IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 23, 2018, accepted July 3, 2018, date of publication July 12, 2018, date of current version August 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2855135

Software Defined Networking Meets Information Centric Networking: A Survey

QING-YI ZHANG[1], XING-WEI WANG[1], MIN HUANG[2], KE-QIN LI[3], (Fellow, IEEE), AND SAJAL K. DAS[4], (Fellow, IEEE)

[1]College of Computer Science and Engineering, Northeastern University, Shenyang 110169, China
[2]State Key Laboratory of Synthetical Automation for Process Industries, College of Information Science and Engineering, Northeastern University, Shenyang 110819, China
[3]Department of Computer Science, State University of New York, New Paltz, NY 12561, USA
[4]Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409, USA

Corresponding authors: Xing-Wei Wang (wangxw@mail.neu.edu.cn) and Min Huang (mhuang@mail.neu.edu.cn)

This work was supported in part by the Major International (Regional) Joint Research Project of NSFC under Grant 71620107003, in part by the National Science Foundation for Distinguished Young Scholars of China under Grant 71325002, in part by the National Natural Science Foundation of China under Grant 61572123, in part by the Program for Liaoning Innovative Research Term in University under Grant LT2016007, and in part by the MoE and China Mobile Joint Research Fund under Grant MCM20160201.

ABSTRACT Information centric networking (ICN) and software-defined networking (SDN) are two emerging networking paradigms that promise to solve different aspects of networking problems. ICN is a clean-slate design for accommodating the ever increasing growth of the Internet traffic by regarding content as the network primitive, adopting in-network caching, and name-based routing, while SDN focuses on agile and flexible network management by decoupling network control logic from data forwarding. ICN and SDN have gained significant research attention separately in the most of the previous work. However, the features of ICN have profound impacts on the design and operation of SDN, such as in-network caching and data-centric security. Conversely, SDN provides a powerful tool for experimenting and deploying ICN architectures and can greatly facilitate ICN functionality and management. In this paper, we point out the necessity of surveying the scattered works on integrating SDN and ICN (SD-ICN) for improving operational networks. Specifically, we analyze the points of SD-ICN strengths and opportunities, and discuss the SDN enablers for deploying ICN architectures. In addition, we review and classify the recent work on improving the network management by SD-ICN and discuss the potential security benefits of SD-ICN. Finally, a number of open issues and future trends are highlighted.

INDEX TERMS Future Internet architecture, ICN, SDN, SD-ICN.

I. INTRODUCTION

The Internet pioneers designed today's Internet architecture for traditional applications, such as file transfer and e-mail. However, being widely used by general public, at present, the Internet actually carries diverse applications and ever increasing traffic, which requires more complex network services and higher demand on transmission capability than ever before. Plenty of new network protocols are continuously patched in existing protocol stack which makes too many additional configurations. Under such background, researchers begin to rethink the design and management of the Internet. In this context, Software-Defined Networking (SDN) [1] and Information-Centric Networking (ICN) [2] are proposed.

Content distribution and retrieval dominates the Internet traffic today. ICN conforms to this trend by treating content as the network primitive. It replaces the IP packet with a named content that can be transmitted across network. In this way, ICN decouples content from its location and moves network away from the host-centric communication paradigm, shifting the users' focus from where information comes to what content they want. Furthermore, by adopting name-based content identification, ICN provides a direct access of in-network caches. Although conceptually simple, realizing ICN as an overlay on the existent network leads to an excessive increased complexity of IP routing, thus it is hard to fully exploit ICN functionality in legacy network.

SDN is characterized by the separation of data plane and control plane. The network control logic is implemented via a (logically) centralized controller. By introducing programmability to network, SDN provides a great opportunity to develop, deploy and improve network services and even

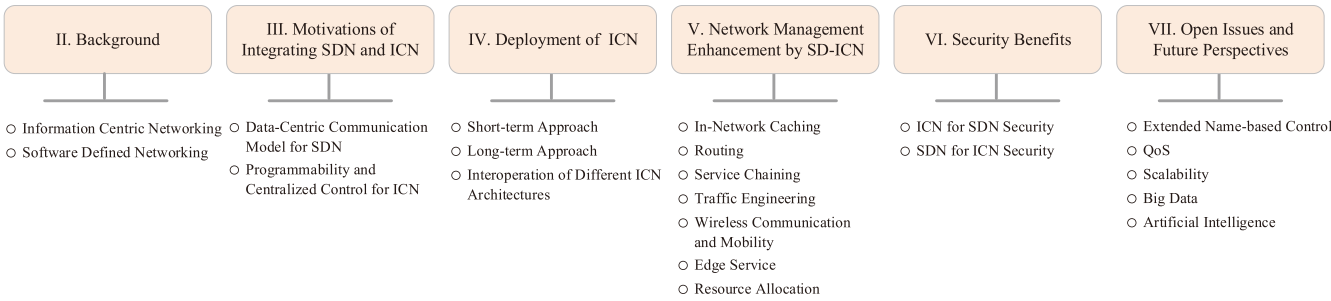


FIGURE 1. The roadmap of this survey.

new network architectures agilely and flexibly compared with traditional networking paradigms. In this way, SDN enables a continuous evolution of network management.

Although a large number of excellent work has been proposed on ICN and SDN, they are addressed separately in most of the previous research. However, as two promising future networking paradigms, they are not competing but complementary, tackling different networking problems and benefiting each other. In this context, research communities get started on integrating of SDN and ICN (SD-ICN).

The benefits of SDN to the current Internet architecture, i.e., being a powerful managing and monitoring tool, centralized control and global view of the network, also apply to ICN. With the help of SDN, ICN can handle efficient data distribution in partially upgraded network, simplify the deployment and smooth the transition phase [3]. Compared with original ICN, SD-ICN can realize the globally optimal resource allocation via the centralized controller. Therefore, ICN's name-based routing and in-network caching will be potentially facilitated. In turn, SDN is able to take advantage of in-network caching capability and quickly adapt to novel data-centric advantages, e.g., data-centric security model. Furthermore, ICN changes the end-to-end communication paradigm and has profound impact on the design and operation of SDN. For example, the name-based control provides SDN a finer-grained control than flow-based control [4].

Although some surveys have been published for ICN and SDN, most of them focus on issues such as mobility [5], energy efficiency [6], caching and forwarding [7] for ICN, traffic engineering [8], security [9] and resilience support [10] for SDN. However, many excellent research dedicated to integrate SDN and ICN are not surveyed. To the best of our knowledge, only Jmal and Fourati [11] go a step further by proposing a survey on this topic. However, this survey is limited in terms of scope, i.e., it does not provide a comprehensive aspects of SD-ICN. They focus more on reviewing ICN caching mechanism and deployment based on SDN but fall short on the analysis of different implementing schemes. Furthermore, they lack a thorough discussion of the enhancement that SD-ICN will bring to network management, such as traffic engineering, routing and service chaining. In this paper, we present a comprehensive literature survey of the emerged SD-ICN paradigm, from its motivations, deployment, network management enhancement, security as well as future perspectives. This survey can help to understand how to make full use of SDN to improve the performance of ICN, and how to make SDN work more effectively by introducing ICN into SDN. The roadmap of this survey is depicted in Fig. 1.

In the rest of this paper, we first present a brief introduction to ICN and SDN basics in Section II. Then, we present motivations of integrating SDN and ICN in Section III. In Section IV, we summarize the ICN deployment technologies based on SDN. In Section V, we discuss the network management enhancement by SD-ICN. Security benefits are presented in Section VI. In Section VII, we highlight several open issues and future perspectives for further research. Finally, conclusions are drawn in Section VIII.

II. BACKGROUND

In this section, we briefly introduce the concept of ICN and SDN.

A. INFORMATION CENTRIC NETWORKING

In ICN, every piece of content is identified by an unique name for name-based routing. The named content can be cached in intermediate network nodes for potential requests. In addition, ICN can secure content itself rather than a specific device. There are some well-known ICN projects. The Scalable and Adaptive Internet Solutions (SAIL) (previously named 4WARD) [12], [13], Publish/Subscribe Internet Technology (PURSUIT) (previously named PSIRP) [14], [15], Content Mediator Architecture for Content-Aware Networks (COMET) [16] and CONVERGENCE [17] are funded by EU Framework Programme 7 (FP7), and Content Centric Networking (CCN)/Named Data Networking (NDN) [18]–[20] is funded by NSF FIA program. There are also some other purpose-designed ICN architectures, such as GreenICN [21] for energy efficiency and MobilityFirst [22] for mobile applications. Table 1 summarizes the main ICN projects.

Although the above ICN architectures have their own features with significant differences from the others, they all obey the same fundamental principles. We briefly summarize these key commonalities as follows.

TABLE 1. Main ICN projects.

Institution	Project	Main Application Scenarios
USA-NSF	CCN/NDN	General data dissemination Internet of Things (IoT) Vehicular Ad-Hoc Networks (VANETs)
	MobilityFirst	Mobile data dissemination
EU-FP7	PSIRP/PURSUIT	General data dissemination
	4WARD/SAIL/NetInf	General data dissemination
	COMET	General data dissemination
	CONVERGENCE/CONET	General data dissemination
EU&Japan-FP7	GreenICN	Energy-efficient data dissemination

• Name-based addressing

Naming is essential for shifting the network paradigm from host-centric to information-centric. For all ICN architectures, the naming conventions, structure of names (hierarchical or flat) and their semantics, has immediate impact on architecture designs. However, they all adopt name to retrieve content rather IP address.

• Publish/subscribe paradigm

ICN decouples the content providers and the content consumers spatially, i.e., the consumer who requests a specific content ignores the provider's location and vice versa [23]. Although ICN is not identical to pub/sub system, they indeed share some similarities. In fact, ICN can be seen as a pub/sub system at network layer. For example, PSIRP/PURSUIT adopts pub/sub as its two basic operations, i.e., PUBLISH and SUBSCRIBE. Similarly, NDN completes content retrieval with a pair of primitives, i.e., Interest and Data packets.

• In-network caching

Decoupling content from location provides a significant feature, i.e., the transferred data can be reused by other requesters (unlike IP packet), making in-network caching possible. Ideally, content replica is cached at the edge of the network which is close to the requesters. Thus network operators are able to make full use of this feature to save bandwidth and alleviate the load of network core.

• Data-centric security

In all ICN architectures, content is authenticated to prevent malicious access without permission. By this way, ICN focuses on the security of the content itself (e.g., information confidentiality and privacy) rather than the safety of its traversed links or devices. Furthermore, ICN allows any network entities to verify the content.

ICN communication is consumer-driven with two basic types of packets: Interest and Data (without loss of generality, we refer to NDN architecture). As described in Fig. 2, consumer sends an Interest to find the content, and ICN routes the Interest (according to Forwarding Information Base (FIB)) towards the content producer(s) using the name that is contained in the Interest. Once the producer or the intermediate node that caches the content copy (in Content Store (CS)) receives the Interest, the node wraps the content in a Data packet and delivers it back to the consumer along the reverse path which the Interest traversed, and the content

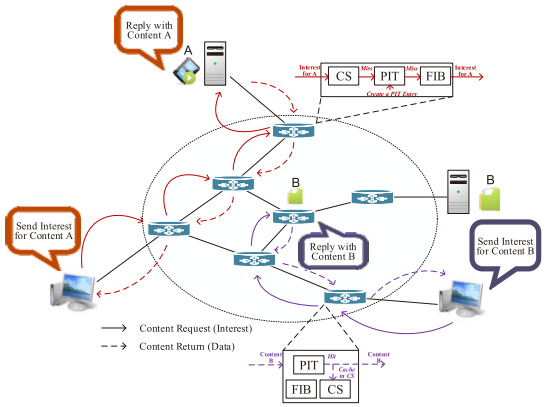


FIGURE 2. Interest/data exchange in ICN.

can be leaved along this delivery path (in Pending Interest Table (PIT)) for future Interests.

B. SOFTWARE DEFINED NETWORKING

In conventional networks, network functions, e.g., routing and traffic engineering, are integrated into a specialized hardware which hosts a specialized operating system, i.e., network devices are vertically integrated. Such a static and ossified architecture is ill-suited to the dynamic resource demands of today's data centers, cloud computing, and heterogeneous network environment. Besides, testing and applying innovative functions becomes very complex and hard.

SDN promises to realize flexible network management and resource configuration by breaking up network vertical integration. This concept is quickly obtained great attention from Google, Facebook, Verizon, Deutsche Telekom, etc. They jointly fund Open Networking Foundation (ONF) [24] with the main goal of promotion and adoption of SDN through open standards development. The basic SDN architecture decouples application, control and data, as depicted in Fig. 3(a). The application plane includes a variety of network applications. In control plane, SDN controller(s), as the network "brain", is (are) mainly responsible for underlying resource scheduling and network state management. The data plane consists of the programmable network devices that only handle data forwarding. These three planes intercommunicate via open APIs, i.e., northbound APIs, e.g., RESTful APIs [25] and southbound APIs, e.g., OpenFlow [26] (Fig. 3(b)).

In conclusion, SDN brings at least four advantages. Firstly, network intelligence can be implemented by software. The network control is completed by Network Operating System (NOS), which can be deployed on generic servers, e.g., X86 platform. The NOS abstracts physical network resource into logical resource that can be configured by upper software applications. Secondly, network operators can configure a variety of network parameters, such as Quality of Service (QoS) and security policy, into network in nearly real-time, and thus can respond to new business requirement quickly. Thirdly, the underlying hardware devices only focus

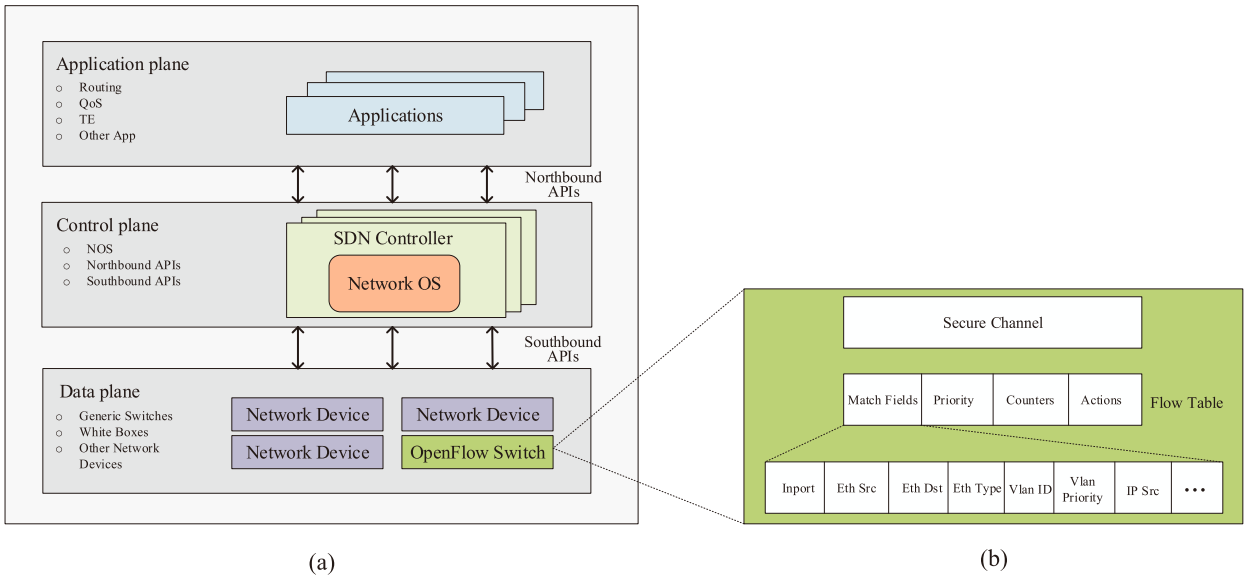


FIGURE 3. (a) SDN architecture. (b) OpenFlow protocol.

on forwarding and storage capacity rather than service logic. Thus, the general-purpose hardware can be adopted. Finally, the network has been turned into a horizontal structure, i.e., various applications, NOS and hardware devices can evolve separately.

III. MOTIVATIONS OF INTEGRATING SDN AND ICN

A. DATA-CENTRIC COMMUNICATION MODEL FOR SDN

The centralized SDN controller has the global view of network status and powerful control of data transfer. SDN promises to simplify network management. However, the current SDN architecture is still based on TCP/IP protocols and thus the longstanding problems of TCP/IP still exist. In fact, it is worth to note that SDN just provides a tool to solve network management problem [27].

Introducing ICN concept into SDN can help overcome the problems that brought by host-centric networking (TCP/IP). The shifting from "where" to "what" brings SDN new opportunities. From the view point of SDN, the integration of ICN and SDN can certainly provide many benefits.

Firstly, SDN can make full use of ICN in-network caching. In particular, data can be obtained from intermediate nodes or original content providers. By adopting name-based data retrieval, SDN is able to exploit in-network caches and multipath routing. Since in-network caching and innate multicast capability of ICN decreases the overall data transmission, the higher network throughput is allowed. Besides, SDN control plane has the ability to monitor in-network caching status, which could be used to optimize the routing of content requests.

Secondly, the ability to directly identify content enables some new data-centric services for SDN. Currently, OpenFlow focuses on establishing flow-based end-to-end communication but ignores the content that goes through. Without further process, for example, Deep Packet Inspection (DPI), different types of content are able to get the same forwarding treatment if their port number, source and destination are identical (i.e., per-flow based forwarding) [28]. This is usually suboptimal. In the context of ICN, content metadata can be easily extracted at the network layer without DPI. As content metadata describes the content length, the type, etc., thus network operators can optimize network behaviors according to such metadata [29].

Thirdly, ICN protects content itself other than communication channels or specific devices. Due to the vulnerability of host-centric security, SDN controller and its northbound and southbound interfaces face serious threats, such as controller spoofing and hijacking. ICN's data-centric security model can eliminate the threats that plague host-centric network. Hence, ICN provides an alternative solution for resolving SDN security problem.

B. PROGRAMMABILITY AND CENTRALIZED CONTROL FOR ICN

The programmability of SDN brings ICN convenient deployment and innovation. SDN is about understanding and managing network as a unified view which can promote ICN's capability of caching, routing, etc. Specifically, ICN can be improved by utilizing SDN from the following aspects.

Firstly, SDN provides ICN with powerful control over the distributed ICN elements. In the context of SDN, through well-defined interfaces, network operators can configure routing strategies and cache replacement policies of the underlying devices flexibly and timely. For different domains, SDN can provide the customized services for various types of contents under different namespaces. In addition, the centralized management, distribution and revocation of keys can strengthen ICN security further.

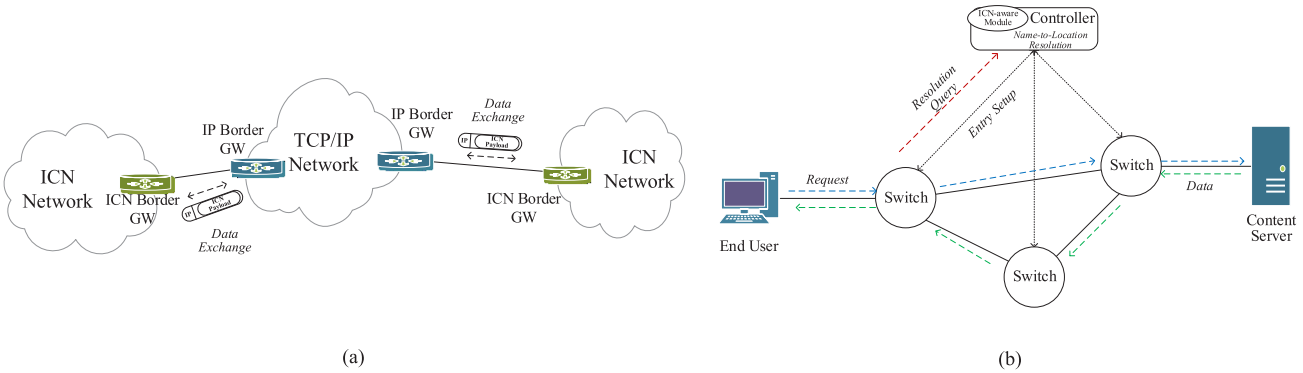


FIGURE 4. Short-term ICN deployment: (a) IP-tunneling approach. (b) Resolution-based approach.

Secondly, SDN enables ICN to be deployed conveniently within the framework of SDN and keep ICN evolving. SDN pursues an approach to opening up agile network innovation via programming interfaces which support programmable control and data planes. Although the early versions of SDN protocols focus on control plane programmability, recent work on SDN is exploring the evolution of SDN protocols, e.g., OpenFlow, to support a wider range of data plane functions [30]. The programmable data and control planes support the construction of the customized functionalities on individual network nodes. ICN definitely can benefit from SDN's programmability, since its unique features (e.g., name-based forwarding) can be implemented in general-purpose devices and thus be deployed easily.

Thirdly, the in-network caching has been regarded as one of the salient features of ICN architectures and some of the claimed performance advantages originate from the widespread caches. However, the indiscriminate caching mechanism has been questioned and argued that it is not optimal as it may imply high content replication [31], locality of caching decision [32], etc. Instead, SDN can take advantage of the centralized control to monitor the network caching status. The control plane is able to be aware of topology and cache related information, which can help optimize resource allocation. Moreover, by leveraging SDN, on-path/off-path caching and even the hybrid approach can be employed to avoid redundant duplications and the overall cache hit ratio can be significantly increased [33].

IV. DEPLOYMENT OF ICN

ICN architectures require a significant upgrade to the existing network infrastructure that almost all network elements need to support ICN functionality. As ICN is a clean-slate architecture, it causes the compatibility issue. Leveraging the programmability of SDN, ICN service and functionality can be implemented by an ICN specific application. It will be simplified to deploy ICN and enable the continuous evolution of ICN architecture. When it comes to support ICN by using SDN, there are two basic approaches: a short-term perspective and a long-term perspective. The detailed analysis is presented as follows.

A. SHORT-TERM APPROACH

The short-term approach is aimed at individual Internet Service Provider (ISP) who desires benefits from ICN without overhauling their entire service offering or forcing their clients to modify or replace their existing devices and software. In this context, the existing protocols (e.g., IP, UDP and HTTP) are used to carry the ICN packet [3]. ICN payload or the whole packet is encapsulated in a certain Protocol Data Unit (PDU) and the header fields should be modified carefully to ensure compatibility. The main difference of constructing the packet header is how to define and locate content identifier. In other words, this is equivalent to build an overlay channel to transport ICN packets. Meanwhile, a number of dedicated border nodes or other external systems are needed to perform a name-to-location resolution, as summarized in Fig. 4.

Based on this idea, a project named POINT [34] was funded to develop technologies, innovations and business value chains for commercially viable IP-over-ICN deployment. POINT adopts a gateway-based approach in order to preserve the IP interfaces towards the existing devices and IP applications and services. The bridging between IP and ICN is performed by the Network Attachment Point (NAP) and the ICN Border Gateway (BGW). The NAP is responsible for bridging ISP customer and the network, and the BGW is responsible for accessing to the general Internet. The interconnection of ICN domains involves setting up IP-encapsulating tunnels, as shown in Fig. 4(a). Furthermore, POINT takes advantage of SDN to directly integrate the emerging equipment at the architecture level. NDNFlow [35] implements a separate communication channel and a ICN-aware controller module parallel to the already existing OpenFlow communication channel and process. All communication and path computation regarding the ICN is handled separately from the IP and Ethernet plane. NDNFlow adopts ICN-enabled switches to set up a specific communication channel to the ICN module of the OpenFlow controller.

ICN module of the controller is responsible for computing paths for ICN flows. ICN-enabled switches receive ICN-specific flows directly and IP-encapsulated tunnels are set up for flows among ICN-enabled switches that are unreachable due to intermediate legacy IP switches.

ContentFlow [28] is a representative proxy-based ICN deployment approach that implements a centralized content management layer in controller to identify content by HTTP header information and make routing decision based on content name. The mapping from ICN name to TCP/IP semantics is performed by a proxy. Specifically, the file name is parsed from the HTTP GET request and the TCP flow information. The proxy at the ingress is to terminate and proxy the TCP sessions and support late binding of the content with its location. Only when a content is requested, the proper route can be set up. In [3], in order to recognize the ICN packets, ICN payload is encapsulated within IP packet in different ways: (i) Exploit the IP options field and define a new IP option which is supposed to include the ICN header. Embed the ICN payload into IP payload. (ii) Dynamically map content names to fixed length tags in ''ingress'' nodes at the border of an ICN/OpenFlow domain to reduce the complexity of matching variable-length content names. (iii) Use a fake UDP transport to run over OpenFlow 1.0. Instead of redesigning a packet format to contain a content name, C-flow [36] extracts a content name from an HTTP request with Deep Packet Inspection (DPI). After that, it assigns an IP address to the content and stores that information in a content-to-IP address table. C-flow creates the path from the end user to the cache node by setting up flow entries in the intermediate switches and uses the assigned IP address instead of the address of the cache node. References [37]–[40] follow the similar design philosophy. We classify this kind of short-term approach as resolution-based deployment approach, as shown in Fig. 4(b).

In conclusion, the two types of the short-term approach to deploy ICN in SDN domain are hardly a desirable way to fully exploit the ICN's advantages. The IP-tunneling and the resolution-based approaches suffer from the low transmission efficiency and complex configurations. Most of the proposals require that every incoming ICN packet must be sent to the SDN controller to compute the path, which can potentially overwhelm the SDN controller with a large number of packets.

B. LONG-TERM APPROACH

The long-term approach is to design ICN capable switches in order to accommodate ICN specific packets. Traditionally, applying a new protocol or new forms of packet encapsulation requires an overhaul of underlying devices, and in the worst case, a total redesign of the hardware and the Application Specific Integrated Circuits (ASICs). SDN also faces the extensible problem in data plane, since current southbound protocols, e.g. OpenFlow, are protocol-dependent. In this context, to develop flexible and even reprogrammable underlying devices becomes a trend among vendors and academics. P4 [41] and POF [42] are respectively representative packet processor description language and forwarding architecture to allow protocol-oblivious forwarding. Consequently, this kind of technology is conducive to solve ICN deployment problem from a long-term perspective.

In particular, P4 supports a programmable parser to allow new headers to be defined. Then, programmers can create target-independent programs that a compiler can map to forwarding devices. Signorello *et al.* [43] program the software-switch through P4 to match NDN packet process. The NDN variable-length Type-Length-Value (TLV) packet format is parsed by splitting TLV into separated headers and parsing them individually. Fig. 5 schematizes the control flow of the ingress pipeline where an Interest is processed by P4 definition. The parser is to parse the NDN header. The count table counts the number of name components which is used to match FIB. The hashname table computes the hash of the full packet name (to address the PIT) as well as the hashes of the name prefixes (to perform the Longest Prefix Match (LPM) on the FIB). The PIT reads a register (maintains the state for pending Interest) indexed by name hash. If a PIT entry is found, the isInPIT metadata contains the content of the selected PIT entry. If not, the FIB is matched with LPM. If route is found, then the updatePIT table is applied to create an entry for the Interest. When the PIT entry is created, a value is stored into a register. The processing workflow for a Data packet is similar to the one for an Interest except for the conditional branch on the packet type. The routeData table is applied last in case of a Data.

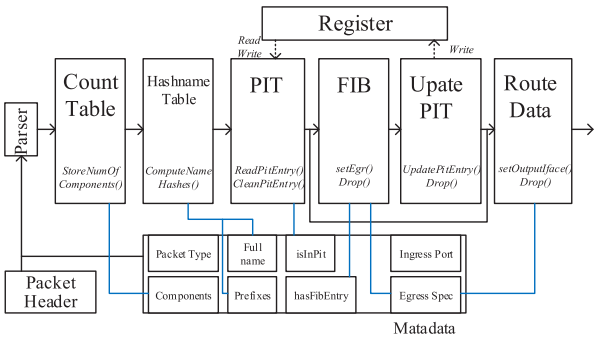


FIGURE 5. P4-based NDN packet control flow of the ingress pipeline.

In [44], Wang *et al.* propose a POF-based deployment scheme of ICN. They proposed a prototype CCN architecture mainly composed by the POF-compliant switches and controller. Due to the lack of caching capacity of the POF-compliant switch, a service node (performing caching and routing functions) is attached to a POF-compliant switch. However, the ICN forwarding logic is of great difference from the general packets forwarding, making designing an ICN switch a challenge. For example, ICN needs cache lookup in forwarding process and variable-length fields match (variable-length name components). Similarly, in SDCCN [45], POF FIB flow table is used for

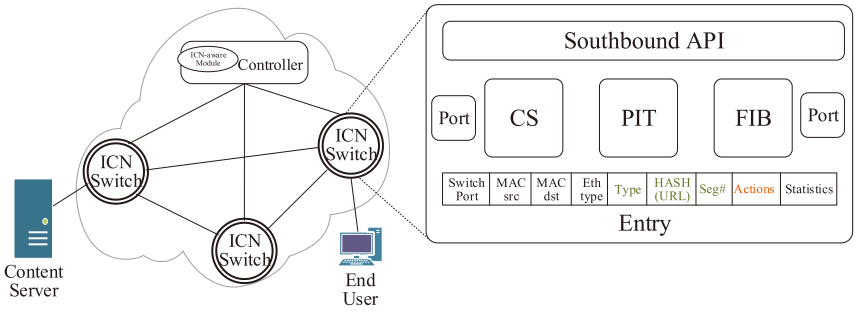


FIGURE 6. OpenFlow-enhanced approach.

FIB implementation. The rule of FIB format has a content name and a bit mask that will indicate whether the corresponding bit of the content name will be considered during forwarding lookup using {*offset; length*} search keys.

Zuraniewski *et al.* [46] propose the Extended Berkeley Packet Filters (eBPF) that matches on NDN packets, which is a run-time data plane agility. Another straightforward way to implement ICN using SDN is to enhance the OpenFlow protocol to support ICN name matching. Suh [47] extend OpenFlow to realize CCN node model. Specifically, they extend OpenFlow protocol to manage in-network caches and enable CCN name matching inside OF-switches flow table, as shown in Fig. 6. There are three newly added fields, i.e., type for identifying packet type (Interest/Data), Hashed URL for content name and Seg# for content segment. Actions are extended to store/load content. In this way, CCN packets can be handled with no modification. Reference [48] follows the similar design philosophy.

Although the above proposals are more flexible than short-term approaches and can be classified into long-term approach, they also face many challenges, e.g., P4 does not allow to skip parsing packet portions (skip verifying the signature) and perform a stateful operation (pending Interest). The OpenFlow-enhanced approach fails to consider the evolution of both ICN protocols and the OpenFlow protocol. The evolution of protocols will increase the complexity of realizing a stable standardization that supports both legacy IP protocols and ICN architecture. Furthermore, the ICN packet format is a complex nested TLV structure with a variable size of not only field's Value but Type and Length, which makes parsing difficult.

C. INTEROPERATION OF DIFFERENT ICN ARCHITECTURES

The aforementioned deployment solutions are based on the particular ICN architectures. However, from the perspective of Network Service Providers (NSPs), there are a number of ICN architectures with their own application values, thus it is essential to enable interoperation of different ICN architectures. Oriented to NDN and PURSUIT, Wang *et al.* [49] propose a unified deployment and interoperability framework for the two ICN architectures over Open vSwitch. By defining a unified packet tagging scheme by exploiting the Multi-Protocol Label Switching (MPLS) tag and extending relevant OpenFlow modules, NDN and PURSUIT are interoperable. In [50], a versatile ICN (VICN) deployment framework for NSPs is proposed, which is suitable for all architectures, as shown in Fig. 7. From the bottom up, VICN partitions the scope of ICN into the following four layers: i) the geography coverage layer, which specifies the coverage of NSPs' networks, ii) the VICN-enabled network infrastructure layer, in which multiple VICN-enabled networks (each network consists of a set of VICN-enabled switches and a VICN-enabled controller) can be established by different NSPs, iii) the ICN instances layer, in which an ICN instance can be created on a particular NSP network, and iv) the ICN users layer, in which each ICN instance can have its own group of users. VICN allows different ICN instances to connect with each other by virtual channels.

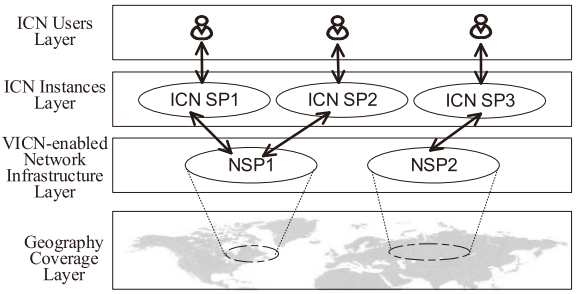


FIGURE 7. An overview of the VICN deployment scenarios.

Realizing ICN features in SDN architecture is not a trivial task due to the immature SDN protocols as well as ongoing research of ICN. Ideally, it can be assumed that ICN protocol is installed in every underlying device. However, considering the off-the-shelf devices deployed widely or some devices not allowed for any modification, such an assumption may not be realistic. Deployment of a promising technology is more likely to be endorsed and driven by powerful players and industries (e.g., equipment manufacturers). For them, it is natural to assess if it is worthwhile, and how ICN can provide benefits to network providers. Thus, tradeoff between cost and benefit must be considered carefully.

TABLE 2. SD-ICN: main benefits and corresponding literatures.

Research Aspect	Main Benefits	Reference
In-Network Caching	• Efficient Cache Awareness • Consistent Network Cache State • Intensive Computing for Caching Optimization • High Level Information for Caching Optimization • Shortening the Content Retrieval Time	[52]–[55]
Routing	• Efficient Cache-Aware Routing • Decreased Routing Signaling Overhead and Simple Path Setup and Maintenance • Multicasting • Multipath Routing	[56]–[60]
Service Chaining	• Improved Flexibility and Scalability	[61]
Traffic Engineering	• Decreased Network Redundant Traffic • Finer-grained Traffic Control • Accurate Estimation of Content Flow	[62], [63]
Wireless Communications and Mobility	• Cooperative Communication among Entities • Direct Communications with UEs • Decreased Redundant Traffic and Duplicated Access • Energy Efficiency • Receiver and Source Mobility Support	[6], [64], [65], [68], [69]
Edge Service	• ICN Service Virutalization • Real-Time Adaption of Services to the Consumer's Context	[71]
Resource Allocation	• Reduced Network Usage • Energy Efficiency	[73]

V. NETWORK MANAGEMENT ENHANCEMENT BY SD-ICN

In this section, we scan the state-of-the-art work on improving network management by SD-ICN and classify them according to the main investigated issues, i.e., caching, routing, service chaining, traffic engineering, wireless communications and mobility, edge service and resource allocation. The main benefits of the SD-ICN paradigm for network management enhancement are discussed in detail. For the reader's convenience, the main benefits and the corresponding literatures are summarized in Table 2.

A. IN-NETWORK CACHING

Explicitly naming content allows for exploitation of in-network caching in any network element. Based on this, ICN can improve network performance by fetching content from the nodes that close to users. There are two approaches to implementing in-network caching, i.e., on-path and off-path. On-path caching is a straightforward approach that requires less computation and communication overhead of placing copies within the network. Content copies are cached in the intermediate nodes along the path from content providers to users. However, continuously accumulated copies on the path results in high redundancy of contents, which may significantly waste the storage of nodes and decrease overall the cache hit ratio. In contrast, off-path caching does not require content copies to be cached on the traversed path. Therefore, retrieving content copies from off-path nodes requires redirection of requests according to the obtained cache information from neighbors. Content copies have to be placed in elaborately selected nodes to reduce the retrieval delay, alleviate redundancy of the same content and increase the overall cache hit ratio. However, the optimal content copy placement needs high level information, such as nodes centrality, traffic patterns and content popularity. It is a computation-intensive optimization problem. Moreover, it is closely related with the routing problem which deflects requests to the appropriate nodes.

In addition to content placement, locating the cached content accurately and timely is critical for exploiting in-network storage in ICN. Advertising all the cached contents in the distributed ICN network will cause consistency problem and huge control traffic. What's worse, an enormous number of cached contents and their volatility nature make it more severe. The slow convergence will lead to inconsistent cache index which results in decreased routing efficiency.

SDN provides the separated control plane. The centralized SDN controller can perform computation-intensive tasks to solve the optimization problem. By leveraging the global view of network cache state, SDN controller can easily obtain and maintain a consistent view of the underlying network cache state at the control layer, which can be used to determine where to cache content copies and compute optimal routes for requests. Recently, Azimdoost *et al.* [51] quantify the minimal amount of information required to keep a (logically) centralized control plane aware of the network cache state and demonstrate the feasibility of keeping the representation of this state up-to-date in the control plane.

Nguyen *et al.* [52] improve caching in NDN based on OpenFlow. A wrapper module was used to allow OpenFlow to support NDN packet. Then, they built three new controller modules, i.e., (i) measurement module infers the content popularity from flow statistics of ingress OpenFLow switches periodically. (ii) Optimization module uses the information of content popularity as an input for the caching optimization and computed the optimal cache location. The objective of the optimization problem is to minimize the sum of the delays over deflected contents. (iii) Deflection module is responsible

to build a mapping between the content name and an outgoing face. Finally the forwarding rules are generated and installed on switches for subsequent requests.

Jmal and Fourati [53] implement a new entity named "Cache Management (CM)" to store the most popular contents instead of caching the requested contents at every traversed node. The CM is managed by SDN controller to realize the centralized content popularity counting and caching strategy. Only storing popular contents provides high performance with saved storage. Cao *et al.* [54] deploy a specified centralized Tracker Server (TS) in each ISP to collect all the content information in a Replica List from all the routers' Top-N popular contents in the network as well as the network cache state. To minimize the overhead of storage and transmission of cache state information, Bloom Filter is adopted to represent the Replica List. Thus, each consumer can fetch popular data directly from the nearest routers under the guidance of Zhang *et al.* [55] propose a intelligent content discovery system based on a centralized Network Management (NM) node. The NM is responsible for collecting traffic data and constructing the learning algorithm as well as distributing the strategies to the forwarding node. The data collected by the NM node are mainly the name prefixes of various Interest packets, which are used to train the Deep Belief Network (DBN) that extracts the topic vector similarity of an Autonomous System (AS). The temporal topic distribution can represent the user's demand for network contents within the AS. Based on this, the NM node obtains the contents that are most likely to be requested in the forwarding nodes, announces them to the forwarding nodes and constructs paths for Interest packets.

B. ROUTING

ICN originally uses name prefix announcement routing scheme to disseminate content location in the network, like NDN. However, the number of content (and content names) are expected to be orders of magnitude larger than the number of hosts. The network wide prefixes announcement increases the routing signaling overhead in proportion to network size and to the number of different name prefixes. Each node stores a huge routing table in its database which is a heavy burden on the storage. More seriously, generating the huge routing table could result in the increased delay as well as table lookup time. By leveraging SDN, Torres *et al.* [56] employ a centralized controller to carry out computation-intensive tasks, such as routing decision and processing name prefixes announcement, and reduce storage consumption of forwarding nodes. SDN ensures that network nodes register and request network information without flooding the network, which reduces routing signaling overhead.

Detti *et al.* [57] propose an ICN routing framework, called COntent NETwork (CONET). It is based on the interaction between the forwarding nodes with Name Routing System (NRS). It confines routing to NRS and reduces the processing delay and increases the throughput of the forwarding node. Its separation between forwarding and routing control naturally maps into the SDN architecture. Therefore, Salsano *et al.* implement CONET over OpenFlow [3].

CRoS-NDN [58] reduces routing overhead by limiting interest flooding in the network. It requires only one controller for end-to-end routing. The requesting node notifies the controller of its identifier and the desired content name. When routing a request, the controller first identifies the requesting node and locates the content producer node. After that, the controller calculates the node identifier sequence path from the consumer to the producer, which is also the data return path. When a request for interest is made, a new FIB entry is established on each node between the consumer and the producer. Topology changes or content mobility may invalidate FIB entries installed on the node. The CRoS-NDN removes invalid FIB entries on each node through data plane feedback.

In order to take advantage of cached content, the requests should be forwarded to the closest nodes which cache the corresponding contents. However, due to the huge amount of copies and dynamic nature of caching, instructions where content copies are cached cannot be broadcast throughout the network. Therefore, exploiting in-network content copies efficiently and accurately is a major challenge in ICN routing scheme.

SDN can enable ICN to locate copies more efficiently due to its rapid collection of knowledge about copy locations and its strong capability of computation. In [59], each node explicitly indicates the cache state (when a copy of content is added or removed) to SDN controller. The controller is responsible for updating its content table and routing information to the closest node. The workload of the controller is in correlation with the volume of content and the size of the network it manages. Thus, a large-scale network should be managed by multiple controllers or split into several domains. To cope with such kind of scalability problem, Lv *et al.* [60] divide ICN into communities based on interest similarity. Multiple SDN controllers are used to conduct interest clustering analysis, community division and intra/inter-community routing. With the help of SDN and community division, the scalability problem of ICN routing can be alleviated and the content retrieval can be significantly improved.

C. SERVICE CHAINING

In today's Internet, various middle-boxes with different network functions (e.g., DPI and firewall) are placed within the network in order to provide additional network services. Network operators require the data flows to go through certain middle-boxes to realize certain goals (such as user experience enhancement and attack prevention) by service chaining. Network Function Virtualization (NFV) enables network function deployment much flexible, as it allows for network functions to be instantiated at general-purpose hardware platform on-demand. With the help of SDN, network operators can optimally place network functions and strongly steer data flows. However, SDN controller has to keep track of

the status of service chaining and make decisions in a timely manner when the flow is required to change the functions (e.g., the results of DPI may require further intrusion detection to be applied on the flow), which results in flexibility and scalability problem [61]. What's more, the unnecessary coupling of routing (which finds the path for a flow to go through) and policy (which determines the service functions needed by a flow) limits service chaining efficiency.

Arumaithurai *et al.* [61] propose a Function-Centric Service Chaining (FCSC) mechanism, as depicted in Fig. 8. It extends the ICN principle of naming content to naming function. It decouples the network function with its location via a naming layer. The naming layer separates the policy module (determine what functions should be applied to a flow) from the routing module. In FCSC, SDN controller determines the policy which a flow needs, and returns the result only to the ingress. Then, the ingress is required to tag the function list (in the form of an ICN name, e.g., /firewall/DPI/...) to the packet header. Not only switches but also middle-boxes are allowed to determine the outgoing face without notifying the controller. In this way, the scalability and the flexibility of service chaining are improved.

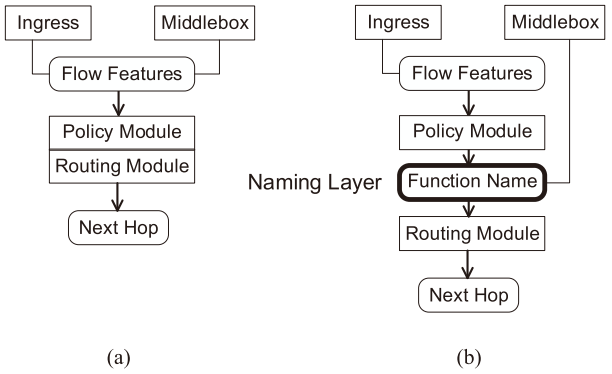


FIGURE 8. (a) SDN. (b) FCSC.

D. TRAFFIC ENGINEERING

Content transmission in ICN does not rely on end-to-end communication, as content can be obtained from in-network caches. Furthermore, the same content requests that received by a router are aggregated. Therefore, the traditional concept of flow is not suitable in ICN, especially when a portion of content is transmitted from in-network caches and the other is from the original content provider. In this case, an effective reordering mechanism is required. However, ICN in turn supports the network to directly exploit the properties of the content, e.g., the beginning and ending of the content can be explicitly extracted as well as its type by its name. Thus, traffic engineering in ICN can make full use of in-network caches and ICN named content semantics.

Chanda *et al.* [62] demonstrate the benefit of obtaining content length using SDN to optimize traffic engineering in ICN. In particular, the SDN controller can locate the proper content copy when a content request arrives. It observes and records the content length on the fly, and uses this information to allocate the proper resource under certain constraints for content flows. Furthermore, since ICN does not require the provider to transfer content metadata, e.g., content length, the overall link bandwidth is saved.

ICN provides the explicit description of content. This allows the network to apply different traffic control policies to different content to achieve specific optimization objectives, such as fairness and load balance. Sun *et al.* [63] design two map-lists, the first is to map various NDN flows to different type/class/priority, and the second is to map different type/class/priority to various queues. The SDN controller monitors and analyzes flow information and then regulates the map-lists. The content flows are aggregated to different queues and various queue management and scheduling algorithms are supported.

E. WIRELESS COMMUNICATIONS AND MOBILITY

Leveraging the in-network caching and innate multicast features of ICN, the SDN's flexibility and reconfigurability of wired network management and the evolutionary on the wireless network side of Cloud Radio Access Network (C-RAN), [64] contributes to a novel integration of the three architectures for the Heterogeneous Network (HetNet). A high-level view of the architecture is shown in Fig. 9, where the application, control, and data planes are separated. The application plane consists of the information-centric applications and services. The control plane abstracts the infrastructure, programming, and even the content naming, addressing for the ICN. Further, C-RAN separates the Baseband Processing Units (BBUs) and the Radio Access Units (RAUs). To alleviate the load of the controllers and seamlessly integrate the wired and wireless sides of SDN, the BBU pool has both control and data forwarding functions. The logically centralized BBU pool has a global view of the RAN and core network as shown in data plane. Every forwarding device is assumed to caching contents, e.g., BBU pool and RAN, which offloads the redundant traffic and duplicated access (the new cache and offload traffic in Fig. 9). Besides, users can even communicate directly without traversing the BBU pool due to ICN capability. However, Huo *et al.* [65] argue that cloud computing services may not provide guarantees to low-latency applications and the transmission of

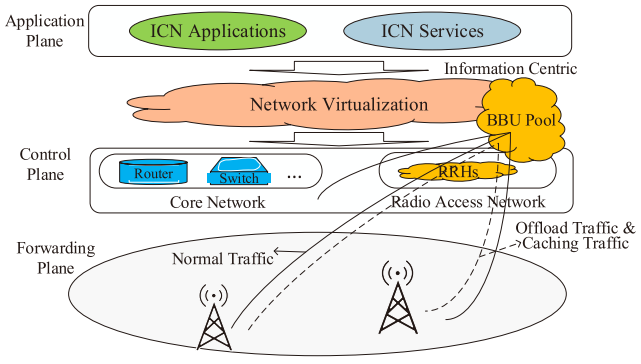


FIGURE 9. The SD-ICN based C-RAN.

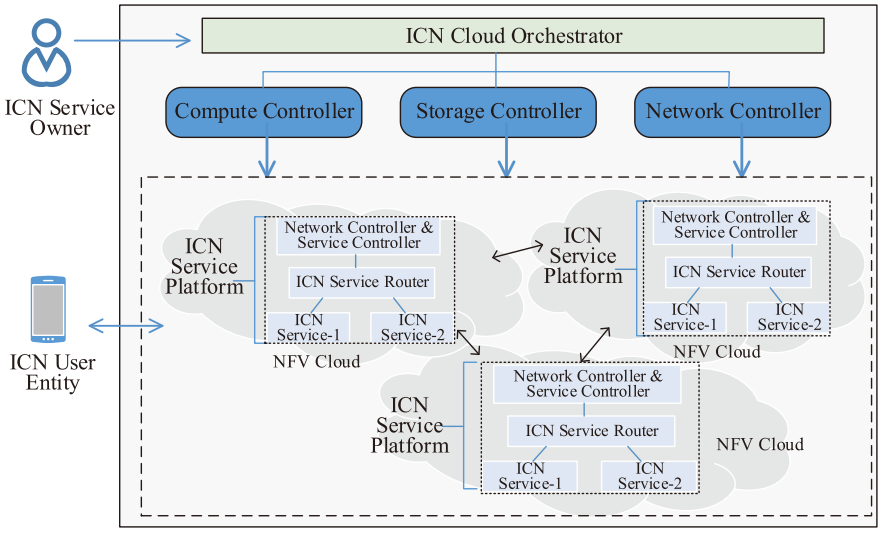


FIGURE 10. ICN edge-cloud service.

a large amount of data from the remote device to the cloud may not be efficient. By jointly considers networking, ICN in-network caching, fog computing [66] and mobile edge computing techniques [67] to realize an energy-efficient content retrieval and computing services for green wireless networks. The energy consumption model includes transmitting energy E_{tr}, caching energy E_{ca} and computing energy E_{ce}, and the total energy consumption can be expressed as $\sum_{allusers}\sum_{allrequests}(E_{tr} + E_{ca} + E_{ce})$. The main goal is to save the energy of using networking resource.

Oriented to Wireless Mesh Network (WMN), Kim *et al.* [68] propose a method for improving the efficiency of content delivery using SD-ICN. In particular, each mesh node of the WMN can utilize its storage as an in-network cache, and SDN enables an optimal cache management. Considering constraints on mesh node computing capability, the computation-intensive tasks, i.e., optimal cache location decision, name-based routing and cache migration are performed in SDN controller. In their design, such cache management is completed at the controller including content identification, since the content is embedded in HTTP message. The cache location decision algorithm is designed to reflect the consideration of WMN's constrained node resources and mesh topology.

Suoa *et al.* [69] propose an efficient content dissemination in highly mobile VANETs using CCN and Floating Content (FC) based on SDN. CCN allows data to be exchanged throughout the network based on their name instead of the location of the hosts. FC is an opportunistic communication scheme which supports infrastructure-less distributed content sharing over a given geographic area. SDN provides wireless resource optimization (channel allocation and interference avoidance), packet routing and forwarding in multi-hop multi-path scenarios. By combining the three technologies, SDN could support the set up of multipath communication. When a vehicle sends or receives a message (Interest or Data), it could send (or receive) it through multiple (redundant) paths.

F. EDGE SERVICE

In recent years, Network Function Virtualization (NFV) has gained attention with the operators demanding for its flexible software realization of network functions over commodity hardware. Combining NFV with SDN enables agile infrastructure, resource management and deployment of new applications. Currently, the integration between ICN and SDN/NFV has emerged. ICN introduces a new way for service provision in SDN and NFV, with a certain extent of data plane programmability, it virtualizes the edge functions [72]. Towards this, Ravindran *et al.* [70] propose an ICN based platform that supports ICN applications as edge-cloud services, e.g. enterprise applications, IoT, Machine-to-Machine (M2M). NFV allows the ICN framework to be realized as a Virtual Network Functions (VNF), while service flows are engineered by the operator or service owner based on SDN. An NFV-based network is an inter-connection of edge-clouds controlled by one or more providers. A high level view of their proposed platform is shown in Fig.10, comprised of three important components: ICN cloud orchestrator, ICN service platform, and ICN service layer. The cloud orchestrator interfaces with the ICN service owners to manage the cloud resource through the ICN service-API. The ICN service platform is composed of VNF instances in the form of ICN service router and service control functions. ICN service router implements an ICN forwarding plane to connecting ICN service instances and service control functions are regarded as control plane (for resolving service requests and managing the name based routing policies). The ICN service layer implements service functions to allow the consumer's application to interact with the ICN service platform.

G. RESOURCE ALLOCATION

Chen *et al.* [71] propose a joint resource allocation mechanism for networking, in-network caching and computing resources using SDN. In particular, as Fig. 11 shows that the data plane contains not only the forwarding devices, but the caching and computing devices. Accordingly, the management modules for the caching and the computing resource are added in control plane. The management plane is used to remotely monitor and configure the control functionality through the SDN controller. By dynamically guiding the different type of Interest requests (for content service and/or computing service) to the corresponding service devices, the mechanism achieves an efficient resource allocation and network orchestration.

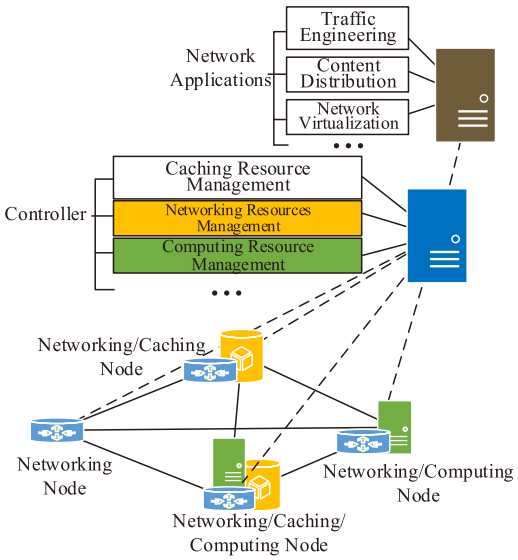


FIGURE 11. Joint resource allocation mechanism for networking, in-network caching and computing.

VI. SECURITY BENEFITS

ICN provides a data-centric security paradigm, avoiding many of the host-based vulnerabilities that plague IP-based networking architecture including the current SDN architecture. However, SDN has some clear security advantages to operational networks. Thus, in this section, we discuss the potential security benefits from ICN for SDN and SDN for ICN perspectives respectively, as summarized in Table 3.

A. ICN FOR SDN SECURITY

Although there are clear advantages of solving security problem gained from SDN architecture (e.g., rapid response to network security threats and centralized anomaly detection), SDN does not change the IP architecture essentially. Thus the longstanding security problems that plague IP networking still remain. In IP-based Internet, security was not conceived by the original design. In order to cope with that, plenty of add-on security mechanisms, such as Transport Layer Security (TLS), DPI and firewall, have been proposed to reject malicious behaviors. In reality, such kind of security model requires the trust in the content provider (host) and the connection between the consumer and content provider rather than the content itself. This is widely recognized as a significant problem [19]. Without the reliable intermediate network node or after the secure connection ends, the trust between the consumer and content provider does not exist. Furthermore, due to the separation of control plane and data plane, SDN also results in a number of new security challenges that directly threaten the controller, southbound/northbound interfaces and network applications as shown in Fig. 12. Attacks on the controllers and applications can easily grant an attacker the control of the network, which can break down the entire network [1].

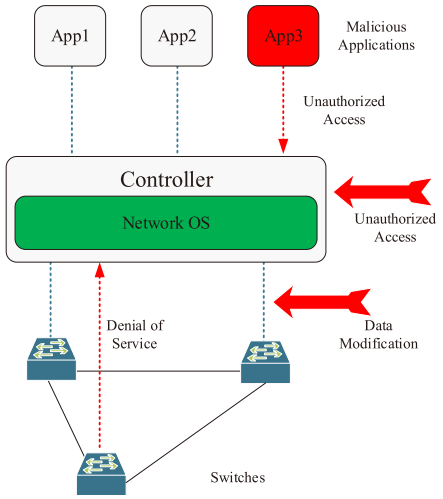


FIGURE 12. Main security challenges of SDN architectures.

By offering data-centric security model, ICN protects content itself rather than communication channel or network device, which intrinsically avoiding many attacks aiming at a specific host or device (Fig. 13(a)). More specifically, all contents are signed by their original providers. Consumers and intermediate ICN nodes can verify the content validity by the designated keys, shifting the trust from network node to content. By allowing content to refer to other content we can allow content to certify other content, which provides a trust chain [19], as present in Fig. 13(b). The private content is protected by encryption, as shown in Fig. 13(a). Data packet is only in response to an Interest request and one Interest retrieves at most one Data packet. This consumer driven communication ensures that the flow is "balanced". ICN potentially promotes SDN security from at least the following three aspects.

First, in data plane, the pub/sub communication model of ICN ensures that there is no unwanted content transmission, which effectively eliminates many existing attacks, such as Spam, malicious content injection and content-based Distributed Denial of Service (DDoS). For ICN architectures which adopt self-certifying names can even filter the malicious content by in-network mechanisms [2]. Moreover, ICN can aggregate the similar Interests which request the

TABLE 3. Potential security benefits of SD-ICN.

	Security Challenges	Potential Benefits by ICN	Potential Benefits by SDN
SDN	• Unauthorized Access • Data Modification • Malicious Application • Fake Controller • DDoS to Switch/Controller	• Self-Certifying Control Message • Immutable Data • Impossible Host-based Attacks • Impossible Data-based DDoS	-
ICN	• Expired Content Revocation • Content Privacy	-	• Rapid Revocation of Expired Content • Expired Content List • Centralized Access Control • Rapid Converging to a consistent Security Configuration

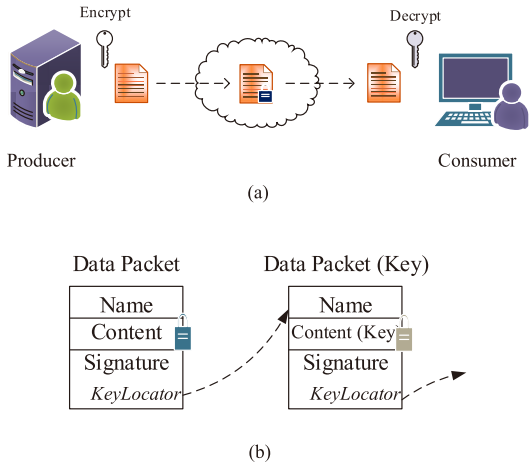


FIGURE 13. (a) Data-centric security. (b) Signatures in NDN.

same content by switches. Theoretically, the same Interest flooding attack is simply impossible. If one attacker generates Interests with randomly name prefixes, such Interests will never retrieve any content. Therefore, SDN controller can monitor the information of Interests which never have content response, and simply limit such kind of transmission.

Second, for SDN southbound interface, without access control support and TLS connection between the controller and switch, the channel between the controller and switch can easily be attacked [9]. For example, the side channel attack targets the flow rule setup process, interception targets the channel and hijacking targets the rule's modification. However, TLS authentication process involves many handshakes and confirmation procedures, which has big delay. In ICN, control message can be regarded as a kind of special request that contains the command's name and parameters. Hence, switches are enabled to authenticate the control messages from SDN controller by asking for signatures on control messages (e.g., OpenFlow Modify-State messages). Only the authenticated control messages from legal controllers can instruct the switch state. Similarly, asynchronous messages (Switch-to-Controller messages), which are used to update the controller about network events and changes to the switch state, can be authenticated by SDN controller. In this way, ICN provides a solid foundation for securing SDN southbound interface against spoofing and tampering. In addition, similar methods could be adopted in protecting SDN application layer and northbound interface.

Third, with the help of ICN, spoofing the address of SDN controller (using a fake controller) to take over the control of the entire network becomes unfeasible, because ICN only talk about content rather than address and the unauthenticated controller messages cannot take effect. To be effective, the attack against SDN controller might be DDoS as follows: launch the resource depletion attack on SDN controller by overwhelming it with huge amount of spurious messages. For instance, as SDN allows transferring the control of a packet to the controller by sending a packet-in message when flow table miss happens in a switch. The attacker can generate a large number of Interests to overwhelm the SDN controller with packet-in messages. However, as mentioned above, the similar Interests can be aggregated by switch and Interests that never result in Data responses are limited, thus the packet-in messages are significantly reduced. As for attackers that generate real Interests on the existent contents, ICN provides name-based policy control to selectively throttle or block abnormal Interest requests.

B. SDN FOR ICN SECURITY

Although ICN helps to solve most of the traditional host-centric security problems, new attacks have emerged in some aspects, such as content privacy and cached content revocation [73]. Due to the pub/sub model and universal in-network caching mechanism in ICN, the content provider is hard to be aware of the locations of their published contents and who will get the contents from the intermediate nodes. Although sensitive data can be protected via encryption in ICN, an attacker can quite easily get the encrypted content which he is interested in. Even for the encrypted data, it is likely to disclose some information, such as content size and request time. However, once the protection scheme is cracked, the cached content is no longer protected. Thus, for example, if the key is cracked, it needs to be marked as unavailable. This information must be spread to all consumers or even all of the routers in time.

As for the content that has been published, its copies will be cached somewhere in the network. Once the content is out of date, it should be revoked timely and efficiently. In order to prevent the expired content from spreading, every

ICN node may need to publish and perform a revocation list. Alternatively, the content could be released with a specific survival time. However, this would cause a higher load in terms of content provider and ICN node, because they are both required to provide service for content copies regularly.

Based on the above analysis, devising an efficient cached content revocation mechanism in a distributed control network is very complicated. By leveraging the centralized control of SDN, it is conducive to implement such mechanism. For instance, if an encrypted content is found cracked or a content has been expired, its provider should send a content revocation request to its adjacent switches within the scope. After receiving that request, the switches can send it to its controller, and then SDN controller can instruct the corresponding switches to evict the expired cached content from their storage and no longer respond to the content request. What a SDN controller needs to do is to maintain a revocation list and issue corresponding instructions to switches. In the same way, the revocation of the cracked keys can be implemented. Hence, SDN can provide a rapid response to the cached content revocation request and revoke the copies in a timely manner. Furthermore, the centralized control eliminates the potential consistency problem of the revocation list.

VII. OPEN ISSUES AND FUTURE PERSPECTIVES

A. EXTENDED NAME-BASED CONTROL

We argue that the usage of ICN name has not been fully exploited by research communities. In fact, a name can identify anything in ICN context, not only a data chunk, but also a network device and even a network function or service. This idea was pioneered by the Named Function Networking (NFN) project [74]. In addition to retrieve content, names can be served to access and invoke functions. For example, users can request a file compression service with the name ''/util/compress/zip/codec/mpeg4(/name/of/media)''. Then, a compressed file will be returned to users. From this point of view, it is reasonable to extend the named data to a more general concept. Concerning this, the way of operating a network by SDN deserves deep thought.

B. QoS

Another important issue of SD-ICN is QoS support. The booming data traffic and higher demands on some future applications (e.g., vehicular ad-hoc network and industrial Internet of things) provide a direct incentive to support better QoS in ICN than ever before. However, the QoS requirements of networked applications, such as low transmission delay, high delivery reliability and low loss rate, have not been well considered in ICN architecture so far. For example, although NDN supports ''nearby access'', it mainly performs as a best effort service. With the aid of SDN, a better decision based on the caching information from multiple sources can be made. More notably, SDN essentially implements a connection-oriented virtual circuit service model, since the controller plans the route upon a specific request arrival. This connection-oriented virtual circuit service model is conductive to QoS support. Thus, it deserves more attention on exploiting how QoS can be improved in the context of SD-ICN.

C. SCALABILITY

Scalability is one of the major concerns of SD-ICN from the outset, since both SDN and ICN face this problem in different ways. The number of content in the Internet are far more than the number of network devices which would cause state space explosion. For example, if we establish datapaths using content names, a great number of entries may quickly exceed the capacity of forwarding table. Moreover, besides the network status awareness, the integration of SDN and ICN requires more capabilities in the control plane, such as cache status awareness and update in a timely manner. Obviously, such extra requirements exacerbate the scalability issue of control plane [75], [76]. In addition, flow setup latency is another major scaling concern in SD-ICN.

D. BIG DATA

Recently, how to process big data in network and how big data affects the future network architecture attract both academia's and industry's interests [77]. Due to extremely large volume, computing complexity and rapid processing requirements, big data applications would not be implemented efficiently without the underlying support of network. Researchers have recognized the need for developing novel network architectures and protocols to support big data management, process, and analytic [78]. The ICN's data centricity intrinsically supports data process at network layer. For example, ICN could be extended to conceive in-network data processing operations, and with the help of SDN, the logically centralized controller can obtain, analyze, and manage the state of ICN's in-network processing. Moreover, SDN helps abstract storage and network capability from user-data interaction and obtain insights from the data by leveraging analytical methods, while big data analytic can help guide decisions of network operations. For instance, SDN and big data aided ICN can dynamically and optimally decide data traffic routes and configure forwarding rules for a given type of traffic according to its name.

E. ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI), especially deep learning [79] (a subset of machine learning), has made a significant breakthrough in achieving high accuracy, efficiency and adaptability in a variety of applications, such as automatic speech recognition, image recognition and natural language processing. However, deep learning applications in network systems gain rather little attention. Although an excellent survey [80] about the deep learning applications for network traffic control is published as a response, traffic control is only one aspect of network research. Recently, Mestres *et al.* [81] elaborate that machine learning can bring automation and recommendation to networking with the help of SDN, which is

enlightening to communities to conduct a thorough research on combination of AI and computer networks. Liu *et al.* [82] pioneer the Stacked Auto-Encoders (SAE) [83] for predicting ICN content popularity based on SDN.

VIII. CONCLUSION

As two promising paradigms, SDN and ICN have gained extensive attention of both academia and industry. SDN merely gives developers a powerful tool to create new applications and find solutions to overcome longstanding problems of networking, however, it faces a lot of challenges, such as resilience, scalability, and security issues. As for ICN, how ICN can concretely be deployed and tested is a critical problem. Merging the philosophy of SDN and ICN into future Internet design is a new trend, which leverages the advantages of ICN in efficient content delivery and SDN in flexible programmability and efficient manageability. They intrinsically have very strong complementarity to each other.

In this paper, we dedicate to draw some guidelines for research community to embrace the research on SD-ICN by analyzing the motivations of integrating SDN and ICN, discussing the advantages and disadvantages of different ICN deployment approaches, reviewing the state-of-the-art work on improving network management from several aspects, such as caching, routing and wireless communications. Furthermore, we provide a valuable discussion of the data-centric security for SDN and centralized control for ICN security.

The continuous development of new network technologies and their integration give birth to better network services. How to make full use of their advantages deserves serious consideration. Hopefully, this paper will foster more profound thinking of future Internet design.

REFERENCES

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

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

[3] 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.

[4] Q. Chen, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "An integrated framework for software defined networking, caching, and computing," *IEEE Netw.*, vol. 31, no. 3, pp. 46–55, May/Jun. 2017.

[5] C. Fang, H.-P. Yao, Z.-W. Wang, W.-J. Wu, X.-N. Jin, and F. R. Yu, "A survey of mobile information-centric networking: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, to be published, doi: 10.1109/COMST.2018.2809670.

[6] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of green information-centric networking: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1455–1472, 3rd Quart., 2015.

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

[8] A. Mendiola, J. Astorga, E. Jacob, and M. Higuero, "A survey on the contributions of software-defined networking to traffic engineering," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 918–953, 2nd Quart., 2017.

[9] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 623–654, 1st Quart., 2016.

[10] A. S. D. Silva, P. Smith, A. Mauthe, and A. Schaeffer-Filho, "Resilience support in software-defined networking," *Comput. Netw.*, vol. 92, no. P1, pp. 189–207, Dec. 2015.

[11] R. Jmal and L. C. Fourati, "Content-centric networking management based on software defined networks: Survey," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 1128–1142, Dec. 2017.

[12] *FP7 SAIL Project*. Accessed: Apr. 2018. [Online]. Available: http://www.sail-project.eu/

[13] *FP7 4WARD Project*. Accessed: Apr. 2018. [Online]. Available: http://www.4ward-project.eu/

[14] *FP7 PURSUIT Project*. Accessed: Apr. 2018. [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/

[15] *FP7 PSIRP Project*. Accessed: Apr. 2018. [Online]. Available: http://www.psirp.org/

[16] *FP7 COMET Project*. Accessed: Apr. 2018. [Online]. Available: http://www.comet-project.org/

[17] *FP7 CONVERGENCE Project*. Accessed: Apr. 2018. [Online]. Available: http://www.ict-convergence.eu/

[18] *NSF Named Data Networking Project*. Accessed: Apr. 2018. [Online]. Available: http://www.named-data.net/

[19] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. ACM CoNEXT*, 2009, pp. 1–12.

[20] L. Zhang *et al.*, "Named data networking," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014.

[21] *EU-Japan GreenICN Project*. Accessed: Apr. 2018. [Online]. Available: http://www.greenicn.org/

[22] *NSF MobilityFirst Project*. Accessed: Apr. 2018. [Online]. Available: http://mobilityfirst.winlab.rutgers.edu/

[23] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox, "Information-centric networking: Seeing the forest for the trees," in *Proc. ACM HotNets*, 2011, pp. 1–6.

[24] *Open Networking Foundation (ONF)*. Accessed: Apr. 2018. [Online]. Available: https://www.opennetworking.org/

[25] L. Richardson and S. Ruby, *RESTful Web Services*. Sebastopol, CA, USA: O'Reilly Media, 2008.

[26] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[27] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN," *Queue*, vol. 11, no. 12, p. 20, 2013.

[28] A. Chanda and C. Westphal, "ContentFlow: Adding content primitives to software defined networks," in *Proc. IEEE GLOBECOM*, Dec. 2013, pp. 2132–2138.

[29] H. Luo *et al.*, "A framework for integrating content characteristics into the future Internet architecture," *IEEE Netw.*, vol. 31, no. 3, pp. 22–28, May/Jun. 2017.

[30] P. Bosshart *et al.*, "Forwarding metamorphosis: Fast programmable match-action processing in hardware for SDN," in *Proc. ACM SIGCOMM*, 2013, pp. 99–110.

[31] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache 'less for more' in information-centric networks (extended version)," *Comput. Commun.*, vol. 36, no. 7, pp. 758–770, Apr. 2013.

[32] S.-W. Lee, D. Kim, Y.-B. Ko, J.-H. Kim, and M.-W. Jang, "Cache capacity-aware CCN: Selective caching and cache-aware routing," in *Proc. IEEE GLOBECOM*, Dec. 2013, pp. 2114–2119.

[33] C. Barakat, A. Kalla, D. Saucez, and T. Turletti, "Minimizing bandwidth on peering links with deflection in named data networking," in *Proc. IEEE Commun. Inf. Technol.*, Jun. 2013, pp. 88–92.

[34] *EU POINT Project*. Accessed: Apr. 2018. [Online]. Available: http://www.point-h2020.eu/

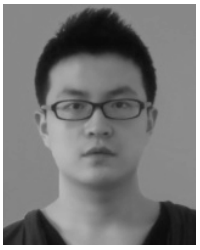
[35] N. L. M. van Adrichem and F. A. Kuipers, "NDNFlow: Software-defined named data networking," in *Proc. Int. Conf. Netw. Softw. (NetSoft)*, 2015, pp. 1–5.

[36] D. Chang, M. Kwak, N. Choi, T. Kwon, and Y. Choi, "C-flow: An efficient content delivery framework with OpenFlow," in *Proc. IEEE Int. Conf. Inf. Netw. (ICOIN)*, Feb. 2014, pp. 270–275.

[37] M. Vahlenkamp, F. Schneider, D. Kutscher, and J. Seedorf, "Enabling ICN in IP networks using SDN," in *Proc. IEEE Int. Conf. Netw. Protocols (ICNP)*, Oct. 2013, pp. 1–2.

[38] A. F. R. Trajano and M. P. Fernandez, "ContentSDN: A content-based transparent proxy architecture in software-defined networking," in *Proc. IEEE Adv. Inf. Netw. Appl.*, Mar. 2016, pp. 532–539.

[39] K. Choumas, N. Makris, T. Korakis, L. Tassiulas, and M. Ott, "Exploiting OpenFlow resources towards a content-centric LAN," in *Proc. Eur. Workshop Softw. Defined Netw. (EWSDN)*, Oct. 2013, pp. 93–98.
[40] S. Eum, M. Jibiki, M. Murata, H. Asaeda, and N. Nishinaga, "A design of an ICN architecture within the framework of SDN," in *Proc. IEEE Int. Conf. Ubiquitous Future Netw.*, Jul. 2015, pp. 141–146.
[41] P. Bosshart *et al.*, "P4: Programming protocol-independent packet processors," *SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 87–95, Jul. 2014.
[42] H.-Y. Song, "Protocol-oblivious forwarding: Unleash the power of SDN through a future-proof forwarding plane," in *Proc. ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, 2013, pp. 127–132.
[43] S. Signorello, R. State, J. François, and O. Festor, "NDN.P4: Programming information-centric data-planes," in *Proc. Int. Conf. Netw. Softw. (NetSoft)*, 2016, pp. 384–389.
[44] Z. Wang, L. Wang, X. Gao, Y. Xia, and S. Wang, "An architecture of content-centric networking over protocol-oblivious forwarding," in *Proc. IEEE GLOBECOM*, Dec. 2015, pp. 1–5.
[45] S. Charpinel, C. A. S. Santos, A. B. Vieira, R. Villaca, and M. Martinello, "SDCCN: A novel software defined content-centric networking approach," in *Proc. IEEE Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2016, pp. 87–94.
[46] P. Zuraniewski, N. van Adrichem, D. Ravesteijn, W. Ijntema, C. Papadopoulos, and C.-Y. Fan, "Facilitating ICN deployment with an extended openflow protocol," in *Proc. ACM ICN*, 2017, pp. 123–133.
[47] J. Suh, "OF-CCN: CCN over OpenFlow," in *Proc. AsiaFI NDN Hands-Workshop*, 2012, pp. 1–20.
[48] A. Kalghoum and S. M. Gammar, "Towards new information centric networking strategy based on software defined networking," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–6.
[49] J. Wang, W. Gao, Y. Liang, R. Qin, J. Wang, and S. Liu, "SD-ICN: An interoperable deployment framework for software-defined information-centric networks," in *Proc. IEEE INFOCOM Workshop*, Apr./May 2014, pp. 149–150.
[50] J. Ren *et al.*, "VICN: A versatile deployment framework for information-centric networks," *IEEE Netw.*, vol. 28, no. 3, pp. 26–34, May/Jun. 2014.
[51] B. Azimdoost, C. Westphal, and H. R. Sadjadpour, "Resolution-based content discovery in network of caches: Is the control traffic an issue?" *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 2943–2955, Jul. 2017.
[52] X. N. Nguyen, D. Saucez, and T. Turletti, "Efficient caching in content-centric networks using OpenFlow," in *Proc. IEEE INFOCOM Workshops*, Apr. 2013, pp. 67–68.
[53] R. Jmal and L. C. Fourati, "An OpenFlow architecture for managing content-centric network (OFAM-CCN) based on popularity caching strategy," *Comput. Stand. Interface*, vol. 51, pp. 22–29, Mar. 2017.
[54] J. Cao, D. Pei, X. Zhang, B. Zhang, and Y. Zhao, "Fetching popular data from the nearest replica in NDN," in *Proc. IEEE Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2016, pp. 1–9.
[55] H. Zhang, R. Xie, S. Zhu, T. Huang, and Y. Liu, "DENA: An intelligent content discovery system used in named data networking," *IEEE Access*, vol. 4, pp. 9093–9107, Dec. 2016.
[56] J. V. Torres, I. D. Alvarenga, R. Boutaba, and O. C. M. Duarte, "An autonomous and efficient controller-based routing scheme for networking named-data mobility," *Comput. Commun.*, vol. 103, pp. 94–103, May 2017.
[57] A. Detti, N. Blefari-Melazzi, S. Salsano, and M. Pomposini, "CONET: A content centric inter-networking architecture," in *Proc. ACM SIGCOMM ICN*, 2011, pp. 50–55.
[58] J. V. Torres, I. D. Alvarenga, A. D. C. P. Pedroza, and O. C. M. Duarte, "Proposing, specifying, and validating a controller-based routing protocol for a clean-slate Named-Data Networking," in *Proc. Int. Conf. Netw. Futur. (NOF)*, 2017, pp. 1–5.
[59] E. Aubr, T. Silverston, and I. Chrisment, "SRSC: SDN-based routing scheme for CCN," in *Proc. Int. Conf. Netw. Softw. (NetSoft)*, 2015, pp. 1–5.
[60] J.-H. Lv, X.-W. Wang, M. Huang, J.-L. Shi, K.-Q. Li, and J. Li, "RISC: ICN routing mechanism incorporating SDN and community division," *Comput. Netw.*, vol. 123, pp. 88–103, Aug. 2017.
[61] M. Arumaithurai, J. Chen, E. Monticelli, X. Fu, and K. K. Ramakrishnan, "Exploiting ICN for flexible management of software-defined networks," in *Proc. ACM ICN*, 2014, pp. 107–116.
[62] A. Chanda, C. Westphal, and D. Raychaudhuri, "Content based traffic engineering in software defined information centric networks," in *Proc. IEEE INFOCOM Workshop Emerg. Design Choices Name-Oriented Netw.*, Apr. 2013, pp. 357–362.
[63] Q. Sun, W. Wendong, Y. Hu, X. Que, and G. Xiangyang, "SDN-based autonomic CCN traffic management," in *Proc. IEEE GLOBECOM*, Dec. 2014, pp. 183–187.
[64] C.-C. Yang, Z.-Y. Chen, B. Xia, and J.-Z. Wang, "When ICN meets C-RAN for HetNets: An SDN approach," *IEEE Commun. Mag.*, vol. 53, no. 11, pp. 118–125, Nov. 2015.
[65] R. Huo *et al.*, "Software defined networking, caching, and computing for green wireless networks," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 185–193, Nov. 2016.
[66] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of Internet of Things," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 46–59, Jan./Mar. 2018.
[67] *Mobile-Edge Computing-Introductory Technical White Paper*, ETSI, Sophia Antipolis, France, Sep. 2014.
[68] W.-S. Kim, S.-H. Chung, and J.-W. Moon, "Improved content management for information-centric networking in SDN-based wireless mesh network," *Comput. Netw.*, vol. 92, no. 2, pp. 316–329, Dec. 2015.
[69] R. Soua *et al.*, "SDN coordination for CCN and FC content dissemination in VANETs," in *Proc. Int. Conf. Ad Hoc Netw.*, 2016, pp. 221–233.
[70] R. Ravindran, X. Liu, A. Chakraborti, X.-W. Zhang, and G.-Q. Wang, "Towards software defined ICN based edge-cloud services," in *Proc. Int. Conf. Cloud Netw. (CloudNet)*, 2013, pp. 227–235.
[71] Q. Chen, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y.-J. Liu, "Joint resource allocation for software defined networking, caching and computing," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 274–287, Feb. 2018.
[72] B. Yi, X.-W. Wang, K.-Q. Li, S. K. Das, and M. Huang, "A comprehensive survey of network function virtualization," *Comput. Netw.*, vol. 133, pp. 212–262, Mar. 2018.
[73] E. G. AbdAllah, H. S. Hassanein, and M. Zulkernine, "A survey of security attacks in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1441–1454, 3rd Quart., 2015.
[74] *Named Function Networking (NFN) Project*. Accessed: Apr. 2018. [Online]. Available: http://www.named-function.net/
[75] S. Gao, Y. Zeng, H. Luo, and H. Zhang, "Scalable area-based hierarchical control plane for software defined information centric networking," in *Proc. IEEE Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2014, pp. 1–7.
[76] S. Gao, Y. Zeng, H. Luo, and H. Zhang, "Scalable control plane for intra-domain communication in software defined information centric networking," *Future Gener. Comput. Syst.*, vol. 56, pp. 110–120, Mar. 2016.
[77] H. Yin, Y. Jiang, C. Lin, and Y. Luo, "Big data transforming the design philosophy of future Internet," *IEEE Netw.*, vol. 28, no. 4, pp. 14–19, Jul./Aug. 2017.
[78] H. Huang, H. Yin, G. Min, H. Jiang, J. Zhang, and Y. Wu, "Data-driven information plane in software-defined networking," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 218–224, Jun. 2017.
[79] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[80] Z. M. Fadlullah *et al.*, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2432–2455, 4th Quart., 2017.
[81] A. Mestres *et al.*, "Knowledge-defined networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 3, pp. 2–10, Jul. 2017.
[82] W.-X. Liu, J. Zhang, Z.-W. Liang, L.-X. Peng, and J. Cai, "Content popularity prediction and caching for ICN: A deep learning approach with SDN," *IEEE Access*, vol. 6, pp. 5075–5089, Dec. 2017.
[83] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.



QING-YI ZHANG received the B.S. degree in computer science from Northeastern University, Shenyang, China, in 2014, where he is currently pursuing the Ph.D. degree. His research interests include future Internet architecture, information centric networking, and software-defined networking.

XING-WEI WANG received the B.S., M.S., and Ph.D. degrees in computer science from Northeastern University, Shenyang, China in 1989, 1992, and 1998, respectively. He is currently a Professor with the College of Computer Science and Engineering, Northeastern University. He has published over 100 journal articles, books and book chapters, and refereed conference papers. His research interests include cloud computing and future Internet. He has received several best paper awards.

MIN HUANG received the B.S. degree in automatic instrument, the M.S. degree in systems engineering, and the Ph.D. degree in control theory from Northeastern University, Shenyang, China, in 1990, 1993, and 1999, respectively. She is currently a Professor with the College of Information Science and Engineering, Northeastern University. She has published over 100 journal articles, books, and refereed conference papers. Her research interests include modeling and optimization for logistics and supply chain systems.

KE-QIN LI (F'15) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1985, and the Ph.D. degree in computer science from the University of Houston, TX, USA, in 1990. He is currently a SUNY Distinguished Professor of computer science with the State University of New York, New Paltz. His research interests include parallel and distributed computing and computer networking.

SAJAL K. DAS (F'15) is currently the Chair of the Computer Science Department and the Daniel St. Clair Endowed Chair with the Missouri University of Science and Technology, Rolla. His current research interests include wireless and sensor networks, mobile and pervasive computing, big data, cyber-physical systems, smart healthcare, distributed and cloud computing, security and privacy, biological and social networks, applied graph theory, and game theory. From 2008 to 2011, he served the U.S. National Science Foundation as a Program Director in the Division of Computer Networks and Systems. He serves as the founding Editor-in-Chief for *Pervasive and Mobile Computing* journal and as an Associate Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING, *ACM Transactions on Sensor Networks*, and several others.