Abstract—IoT connects a large number of physical objects with the Internet that capture and exchange real-time information for service provisioning. Traditional network management schemes face challenges to manage vast amounts of network traffic generated by IoT services. Software-defined networking (SDN) and information-centric networking (ICN) are two complementary technologies that could be integrated to solve the challenges of different aspects of IoT service provisioning. ICN offers a clean slate design to accommodate continuously increasing network traffic by considering content as a network primitive. It provides a novel solution for information propagation and delivery for large-scale IoT services. On the other hand, SDN allocates overall network management responsibilities to a central controller, where network elements act merely as traffic forwarding components. An SDN-enabled network supports ICN without deploying ICN-capable hardware. Therefore, the integration of SDN and ICN provides benefits for large-scale IoT services. This article provides a comprehensive survey on software-defined information-centric Internet of Things (SDIC-IoT) for IoT service provisioning. We present critical enabling technologies of SDIC-IoT, discuss its architecture, and describe its benefits for IoT service provisioning. We elaborate on key IoT service provisioning requirements and discuss how SDIC-IoT supports different aspects of IoT services. We define different taxonomies of SDIC-IoT literature based on various performance parameters. Furthermore, we extensively discuss different use cases, synergies, and advances to realize the SDIC-IoT concept. Finally, we present current challenges and future research directions of IoT service provisioning using SDIC-IoT.

Index Terms—Information-centric networking (ICN), Internet of Things (IoT), service provisioning, software-defined networking (SDN).

I. INTRODUCTION

INTERNET of Things (IoT) aims to connect physical objects with the Internet, providing any time, anywhere, and any path connectivity [1]. These connected physical objects include smartwatches, health meters, temperature sensors, proximity sensors, accelerometers, and optical sensors. The connectivity of these objects gives rise to novel real-life services over a wide range of fields, including healthcare, smart cities, smart manufacturing, and industrial control systems. IoT can be deployed with sensing and intelligent capabilities to communicate, collaborate, and exchange real-time data for autonomous control of infrastructures. Diverse IoT applications and services require higher transmission capability than ever before. Services provided by IoT vary according to their use cases; however, the infrastructure used for communication is shared across all networks. The current Internet model was designed many decades ago [2]. Most Internet communication is performed between clients and servers using unique IP addresses. However, the majority of IoT services rely on content sharing between content providers and consumers rather than interacting with particular servers [3]. The sensory infrastructure is highly mobile, and the data dissemination from IoT devices depends on different wireless networking technologies, including cellular, WiFi, low power wide area networks (LPWANs), Bluetooth low energy, ZigBee, and WLAN [4]. Hence, different add-ons are continuously developed in the existing protocol stack to support novel network protocols [5].

The connectivity of IoT devices produces a massive amount of data giving rise to real-time big data streams. The access, discovery, and analysis of data put more constraints on the current TCP/IP architecture causing many problems [6]. One of the main concerns is the naming of IoT devices. Currently, the IPv4 naming space is exhausted, further portraying threats on the IPv6 address space that could be exhausted in the future [7]. Another challenge arises due to the long address length of IPv6, which is challenging to implement on constrained IoT devices. In this regard, efficient naming and addressing schemes for IoT devices are not duly available in this infrastructure [8]. Moreover, IoT suffers from different constraints in battery power, processing, size, usage (in-body/out-of-body), memory, and storage. IoT devices comprise heterogeneous hardware and vendor-specific configurations requiring device-specific operating systems. Due to the limited size, battery, and storage constraints, data often become unavailable [9]. To these challenges, solutions, like in-network caching to make data available, tend to be missing in the traditional IP-based networks [10]. Moreover, IoT services require more security and extra privacy based on accessed data and its usage [11]. Furthermore, IoT services require...
data mobility, which suffers from mobility management, fault-tolerance, and hand-off issues. IoT requires network architectures that facilitate changing network paths, provide technological abstractions, and consider data exchanges instead of access nodes [12]. This increases network complexities making it challenging to implement novel network configurations [13]. In this context, software-defined networking (SDN) and information-centric networking (ICN) are proposed to address the limitations of current communication technologies at the design level [14].

ICN has proved to be a novel communication model for content distribution challenges faced by IoT [15]. It is sustainable for a wide range of IoT applications that are information-centric as they focus on data regardless of the origin of the data [16]. It offers flexibility to address the content directly by name instead of the address of the content [6]. It introduces named content instead of IP packets that are transmitted in the network toward the destination. This offers a flexible communication model shifting away from a host-centric environment to a content-centric paradigm [17]. In this case, the actual target is the content itself rather than the address of the content. Hence, in-network caching can be implemented by storing content near the user vicinity to improve data retrieval and reduce network congestion. ICN efficiently manages mobility by reissuing any unsatisfied requests considering the fact that the content is independent of its location [18]. ICN provides instant data access using a request-reply exchange mechanism, content-based security, and multicast support that is specifically beneficial for IoT [19]. Although this seems to be conceptually simple; it becomes challenging to implement as an overlay on existing networks [9]. ICN creates increased complexity for IP routing, which makes it hard to fully utilize in legacy networks [20].

Currently, Internet architecture faces challenges to scale due to the content-centric requirements of the underlying applications [28]. A vital aspect of the future Internet is that it should support a content-driven paradigm rather than a connection-driven paradigm. Internet was designed for the entity-based standpoint providing a communication platform through established connections [29]. A wide variety of content-centric IoT services cause communication overhead on the current Internet architecture [30]. Although the content-centric nature of IoT provides the utility of access, it causes discrepancies in the current Internet architecture [31]. Content-centric systems exchange content without considering the origin of the content. However, their implementation requires designing novel architectures based on various prioritization schemes for the content regardless of its origin. SDN becomes helpful in this situation by supporting network programmability and flexible network function implementation.

SDN aims at increasing network programmability by separating control and data planes [1]. It develops a layered network architecture by dividing into the application, control, and data planes. In addition to the layered architecture, multiple application programming interfaces (APIs) exist, including northbound, southbound, and east/westbound interfaces supporting the flexible programmability of SDN. The northbound API provides an abstract view of the network to the application plane, whereas the southbound API provides a communication interface between data and control planes. This enables the controller to deploy traffic forwarding rules in the network elements. The east/westbound API allows multiple controllers to coordinate in order to present a logically centralized controller view and support integrated decisions. The controller implements adequate control logic, on the forwarding elements, based on service requirements to forward network traffic. SDN provides flexibility to develop, deploy, and improve network services in an agile and flexible way.

A. Motivation of This Survey

Fig. 1 illustrates a use case of IoT service provisioning using SDN and ICN. It shows sensors connected to the physical objects capturing real-time data and delivering services to the intended stakeholders. The motivation of this survey is prompted by the need for SDN and ICN for IoT services and the lack of existing surveys on this topic. IoT devices can deliver content-based services where SDN and ICN could act as complementary technologies for IoT service delivery. These services overwhelm the network with huge amounts of content, posing network management challenges. To overcome these challenges, several techniques use SDN and ICN for IoT service provisioning separately. However, SDN and ICN are promising paradigms that could benefit from each other to solve multiple networking challenges of IoT services. Therefore, researchers have already proposed to integrate these paradigms to develop a software-defined information-centric Internet of Things (SDIC-IoT) architecture for seamless service provisioning [32], [33], [34]. SDN benefits from centralized control, robust management, and a global view of the network. SDN-enabled ICN can handle the distribution of IoT data in partially upgraded network, simplify the deployment, and consolidate the transition phase. Moreover, the ICN feature of in-network is capable of information delivery from unavailable IoT by caching content at intermediate nodes [35]. It minimizes delay caused by data retrieval, in the
TABLE I
COMPARISON OF THIS SURVEY BASED ON THE DISCUSSED CHARACTERISTICS WITH EXISTING SURVEYS

| Reference   | Year | SDN | ICN | IoT | NFV | SDIC-IoT | Network Management | Scalability | Mobility | 5G-6G Networks |
|-------------|------|-----|-----|-----|-----|----------|--------------------|-------------|----------|---------------|
| Amadeo et al. [3] | 2016 | ✗   | ✓   | ✓   | ✗   | ✗        | ✗                  | ✓           | ✓        | ✓             |
| Arshad et al. [16] | 2018 | ✗   | ✓   | ✓   | ✗   | ✗        | ✓                  | ✓           | ✗        | ✓             |
| Salman et al. [21] | 2018 | ✓   | ✓   | ✗   | ✓   | ✗        | ✓                  | ✓           | ✓        | ✓             |
| Fang et al. [22] | 2018 | ✗   | ✓   | ✓   | ✗   | ✗        | ✓                  | ✓           | ✓        | ✗             |
| Fang et al. [23] | 2017 | ✗   | ✓   | ✓   | ✗   | ✗        | ✓                  | ✓           | ✓        | ✓             |
| Zhang et al. [24] | 2015 | ✗   | ✓   | ✓   | ✗   | ✗        | ✓                  | ✓           | ✓        | ✗             |
| Bekri et al. [5] | 2020 | ✓   | ✓   | ✓   | ✓   | ✓        | ✓                  | ✓           | ✓        | ✓             |
| Farris et al. [25] | 2018 | ✓   | ✓   | ✓   | ✓   | ✓        | ✓                  | ✓           | ✓        | ✓             |
| Bizanis et al. [26] | 2016 | ✓   | ✓   | ✗   | ✓   | ✓        | ✓                  | ✓           | ✓        | ✓             |
| Zhang et al. [14] | 2018 | ✓   | ✓   | ✗   | ✓   | ✓        | ✓                  | ✓           | ✓        | ✓             |
| Jmal et al. [27] | 2017 | ✓   | ✓   | ✓   | ✓   | ✓        | ✓                  | ✓           | ✓        | ✓             |
| This Survey | 2022 | ✓   | ✓   | ✓   | ✓   | ✓        | ✓                  | ✓           | ✓        | ✓             |

In this context, Zhang et al. [14] proposed a survey on the integration of SDN and ICN. This survey focuses on communication aspects of SDN and ICN for general-purpose computing. They survey ICN caching mechanisms based on SDN; however, their study does not cover implementation schemes. Jmal and Fourati [27] surveyed content-centric network (CCN) management using SDN. However, it has a limited scope and lacks comprehensive aspects of SDN and ICN integration. Furthermore, this study [27] does not cover the novel characteristics of software-defined ICN (SDICN) that could be beneficial for the resource-limited IoT. There is a lack of existing surveys that address the integration of SDN and ICN for IoT service provisioning. To the best of our knowledge, this survey is the first one to cover SDICN for IoT by considering core issues faced by IoT in TCP/IP networks for service provisioning. Table I shows the comparison of this survey with existing surveys based on various parameters that we address in this article.

C. Contributions of This Survey

In this survey, we discuss the integration of SDN and ICN to solve communication, content management, and service provisioning challenges faced by IoT. We provide a comprehensive literature survey on existing SDN and ICN-based approaches of in-network caching, content naming, and mobility management mechanisms for IoT. The contributions of this survey can be summarized as follows.

1) We discuss the underlying motivation of integrating SDN and ICN and the key benefits that it brings for IoT service provisioning.
2) We present and critically analyze the requirements of SDIC-IoT and its adaptability toward IoT for content naming, caching, security, and mobility management.
3) We categorize existing literature that focuses on integrating SDN and ICN and classify it based on different parameters, including mobility, virtualization, caching, and implementation.
4) We discuss novel use cases of SDIC-IoT that provide a model for future SDIC-IoT systems.
5) We present several open issues, challenges, and future research directions regarding the implementation of future SDIC-IoT systems.
A. Information-Centric Networking

The key advantage of ICN is its capability of addressing network content by names instead of addresses [6]. ICN is based on the publish/subscribe Internet architecture, which focuses on the efficient delivery of content to the user regardless of the origin of the content. Compared to the host-centric connectivity in IP networks, ICN allows its connected users to obtain content from any location in the network [16]. Therefore, ICN could be exploited to solve the challenges in current IP networks, including mobility, energy efficiency, congestion, and security [32].

1) Evolution of ICN: Fig. 2 shows the evolution of ICN since it was first introduced in 1999 by Cheriton while addressing name-based routing in Translating Relaying Internet Architecture Integrating Active Directories (TRIAD) architecture [38]. National Science Foundation (NSF) of the USA has proposed different projects that target future Internet Architectures (FIAs), including named data networking (NDN), Nebula, expressive Internet architecture (XIA), and ChoiceNet [39]. UC Berkeley and the European commission (EU) also introduced numerous novel projects that include Data-Oriented Network Architecture (DONA), COMET, CONVERGENCE, scalable and adaptive Internet solutions (SAILS), and publish-subscribe Internet technology (Pursuit) [40], [41]. Subsequently, in 2007 the concept of DONA was introduced as the first clean slate architecture of ICN [42]. In 2009, CCN was proposed based on replacing the IP protocol stack [43]. Later CoNet, convergence of fixed and mobile broadband access (COMBO), and Content-centric fashion MANET (CHANET) adopted the concept of CCN [44]. The architecture of CHANET is fundamentally the same as of CCN; however, it has not been taken much into consideration [45]. Later in 2010, NSF introduced a novel program for FIA, which introduced MobilityFirst [46]. This program devised an architecture based on mobility to offer seamless communication among users based on ICN principles. Another add-on of the ICN program was NDN [12] that was an extension of the CCN approach. Similar projects include GreenICN and named function networking (NFN) [12]. However, NFN does provide specific differences compared to NDN. As part of the FP7 European program, COMET proposed a content-oriented architecture based on CCN and NDN concepts for efficient content access and dissemination [42]. In 2015, the EU funded two more projects, including iP Over IcN the betTer IP (POINT), and UMobile [47]. These programs were based on the assumption that some IP-based applications could benefit from ICN-based networks. POINT provides a platform, which supports IP-based applications, such as legacy applications during the interim period of a globally deployed ICN network. UMobile focused on an ICN-based mobile-centric architecture to assist communication among mobile participants, such as sensors, actuators, and mobile devices [48]. Both projects are in their evolutionary stage; therefore, the final version of this architecture is not yet available. This is also implemented in the ICN 2020 project that started in mid-2016; one of the goals is to improve underlying ICN solutions for IoT.

A robust architecture of ICN is CCN; it offers communication using two specialized packets that include Interest and data packets [20]. These packets use name to uniquely address the desired content. The Interest packet, issued by the user, identifies the content of interest while the data packet include the corresponding content to be returned to the user. ICN aims at ensuring efficient data dissemination and integration of the overall network features around the content. The fundamental element in ICN is the content name, which has the same significance as IP having unique, persistent, secure, and location-independent in the overall network context. Considering the content-oriented nature of ICN, content names could be represented in various ways, such as attribute-value, flat, or hierarchical. Therefore, the network layer of traditional IP-based networks could be modified; and it can use content names for name-based routing. Moreover, in-network
caching and content replication allow data access and retrieval during communication.

B. Internet of Things

IoT connects billions of devices, to the Internet, that collect and share data and provide services in various areas of life. The sensing capability, along with the availability of wireless sensor networks, turns these objects into IoT. Things in IoT can be humans and other objects connected using machine-to-machine (M2M), machine-to-human, and human-to-human interfaces. Many definitions exist in the literature for IoT. IETF defines IoT as uniquely identifiable objects connected using TCP/IP and non-TCP/IP suites [49]. IEEE defines IoT as uniquely identifiable physical and virtual objects employing existing and novel communication protocols [50]. IoT is an umbrella term having underlying use cases, including M2M, Industrial IoT (IIoT), Social IoT (SIoT), Web of Things (WoT), Internet of Medical Things (IoMT), and Internet of Everything (IoE).

Users of IoT applications are more interested in getting access to content instead of the address of the content source. This is because of the vast proliferation of content-oriented applications during the past few years. For example, every device in IoT collects data at a larger and continuous scale. TCP/IP was initially designed to serve the expensive network resources using a limited address space [51]. It was not originally intended to connect billions of uniquely identifiable IoT objects. Moreover, the sophisticated nature of IoT puts vast constraints on content dissemination and scalability of the underlying architecture. Therefore, ICN has emerged as an ideal candidate to fulfilling the needs of IoT infrastructure [52]. The primary characteristics of ICN include in-network caching, naming, security, mobility, and scalable information delivery that are highly suitable for IoT services [8]. Furthermore, the ICN-oriented hourglass framework supports thin-waist like TCP/IP [53]. Additionally, ICN could mask the traditional TCP/IP layer or media access control (MAC) layer. Although existing literature [46], [54] argue that ICN is poised to replace TCP/IP, it is not the case since ICN is an overlay network that sits on an IP network. ICN maps IP addresses with content names and delivers content using MAC addresses [54].

C. Software-Defined Networking

Communication networks weave the fabric of the current connected world. SDN is a novel paradigm of network management aiming at separating data and control planes to manage network components flexibly [28]. The controller in SDN acts as the brain of the whole network performing most of the computations. The data plane contains forwarding devices (e.g., routers and switches) responsible for flow rule matching to forward incoming packets. Forwarding devices send packets to the controller if the relevant flow rules are not available in the limited ternary content addressable memory (TCAM) memories. The controller analyzes the flow and installs the corresponding flow rule in the data plane device, which then forwards the packet toward its destination. SDN brings independence between control and data planes, offering flexibility to service providers and application developers to proactively manage network resources and support network expansion [28]. It provides flexible network evolution where network operators configure, upgrade, and maintain network resources dynamically. SDN breaks the vertical architecture of network administration and separates the data and control planes for flexible network management. Thus, network operators and application developers can configure, upgrade, deploy, and flexibly maintain network services. The logical centralization of the network enables the controller to gain access to all network resources where traffic can be efficiently managed. Traditional networks handle packets using MAC addresses, IP prefixes, and combinations of IP and TCP/UDP port numbers using the control-data plane interface (C-DPI) [55].

D. Software-Defined Information-Centric Networking for IoT

Table II shows challenges that are unique for IoT service provisioning. It includes SDICN characteristics that could be beneficial in addressing these challenges. A practical realization of ICN for IoT requires an efficient architecture hiding all network complexities. In SDN, the centralized controller contains the network’s global view, enabling efficient network resource management and control. SDN becomes a key candidate in the SDICN paradigm, providing greater network programmability by separating control and data planes. The control plane of SDN lowers down the implementation and architectural complexities of ICN, allowing novel mechanisms for resource management [9]. Although SDN simplifies network management, its architecture is still based on TCP/IP. Bringing ICN into SDN alleviates the challenges of host-centric networking. Shifting from “where” to “what” brings novel opportunities for efficient traffic management. SDN can optimally exploit in-network caching and multipath routing using name-based content retrieval with the global view of the network. SDN establishes flow-based end-to-end
communication that ignores the content going through it. Without further processing, such as deep packet inspections (DPIs), different types of contents get the same treatment when their port number, source address, and destination addresses are identical; this is usually suboptimal. ICN provides the capability to inspect content metadata at the network layer without DPI [34]. The content metadata describes the content length and type, enabling network service providers to optimize network behaviors accordingly. The OpenFlow channel and its north and southbound interfaces are prone to attacks [56]. To counter this, ICN can protect the content itself, securing the overall network. Indeed, the content-centric security model of ICN could protect the overall network enhancing efficient access to data generated by IoT. Both SDN and ICN complement each other, paving the way for efficient IoT service delivery bringing the novel idea of SDIC-IoT.

Fig. 3 illustrates the proposed architecture of SDIC-IoT where IoT devices are connected at the data plane. These devices interact with data plane components to transfer the captured data. Forwarding switches at the data plane act as caching components to store data near content consumers. The controller in this architecture determines optimal caching strategies based on the central view of the connected components. The modified application plane in the SDIC-IoT architecture contains ICN-specific applications, including routing, caching, and performance optimization. IoT devices communicate with the data plane devices for content transfer. Upon receiving a content request, the content is forwarded to the controller, which installs relevant flow rules at the data plane devices. Data plane devices then deliver the content to the content consumers based on the flow rule. In this architecture, in-network caching and multicasting characteristics reduce overall transmission time and increase network throughput. Directly identifying services by content names overcomes the domain-specific heterogeneity of IoT and provides an abstraction to deliver the content between content producers and consumers. Fig. 4 shows the content caching mechanism using the SDIC-IoT architecture. A consumer generates a request for content C2. This request is directed to the caching node storing content C2. This content is then forwarded to the consumer. Hence, this architecture efficiently manages network resources by storing the content on fewer nodes. In the following, we present the benefits of SDIC-IoT.

1) IoT Content Management: ICN tries to solve the problems of content distribution efficiency, security, and network congestion (exacerbated by the increase of content generated by IoT and the shift in multimedia technology usage). At the same time, SDN brings flexibility in network management. The motivation for using the SDIC-IoT architecture comes from the large-scale data generation capabilities of IoT. SDIC-IoT provides the flexibility to cache content on important switches that are frequently traversed by the network traffic. Hence, this architecture reduces frequent content request redirection and reduces latency in accessing the content. In traditional caching, content unavailability, replication, and request redirections are significant challenges. Content becomes unavailable when content gets no-hit in a specific time interval. Hence, it causes content redundancy and unavailability at the same time. This results in inefficient utilization of network resources and an increase of network management costs. SDN guides network traffic efficiently, since it has the global view of the network; this supports ICN capability of caching and routing simultaneously. Routing strategies can be flexibly configured using well-defined interfaces offered by SDN (e.g., OpenFlow, RESTful, and east/westbound interfaces). SDN also provides various types of customized services using different namespaces supported by ICN. Fig. 5 illustrates a taxonomy of caching management schemes. It shows how centralized caching using SDN overcomes the challenges of distributed caching strategies.

2) Content-Oriented Communication Model for IoT: SDN simplifies network management by providing a global view of the network. However, the current SDN architecture is
still based on TCP/IP. Therefore, the challenges of TCP/IP still exist in SDN. In this case, ICN solves the challenges of the host-centric paradigm of TCP/IP by relying on content names rather than the content location. Furthermore, IoT service provisioning relies on content transfer from source to destination. This content can be obtained directly from producers or through intermediate nodes storing the content. Due to the content-oriented model of ICN, SDN can efficiently utilize in-network caching and multipath routing to transfer IoT-generated content. In-network caching and multicast routing increase network throughput by decreasing overall data transmission throughout the network. Moreover, the SDN controller can find optimal routes for data caching and transfer for IoT services.

3) Centralized Control for IoT Services: The centralized control of SDN allows for robust control of route selection and caching nodes management. Network operators can program networks (via A-CPI) using optimal route selection strategies supported by the controller. SDN provides content using different namespaces for different domains of IoT applications. Novel security services can be provided by key distribution and revocation, strengthening security in the resource-limited IoT. Evolving caching mechanisms of ICN for IoT services can be efficiently managed using dynamic control supported by SDN. Various implementations of A-CPI and C-DPI support programmability to route traffic according to the user needs. Programmable control and data plane support the development of customized functionalities on specific network nodes. IoT services can benefit through centralized control of general-purpose networking devices by flexibly forwarding name-based traffic. ICN mechanisms of in-network caching suffer from content replication and proximity of caching decisions. In this case, SDN can manage content caching status through a centralized view of the caching nodes. Moreover, SDN always has the topology information of the network, which helps optimize resource allocation. Indeed, SDN on-path and off-path caching schemes reduce caching redundancies and increase the overall cache hit ratio.

4) Energy Efficiency: Resource-limited IoT devices have limitations with respect to power and computing capabilities. Most IoT devices spend a significant part of their life in sleep mode and only awake when they exchange data. Existing IoT networks handle energy at the medium access layer or above the transport layer. ICN caching coupled with SDN management capabilities can reduce extra energy consumption. A service request could be processed by an intermediate node that caches the content when the content producer is idle. Having a local copy of the content reduces the extra energy consumption required to communicate with the producer.

E. Summary and Insights

The SDIC-IoT architecture provides a promising solution for IoT service provisioning. However, it creates other challenges of fault-tolerance, privacy, security, and scalability of the heterogeneous SDIC-IoT infrastructure. A hybrid flow rule installation strategy must be used to install flow rules on the data plane devices to fulfill the installation needs of the large-scale flow rule. The application plane in SDIC-IoT could be used to develop novel applications to implement various network functions. The OpenFlow channel in this architecture may become a bottleneck due to the centralized control [57]. Although distributed control strategy could be exploited to address this challenge, it causes increased latency, lower throughput, controller-state synchronization, and security challenges. Hence, a practical realization of SDIC-IoT should consider the requirements of all the involved infrastructures, communication challenges, and resource allocation concerns.

III. REQUIREMENTS OF FUTURE SOFTWARE-DEFINED INFORMATION-CENTRIC INTERNET OF THINGS

SDIC-IoT offers tremendous opportunities for IoT content management by leveraging ICN capabilities of an independent data plane and SDN benefits of central control of data plane elements. However, this complex resource integration of IoT, SDN, and ICN on a single platform requires the consideration of heterogeneous domain-specific requirements. We discuss these requirements in the following.

A. Requirements of Heterogeneous Network Management

Continuous growth and applications of IoT produce huge amounts of content requiring latency-aware and efficient processing. An important aspect is the management of huge amounts of devices and content produced by these devices. Thus, suitable communication techniques are needed to distribute efficiently and optimally control network traffic. Traffic management involves techniques for load balancing and delay reduction. These requirements can be satisfied by SDIC-IoT using hybrid capabilities to manage underlying networks. In this situation, SDN provides a global view of the network to route traffic toward its destination optimally. ICN facilitates unique, persistent, and location-independent naming that IoT applications could directly address. Thus, SDICN can be used for IoT network management for load balancing, reducing delays, optimized traffic forwarding, and efficient bandwidth utilization. Moreover, data analytics techniques could be developed at the network’s edge to reduce network traffic and simplify network management. However, centralized management of SDN poses significant challenges to the implementation of SDIC-IoT. Therefore, network management
requirements ask for efficient management of SDIC-IoT using efficient strategies that have been tested on large-scale networks [58], [59].

B. Network Function Virtualization Requirements

Traditional networks rely on the predefined programmability of network devices, making it difficult to program them for multitasking [60]. Virtualization of network devices supports network function update in real time, allowing them to perform multitasking. Implementation of network functional virtualization (NFV) for Internet service providers (ISPs), becomes easier due to the separation of control and data plane devices. Furthermore, SDN allows optimal network function placement for guiding flexible traffic flow. Function-centric service chaining extends ICN by transferring content naming to function names [61]. It separates network functions from their locations, It improves the scalability and flexibility of service chaining. Combining NFV with SDN enables novel application development through agile network resource management [33]. ICN establishes a novel way of network management by data plane programmability offered by SDN and NFV.

C. Requirements of Location-Independent Information Access

Device owners connected using IoT need to access their data independent of the location. An information access service must allow efficient content access based on content name, context, and information metadata. The requirements for such services should allow scalability, fault-tolerance, and acceptable latency of content access. An API could be developed for efficient content access by hiding the complexities of the underlying hardware and providing information advertising and lookup. However, the requirements of such a service to identify the capabilities of end devices should be proactively considered before developing such an API. Although IoT may implement the API or use gateways, security issues arise due to insecure transmission mediums. Therefore, security and privacy requirements should be considered for seamless information access on attack-vulnerable networks.

D. Security and Privacy Requirements

IoT establishes interoperability among different cyber-physical systems for service provisioning. The resource-limited nature of these devices exposes them to different types of security and privacy attacks [62]. IP does not support security by design; it was later included to support authentication and data integrity [63]. In this case, wireless communication among low-powered nodes degrades the performance of existing protocols. IP-based security protocols (e.g., IKEv2/IPsec, TLS, and EAP), IPv6 over 6LoWPAN, and CORE vastly depend on the location of nodes in the network [64]. These protocols secure the communication channel rather than the content. When security is supported with other capabilities (e.g., mobility), it creates more network overhead, and complex management [65]. SDIC-IoT supports secure content sharing among nodes since authentication and integrity can be ensured locally. It eliminates the lack of trust issues that arise due to content dissemination among intermediary nodes. The security of content reduces the overhead of securing the whole communication medium.

E. Interoperability Requirements Among SDN, ICN, and IoT

IoT traffic passes through different network infrastructures arising interoperability issues. Network owners implement domain-specific policies, including data format restrictions, security strategies, and waypoint enforcement schemes [57]. These policies exacerbate interoperability issues due to the lack of a centralized governing authority. Furthermore, interoperability is sought for the heterogeneous IoT infrastructure to operate seamlessly among heterogeneous network entities. ICN characteristics, of content-based addressing which is independent of location, allow for improving interoperability. Thus, the controller could be programmed to forward traffic based on content names rather than content addresses [66]. Interoperability gateways can be deployed between different network domains that may perform content translations requiring unified interoperability mechanisms. The southbound OpenFlow interface of SDN provides a standard way to address hardware elements attached to the network. These elements merely act as forwarding components, thereby alleviating any hardware dependencies in addressing the content. However, flow rule installation strategies, limited TCAM memory management, and flow setup latency heightens interoperability challenges. New protocols and virtualization standards can be developed to support interoperability.

F. Mobility Requirements of SDIC-IoT

Mobility has become a salient feature of IoT due to device transformation into smart gadgets producing and transmitting data in real time. Mobility can be of two types: 1) consumer mobility and 2) producer mobility [67]. ICN supports consumer mobility by Interest retransmission and in-network caching [68]. In a mobility situation, the data consumer requests the required data while moving to another location far from the caching router. In this case, the consumer retransmits Interest for the same content. An intermediate router responds to the request. If data is not available at any caching node, Interest is forwarded to the data producer, which delivers the content with a longer delay. However, the situation worsens if the producer moves to another location before receiving Interest [69]. The new name prefix of the producer might be unknown to any ICN router until the network converges. This halts the communication for at least the sum of the handover time and the time taken for the network to (re)converge. In this case, SDN provides centralized control to efficiently select the closest router to cache content based on consumer/producer mobility. However, the processing of IoT service requests in a mobile environment without distributing services, that are being executed, is challenging. Therefore, seamless content delivery to routers without worrying about services is crucial to enhancing the Quality of Service (QoS) in SDIC-IoT.
G. Scalability Requirements of SDIC-IoT

IoT promises to connect billions of smart objects, instigating the need for scalable networks capable of supporting large-scale infrastructure. Although current solutions such as IPv6 has a greater address space capable of addressing IoT devices, large-scale production of IoT poses future addressing challenges [16], [70]. Another challenge is the enormous amounts of data produced by IoT, requiring efficient, and scalable management. Both SDN and ICN face challenges of scalability in different ways. IoT services generate a huge amount of content. In this case, the amount of content might be greater than the capacity of network devices. Thus, managing this content might cause a state space explosion. Establishing content forwarding at the data plane of SDIC-IoT, based on content names, might overflow limited TCAM memory of forwarding devices. Although the SDN controller provides the capability of state-awareness, cache status, and latency-aware cache updates make it difficult to provide efficient IoT service provisioning. These additional requirements cause scalability issues to the SDN controller. Moreover, flow setup latency based on content names at the data plane devices also causes major scalability issues [71]. Centralized management of SDIC-IoT poses restrictions on the single controller. Therefore, the architecture of SDIC-IoT should be scalable enough to efficiently support network content delivery.

H. Requirements for Fault-Tolerant SDIC-IoT

Even in the failure of some of the SDIC-IoT components, the ability to provide services is essential to establishing fault tolerance. This involves fault detection, damage assessment, error recovery, and fault-treatment [72]. IoT relies on continuous Internet connectivity to provide services to the underlying systems. Fault tolerance in IoT services becomes challenging due to the asymmetric communication between IoT and communication infrastructure. SDN can route traffic toward available data plane devices even if some parts of the network fail. Faults in IoT are particularly probable due to the highly connected nature of IoT. Therefore, proactive fault-tolerant approaches should be implemented so that the system may act before faults. SDN techniques could provide device backup paths enabling the network to use them in case of failures. The programmed strategy is activated in case of faults, and the traffic is routed toward the backup links. Backup controllers could also be implemented for fault recovery. The SDIC-IoT controller could be programmed to self-heal from different kinds of attacks. Store and supply mechanisms for content caching could be used to ensure end-to-end delivery of content. Moreover, to improve fault tolerance, SDIC-IoT can cache content on intermediate nodes if some of the caching nodes become unavailable.

I. Lessons Learned: Summary and Insights

Table III shows a taxonomy of requirements for the implementation of future SDIC-IoT systems. In this section, we discussed various requirements of implementing SDIC-IoT for end-to-end IoT service provisioning. SDN capabilities of programmability, centralized control, and flexible innovation could be beneficial for efficient ICN implementation. Combined SDN and ICN characteristics can provide suitable mechanisms for service provisioning in smart infrastructures. The application plane of the SDIC-IoT architecture provides a platform to develop novel applications for customized traffic forwarding based on the requirements of the underlying infrastructures.

IV. Taxonomy of Software-Defined Information-Centric Internet of Things

This section discusses the taxonomies of SDIC-IoT using various parameters. We divide existing literature into eight categories; each category addresses different aspects of IoT service provisioning.

A. SDIC-IoT Frameworks for IoT

SDN and ICN bring significant innovations in the future IoT service provisioning architecture using various framework implementations. Each of the approaches, we cover in this section, implements a framework for SDIC-IoT integration for IoT service delivery. Veltri et al. [73] developed the coCONET framework to deploy ICN in the existing SDN architecture. They modify the OpenFlow channel to support coCONET operations. The coCONET framework is responsible for Interest propagation in the network toward the host which stores the content. The host replies with the content by sending the requested content to the source. Wang et al. [74] developed a novel framework to integrate different ICNs on top of the same physical network. They deploy ICN architecture on an Open vSwitch at the data plane. This framework uses content distribution management, forwarding path calculation, name to tag mapping, and OpenFlow strategy dissemination for IoT service delivery. Eum et al. [75] distributed SDICN framework layout into edge networks, the OF-ICN forwarding network, and the OF-ICN control network. The OF-ICN forwarding network transfers data collected through sensors at the edge networks. The OF-ICN control network acts as a centralized control authority to optimally route network traffic. Arumaiturai et al. [76] proposed an ICN-based service chaining framework that complements SDN characteristics. It supports scalability, flexibility, and reliability in the underlying network. The controller selects an optimal path for traffic forwarding through Open vSwitches, which implement service chaining. However, this technique stacks the data packet labels casing delays while searching accurate service labels from the stack. Xiulei et al. [77] proposed a low complexity SDICN framework based on OpenFlow. This framework ensures optimal data collection from target data centers using predefined rules installed at the data plane devices. It also provides load balancing by selecting optimal paths from source to destination. Siracusano et al. [78] deployed an experimental framework of ICN over SDN using virtual and physical testbeds. They develop a reference environment by updating control logic to implement optimal caching policies over SDN. However, the scope of the proposed caching strategy is limited, and experimental evaluation suffers from simulation errors when deployed on an actual service provisioning environment.
TABLE III
TAXONOMY OF IOT SERVICE PROVISIONING REQUIREMENTS USING FUTURE SDIC-IoT SYSTEMS, THEIR CHALLENGES, AND PROPOSED SOLUTIONS

| Requirements of heterogeneous network management | Challenges | Solutions |
|-------------------------------------------------|------------|-----------|
| • Huge IoT content. | • Hybrid capabilities of SDN and ICN. | | |
| • Continuous growth of the network. | • SDN capability of centralized control. | | |
| • Lack of data analytics techniques. | • Location-independent addressing of ICN. | | |
| • Distribution and control of traffic. | • Load balancing/traffic management. | | |

| Network function virtualization requirements | Challenges | Solutions |
|----------------------------------------------|------------|-----------|
| • Programmability of network devices. | • Implementation of NFV for ISP. | | |
| • Lack of centralized view of the network. | • Decoupling of control and data planes. | | |
| • Lack of agile network control. | • NFV-based network function placement. | | |
| • Network function dependence on location. | • Agile network resource placement. | | |

| Requirements of location-independent information access | Challenges | Solutions |
|----------------------------------------------------------|------------|-----------|
| • Location-dependent information access. | • Location-independent data access. | | |
| • Fault-tolerance/latency issues. | • Flexibly changing network functions. | | |
| • Lack of name-based addressing at data plane. | • Data plane name-based addressing. | | |
| • Lack of location-identification at data plane. | • Seamless API implementation at end devices. | | |

| Security and privacy requirements | Challenges | Solutions |
|----------------------------------|------------|-----------|
| • Heterogeneous infrastructures. | • Local security policies implementation. | | |
| • Lack of security in IP by design. | • Lightweight security and privacy solutions. | | |
| • Complexity in securing mobile devices. | • Content level security solutions. | | |
| • Lack of content-protection schemes. | • Defined security policies. | | |

| Interoperability requirements among SDN, ICN, and IoT | Challenges | Solutions |
|-------------------------------------------------------|------------|-----------|
| • Diversity of infrastructures. | • Centralized governance of the network. | | |
| • Lack of centralized governance. | • Content-based addressing of traffic. | | |
| • Content translation among networks. | • Development of interoperability gateways. | | |
| • Resource management issues at data plane. | • Global communication protocol. | | |
| | • Virtualization of hardware elements. | | |

| Mobility requirements of SDIC-IoT | Challenges | Solutions |
|-----------------------------------|------------|-----------|
| • Large-scale development of mobile IoT. | • Interoperability using Interest re-transmission. | | |
| • Challenges associated with producer mobility. | • Optimal caching node selection. | | |
| • Latency in request propagation. | • Seamless maneuvering of services. | | |
| • Seamless maneuvering of services. | • Solutions for producer mobility. | | |

| Scalability requirements of SDIC-IoT | Challenges | Solutions |
|--------------------------------------|------------|-----------|
| • Huge traffic to and from Service providers. | • Intelligent content caching strategies. | | |
| • Addressing of large-scale IoT objects. | • Virtual implementations of single controller. | | |
| • Overflow of TCAM memory. | • Optimizing east-west bound API of SDN. | | |
| • Bottleneck of cache state update. | • Content scope definition using content names. | | |

| Requirements for fault-tolerant SDIC-IoT | Challenges | Solutions |
|-----------------------------------------|------------|-----------|
| • Asymmetric communication between. | • SDN capability of routing traffic. | | |
| • Highly connected nature of IoT. | • Proactive fault-tolerant solutions. | | |
| • Lack of backup paths in case of failures. | • Pre-programmed fault-recovery strategies. | | |
| • Reactive solutions to failures. | • Store and supply caching using ICN. | | |

Liu et al. [79] simplified the control of space-terrestrial networks using SDN and ICN. They use request aggregation and data distribution of forwarding nodes to control the network and reduce delays. However, least recently used (LRU) and least frequently used (LFU) caching strategies consume higher network resources. Thus, efficient cache replacement strategies are still required to limit the network resource consumption. Liu et al. [80] proposed a framework for secure cache resource trading between caching nodes using blockchain. When two caching nodes want to trade resources, they exchange their public keys. If the public key does not relate to the node’s identity, the identity is not trusted, and the process is terminated. However, this framework increases service delay during authentication and suffers from scalability issues.

Existing SDIC-IoT frameworks support virtualization of network resources and deploy ICN characteristics at data plane components. However, most of these frameworks do not consider the various resource-limitation constraints of IoT devices. Therefore, novel SDIC-IoT frameworks must consider the resource-limitation constraints of IoT.

B. SDIC-IoT for IoT Mobility

Mobility is an essential aspect of IoT due to the mobile nature of content-capturing devices. SDIC-IoT efficiently addresses content mobility issues using global content naming, routing, and caching strategies. Ren et al. [81] proposed a mobility management approach for CCN using SDN control logic. The controller ensures mobility by dynamically updating routing tables at the data plane, allowing simultaneous updates of content locations and names. However, weak experimental evaluation limits its applicability in a real environment. Zhang and Wang [82] proposed a QoS-aware SDICN integration for satellite-terrestrial networks. This scheme supports content delivery, mobility, in-network caching, and content-based traffic engineering. However, latency and energy consumption increases during continuous interactions of the data plane and control plane. Therefore, this approach requires design considerations for a low latency control plane. Xing et al. [83] proposed a wide area deployable SDICN approach, which separates wireless access network and core transmission network. Edge services network connects resource-limited devices and implements intelligent traffic routing and caching. However, the authors perform small-scale experiments that cannot ensure its applicability in a real environment. Further, optimized path selection for content has not been considered. Ravindran et al. [84] proposed mobility as a service over 5G-ICN using application-driven network slices. This technique delivers services over ICN in a virtualized control and data plane to overcome complex network logic. However, communication among different
network slices causes scalability issues. He et al. [85] proposed a mobility-aware low-complexity resource allocation strategy in SDICN. They develop an integration scheme, which jointly considers networking, caching, and computing for performance enhancement in vehicular networks. However, the integration of heterogeneous platforms increases the complexity of the system. Leal et al. [86] proposed an integration framework of SDICN to meet mobility requirements of command and control on the battlefield IoT. SDN deals with traditional IP networks, whereas ICN supports the heterogeneity of the battlefield IoT. However, smooth SDN controller migration in SDN and ICN control still needs novel solutions. Benedetti et al. [67] proposed SDICN for mobility support to overcome the limitations of the lack of state forwarding information in traditional ICN. This technique aims to evaluate the energy consumed by intermediate routers considering different mobility and application settings. However, Benedetti et al. [67] does not considered optimization strategies in the presence of multiple mobile consumers and SDN controllers. Ban [87] proposed a behavioral analysis strategy using convolutional neural networks on the data collected in SDICN. However, this technique suffers from low convergence speed and unawareness of data size intervals. Zhao and Zhu [88] formulated an Knapsack 0–1 optimization problem for D2D content caching and update in an IoT mobility scenario. This scheme optimizes low-capacity content sharing between D2D; however, it suffers from limited resource issues when large multimedia contents are shared.

Content forwarding strategies using SDIC-IoT must be adaptive enough to deal with the challenges of node failure in the context of large IoT deployments. ICN supports consumer mobility by design; therefore, novel techniques for IoT content producer mobility should be developed [89].

C. SDIC-IoT OpenFlow Implementations for IoT

These techniques utilize SDN OpenFlow to implement caching, routing, and traffic forwarding strategies. The generated IoT content could be cached based on optimal route selection strategies supported by the controller. OpenFlow then deploys content forwarding rules on the data plane devices and delivers content from producers to consumers. Vahlenkamp et al. [90] deployed ICN over existing ISP and data center networks using SDN. The state is stored in network elements reducing the extra time required to contact the controller. Salsano et al. [55] deployed CONET as an extension of the southbound OpenFlow protocol. Similar extensions could be developed for northbound API offering more possibilities to program SDN. Li et al. [91] deployed ICN over SDN-controlled IP networks and build a testbed for content routing and delivery. However, it suffers from scalability issues when multiple content holders request the controller for forwarding information base (FIB). Mougy [92] discussed architectural aspects of ICN over SDN by specifically focusing on the standardization of southbound OpenFlow. The authors demonstrate the need for the distributed control plane for the scalability of the SDICN architecture. Gao et al. [93] proposed a scalable architecture for intradomain communication to address controller scalability challenges. This technique uses multiple domain controllers to allocate resources at different levels. Petropoulos et al. [94] proposed a scalable and dynamic network topology bootstrapping to implement ICN over SDN. This technique extends OpenFlow to decouple content requests and forwarding phases. It forms a scalable and efficient interoperability framework, which could be deployed over SDICN for IoT service provisioning. Vassilakis et al. [95] proposed an ICN protocol for SDN-based cellular networks and develop request resolution and content routing functions. However, this approach suffers from the lack of multisource content delivery. Kalghoum and Gammar [96] proposed an NDN architecture based on SDN for network service provisioning. They devise a routing strategy by decoupling the control and data planes. They analyze the switch flow tables and develop a cache replacement policy. Ai et al. [60] proposed network recovery by alleviating network congestion using SDICN. The system continuously performs network surveillance, detects discrepancies, and performs recovery of the network. Amadeo et al. [97] proposed an architecture of the NDN data plane coupled with the SDN control plane for independent traffic routing. However, this technique lacks service allocation strategies that are crucial in the optimal delivery of content. Rastegar et al. [98] proposed optimal flow rule caching in TCAM of forwarding devices based on traffic pattern analysis. They consider a scenario where IoT devices generate bursty traffic at different time intervals and cache flow rules for efficient traffic forwarding. However, the authors predict traffic patterns for bursty traffic, whereas IoT devices often generate continuous data streams for real-time applications.

ICN-specific implementations of OpenFlow require name matching for content forwarding at the data plane. OpenFlow should be able to address flows stored at the data plane components through content names. Novel techniques for flow handling should be developed to identify the packet type, content name, and content segment.

D. SDIC-IoT Virtualization Approaches for IoT

In these techniques, NFV supports ICN by virtualizing network elements. NFV enables flexible network virtualization by offering on-demand network function deployment at general-purpose hardware. This way, commodity hardware could be programmed to offer ICN capabilities of in-network caching and name-based routing. Zhang and Zhu [99] proposed a solution for the challenge of time-sensitive data transmission using ICN virtualization. The user initially requests data from the controller. The controller then finds the data source and identifies a data delivery path for efficient delivery. However, the proposed solution suffers from network congestion issues when multiple nodes request the same content. Li et al. [100] proposed an integration of ICN with NFV to support augmented reality-based content sharing. This scheme eliminates the imbalanced distribution of content requests and available resources by virtualizing.
the resources and offering on-demand network slices to the requesting nodes. However, this technique suffers from the lack of optimization strategies. Liao et al. [101] proposed a cognitive popularity-based service distribution architecture using SDICN. They develop an SDICN-enabled edge training scheme to generate an accurate AI service model over decentralized big data samples. AI service caching and distribution schemes provide user request-oriented cognitive popularity models for content caching and distribution optimization. Xue et al. [102] split network resources using NFV and allocate on-demand resources using ICN in driving assistance systems. They propose a two-layer content caching scheme for vehicular ad hoc networks by jointly allocating caching on roadside units and vehicles. They formulate it as a nonlinear integer linear program to reduce cost and transmission delay. However, driving assistance systems require augmented reality/virtual reality (AR/VR) content, in a mobile environment, making it challenging to store content in resource-limited caching nodes. Moreover, distributed content management, in mobility scenarios, becomes a critical challenge.

Combining SDN and NFV lead to an agile infrastructure capable of orchestrating IoT services. New NFV techniques should consider ICN implementations over SDN through a certain amount of programmability of data plane elements. A novel virtualization platform based on the already available implementation of SDN could be explicitly developed, including the ICN layer capable of addressing the requirements of IoT.

E. SDIC-IoT Layered Implementations for IoT

The current implementation of OpenFlow might not provide an abstraction for ICN content. In this case, per-flow granularity might not work for content where an operator must assign different forwarding rules to different contents. The techniques discussed in this section use layered extensions in SDN to integrate ICN. Syrivelis et al. [103] proposed a layered architecture, which converts the flow concept of SDN into forwarding identifiers to address the content. However, the scalability of this approach depends on the capacity and the ability of the OpenFlow channel to respond to incoming requests. Thus, it suffers from a longer response time when the number of content requests increases. Huang et al. [104] developed a data-driven information layer in the SDN reference architecture to promote network intelligence, configuration management, deployment, and programmability. It consists of four platforms, including the measurement, data management, data analysis, and server platforms. A coordinated effort of these platforms provides higher performance and flexibility during content delivery. Saadeh et al. [32] proposed a privacy-aware IoT architecture based on the combination of ICN and SDN. This architecture distributes control functionalities over different layers, including the operational, tactical, and strategic layers. It provides more abstraction levels for node interactions in the network, supporting interoperability. However, this architecture does not consider protocol upgrades and implementation. Bracciale et al. [6] proposed a lightweight named object solution to represent IoT objects in a derived namespace using ICN. They propose an abstraction layer capable of ICN-based interactions among IoT objects. This technique provides easy discovery, programmability, and management procedures for IoT. This technique defines forwarding agent-specific rules to route traffic securely. Luo et al. [105] proposed a layered approach for ICN and SDN integration to reduce flow setup delay, flow setup request handling, and flow rule installation. Zhang and Zhu [106] proposed a layered approach using fifth-generation (5G) for optimal in-network caching, NFV to derive content, and SDN to dynamically configure wireless resource allocation strategies. This technique jointly optimizes the effective user capacity and resource allocation fairness and coordinates noncooperative naming among all users. Tariq et al. [107] proposed a layered approach to integrate ICN, SDN, and IoT to devise an energy-efficient priority-based forwarding. They use flow management techniques of SDN to efficiently forward priority-based packets. Sun and Ansari [108] proposed a publish/subscribe model to cache popular content in a broker. However, using a broker to trade caching content becomes a single point of failure, increases delays, and suffers from trust issues.

An ICN service layer could be implemented on top of the already available layered architecture of SDN. This could lead to a solid implementation of a southbound interface that provides security against data plane tampering and spoofing attacks. ICN layer extracts content metadata at the network layer without DPI. The content-centric nature of ICN supports data processing at the network layer. Content metadata contains name, length, and type, making it flexible for network operators to optimize network behaviors based on content.

F. SIC-IoT Optimal Caching Node Selection for IoT

Caching techniques support instant information access, storage, and efficient utilization of network resources. Conventional cache replacement systems send multiple client requests through the same connection to obtain the same content. However, SDN-based techniques optimize content caching strategies using centralized control of the network infrastructure. Liu et al. [109] proposed a deep learning (DL)-based content popularity prediction based on the computing resources of switches. This technique improves caching performance by predicting content popularity. Liu et al. [110] proposed cache placement for popular high-resource-intensive content. The system design includes a centralized control architecture duly equipped with a cache controller. Cache placement is formulated as an integer programming problem. An integer-based knapsack strategy solves this problem and finds optimal cache placement nodes. Al-Turjman [111] proposed a cache replacement strategy based on the periodic request popularity of on-demand requests and IoT active mode to transmit the captured content. A number is assigned to cached data based on a cached timestamp to retain the most valuable information in the cache. Nascimento et al. [112] proposed a cache replacement strategy using SDN to avoid duplicate requests in the same connection. They use the least
connection load balancing algorithm to optimize resource consumption in the network. Xu et al. [113] proposed a joint resource scheduling strategy to manage the compute and caching resources uniformly. Resource allocation provides short-term and long-term caching strategies. The short-term strategy caches the content on intermediate nodes, whereas the long-term strategy caches the AI model. Zhang et al. [114] proposed a caching strategy for resource-intensive network traffic. They transform the resource allocation problem into a dynamic integer linear programming problem. They optimize caching strategies based on video popularity and transmission delay. Zhang and Liu [115] proposed a probabilistic caching scheme for big data. They formulate the cache hit probability based on stochastic geometry and propose caching probability conversion algorithms to determine the probability of caching a particular content. However, caching big data popular content requires high capacity caching resources and higher network bandwidth to transmit big data streams.

Fig. 6 shows components of SDN-based caching. The controller selects an optimal path for caching node selection. This achieves efficient content placement and retrieval. Although caching strategies enhance content delivery, indiscriminate caching may cause content replication and localize caching decisions. Novel techniques need to be developed using SDN to identify cache status, thereby optimally avoiding content replications.

G. SDIC-IoT Traffic Management for IoT

ICN characteristics of in-network caching and name-based routing significantly change routing and transmission of Internet traffic. Therefore, traffic engineering techniques need to be revisited. Data plane routers can collect content metadata, which makes it easier to keep track of the data volume transferred on each link. Chanda et al. [116] proposed a content metadata access-based traffic engineering scheme. They observe content metadata at the network layer to optimize traffic routing among different nodes. The optimization criteria could be different based on the network operator priorities, such as caching and bandwidth utilization. Chen et al. [117] proposed a traffic engineering technique to support service provisioning in IoT. They use global name resolution and globally unique identifiers of MobilityFirst to represent IoT services. Lv et al. [118] proposed a traffic routing technique by dividing ICN topology into different communities based on the maximal tree. The information is kept in the relevant information center community, where information forwarding is performed using intracommunity and intercommunity schemes. The intracommunity traffic scheme forwards network traffic based on the same community while the intercommunity scheme routes network traffic based on social relationships. Cai et al. [119] analyzed traffic forwarding priorities based on packet delay and node load. The controller updates forwarding priorities of the matching interfaces according to the network status. The highest priority Interest packet is delivered through the matching interface. Zhang et al. [20] proposed an intelligent routing mechanism based on SDN and community division. This technique uses eigenvector centrality and label propagation for content retrieval. A controller is then deployed for each community, which efficiently routes network traffic. Ant Colony Optimization moderates intracommunity communication, and intercommunity routing is based on the frequency of interactions. Li et al. [66] proposed SDICN to cope with rapid access requests and management of heterogeneous networks. They develop a three-layer architecture. A novel onboard switch based on protocol-oblivious forwarding (POF) is designed, which performs cooperative and coded caching. Naeem et al. [120] proposed an NDN-IoT content caching and content replacement strategy. They use historical content access patterns to select and revoke caching content. However, they do not consider temporal aspects of content preferences while selecting popular content.

Traffic engineering techniques deal with traffic burstiness and improve resource utilization by finding optimal routes to reduce content delivery. Name-based data packets can be viewed as flows; this transforms the definition of flows in IP networks. Therefore, fine-grained flow management techniques are required in SDIC-IoT. Since the number of content names in ICN is enormous, cache placement, and traffic engineering are NP-hard problems. Therefore, novel techniques need to be proposed to solve traffic engineering and cache placement problems.

H. Summary and Insights

Table IV shows the taxonomy of existing SDIC-IoT schemes based on different features. Any IoT service provisioning solution should consider scalability, mobility, and virtualization. Therefore, we divide the level of these characteristics into low, medium, and high among existing approaches. QoS support and consistency of content delivery in SDIC-IoT need new solutions. Efficient middleware techniques are required that ensure expressiveness while maintaining information forwarding at a feasible rate. The relationship between resource-awareness and resource adaptation needs to be explored in future studies. The programmability characteristics of the controller can be used to control network resources intelligently. Other technologies, such as
big data, cloud computing, and wireless access networks can be integrated to make SDIC-IoT more secure and reliable. One of the critical characteristics of ICN is the ability to cache information at intermediate nodes and use it collaboratively for information sharing throughout the network. Optimizing caching through cloud replication brings benefits for efficient bandwidth utilization and reduces transmission delays.

V. USE CASES OF SOFTWARE-DEFINED INFORMATION-CENTRIC INTERNET OF THINGS

In this section, we present an overview of recently developed synergies and case studies of SDIC-IoT. A summary of these case studies is shown in Table V.

A. Smart City

IoT improves citizens’ quality of life by supporting smart city systems, including smart power control, smart parking systems, smart building management, and smart security services. Smart cities use a massive number of IoT objects which require a large namespace, interoperability, and security services. SDIC-IoT provides publish/subscribe communication model where citizens can subscribe to services they need. These services could be provided at lower latency and using a few network resources through SDIC-IoT.

He et al. [121] proposed an SDN-based solution for smart cities where content caching is enforced on edge cloudlets. Deep reinforcement learning identifies resource caching policy considering content location and consumption. The SDN controller manages the underlying network equipped with content caching and computing capacities. A cache hit occurs when a user wants to access the data, or smart city applications need to process the data. Due to the large number of network states, DL can be used to identify which particular virtualized resources are assigned to a specific user. A significant limitation of this system is the consumption of a vast amount of energy when implementing DL models on various edge servers in the network. Moreover, a long naming convention produces a higher overhead in the network. Fekih et al. [122] proposed a comparative study of using CCN...
TABLE V
OVERVIEW OF RECENTLY REPORTED SYNERGIES AND CASE STUDIES OF SDIC-IoT

| Use Case                | Research            | ICN Variant                  | Scope                                | Limitation                        |
|-------------------------|---------------------|------------------------------|--------------------------------------|-----------------------------------|
| Smart cities            | He et al. [121]     | Dynamic orchestration resources. | Deep learning based edge caching. | Energy consumption.               |
|                         | Pehin et al. [122]  | Modification of root controller. | CCN and SDN for smart cities.       | Limited evaluation.               |
|                         | Liao et al. [101]   | Contest popularity.           | Cognitive AI service distribution.   | Increased delay.                  |
| Smart homes             | Naeem et al. [123]  | Mobility and content dissemination. | ICN and IoT integration. | Communication overhead.           |
|                         | Amadeo et al. [54]  | Naming, caching, forwarding.  | Flexible and expressive naming.      | Scalability issues.               |
|                         | Burke et al. [124]  | CCN, naming, security.        | SDN-based security.                  | Limited access control scheme.    |
| Smart healthcare        | Hall et al. [125]   | NDN service caching, naming structure. | Abstract caching for IoT. | Limited mobility.                  |
| systems                 | Nour et al. [32]    | NDN for traffic engineering.  | Mapping and hand-off mobility.       | Lack of security.                 |
|                         | Zhang et al. [196]  | Data access using NDN.        | Hierarchical naming and content security. | Communication overhead.          |
| Smartgrid               | Han et al. [127]    | On-path caching routes.        | Layered publish/subscribe model.     | Implementation challenges.        |
|                         | Katsaros et al. [128] | ICM solutions.              | In-network processing.              | Lack of resource optimization.    |
|                         | Zhang et al. [125]  | ICM-based for energy management. | Informed decision making.              | Scalability issues.               |
| Smart transportation    | Li et al. [130]     | NFV/IC for service orchestration. | Content sharing. | Greater configuration cost.        |
| systems                 | Mou et al. [129]    | NDN and smart transportation systems | Secure and reliable communication. | Added complexity.                |
|                         | Amadeo et al. [3]   | ICN caching, naming, and mobility. | ICN caching. | Increased latency.                |
| AR/VR services          | Liao et al. [100]   | Massive caching for 6G AR/VR. | Cooperation among caching routers.   | Trust management overhead.        |
|                         | Zehlbian et al. [130] | ICN/CCN content popularity. | Content popularity prediction based on user historical requests. | Do not consider temporal changes in user preferences. |
|                         | Dong et al. [131]   | Caching popular content.       | Caching content on URI.              | Limited URI expressivity.         |
| Smart industries        | Li et al. [133]     | Content popularity prediction. | Aggregating user interests to determine content popularity. | Evaluation using traditional recommendation datasets. |
|                         | Duan et al. [134]   | Partitioning caching space.   | Prioritizing and buffering content.  | Missing content popularity.       |
| 6G SDIC-IoT             | Stergiou et al. [135] | Adaptive caching.            | Caching popular content.              | Higher complexity.                |
|                         | Zhao et al. [136]   | Optimizing caching efficiency. | Reinforcement for caching.           | Lack of mobility.                 |
|                         | Fadullah et al. [137] | Federated learning for cache placement. | Optimizing content popularity, distribution, and mobility. | Lack of global minimization of latency. |

and SDN for smart cities. They utilize a modified OpenFlow to realize ICN functionalities. They divide the network into subnetworks where distributed controllers manage the subnets. A root controller gathers information from distributed controllers to optimize resource utilization and cache management. A major limitation of this system is the lack of actual implementation. Although the theoretical model provides an insightful understanding of the proposed solution, the lack of experimental evaluation does not allow to conclude about the practicality of the solution. Liao et al. [101] proposed a cognitive popularity-based intelligent service delivery framework. They design an SDICN-enabled training scheme implemented on the edge cloud. The training scheme uses decentralized big data samples collected throughout the network to generate accurate models. The caching scheme automatically identifies content popularity based on the number of content requests. Although the scheme shows promising results for AI caching and distribution, training of big data models causes huge network overhead causing delays in content caching and distribution.

B. Smart Homes

Smart home services aim at improving personal lifestyle by capturing real-world user information supported by IoT. Smart home automation does not require the addresses of the hosted sensors or services; instead, it needs information about essential services. Access control needs an authorization mechanism based on the content itself rather than the location of the sensor. SDIC-IoT, as a communication model, helps faster deployment, and management of smart homes services.

Naeem et al. [123] compared the IPv6 proposal with ICN in smart homes using control, service, and complexity parameters. The authors discuss heterogeneity challenges in ICN and IoT integration. They use ICN features of naming and in-network caching for content dissemination and mobility management. The authors propose a service discovery protocol and content-based policy for user interactions with IoT devices. Amadeo et al. [54] proposed an ICN framework for smart homes containing a flexible and expressive naming scheme, which supports information exchange and configuration management. The proposed framework applies to common smart home service models, including push, pull, and multiparty. Nodes in the network implement NDN characteristics developed explicitly for data and management planes of the network. The data plane controls the content naming, caching, retransmissions, and security. The management plane configures, controls, and manages network functions on data plane elements. Burke et al. [124] identified security requirements in home automation and develop an SDN-based security framework. They integrate the trust with key traffic features and implement access control using the name of the content. Thus, services are identified and authorized using content names. The sensor analyzes the access control signature, processes it, and replies with a data packet in response.

Although existing literature focuses on SDIC-IoT in smart homes, more research is required to develop large-scale solutions. Users of unsolicited content interrupt Interest configuration as persistent Interest is not always enough to handle an increasing number of content subscribers. Moreover, secure content dissemination is necessary to enable authorized subscribers to consume the content. Privacy of user information is also highly essential as smart homes deal with highly personal user information.

C. Smart Healthcare Systems

Smart healthcare is a coordinated strategy involving sensors and actuators, wearables, smart pills, wireless networks, and mobile applications. In smart healthcare, IoT captures content...
from physical devices, in-body sensors, and wearables. This content could be cached at different network locations to gain low latency access. OpenFlow of SDIC-IoT could be combined with ICN to optimize content caching in smart healthcare systems.

Hail and Fischer [125] proposed an IoT-based architecture for healthcare service caching over NDN. They develop a special naming structure to solve heterogeneity issues and integrate different services into heterogeneous infrastructures. However, a significant limitation of this study is that it provides an abstract implementation of the caching scenarios. However, it does not consider the secure mobility of patients for seamless publish/subscribe content dissemination. Nour et al. [52] proposed an NDN for healthcare services for different traffic models supporting service management. This approach supports producer and cache mobility using mapping-based and hand-off mechanisms. However, network content can always be accessed by any node, thereby requiring the security of the communication channel. Zhang et al. [126] proposed an NDN-based distributed mobile healthcare architecture. It supports secure content distribution rather than the expensive communication medium. They use a hierarchical naming strategy independent of the content producer; it implements security on the content, making it self-secure. Saxena et al. [138] proposed an NDN-M2M communication strategy for smart healthcare. It captures and transmits content produced by sensors to the local cloud server and identifies emergency events using a Hidden Markov Model. An adaptive content-aware forwarding scheme sends events to the home cloud, and exchanges the content using the publish/subscribe strategy.

Healthcare uses different content access levels based on the sensitivity of healthcare data. Therefore, novel access control techniques should be developed with name-based access control for different stakeholders. Emergency situations require specific QoS parameters for patient treatment; thus, event tags should be generated to distinguish various medical conditions. Furthermore, reliable content delivery should be ensured from mobility-prone medical equipment.

D. Smart Grid Systems

The smart grid is an electricity supply network, which uses computing infrastructure to optimize power management [139]. They collect, analyze, control, and manage energy transmission networks to ensure fault-tolerance, service delivery, and improved QoS. Content naming characteristics of SDIC-IoT support optimized multisource content delivery, which reduces traffic congestion. It offers targeted in-network caching for efficient M2M communication. Machine-critical smart grid systems suffer from network intrusions resulting in severe economic loss. SDIC-IoT security solutions can particularly help since host locations are not exposed.

Han et al. [127] proposed on-path time synchronization for smart grids using SDIC-IoT. The SDN controller computes on-path caching routes before choosing the nearest nodes to implement accurate time synchronization. They use the publish/subscribe model to organize communication independent of locations. The proposed system consists of three planes: 1) OpenFlow ICN control network; 2) OpenFlow ICN network; and 3) edge networks. OpenFlow ICN control network replicates the control plane whereas OpenFlow ICN network corresponds to forwarding plane in traditional SDN. Edge networks cover smart grid entities capturing physical world parameters. Katsaros et al. [128] employed ICN in smart grids to solve various communication challenges, including fault management, energy management, and resource optimization. They compare the performance of ICN and host-centric solutions and report that ICN performs better in emergencies using multirendezvous point selection and in-network processing. Hail and Fischer [125] proposed an ICN-based framework for a home energy management system. This system collects data from different household elements in smart homes and performs aggregation and analysis for informed decision making. The architecture of this framework uses two layers, including the publish/subscribe layer and the security layer. The security layer is responsible for sharing security keys among subscribers. However, a significant drawback of this technique is that it does not scale with the increase in data. More investigations are needed to define M2M communication on top of content names. Moreover, secure publish/subscribe content flows between active subscribers require flexible key exchange protocols.

E. Smart Transportation Systems

Smart transportation uses communication and computation infrastructure to manage transport networks [140]. Efficient data exchange techniques should be developed to avoid network congestion [141], [142]. Thus, caching schemes using SDIC-IoT help in data availability near vehicle locations. Another scenario where SDIC-IoT could be applicable is when users subscribe to different transportation services to identify vehicle routes, delays, and congestion. These services may combine both real-time and legacy information for personal needs and fulfillment of the service demands. Transportation services could be delivered based on content name regardless of the location where these services originate. SDN deals with heterogeneous devices using OpenFlow. It hides the complexities of the mobile nodes connected using different infrastructure elements and communication protocols. Li et al. [100] proposed optimal resource allocation and service management strategies for vehicular networks using NFV and ICN. They develop a content-sharing scheme that eliminates the imbalance between content requests and delivery by allocating on-demand virtual resources. Bouk et al. [129] proposed a novel architecture for integrating NDN in smart transportation systems for mobility-aware security. They implement a secure and reliable communication mechanism among mobile devices using NDN characteristics of in-network caching and content-based security. Amadeo et al. [3] proposed a framework for integrating ICN and vehicular networks. They use ICN characteristics of naming, caching, and multicasting to overcome smart transportation challenges. Saxena and Raychoudhury [143] discussed NDN-based content routing techniques in vehicular networks for content
dissemination by implementing different IP-based forwarding schemes.

Apart from existing literature, mobility is one of the key characteristics of intelligent transportation systems, which influences QoS. Collaborative caching schemes can be developed by integrating edge computing to provide content in mobility scenarios. Moreover, adequate traffic forwarding schemes are required for vehicles using centralized management characteristics of SDN.

F. Augmented Reality/Virtual Reality Services

AR/VR applications consume higher bandwidth and require high-quality images and better video streaming to provide a satisfactory quality of experience to users. Thus, it is essential to provide content at the edge to achieve ubiquitous AR/VR capabilities. Content caches could be deployed at the edge servers to reduce transmission delays and bandwidth consumption. Content access patterns in AR/VR tend to be different from traditional video content such as YouTube. For instance, in an AR/VR-supported museum visit, locations, maps, 3-D museum models, and popular video content could be cached near the requesting users. The user equipment (UE) could also act as a content cache in AR/VR applications. For instance, consider a traveling user who possesses AR/VR content and another user who wants to access this content unavailable in the caching node. Therefore, the first user can provide the required content instead of contacting the caching node. This exchange of AR/VR content between devices reduces bandwidth consumption, decreases latency, and ensures efficient resources management.

Liao et al. [144] proposed a trust-aware content caching scheme for massive content distribution in AR/VR applications using blockchain. They ensure trust and cooperation between caching routers and propose a new consensus mechanism called Proof of Cache Offloading. Content requests between IoT and caching nodes are first verified and stored in the blockchain. They optimize caching space of routers to deliver highly popular content with lower latency. Zehtabian et al. [130] proposed popular content prediction based on historical user demands in AR/VR applications. They use ML models on user daily content consumption to predict popular AR/VR content. They use synthetic content requests based on observing a limited number of content request patterns. However, they do not use temporal aspects of user preferences to influence content popularity. Dang et al. [131] proposed caching on UE using contract theory to provide device-to-device AR/VR services. They optimize the overall utility of the network using on-device caching. However, the limited capacities of UE require the cooperation of multiple UEs for caching, making it complex to divide and distribute the content.

In the context of AR/VR, users are immersed in a ubiquitous, shared digital world where caching devices receive a massive amount of continuous data streams. The problem of how to efficiently use caching and ensure the uniformity of AR/VR content across geographically distributed caching nodes is yet to be resolved. Therefore, integrating AR/VR with distributed caching is an interesting research issue.

G. Smart Industries

Smart industries aim at improving industrial efficiency by using smart services in smart factories. Smart industries generate massive amounts of data causing higher energy consumption and longer access delays. Caching content near the edge of smart industry devices reduces energy consumption, lowers delays, reduces cost, and helps overcome environmental harm. The smart industry concept represents a paradigm shift requiring distributed intelligence, faster content acquisition, and secure data transmission.

Zahed et al. [132] proposed trusted caching for smart industries. They optimize energy consumption and security damage costs using a secure green caching heuristic. Li et al. [133] proposed a content popularity scheme based on recommendation systems that consider the diversity and temporal Interest of users. They model user behavior based on historical interaction sequences and perceive the temporal aspects of user interests for different contents. They propose a group Interest aggregation algorithm to merge the Interests of users covered by the same caching node and generate a content cache list. However, they evaluate the scheme using traditional recommendation data sets; this raises questions about its efficacy in a real-world caching environment. Duan et al. [134] proposed a green caching scheme for IIoT, which reduces energy consumption. They divide and optimally allocate caching space into two parts. One part is used to store prefetched content, and the other part is reserved as a buffer for wireless transmission. However, the authors do not consider content popularity which greatly increases the cache hit ratio and lowers redundancy in caching.

Content in smart industries is mainly used for analytics, e.g., for predictive maintenance and for improved logistics; large amounts of content are generated. In smart industries, timely content delivery is critical for industrial automation, and any discrepancy might lead to the suspension of the workflow on the shop floor. Therefore, novel caching schemes supporting large caching capacities at the edge of smart industrial equipment need to be developed for lower latency and high data volumes. Smart industry caching schemes should be fault-tolerant since a failure may lead to high-risk situations, including threat to life, equipment failure, and revenue loss.

H. 6G Software-Defined Information-Centric Internet of Things

The development of 6G will offer extremely high data rates with exceptionally low latency enabling novel ubiquitous applications and services. In the 6G era, large-scale IoT devices will demand content-intensive services and excessive resource requirements, putting unprecedented pressure on the network to meet these requirements. Novel caching techniques will be essential to cache large-scale contents to increase the efficiency of IoT and support QoS for the users. SDIC-IoT will be a key element in this environment, allowing for efficient use of caching and communication resources. SDIC-IoT will bring instant wireless connectivity and support highly digitized, content-driven, and massively connected services. The combination of AI and SDIC-IoT can support dynamic and zero-touch network orchestration,
management, and optimization, promoting the 6G evolution. AI-enabled 6G SDIC-IoT network orchestration will allow for dynamic orchestration of network architecture and slices and autonomous aggregation of different radio access technologies; the objective is to achieve interoperability in different networks and fulfill the demands of constantly changing applications and services. AI in 6G SDIC-IoT, will support autonomous management of caching demands, enable dynamic control of content caching and replacement.

Stergiou et al. [135] proposed a secure big data caching decision system in 6G smart buildings. They use an adaptive content replacement scheme to cache popular content on edge nodes. However, this scheme requires cooperation among multiple caching nodes due to resource limitations of edge devices, causing increased latency. Zhao et al. [136] proposed a reinforcement learning-based approach to optimize the efficiency of content caching across a variety of caching nodes such as base stations, local caches, and edge nodes. They consider 6G-enabled autonomous vehicles for content provisioning. However, the authors do not consider vehicle mobility, the prime concern in autonomous driving. Fadlullah and Kato [137] proposed a two-stage federated learning algorithm to predict cache placement by jointly considering content popularity, content distribution, and vehicle mobility. The learned model is transferred to UEs for edge intelligence in the considered 6G tiny cell. However, the authors do not consider minimizing global latency, which is paramount in 6G networks.

In 6G, there will be more demand for content-intensive applications enabled by massive IoT, including things, users, and vehicles. Large-scale usage of IoT will require increased communication access across geographical space. In 6G-enabled living, smart services will penetrate every corner of our lives. Therefore, there will be a higher need for content caching near the edge of IoT to provide speedy content access. 6G SDIC-IoT will extend network depth from single information transmission to global information transmission, caching, and processing, increasing network utility and improving the quality of experience. Therefore, 6G SDIC-IoT will require the following capabilities.

1) Complex communication resources, including multidimensional radio and x-haul resources.
2) Multilevel computing resources, including cloud, fog, edge, and mist and multilevel caching resources, including UEs and dedicated caching servers.
3) Novel techniques to ensure trust among multiple caching nodes when multiple caching nodes are used to cache large-scale contents.

I. Lessons Learned: Summary and Insights

Content produced by resource-intensive and delay-sensitive IoT services could be executed on edge cloudlets in the vicinity of IoT. ICN adaptation could provide benefits of implementing caching strategies on edge nodes. Collaborative caching schemes optimize resource placement, such as edge nodes storing actual content whereas routers storing data indexes. Optimization techniques could be developed to create cache indexes considering content popularity and cache-hit ratio.

VI. CURRENT ISSUES AND FUTURE RESEARCH DIRECTIONS

It is critical to identify challenges associated with different SDICN components from an IoT perspective. We discuss various issues of SDIC-IoT and highlight guidelines and future research directions. Table VI summarizes these research directions.

A. Content Naming

One of the significant issues of SDIC-IoT is that the number of content items is expected to grow by an order of magnitude greater than the number of hosts [3]. Therefore, the execution time increases for name-based routing between each node from source to destination. This increases transmission delays and decreases throughput. A future direction could be to develop naming strategies based on CONET and NDN for efficient monitoring and management of content. Service naming is also essential to distinguish services running on the same device. Content produced by IoT is often smaller in size than the content names. Thus, better optimization solutions are required to handle naming size issues in the future.

Saving numerous copies of the same content creates storage, processing, and communication overhead. Therefore, future research should develop solutions to bind names with the content to avoid content replication (e.g., ML to detect content duplication).

B. Standardization of Northbound API of SDIC-IoT

The standardization efforts for the northbound API foster the SDIC-IoT idea of bringing innovation in a software-defined way rather than making changes in the hardware components. Future research should be directed toward REST API implementations and defining constraints to communicate efficiently in SDIC-IoT. A future research direction can be limiting hardware access through applications by implementing an authorization mechanism for resource access. Another vital future research direction could be to standardize northbound API to support the portability and interoperability of services. A critical issue in this direction is to populate the vendor-specific list of functions that optimize content caching. An interesting future direction will be to intelligently prefetch content in caching nodes by capturing traffic from the controller using REST API. ML techniques can then be applied to traffic metrics to extract popular caching content.

C. IoT Mobility Management in SDICN

The concept of mobility is different from the producer and consumer perspectives. From the standpoint of producer mobility, future research should be directed to design scalable name resolution schemes with an efficient mechanism to update the entire system; the new name of the content could be synchronized throughout the network. From a consumer mobility perspective, future research should be directed to develop probabilistically optimized caching techniques to allocate the nearest caching nodes to the consumers. New FIB entries could be recorded for content producers and consumers upon receiving an Interest request. Any changes in
TABLE VI

| Current Issues                  | Causes                                                                 | Future Directions                                                                 |
|--------------------------------|------------------------------------------------------------------------|----------------------------------------------------------------------------------|
| Content Naming                 | • Greater number of content names.                                     | • Routing on specific nodes using CONET.                                          |
|                                | • Increased execution time among nodes.                                | • Optimized solutions for content naming.                                         |
|                                | • Increased delay and reduced throughput.                              | • Version naming of content.                                                      |
|                                | • Smaller data size compared to name.                                  | • Schemes for name binding with content.                                          |
| Standardization of northbound API of SDIC-IoT | • Lack of standardization of northbound API.                           | • Implementation of NFV for ISP.                                                  |
|                                | • Lack of portability and interoperability.                            | • Feature extraction using northbound API.                                        |
|                                | • Lack of vendor-specific functions for caching.                      | • Programmability of the controller.                                              |
|                                | • Hardware-based solutions.                                            | • Development of novel applications for content caching.                         |
| IoT mobility management in SDIC-IoT | • Address resolution in different networks.                           | • Re-routing incomplete requests.                                                 |
|                                | • Lack of producer mobility due to scalability.                       | • Scalable name resolution.                                                       |
|                                | • Movement of caching nodes.                                           | • Probabilistic caching node selection.                                            |
|                                | • Impact of topology changes on FIB entries.                           | • Replacing invalid FIB entries.                                                  |
| QoS support                    | • Large amounts of on-demand services.                                 | • IoT-specific QoS considerations.                                               |
|                                | • Lack of IoT-specific QoS requirements.                              | • SDN-based solutions for cache placement.                                        |
|                                | • Best effort approach of NDN.                                        | • Prioritizing QoS parameters.                                                    |
|                                | • Lack of virtualization in current networks.                         | • Routing traffic based on QoS parameters.                                        |
| Scalability of SDIC-IoT        | • Controller overload.                                                 | • Group table update.                                                            |
|                                | • Higher flow setup latency.                                           | • Distributed controllers.                                                       |
|                                | • Inefficient naming strategies.                                       | • Hybrid flow route implementation.                                              |
|                                | • Centralized control of the network.                                  | • Hierarchical naming strategies.                                                 |
| Big data challenges for SDIC-IoT | • Large amounts of data.                                              | • In-network content processing.                                                  |
|                                | • Increased computational complexity.                                  | • Analytics techniques for big data.                                              |
|                                | • Data delivery in heterogeneous networks.                             | • Network slicing using NFV.                                                      |
|                                | • QoS management in big data.                                         | • Optimal delivery path selection.                                                |
| Caching location prediction using DL | • Lack of content popularity prediction.                               | • Intelligent caching strategies.                                                 |
|                                | • Sequential network content access.                                  | • Centralized control using SDN.                                                  |
|                                | • Increased delay and poor caching.                                   | • DL based content popularity prediction.                                          |
|                                | • Local content popularity.                                           | • Caching DL neurons.                                                            |
| SDIC-IoT caching for mobile networks Using 5G | • Delay-sensitive resource-intensive tasks.                         | • Integrating SDN ICN and NFV.                                                    |
|                                | • Channel fading and inferences.                                       | • Caching content near the users.                                                 |
|                                | • Statistical and delay bounded QoS.                                   | • Interest finding using ML techniques.                                           |
|                                | • Utilization efficiency of the spectrum.                             | • SDN QoS requirements.                                                          |
| Security and privacy           | • Heterogeneity of IoT applications.                                   | • New security applications.                                                      |
|                                | • Limited resources.                                                  | • Traffic monitoring and ML.                                                     |
|                                | • Lack of device-specific implementations.                            | • Secure update patching.                                                        |
|                                | • Lack of secure communication medium.                                | • On-device security.                                                            |
| Development of novel business models | • Host-centric revenue generation.                                   | • Novel design conventions.                                                      |
|                                | • Lower consumption measurement.                                      | • Optimal content caching/discovery.                                              |
|                                | • Lack of consumption pricing.                                        | • Higher level of granularity.                                                    |
|                                | • Lack of copyright of produced content.                              | • Novel business models development.                                              |
| Development of caching marketplace | • Single point of failure platforms.                                 | • Distributed caching platform.                                                  |
|                                | • Distributed caching resources.                                      | • Blockchain-based cache resource trading.                                        |
|                                | • Different cache pricing policies.                                   | • Incentivizing content production.                                               |
|                                | • Transparent cache resource trading.                                 | • Trust between producers/consumers.                                              |
| Federated caching and learning | • Content duplication issues.                                         | • Federated caching.                                                             |
|                                | • Proprietary network infrastructure.                                 | • Optimal resource sharing.                                                       |
|                                | • Different monetization strategies.                                  | • Novel cost and payment models.                                                  |

D. QoS Support

A large amount of network traffic and demand for future smart IoT solutions (e.g., industrial control, healthcare, and smart city) instigate QoS enhancement for caching node selection, path determination, caching content selection, and mobility management. Current SDIC-IoT systems do not consider the requirements of low latency, higher delivery rate, and low loss rate. Although NDN supports nearby access, it works based on best-effort services. Future research should develop SDN-based scalable caching location selection schemes. Connection-based virtual circuit service models could be used to remove invalid FIB entries on each node using data plane feedback. Novel ML techniques could be used to select candidate caching nodes. From the consumer mobility perspective, a promising solution is to resend Interests for required content. However, this is not an optimal solution as it produces massive network overhead due to Interest looping. Future research should develop novel mapping techniques with lower communication overhead and lower internode communication delay to support streamed and subscribed data.
to find routes based on service QoS requirements. ML could be used to compare QoS requirements for specific IoT services and develop policies to meet minimum QoS requirements. Future research should focus on adaptive strategies that could adjust QoS in changing network conditions. Intelligent systems should be open and adapt to the rapidly changing service requirements. Therefore, future systems require a convergence of communication, computing, and caching for better QoS.

E. Scalability of SDIC-IoT

Scalability is one of the prime concerns in SDN-based systems. ICN also suffers from scalability issues due to a large number of content names and caching strategies. The content present in a network is far greater than the number of devices; this causes a state space explosion. Forwarding tables at the data plane have limited TCAM memory, which may overflow when defining data paths using content names. Heterogeneous IoT services require controllers to store more parameters, such as cache status awareness and updates. This may overwhelm the controller and cause a network breakdown. Therefore, a backup controller may be deployed to take control in case of the failure of the active controller. Flow setup latency requires a significant time, causing delays in traffic delivery. The SDN group table scheme could be used to install content forwarding rules for multiple contents. Future research should be directed to define hierarchical naming strategies to solve content naming challenges, providing rich semantics leading to smaller routing table sizes and reduced storage and computation requirements.

F. Big Data Challenges for SDIC-IoT

Due to the substantial volume of data, computing complexity, and rapid processing times, caching big data becomes challenging. Specifically, when content is cached on multiple caching nodes, it becomes challenging to search content across multiple nodes. Future research should focus on developing caching indexes. Novel search models could be used on the caching indexes to search the desired content efficiently. Traditional ICN could be extended to process in-network data. Future research should be directed to develop SDN-based big data analytics to proactively compute traffic routes and configure forwarding rules for large-scale traffic. Time-sensitive and bandwidth-intensive time-varying big data services pose more challenges in constrained wireless networks. In the future, novel network slicing schemes using NFV could be developed to offer dynamic reconfiguration of wireless resources and find optimal content delivery paths.

G. Caching Location Prediction Using Deep Learning

In SDIC-IoT, predicting content popularity plays a key role in establishing efficient content-caching strategies. Some previous techniques [145], [146], [147] propose that all nodes know the content popularity in the network. However, this is not always feasible since it may result in increased delay and poor caching management. Future research should be directed to develop DL techniques to predict content popularity at different nodes to design content caching strategies. SDN in this situation can be used to develop multiscale traffic prediction with higher accuracy. SDN switches at the data plane store different traffic features, including spatial, temporal, and social correlation. The DL model can understand traffic features from a global network perspective. Thus, distributed patterns of content popularity can be extracted due to the model unsupervised learning strategy, where slight variations in the network behavior do not impact the network performance. DL neurons could be easily changed on different nodes in the network using SDN. This could provide the flexibility to deploy, modify, and update DL models in the network to predict content popularity.

H. SDIC-IoT Caching for Mobile Networks Using 5G

In-network caching becomes a critical factor in 5G networks because high-speed network content could be cached to optimize resource consumption. Supporting multimedia data on 5G wireless networks brings novel challenges that were not present in traditional networks. Multimedia data transmission over 5G wireless networks is both delay sensitive and resource intensive, requiring optimized data delivery. 5G wireless channels are more time-varying due to the wireless-channel fading, outside inferences, and mobility. This involves novel statistical and delay bounded QoS ensuring mechanisms. 5G wireless networks consist of heterogeneous wireless and wire-line networks and diverse interfacing architectures. End devices connecting to these networks require cost-effective control mechanisms to optimize their overall operating performance. To address these challenges, future research could be directed to integrate SDN, ICN, and NFV to optimize the performance of 5G wireless networks. The frequently requested content can be cached near user groups with social connections. Common Interest could be identified using ML techniques. NFV techniques can divide the underlying network into multiple virtual networks to execute various applications at different networks. These virtual networks may use the high-speed 5G networks to enable parallel task execution of various applications. SDN could flexibly program the network for statistical QoS requirements of different mobile users.

I. Security and Privacy

Fig. 7 shows SDIC-IoT vulnerabilities against various attacks. Flooding attacks on communication channels between A-CPI and C-DPI can bring down the SDIC-IoT network. Attacks include distributed denial of service attacks, link flooding attacks, man-in-the-middle attacks, and privacy breaching attacks [148]. Attackers can attack communication channels between different planes. A secure medium is of utmost priority for implementing IoT applications. Most IoT devices are always connected to the Internet. The resource-limited nature of IoT makes it vulnerable to being used as an attack source. The ability to generate network traffic makes IoT a readily available source for flooding attacks. The application plane of SDIC-IoT could be used to develop novel applications against security and privacy-breaching attacks. The capability of the controller to extract network statistics could be used to develop pattern analysis-based security solutions [149], [150], [151].
Moreover, IoT devices use different operating systems, making it challenging to patch security updates on all devices.

J. Development of a Caching Marketplace

An important research direction is developing a marketplace that ensures a secure exchange of cached content [152], [153]. Content caching resources are distributed and allocated dynamically; thus, pricing policies differ from legacy systems. The caching rewards for all stakeholders should be balanced. Therefore, an incentive mechanism is desired to attract caching nodes to participate in budget-limited caching applications. Existing solutions utilize a centralized approach to attract content producers, assess content quality, and determine rewards [154], [155]. Centralization allows using global optimization algorithms to address caching problems. However, it suffers from several issues, including a single point of failure and a lack of trust between content producers and consumers. Therefore, developing a decentralized marketplace for cache resource trading is an important future direction. Blockchain could be used to process caching transactions and distribute caching rewards transparently.

K. Development of Novel Business Models

In the traditional host-centric networking paradigm, service requests must be processed directly by content producers; this gives full control of the content to producers. However, in SDIC-IoT, the network layer caches the content on different nodes regardless of the producer availability; the decoupling of content from producers allows for open content distribution. Thus, existing host-centric revenue generation business models cannot be applied in this context. Revenue generation elements in SDIC-IoT, for optimal cache storage and content discovery, might require the development of novel business models. In the future, SDN could be vital in developing a completely distributed platform using multiple controllers at different network locations. Controllers can track content consumption at a higher level of granularity. Another important research direction is to implement copyright at different content publishing, delivery, and consumption levels. However, this should not invade the privacy of content consumers.

L. Federated Caching

Content duplication becomes a significant issue in IoT environments when content is cached in different locations in the network. In federated caching, the content is cached and shared among multiple network operators [156]. It works by using an optimal resource-sharing mechanism in terms of cost and resource management. Federated caching can achieve a higher cache hit ratio and reduce cost. However, different communication networks may have different proprietary network equipment and architectures; this may require novel cost and monetization models considering complex access patterns. There is a lack of architectures that present cost and payment models in federated caching. Therefore, future research needs to explore interoperability among different network architectures for efficient federated caching implementations. Moreover, novel pricing models for content payment need to be developed in federated caching.

VII. Conclusion

IoT uses communication technologies to connect physical objects with the Internet to realize a wide variety of services. IoT services challenge current networks due to heterogeneity, a higher number of devices, and diverse implementations. SDN and ICN are two complementary technologies that tackle different networking challenges to deliver IoT services. SDN ensures the programmability of the network by implementing virtualized network functions over general-purpose hardware. It gives the flexibility of implementing ICN functionality of in-network caching and name-based routing in an SDN-controlled environment. Thus, the SDICN paradigm provides an effective solution to support IoT services.

This article covered guidelines to integrate SDIC-IoT for IoT service provisioning. It analyzed the motivation of SDIC-IoT integration and did show its benefits for IoT service provisioning. We discussed critical requirements in designing the SDIC-IoT architecture while considering resource-limitation constraints, the content-centric nature, and the massive amounts of IoT devices. We described the taxonomies of existing approaches from several aspects, such as caching, mobility, traffic engineering, implementation, and virtualization. Furthermore, we discussed several use case implementations of SDIC-IoT that provide guidelines for future SDIC-IoT systems. Finally, we presented current challenges that impact SDIC-IoT implementations and future research directions.

REFERENCES

[1] W. Rafique, L. Qi, I. Yaqoob, M. Imran, R. U. Rasool, and W. Dou, “Complementing IoT services through software defined networking and edge computing: A comprehensive survey,” IEEE Commun. Surveys Tuts., vol. 22, no. 3, pp. 1761–1804, 3rd Quart., 2020.

[2] B. Nour, S. Mastorakis, R. Ullah, and N. Stergiou, “Information-centric networking in wireless environments: Security risks and challenges,” IEEE Wireless Commun., vol. 28, no. 2, pp. 121–127, Apr. 2021.
[3] 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.

[4] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, “Internet of Things (IoT) communication protocols,” in *Proc. 8th Int. Conf. Inf. Technol. (ICT)*, 2017, pp. 685–690.

[5] W. Bekri, R. Jmal, and L. C. Fourati, “Internet of Things management based on software defined networking: A survey,” *Int. J. Wireless Inf. Netw.*, vol. 27, pp. 385–410, Jul. 2020.

[6] L. Bracciole, P. Loret, A. Detti, R. Paolillo, and N. B. Melazzi, “Lightweight named object: An ICN-based abstraction for IoT device programming and management,” *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5029–5039, Jun. 2019.

[7] B. Nour, K. Sharif, F. Li, H. Moungla, and Y. Liu, “A unified hybrid information-centric networking scheme for IoT applications,” *Comput. Commun.*, vol. 150, pp. 103–114, Jan. 2020.

[8] S. Arshad, M. A. Azam, S. H. Ahmed, and J. Loo, “Towards information-centric networking (ICN) naming for Internet of Things (IoT) the case of smart campus,” in *Proc. Int. Conf. Future Netw. Distrib. Syst.*, 2017, pp. 1–6.

[9] S. Mastorakis, A. Milbaa, J. Lee, and S. Misra, “Icedge: When edge computing meets information-centric networking,” *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4203–4217, May 2020.

[10] Y. Thomas, N. Fiotiou, S. Toumpis, and G. C. Polyzos, “Improving mobile ad hoc networks using hybrid IP-information centric networking,” *Comput. Commun.*, vol. 156, pp. 25–34, Apr. 2020.

[11] B. K. Mohanta, D. Jena, U. Satapathy, and S. Patnaik, “Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology,” *IEEE Internet Things J.*, vol. 11, Sep. 2020, Art. no. 100227.

[12] B. Wissingh, C. A. Wood, A. Afanasiev, L. Zhang, D. Oran, and C. F. Tschanz, “Information-centric networking (ICN): Content-centric networking (CCN) and named data networking (NDN) terminology,” RFC 8793, 2020, pp. 1–17.

[13] S. Tayeb, S. Latifi, and Y. Kim, “A survey on IoT communication and computation frameworks: An industrial perspective,” in *Proc. IEEE 7th Annu. Comput. Commun. Workshop Conf.* (CCWC), 2017, pp. 1–6.

[14] Q.-Y. Zhang, X.-W. Wang, M. Huang, K.-Q. Li, and S. K. Das, “Software defined networking meets information centric networking: A survey,” *IEEE Access*, vol. 6, pp. 39547–39563, 2018.

[15] C. Gündoğan, P. Kietzmann, T. C. Schmidt, and M. Wählisch, “Information-centric networking for the industrial Internet of Things,” in *Proc. Wireless Netw. Ind. IoT*, 2021, pp. 171–189.

[16] S. Arshad, M. A. Azam, M. H. Rehmanni, and J. Loo, “Recent advances in information-centric networking-based Internet of Things ICN-ICoT,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2128–2158, Apr. 2019.

[17] F. Al-Turjman, E. Ever, Y. B. Zikria, S. W. Kim, and A. Elmahgoubi, “A data-oriented (and beyond) network architecture,” *Proc. 6th SDN-ICN architecture design for the Internet of Things*, in *Proc. 6th Int. Conf. Softw. Defined Syst. (SDS)*, 2019, pp. 96–101.

[18] P. P. Ray and N. Kumar, “SDN/NFV architectures for edge-cloud oriented IoT: A systematic review,” *Comput. Commun.*, vol. 169, pp. 129–153, Mar. 2021.

[19] R. Jmal and L. C. Fourati, “Emerging applications for future Internet approach based-on SDN and ICN,” in *Proc. IEEE/ACS 14th Int. Conf. Comput. Syst. Appl. (AICCSA)*, 2017, pp. 208–213.

[20] K. Kaur and R. Kapoor, “MCPCN: Multi-hop clustering protocol using cache nodes in WSN,” *Wireless Pers. Commun.*, vol. 109, no. 3, pp. 1727–1745, 2019.

[21] O. Serhane, K. Yahyaoui, B. Nour, and H. Moungla, “A survey of ICN content naming and in-network caching in 5G and beyond networks,” *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4081–4104, Mar. 2021.

[22] M. Amadeo, C. Campolo, G. Ruggeri, A. Molinaro, and A. Iera, “Understanding name-based forwarding rules in software-defined named data networking,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[23] D. R. Cheriton and M. Gritter, “A scalable deployable NAT-based Internet architecture,” *Stanford Comput. Sci., Stanford, CA, USA*, Rep., 2000. [Online]. Available: http://ceng.anadolu.edu.tr/skinlan/BIL555.iceri2000-Triad.pdf

[24] “National science foundation of the USA.” Accessed: Jun. 6, 2021. [Online]. Available: http://www.nets-fia.net/

[25] T. Koponen et al., “A data-oriented (and beyond) network architecture,” in *Proc. Conf. Appl. Technol. Archit. Protocols Comput. Commun.*, 2007, pp. 191–192.

[26] A. Köpsel and H. Woesner, “OFELIA—Pan-European test facility for OpenFlow experimentation,” in *Proc. Eur. Conf. Service Based Internet*, 2011, pp. 311–312.

[27] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, “A survey of information-centric networking,” *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, Jul. 2012.

[28] G. Carofiglio, M. Gallo, and L. Muscariello, “ICP: Design and evaluation of an interest-centric protocol for content-centric networking,” in *Proc. IEEE INFOCOM Workshops*, 2012, pp. 304–309.

[29] A. Detti, N. B. Melazzi, S. Salsano, and M. Pomposini, “CONET: A content centric inter-networking architecture,” in *Proc. ACM SIGCOMM Workshop Inf. Centric Net.*, 2011, pp. 50–55.

[30] M. Amadeo, A. Molinaro, and G. Ruggeri, “E-CHANET: Routing, forwarding and transport in information-centric multihop wireless networks,” *Comput. Commun.*, vol. 36, no. 7, pp. 792–803, 2013.

[31] G. Xylomenos et al., “A survey of information-centric networking research,” *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024–1049, 2nd Quart., 2013.

[32] D. Trossen, M. J. Reed, J. Riihijärvi, M. Georgiades, N. Fotiou, and G. Xylomenos, “IP over ICN-the better IP?” in *Proc. EUC Conf. Netw. Commun. (EuCNC)*, 2015, pp. 413–417.

[33] A. Rahman, D. Trossen, D. Kutscher, and R. Ravindran, “Deployment considerations for information-centric networking (ICN),” *ICNRF, IETF, Fremont, CA, USA*, Rep., 2018. [Online]. Available: https://tools.ietf.org/html/draft-irtf-icnrg-deploymentguidelines-04
[49] Z. Sheng, S. Yang, Y. Yu, A. V. Vasilakos, J. A. McCann, and K. K. Leung, “A survey on the IETF protocol suite for the Internet of Things: Standards, challenges, and opportunities,” *IEEE Wireless Commun.*, vol. 20, no. 6, pp. 91–98, Dec. 2013.

[50] S. Debroy, P. Samanta, A. Bashir, and M. Chatterjee, “SpEED-IoT: Spectrum aware energy efficient routing for device-to-device IoT communication,” *Future Gener. Comput. Syst.*, vol. 93, pp. 833–848, Apr. 2019.

[51] S. L. Levin and S. Schmidt, “IPv4 to IPv6: Challenges, solutions, and roadmaps,” *Telecomm. Policy*, vol. 38, no. 11, pp. 1059–1068, 2014.

[52] B. Nour, K. Sharif, F. Li, and H. Moungla, “A distributed ICN-based IoT network architecture: An ambient assisted living application case study,” in *Proc. IEEE Global Commun. Conf.*, 2017, pp. 1–6.

[53] A. Ravi, P. Ramathan, and K. Swalldingam, “Integrated network coding and caching in information-centric networks: Revisiting paver- cising in the ICN framework,” *Photon. Netw. Commun.*, vol. 30, no. 3, pp. 416–427, 2015.

[54] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, “Information centric networking in IoT scenarios: The case of a smart home,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2015, pp. 648–653.

[55] S. Salsano, N. Blefari-Melazzi, A. Detti, G. Morabito, and L. Veltri, “Information-centric networking over SDN and OpenFlow: Architectural aspects and experiments on the OFELIA testbed,” *Comput. Netw.*, vol. 57, no. 16, pp. 3207–3221, 2013.

[56] R. U. Rasool, K. Ahmed, Z. Anwar, H. Wang, U. Ashraf, and W. Rafique, “CyberPulse+i+: A machine learning-based security framework for detecting link flooding attacks in software defined networks,” *Int. J. Intell. Syst.*, vol. 36, no. 8, pp. 3852–3879, 2021.

[57] W. Rafique, X. He, Z. Liu, Y. Sun, and W. Dou, “CFADefense: A security solution to detect and mitigate crossfire attacks in software-defined IoT-edge infrastructure,” in *Proc. IEEE 21st Int. Conf. High Perform. Comput. Commun. IEEE 17th Int. Conf. Smart City IEEE 5th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, 2019, pp. 500–509.

[58] Z. Eghbali and M. Z. Lighvan, “A hierarchical approach for accelerating IoT data management process based on SDN principles,” *J. Netw. Comput. Appl.*, vol. 181, May 2021, Art. no. 103027.

[59] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, “Soft-WSN: Software-defined WSN management system for IoT applications,” *IEEE Syst. J.*, vol. 12, no. 3, pp. 2074–2081, Sep. 2018.

[60] Z.-Y. Ai, F. Song, and X. Wang, “Combining SDN and ICN for network survivability improvement,” *J. Internet Services Inf. Secur.*, vol. 8, no. 1, pp. 18–30, 2018.

[61] M. Arumaiturai, J. Chen, E. Monticelli, X. Fu, and K. K. Ramakrishnan, “Exploiting ICN for flexible management of software-defined networks,” in *Proc. 1st ACM Conf. Inf. Centric Netw.*, 2014, pp. 107–116.

[62] F. Ren, Y. Qin, H. Zhou, and Y. Xu, “Mobility management scheme based on software defined controller for content-centric networking,” in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2015, pp. 149–150.

[63] Z. Liu, J. Zhu, C. Pan, and G. Song, “Satellite network architecture design based on SDN and ICN technology,” in *Proc. 8th Int. Conf. Electron. Inf. Enter. Commun. (ICEIEC)*, 2018, pp. 124–131.

[64] J. Liu, S. Guo, Y. Shi, L. Feng, and C. Wang, “Decentralized caching framework toward edge network based on blockchain,” *IEEE Internet Things J.*, vol. 7, no. 9, pp. 9158–9174, Sep. 2020.

[65] F. Ren, Y. Qin, H. Zhou, and Y. Xu, “Mobility management scheme based on software defined controller for content-centric networking,” in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2016, pp. 193–198.

[66] Y. Zhang and Y. Wang, “SDN based ICN architecture for the future integration network,” in *Proc. 16th Int. Symp. Commun. Inf. Technol. (ISCIT)*, 2016, pp. 474–478.

[67] C. Xing, K. Ding, C. Hu, M. Chen, and B. Xu, “SD-ICN: Toward wide area deployable software defined information networking,” *KSII Trans. Int. Inf. Syst.*, vol. 10, no. 5, pp. 2267–2285, 2016.

[68] R. Ravindran, A. Chakraborti, S. O. Amin, A. Arzign, and G. Wang, “5G-ICN: Delivering ICN services over 5G using network slicing,” *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 101–107, May 2017.

[69] A. He et al., “Resource estimation in software defined and information-centric vehicular networks with mobile edge computing,” in *Proc. IEEE 86th Veh. Technol. Conf. (VTC-Fall)*, 2017, pp. 1–5.

[70] G. M. Leal, I. Zacarias, J. M. Stocchero, and E. P. de Freitas, “Empowering command and control through a combination of information-centric networking and software defined networking,” *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 48–55, Aug. 2019.

[71] Y. Ban, “Software-defined information-centric networking based exercise intensity evaluation of volleyball player: An efficient convolutional neural network method,” *Internet Technol. Lett.*, vol. 4, no. 5, p. e246, Sep. 2021, doi: 10.1002/itl2.246.

[72] Z. Xiao and Q. Zhu, “Mobility-aware and interest-predicted caching strategy based on IoT data freshness in D2D networks,” *IEEE Internet Things J.*, vol. 8, no. 7, pp. 6024–6038, Apr. 2020.

[73] D. Raychoudhuri, K. Nagaraja, and A. Venkatakrishni, “MobilityFirst: A robust and trustworthy mobility-centric architecture for the future Internet,” *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 16, no. 3, pp. 2–13, 2012.

[74] M. Vahrenkamp, F. Schneider, D. Kutschke, and J. Seedorf, “Enabling ICN in IP networks using SDN,” in *Proc. 21st IEEE Int. Conf. Netw. Protocols (ICNP)*, 2013, pp. 1–2.

[75] P. Li, W. Muqing, W. Ning, and L. Hongbao, “Supporting information-centric networking in SDN,” *Int. J. Future Comput. Commun.*, vol. 4, no. 6, pp. 386, 2015.

[76] A. E. Mougy, “On the integration of software-defined and information-centric paradigms,” in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, 2015, pp. 105–110.
