long content should be stored, N_n is the cache size of the current router, while N_x is the average cache size of the path between the publisher and the subscriber. ProbCache may result in lower delay and is potentially energy efficient as it results in fewer hops to the subscribers.

Nevertheless, probabilistic techniques do not address the problem of stale content. This is addressed in hop-based probabilistic caching (HPC) [67], where caching content depends on a probabilistic metric that includes the distance to the destination as well as a time interval. The following equation is used to calculate the probability metric in HPC:

$$\text{HPC} = \text{CacheWeight}_y \times \text{CacheWeight}_{\text{MRT}} \tag{7}$$

$$\text{CacheWeight}_y = \frac{1}{y+\alpha},\ a \geq 0 \tag{8}$$

$$\text{CacheWeight}_{\text{MRT}} = \begin{cases} \dfrac{\text{MRT}_m}{\text{MRT}_{\text{exp}}}, & \text{MRT}_m < \text{MRT}_{\text{exp}} \\ 1, & \text{else} \end{cases} \tag{9}$$

where $\text{CacheWeight}_{\text{MRT}}$ stores content for a specific period of time, while CacheWeight_y stores content closer to the subscriber. Here, y is the node that satisfied the content request, α is determined according to the network's cache capacity, and MRT_m and MRT_{exp} are the meantime and the expected mean time that contents spend on a node, respectively. ProbCache and HPC are not scalable as they only consider caching content close to the user.

Scalability is better addressed by graph-based caching, which considers network topology in caching decisions. A family of graph-based caching policies employs centrality metrics [62], where a cache is chosen based on a metric that reflects its position in the network. For example, the degree [68] centrality metric reflects how many connections a node has, which has an implication on the ability to distribute content. Another known centrality metric is closeness [68], which reflects how close a node is to others in the network.

These policies have the advantage of considering the network topology and not performing caching decisions independently at each node. However, graph-based policies are vulnerable to duplication and redundancy, as caching decisions do not consider content. This is addressed in label-based caching [69], where nodes are assigned topics that they can cache. This creates a diversity of content in the network, which may be useful when the traffic demand is not known.

One of the most important caching metrics for personalization in the IoT is popularity, where storing an object depends on how many times it was requested. Typically, techniques define a threshold to determine when an object is considered popular. In this category, techniques mainly differ from one another by how popularity is determined. In [70], popularity is calculated independently at every node using a counter, and a static threshold is defined to cache content. The equation for calculating popularity is shown in the following equation:

$$\text{Popularity} = \frac{1}{1+e^{\frac{p-R_c}{q}}} \tag{10}$$

where R_c is the popularity counter, and p and q are the parameters that define the logistics of when content should be cached.

In IoT applications, having a static threshold may not be suitable for short-lived data. This is addressed in [71], where the threshold is recalculated every period of time and is only used to determine how many objects will be cached out of all received ones. Another dynamic popularity caching mechanism is proposed in [72], where content is divided into categories based on a popularity metric. The proposed metric allows for defining types of content that are either always cached or never cached, which provides more filtering capabilities for certain content types.

None of the aforementioned caching solutions were proposed specifically for the IoT. In this case, the traffic characteristics and the content itself should be considered. For example, the coherent caching technique proposed in [73] considers the validity of content in caching decisions. Thus, cached content is refreshed every time new content is received. This ensures freshness but runs a risk of limited content diversity, which leads to low cache hit rates. Another IoT-based strategy is proposed in [74], which stores objects if they belong to the same topic. This has the advantage of providing resource assurances to specific services, but it may result in fairness problems between users requesting different services.

Finally, caching can also be used to address mobility challenges in the IoT. The survey in [75] has identified three categories of mobility in ICNs: subscriber mobility, publisher mobility, and subscriber/publisher mobility. Subscriber mobility is relatively easier to address, as subscribers may simply need to retransmit their requests after a timeout interval. The main challenge is to determine the timeout interval in a dynamic environment. Solutions such as proactive caching have been proposed in [76] to reduce delay. However, these solutions mainly work under a sufficient degree of predictable mobility (a vehicle moving on a road, for example). Other solutions have proposed using anchor points to support subscriber mobility [77], or using multiple routing paths to increase the probability of delivery [78].

On the other hand, publisher mobility is more challenging, especially in ICN architectures that utilize a dedicated

name resolution service. This is because, as publishers move, these databases have to be updated accordingly, possibly on a global scale. Several solutions have been proposed to this problem. Constrained flooding was proposed in [78], where the publisher floods a defined neighborhood after a handoff is detected. Another solution was proposed in [79], where publishers advertise their new location in the first packet after a handoff. However, the most prominent solutions mimic the approach utilized in Mobile IP [78], [80]. Here, an anchor point is used, either to support mobility aspects in the name resolution service [78] or create a tunnel to redirect the packets to the correct destination [80].

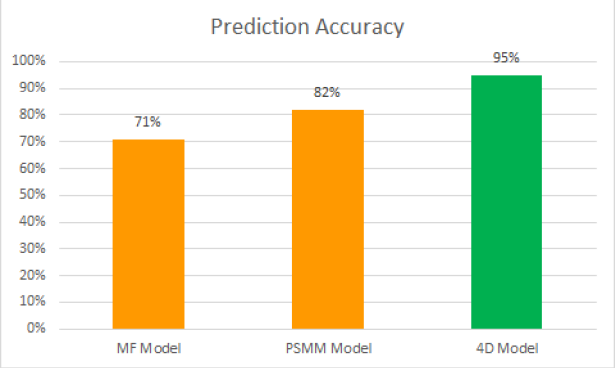

Fig. 3. ***Prediction accuracy of the proposed 4-D mobility model.***

VI. INTELLIGENCE LAYER

In the previous sections, we have discussed the most prominent research efforts in developing an adaptive and scalable IoT infrastructure. In order for this network to be personalized, there has to be tight coupling between services and all sensing and network components in a user-centric environment. In the era of big sensed data, this may be quite challenging due to the volume or dynamism of this data. Developing personalized decision-making systems does not only depend on the availability of situational data but also on the ability to extract meaningful information out of this data and reason for this information.

The fourth layer of the architecture shown in Fig. 1 addresses these challenges. Here, we discuss how AI and context awareness can lead to intelligent systems. We particularly investigate two types of AI techniques: reasoning and machine learning. Reasoning can determine decisions in the presence of conflicting constraints, while machine learning can discover patterns and establish a knowledge base to support decision-making. It is of no surprise that AI has found many applications in the IoT domain. Recently, there has been increasing interest in deploying AI agents close to the edge, in order to improve the utilization of the raw data and provide higher levels of context awareness to the users. In addition, deploying AI can lead to proactive and predictive network operations, by estimating future user states and requirements ahead of time. This can lead to efficient resource provisioning and user-centric networking on a large scale.

Fortunately, there exists now a wide variety of AI tools to address these challenges [9]. In this section, we demonstrate different ways in which AI can enable personalized IoT systems. It is difficult to provide a comprehensive survey of all ways AI can be used in IoT. Rather, we select areas where AI can play a prominent and irreplaceable role.

One of the most valuable applications of AI in IoT networks is mobility prediction. If performed accurately, mobility prediction can support many network functions such as predictive resource management, personalized caching, adaptive duty cycling, and proactive VM migration in Fog computing. For these reasons, mobility prediction has been the focus of intense research. For example, Calabrese *et al.* [81] proposed a model that predicts locations based on previous preferences. On the other hand, the work in [82] explored the impact of social networking on mobility. The impact of location semantics was also explored in [83]. Nevertheless, accurate mobility prediction remains illusive, due to the unpredictable nature of human mobility, and the fact that it is affected simultaneously by many factors such as time, social network, and personal preference. Existing research efforts typically focus on a small subset of these factors, leading to inaccurate predictions. Accordingly, we presented a comprehensive mobility model in [84], which considers four dimensions of human mobility: social network, time, preference, and location semantics. In [84], we studied the impact of each of these dimensions on the mobility of each user, using a data set that we collected in the city of Ulm, Germany. Based on our findings, we proposed an optimized 4-D mobility model that determines the weights that should be given to each of these factors for every user. In addition, we use supervised learning to tune the model to unpredictable deviations that a user may have from their patterns. Fig. 3 shows that our model is able to achieve a prediction accuracy up to 95%, making it superior to its counterparts. Such an accurate model can have limitless applications in personalized IoT networks.

Predicting user mobility can also be quite valuable in data collection, especially in the scenarios of opportunistic or public sensing. Here, users directly collect data form the sensors, by leveraging the ubiquitous connectivity of smartphones, either for direct usage or for relaying to the cloud. The opportunistic nature of these connections can cause significant energy losses if user mobility is not considered, since sensors may be waking up needlessly without there being a transmission opportunity. On the other hand, sensors extending their duty cycles for too long to save energy may end up missing opportunistic connections, thereby reducing service quality. For personalized applications, users would ideally want the sensors to be awake at the precise time when they need data. We addressed this tradeoff in [85], where we proposed a data collection protocol for public SNs. Here, fuzzy

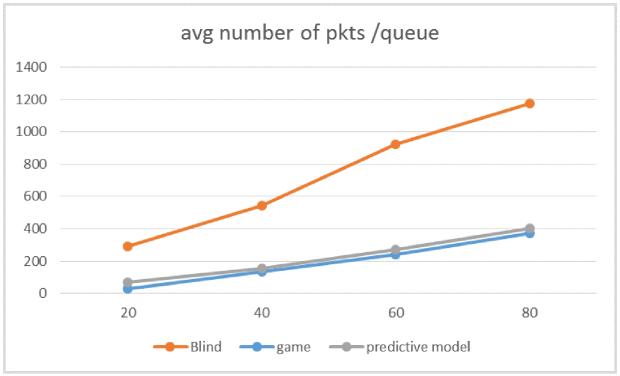

Fig. 4. ***Average size of queues in the predictive data collection protocol.***

logic was used to estimate the existence of opportunistic connections based on the history of connections seen by a sensor within a particular time window. We also used game theory to determine the optimum set of sensors to transmit in any given time slot, in order to prevent redundant sensor transmissions when their data is correlated. Thus, if a smartphone comes in contact with several sensors within close proximity, only a small subset of these sensors will need to transmit. This saves energy without compromising service quality.

Our approach was evaluated using computer simulations, which showed excellent energy savings and low latency in data collection, as can be seen in Figs. 4 and 5. Fig. 4 shows that the average queue length in smartphones (containing the collected packets from sensors to be relayed to the cloud) is significantly reduced with the proposed model (which uses the combined game theory and fuzzy logic techniques) compared to the blind scenario (without any technique to predict user mobility). This means that latency is reduced as packets have to wait less time in the queue. On the other hand, Fig. 5 shows that the predictive model based on fuzzy logic results in significant energy savings, compared to the blind scenario and the scenario where game theory is used without prediction.

A promising direction in personalization has been to employ cognition in IoT [4]. This Cognitive IoT is built on concepts such as perception-action cycle, memory, intelligence, and language. Thus, building this system involves a variety of AI tools. An example of such a system is the cognitive smart home platform proposed in [86]. This platform implements two kinds of memory. A perceptual memory is implemented using a Bayesian model to ensure accurate understanding of the observed sensory input. This perceptual memory provides the necessary feedback to an executive memory that is built using reinforcement learning in order to keep track of the outcomes of the system actions. In addition, a multilayer decision support system provides strategic resource allocation based on the current context. The system applies two kinds of cognitive actions: those applied to the environment (closing or opening shutters, for example) and those applied to the system itself (configuring the sleep cycle of sensors/actuators).

The concept of cognition has also been applied to improve system performance through increased environmental awareness and system adaptability. For example, the work in [87] proposes a cognitive framework for wireless networks titled app-aware weighted cognitive map (WCM) that balances tradeoffs between conflicting user requirements. The proposed framework manages the available network resources while considering dynamic user requirements and environment variables. It consists of two key components: a reasoning engine and a learning module. The reasoning engine is built using WCM [87], which models the interaction between controllable system components and environmental variables. In addition, the learning module uses reinforcement learning to build a knowledge base of system states. Using this module, the system can identify the most rewarding actions in each state.

The performance of this system was evaluated in [87] using computer simulations. These simulations modeled a network where users are running dynamic and resource-hungry applications. Each application defines a set of metrics to be supported by the network. The simulations compared the performance of the WCM framework against other adaptive networking schemes, namely QoS-Aware Routing and Admission Control, Adaptive Admission Control, and the transmit power and data rate control protocol known as Symphony [87]. To measure the ability of these systems to support different user requirements, the simulations calculated the percentage of service disruption for each protocol, which is measured as the percentage of time the application requirements were not satisfied. This was measured individually for throughput, latency, and reliability, against varying number of nodes. The results are illustrated in Fig. 6 and clearly show the superiority of the cognitive system over its counterparts. This is owed to its ability to balance between conflicting actions and application requirements,

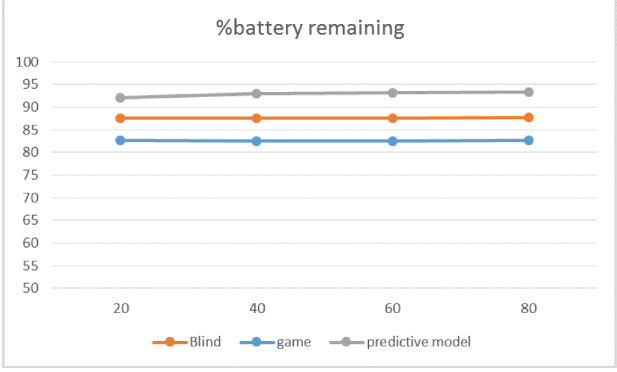

Fig. 5. ***Average remaining battery in the predictive data collection protocol.***

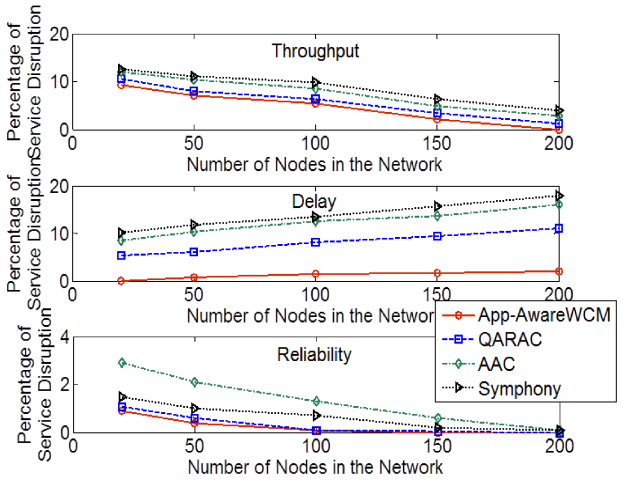

Fig. 6. ***Performance evaluation of the cognitive wireless network framework.***

while learning which actions lead to better overall system performance.

VII. OPEN PROBLEMS

The vision of a ubiquitous and personalized smart city is yet to be realized. In previous sections, we have discussed several key technologies for addressing some of the challenges in this vision. However, several open problems still require investigation. This section highlights some of the key problems.

A. Decoupling of Resources and Applications

One of the biggest challenges for the realization of a scalable IoT is going to be hardware reuse. Currently, deploying IoT systems on any scale typically incurs infrastructure costs. If hardware reuse is allowed on a large scale (for example, using a sensing-as-a-service model), this will support innovation on an unprecedented scale. Due to the typically limited resources of IoT devices, they generally remain application specific. This decoupling between the hardware and what can run on it is still a topic of intense research, and likely to remain so in the near future.

In addition, discovering, customizing, and sharing these resources are another open problem. AI techniques will play an important role is probing IoT resources to find opportunities for composite services that are suitable for the changing context of the user. It will then be a big challenge to determine how these resources can be simultaneously customized and utilized by multiple users simultaneously. Semantic technologies are likely to be a great enabler for supporting such a shared platform, particularly in discovering and probing the resources.

If one day this decoupling exists, there is going to be a need for cost and revenue sharing models. This means that the system has to somehow track the resources utilized by each user and the manner in which these resources were used. This can be an important application for blockchains. Their distributed nature allows for distributed and shared operation that may even be multitenant. For example, utilizing a resource can be implemented as a smart contract on a blockchain that is fired up upon request, and this transaction can include payment to the owner(s) of the hardware, depending on its use.

B. Distributed Coordination and Management

Scalable IoT applications will be quintessentially distributed and likely multitenant. Within this distributed environment, users expect seamless services that are context aware and personalized. This introduced many challenges as the users move across the boundaries of a single management domain. Here, the networks have to coordinate how service provisioning is going to continue. This is a well-established topic in wireless connectivity, but it requires much investigation in the context of the IoT. For one, the user may be requiring sensing/actuating services that are only available in a different network. Thus, this data must be made available across network boundaries. Fog computing can address this issue in a distributed and scalable way, but resource management that limits blocking probability on handovers is required.

Heterogeneity can also be a limiting factor in distributed management. Incompatible technologies might cause service disruptions when moving across networks. In addition, the spectrum of device capabilities in the IoT is quite large: some devices are quite limited while others have good processing, energy, and memory capabilities. Thus, moving across network boundaries can encounter limited resources that are inadequate for application requirements.

Moreover, it may be difficult to support specific context-awareness features across network boundaries. This may be due to the different capabilities of networks to measure or identify context information. Even though this may not lead to service disruptions, it may lead to varying service quality. We are all familiar with the well-known bar scale on our WiFi or mobile connections that reflect the quality of the received signal. In the IoT, there may be a need for a similar bar scale that reflects the service quality, since smart applications are the primary objective of the IoT.

C. Security and Privacy

Security and privacy challenges in the IoT are well known and have been covered in previous survey papers [88]. However, there are some challenges that are particularly related to the personalization aspects of IoT networks.

One of the key challenges is security and privacy across network boundaries. Heterogeneity in the IoT may imply that some networks may utilize devices that have limited security capabilities, and other networks may not have

strict considerations for the privacy of the user. In personalized networks, the user should have the ability to choose appropriate security and privacy settings. These may not be achievable across network boundaries. Thus, there is a need to provide awareness for the user as to how their privacy may be violated as they move from one network to the other. Also, automated and context-aware services may interact with the network autonomously on behalf of the user. In all these interactions, the user must be made aware of the implications and given the choice to terminate any suspicious activity.

D. Memory and Retrospective Operation

As mentioned before, scalable IoT networks are often memoryless, where they forget about the user after the service is finished. In addition, IoT devices may have limited memory capabilities, preventing them from storing significant amounts of data about the user. This may limit the level of personalization and context awareness that can be achieved with distributed IoT networks.

One solution is to allow the devices to store a limited and sparse data set about each user (for example samples of mobility data) and then use machine learning to interpolate the missing data upon occurrence of a related event. This retrospective operation allows the network to estimate historical data without the need for saving it. If this process is accurate enough, it may lead to significant memory savings. This idea can also be used to limit the amount of data transferred from sensors to any device, thus also leading to higher energy efficiency. Nevertheless, retrospective operation requires the coordination between nodes to determine the optimal time and place to store the data.

VIII. CONCLUSION

In this paper, a comprehensive investigation of scalable personalized IoT networks has been presented. We discussed the main challenges to achieve scalability and personalization, and identified four main layers where these challenges may be addressed: sensing, fog, core networking, and intelligence. In each of these layers, we offered insights and analysis of the most prominent solutions in research.

In the sensing layer, we discussed how virtualization and programmability can address a very important requirement in scalable IoT networks, which is hardware reuse. We showed how composite services and personalized applications can be leveraged by virtualizing different components of the sensing layer, from individual devices to entire networks. In the Fog layer, we highlighted the main challenges that need to be addressed in order to migrate services to the Fog. These challenges include heterogeneity, resource management, and task scheduling. Several solutions to these challenges were discussed, including proactive solutions that have great potential in ensuring high-quality context awareness services.

We also studied the suitability of ICNs in scalable personalized IoT networks. Here, proactive caching coupled with event-based publish/subscribe can support many IoT requirements such as mobility, dynamic requirements, and traffic congestion. Finally, in the intelligence layer, we showed how AI and context awareness techniques can lead to significant performance improvements across several layers. Mobility prediction, proactive duty cycling, and cognitive networking are among the few examples that demonstrate how the intelligence layer has become indispensable in IoT networks. ■

REFERENCES

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[2] M. Ibnkahla, *Wireless Sensor Networks: A Cognitive Perspective*. Boca Raton, FL, USA: CRC Press, 2016.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[4] A. El-Mougy, M. Ibnkahla, G. Hattab, and W. Ejaz, "Reconfigurable wireless networks," *Proc. IEEE*, vol. 103, no. 7, pp. 1125–1158, Jul. 2015.

[5] S. Chen *et al.*, "Butler, not servant: A human-centric smart home energy management system," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 27–33, Feb. 2017.

[6] D. Casado-Mansilla *et al.*, "A human-centric & context-aware IoT framework for enhancing energy efficiency in buildings of public use," *IEEE Access*, vol. 6, pp. 31444–31456, 2018.

[7] M. Amadeo *et al.*, "Information-centric networking for the Internet of Things: Challenges and opportunities," *IEEE Netw.*, vol. 30, no. 2, pp. 92–100, Mar./Apr. 2016.

[8] G. Xylomenos *et al.*, "A survey of information-centric networking research," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024–1049, 5th Quart., 2014.

[9] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "Context-aware computing, learning, and big data in Internet of Things: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 1–27, Feb. 2018.

[10] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[11] C. Perera, C. H. Liu, and S. Jayawardena, "The emerging Internet of Things marketplace from an industrial perspective: A survey," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 4, pp. 585–598, Dec. 2015.

[12] W. Sun, Z. Yang, X. Zhang, and Y. Liu, "Energy-efficient neighbor discovery in mobile ad hoc and wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1448–1459, 3rd Quart., 2014.

[13] V. P. Kafle, Y. Fukushima, P. Martinez-Julia, and H. Harai, "Scalable directory service for IoT applications," *IEEE Commun. Standards Mag.*, vol. 1, no. 3, pp. 58–65, Sep. 2017.

[14] S. Cirani *et al.*, "A scalable and self-configuring architecture for service discovery in the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 508–521, Oct. 2014.

[15] S. Pattar, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Searching for the IoT resources: Fundamentals, requirements, comprehensive review, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2101–2132, 3rd Quart., 2018.

[16] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the Internet of Things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[17] A. I. Sulyman, S. M. A. Oteafy, and H. S. Hassanein, "Expanding the cellular-IoT umbrella: An architectural approach," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 66–71, Jun. 2017.

[18] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, "IOT gateway: Bridging wireless sensor networks into Internet of Things," in *Proc. IEEE/IFIP Int. Conf. Embedded Ubiquitous Comput.*, Dec. 2010, pp. 347–352.

[19] D. Kreutz, F. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[20] Z.-J. Han and W. Ren, "A novel wireless sensor networks structure based on the SDN," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 3, pp. 1–7, 2014.

[21] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, 1st Quart., 2018.

[22] S. M. Oteafy and H. S. Hassanein, "IoT in the fog: A roadmap for data-centric IoT development," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 157–163, Mar. 2018.

[23] I. Leontiadis, C. Efstratiou, C. Mascolo, and J. Crowcroft, "SenShare: Transforming sensor networks into multi-application sensing infrastructures," in *Proc. Eur. Conf. Wireless Sensor Netw.* Springer, 2012, pp. 65–81.

[24] P. Levis and D. Culler, "Maté: A tiny virtual machine for sensor networks," *ACM SIGPLAN Notices*, vol. 37, no. 10, pp. 85–95, 2002.

[25] J. Koshy and R. Pandey, "Vmstar: Synthesizing scalable runtime environments for sensor networks," in *Proc 3rd Int. Conf. Embedded Netw. Sensor Syst.*, 2005, pp. 243–254.

[26] D. Simon, C. Cifuentes, D. Cleal, J. Daniels, and D. White, "Java on the bare metal of wireless sensor devices: The squawk Java virtual machine," in *Proc. 2nd Int. Conf. Virtual Execution Environ.*, 2006, pp. 78–88.

[27] C. Mouradian, T. Saha, J. Sahoo, R. Glitho, M. Morrow, and P. Polakos, "NFV based gateways for virtualized wireless sensor networks: A case study," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 1883–1888.

[28] C. Mouradian *et al.*, "Network functions virtualization architecture for gateways for virtualized wireless sensor and actuator networks," *IEEE Netw.*, vol. 30, no. 3, pp. 72–80, May/Jun. 2016.

[29] M. Ojo, D. Adami, and S. Giordano, "A SDN-IoT architecture with NFV implementation," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2016, pp. 1–6.

[30] A. B. G. Hernando, A. D. S. Fariña, L. B. Triana, F. J. R. Piñar, and D. F. Cambronero, "Virtualization of residential IoT functionality by using NFV and SDN," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2017, pp. 86–87.

[31] R. Morabito, R. Petrolo, V. Loscri, and N. Mitton, "LEGIoT: A lightweight edge gateway for the Internet of Things," *Future Gener. Comput. Syst.*, vol. 81, pp. 1–15, Apr. 2018.

[32] I. Khan, F. Belqasmi, R. Glitho, and N. Crespi, "A multi-layer architecture for wireless sensor network virtualization," in *Proc. 6th Joint IFIP Wireless Mobile Netw. Conf. (WMNC)*, Apr. 2013, pp. 1–4.

[33] J. Hoebeke, E. De Poorter, S. Bouckaert, I. Moerman, and P. Demeester, "Managed ecosystems of networked objects," *Wireless Pers. Commun.*, vol. 58, no. 1, pp. 125–143, 2011.

[34] I. Ishaq, J. Hoebeke, I. Moerman, and P. Demeester, "Internet of Things virtual networks: Bringing network virtualization to resource-constrained devices," in *Proc. Int. Conf. Green Comput. Commun. (GreenCom)*, Nov. 2012, pp. 293–300.

[35] H. M. N. Bandara, A. P. Jayasumana, and T. H. Illangasekare, "Cluster tree based self organization of virtual sensor networks," in *Proc. IEEE GLOBECOM Workshops*, Nov. 2008, pp. 1–6.

[36] Q. Han, A. P. Jayasumana, T. Illangaskare, and T. Sakaki, "A wireless sensor network based closed-loop system for subsurface contaminant plume monitoring," in *Proc. IEEE Int. Symp. Parallel Distrib. Process.*, Apr. 2008, pp. 1–5.

[37] L. Sarakis, T. Zahariadis, H.-C. Leligou, and M. Dohler, "A framework for service provisioning in virtual sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, p. 135, 2012.

[38] M. Haghighi and D. Cliff, "Multi-agent support for multiple concurrent applications and dynamic data-gathering in wireless sensor networks," in *Proc. 7th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput.*, Jul. 2013, pp. 320–325.

[39] E. Baccelli, O. Hahm, M. Gunes, M. Wahlisch, and T. C. Schmidt, "RIOT OS: Towards an OS for the Internet of Things," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2013, pp. 79–80.

[40] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, Nov. 2004, pp. 455–462.

[41] S. M. Oteafy and H. S. Hassanein, *Dynamic Wireless Sensor Networks*. Hoboken, NJ, USA: Wiley, 2014.

[42] S. M. A. Oteafy and H. S. Hassanein, "Re-usable resources in wireless sensor networks: A linear optimization for a novel application overlay paradigm over multiple networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–5.

[43] S. M. A. Oteafy and H. S. Hassanein, "Utilizing transient resources in dynamic wireless sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2012, pp. 2124–2128.

[44] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[45] O. Fratu, C. Pena, R. Craciunescu, and S. Halunga, "Fog computing system for monitoring mild dementia and COPD patients-romanian case study," in *Proc. 12th Int. Conf. Telecommun. Mod. Satell., Cable Broadcast. Services (TELSIKS)*, Oct. 2015, pp. 123–128.

[46] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.

[47] S. M. A. Oteafy and H. S. Hassanein, "Leveraging tactile internet cognizance and operation via IoT and edge technologies," *Proc. IEEE*, to be published.

[48] M. Dohler *et al.*, "Internet of skills, where robotics meets AI, 5G and the tactile Internet," in *Proc. Eur. Conf Netw. Commun. (EuCNC)*, Jun. 2017, pp. 1–5.

[49] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the internet of things," in *Proc. 2nd ACM SIGCOMM Workshop Mobile Cloud Comput. (MCC)*, 2013, pp. 15–20.

[50] L. F. Bittencourt, M. M. Lopes, I. Petri, and O. F. Rana, "Towards virtual machine migration in fog computing," in *Proc. 10th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput. (3PGCIC)*, Nov. 2015, pp. 1–8.

[51] D. Gonçalves, K. Velasquez, M. Curado, L. Bittencourt, and E. Madeira, "Proactive virtual machine migration in fog environments," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 742–745.

[52] J. Oueis, E. C. Strinati, S. Sardellitti, and S. Barbarossa, "Small cell clustering for efficient distributed fog computing: A multi-user case," in *Proc. IEEE 82nd Veh. Technol. Conf. (VTC-Fall)*, Sep. 2015, pp. 1–5.

[53] J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in *Proc. Veh. Technol. Conf.*, May 2015, pp. 1–6.

[54] P. Marie, T. Desprats, S. Chabridon, and M. Sibilla, "Enabling self-configuration of QoC-centric fog computing entities," in *Proc. IEEE Conf. Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People, Smart World Congr.*, Jul. 2016, pp. 526–533.

[55] T. Koponen *et al.*, "A data-oriented (and beyond) network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 181–192, 2007.

[56] FP7. *PURSUIT Project*. [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/

[57] FP7. *Converg. Project*. [Online]. Available: http://www.ictconvergence.eu

[58] FP7. *SAIL Project*. [Online]. Available: http://www.sail-project.eu/

[59] NSF. *Mobility First Project*. [Online]. Available: http://mobilityfirst.winlab.rutgers.edu/

[60] NSF. *NDN Project*. [Online]. Available: http://www.named-data.net/

[61] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1473–1499, 3rd Quart., 2015.

[62] A. Ioannou and S. Weber, "A survey of caching policies and forwarding mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2847–2886, 4th Quart., 2016.

[63] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, "Caching in information-centric networking: Strategies, challenges, and future research directions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 2, pp. 1443–1474, 2nd Quart., 2018.

[64] K. V. Katsaros, W. K. Chai, N. Wang, G. Pavlou, H. Bontius, and M. Paolone, "Information-centric networking for machine-to-machine data delivery: A case study in smart grid applications," *IEEE Netw.*, vol. 28, no. 3, pp. 58–64, May/Jun. 2014.

[65] S. Vural, N. Wang, P. Navaratnam, and R. Tafazolli, "Caching transient data in Internet content routers," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 1048–1061, Apr. 2017.

[66] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proc. 2nd Ed. ICN Workshop Inf.-Centric Netw. (ICN)*, 2012, pp. 55–60.

[67] A. Ioannou and S. Weber, "Towards on-path caching alternatives in information-centric networks," in *Proc. 39th Annu. IEEE Conf. Local Comput. Netw.*, Sep. 2014, pp. 362–365.

[68] A. Anand, V. Sekar, and A. Akella, "SmartRE: An architecture for coordinated network-wide redundancy elimination," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 87–98, Aug. 2009.

[69] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *Proc. 15th Annu. Joint Conf. IEEE Comput. Soc. Netw. Next Gener.*, vol. 2, Mar. 1996, pp. 594–602.

[70] I. Psaras, W. K. Chai, and G. Pavlou, "In-network cache management and resource allocation for information-centric networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2920–2931, Nov. 2014.

[71] G. Domingues *et al.*, "Enabling opportunistic search and placement in cache networks," *Comput. Netw.*, vol. 119, pp. 17–34, Jun. 2017.

[72] T. Janaszka, D. Bursztynowski, and M. Dzida, "On popularity-based load balancing in content networks," in *Proc. 24th Int. Teletraffic Congr. (ITC)*, Sep. 2012, pp. 1–8.

[73] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, and K. Drira, "Cache coherence in machine-to-machine information centric networks," in *Proc. IEEE 40th Conf. Local Comput. Netw. (LCN)*, Oct. 2015, pp. 430–433.

[74] Y. Song, H. Ma, and L. Liu, "TCCN: Tag-assisted content centric networking for Internet of Things," in *Proc. IEEE 16th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2015, pp. 1–9.

[75] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, and F. R. Yu, "A survey of mobile information-centric networking: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2353–2371, 3rd Quart., 2018.

[76] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V. A. Siris, and G. C. Polyzos, "Caching and mobility support in a publish-subscribe Internet architecture," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 52–58, Jul. 2012.

[77] J. Lee and D. Kim, "Proxy-assisted content sharing using content centric networking (CCN) for resource-limited mobile consumer devices," *IEEE Trans. Consum. Electron.*, vol. 57, no. 2, pp. 477–483, May 2011.

[78] D.-H. Kim, J.-H. Kim, Y.-S. Kim, H.-S. Yoon, and I. Yeom, "Mobility support in content centric networks," in *Proc. 2nd Ed. Workshop Inf.-Centric Netw. (ICN)*, Aug. 2012, pp. 13–18.

[79] F. Hermans, E. Ngai, and P. Gunningberg, "Global source mobility in the content-centric networking architecture," in *Proc. 1st ACM Workshop Emerg. Name-Oriented Mobile Netw. Design-Archit., Algorithms, Appl.*, 2012, pp. 13–18.

[80] J. Lee, S. Cho, and D. Kim, "Device mobility management in content-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 12, pp. 28–34, Dec. 2012.

[81] F. Calabrese, G. Di Lorenzo, and C. Ratti, "Human mobility prediction based on individual and collective geographical preferences," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2010, pp. 312–317.

[82] M. Musolesi and C. Mascolo, "Designing mobility models based on social network theory," *ACM Mobile Comput. Commun. Rev.*, vol. 11, no. 3, pp. 59–70, Jul. 2007.

[83] H. Abdel-Fatao, J. Li, and J. Liu, "STMM: Semantic

and temporal-aware Markov chain model for mobility prediction," in *Proc. Int. Conf. Data Sci.*, vol. 9208. New York, NY, USA: Springer-Verlag, 2015, pp. 103–111.

[84] N. Basta, A. ElNahasa, H.-P. Grossman, A. El Mougy, and S. Abdennadher, "An adaptable four-dimensional destination predictor for smart vehicles," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2019.

[85] G. Younes and A. El Mougy, "Predictive resource management for opportunistic networks using game theory and fuzzy logic," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2019.

[86] S. Feng, P. Setoodeh, and S. Haykin, "Smart home: Cognitive interactive people-centric Internet of Things," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 34–39, Feb. 2017.

[87] A. El Mougy, A. Kamoun, M. Ibnkahla, S. Tazi, and K. Drira, "A context and application-aware framework for resource management in dynamic collaborative wireless M2M networks," *J. Netw. Comput. Appl.*, vol. 44, pp. 30–45, Sep. 2014.

[88] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1294–1312, 3rd Quart., 2015.

ABOUT THE AUTHORS

Amr El-Mougy (Member, IEEE) received the M.Sc. degree from Concordia University, Montreal, ON, Canada, in 2006, and the Ph.D. degree from Queen's University, Kingston, ON, in 2013.

He was a Postdoctoral Fellow with Ottawa University, Ottawa, ON, where he was involved in a research project on local thermal equilibrium-based public safety networks. He is currently an Assistant Professor with German University, Cairo, Egypt, where he is also the Head of the IoT Lab and currently leading several projects such as an IoT Testbed for Smart Energy Management, Networked Appliances, Applications, and Sensing Systems for the Smart City, and iTram: An Information and Communication Technology Framework for Intelligent Transportation Systems. He has co-authored several book chapters and more than 30 publications.

Ismael Al-Shiab (Student Member, IEEE) received the M.Sc. degree in computer engineering from the American University of Sharjah, Sharjah, United Arab Emirates, in 2015. He is currently working toward the Ph.D. degree at the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada.

He is currently a Research Assistant with Carleton University. He is a Cisco Instructor Trainer and a Computer Engineer with more than ten years of academic and industrial experience in the fields of computer networks and routing, computer security, firewalls, and embedded systems. His current research interests include sensor networks virtualization, large-scale resource allocation optimization, software-defined networks, and information-centric networks under the umbrella of the Internet of Things.

Mohamed Ibnkahla (Senior Member, IEEE) received the Ph.D. and Habilitation a Diriger des Recherches degrees from the National Polytechnic Institute of Toulouse, Toulouse, France, in 1996 and 1998, respectively.

He joined the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada, in 2015, as a Full Professor, where he holds the Cisco Research Chair in Sensor Technology for the Internet of Things (IoT) and the NSERC/Cisco Industrial Research Chair in Sensor Networks for the IoT. Prior to joining Carleton University, he was a Professor at the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada, from 2000 to 2015. Over the past ten years, he has been conducting multidisciplinary research projects designing, developing, and deploying IoT systems targeting various application domains, including e-Health, smart buildings, public health, renewable energies and smart grid, public safety, security, intelligent transportation systems, environment monitoring, and smart cities. He published six books and more than 70 peer-reviewed journal papers and book chapters, 20 technical reports, 110 conference papers, and four invention disclosures. He has authored *Wireless Sensor Networks: A Cognitive Perspective* (CRC Press-Taylor and Francis, 2012) and *Cooperative Cognitive Radio Networks: The Complete Spectrum Cycle* (CRC Press-Taylor and Francis, 2015). In the past five years, he gave more than 30 keynote talks and invited seminars.

Dr. Ibnkahla received the Leopold Escande Medal, France, in 1997, and the Premier's Research Excellence Award, Canada, in 2001. He is the joint holder of five Best Paper Awards.