electronics

Review

Leveraging ICN and SDN for Future Internet Architecture: A Survey

Manar Aldaoud [1,*], Dawood Al-Abri [1], Medhat Awadalla [1,2] and Firdous Kausar [1]

[1] Department of Electrical & Computer Engineering, Al-Khoudh, Sultan Qaboos University, Muscat P.O. Box 33, Oman
[2] Department of Computers and Systems, Helwan University, Cairo 12612, Egypt
* Correspondence: s117905@student.squ.edu.om; Tel.: +968-99565201

Abstract: Information-Centric Networking (ICN) and Software-Defined Networking (SDN) are both new evolving network architectures that are receiving a lot of attention from researchers. ICN is a Future Internet architecture which tries to transform the current Internet architecture from location- and host-centric to content-centric, where obtaining requested data is achieved by the contents' names regardless of the location of the data. From another angle, SDN is considered a new Internet architecture that moves the control plane management from network devices to a centralized controller. The SDN controller enhances network robustness and improves its scalability, reliability, and flexibility. The integration of ICN and SDN results in massive benefits, where SDN enhances ICN networks' manageability, controllability, and functionality, and ICN reshapes the SDN design to make it compatible with ICN features and to enhance ICN in terms of network caching, routing, mobility, and security.. In this review paper, a comprehensive survey of the issues and challenges of integrating ICN and SDN is presented. Firstly, ICN's main characteristics are summarized, and a short comparison between different ICN architectures is completed. Then, the key details of SDN are highlighted. Moreover, the motivation and benefits of merging ICN with SDN are summarized and the state-of-the-art work on merging ICN and SDN is reviewed and classified from several aspects. Finally, several open research issues are highlighted.

Keywords: Information-Centric Networking (ICN); Software-Defined Networking (SDN); Named-Data Networking (NDN); Future Internet

Citation: Aldaoud, M.; Al-Abri, D.; Awadalla, M.; Kausar, F. Leveraging ICN and SDN for Future Internet Architecture: A Survey. *Electronics* **2023**, *12*, 1723. https://doi.org/10.3390/electronics12071723

Academic Editor: Dimitris Kanellopoulos

Received: 15 February 2023
Revised: 12 March 2023
Accepted: 17 March 2023
Published: 4 April 2023

1. Introduction

Many new networking paradigms have been proposed to revolutionize the current network architecture. Two promising technologies are Information-Centric Networking (ICN) and Software-Defined Networking (SDN). ICN is a new paradigm that overlays today's Internet architecture. Nowadays, the Internet is host-centric that is based on the data location, i.e., the requester must specify the IP address (location) of the device that holds the required data. In contrast, ICN is a clean-slate design that focuses on the content itself ("what") instead of the data location ("where"), which increases the chance of retrieving the requested data from the network's optimal source. Moreover, the need for many network services is reduced in ICN since there is no need for the data location and IP addresses such as NATing (Network Address Translation) and Domain Name System (DNS) Another emerging technology is SDN, which depends on having an SDN controller that takes all decisions and forwards them to the forwarding devices using a communication protocol. Merging these two new technologies enhances the Future Internet in many ways: reliability, scalability, mobility, flexibility, and robustness. Nevertheless, such merging introduces new challenges and obstacles that must be considered in order to reap the benefits that come from merging ICN and SDN.

Although this is not the first paper that has surveyed the merging of ICN and SDN, our survey differs in many aspects, as summarized in Table 1. Our survey presents a large-scale, thorough, and comprehensive up-to-date review of leveraging SDN controllers in ICN networks. Both surveys [1,2] focus on a specific ICN architecture. The work in [1] focuses on the Named Data Networking (NDN) architecture and briefly introduces some of the SDN-based applications in NDN networks. In contrast, the work in [2] focuses on the Content-Centric Networking (CCN) architecture and reviews works that combine it with SDN from an OpenFlow point of view.

Table 1. A Brief Comparison of Our survey With Current Surveys.

Reference	Year	ICN Model	Main Areas
[1]	2016	NDN	Applications
[2]	2017	NDN	OpenFlow
[3]	2021	ICN	Challenges and open issues
[4]	2018	Multiple models	Applications
[5]	2018	Multiple models	Multiple aspects: deployment of ICN, caching, routing, service chaining, Traffic Engineering, wireless, edge service, and resource allocation
This survey	2023	Multiple models	Multiple aspects and more works than mentioned in [5]: ways and techniques to make OF SDN compatible with ICN, ICN with P4, and POF SDN, Traffic Engineering, routing, forwarding, caching, scalability, satellites networks, 5G networks, multimedia delivery, energy consumption, mobility, and QoS.

A short survey conducted in [3] focuses on issues related to integrating ICN and SDN. However, it only covers a few works and most of these works are before 2018.

The surveys in [4,5] and ours introduce relevant works taking into account different types of ICN architectures. However, the work in [4] focuses on approaches that use SDN to improve the performance of ICN applications.

The most closely related survey to ours is the one in [5]. However, our work differs from [5] in terms of breadth, depth, and analysis point of view. In this regard, the major differences and the major contribution points of our work are shortened as follows:

- We present a brief review and comparison between different ICN platforms;
- We present a thorough and in-depth overview of the approaches that make ICN compatible with SDN controllers in terms of the used technology, e.g., OpenFlow extending with an identifier, wrapping, and overlaying. On the other hand, the work in [5] discusses some of these approaches based on the solution type, either a long-term or short-term solution. Moreover, in this work, we went through approaches that use different languages and protocols to communicate with the SDN controller, such as P4 language and Protocol Oblivious Forwarding (POF);
- We broadly and systematically survey the state-of-the-art that combines SDN with ICN networks. We cover many aspects that are not covered in [5], such as scalability, mobility, multimedia delivery, SDN controller clustering, QoS, energy consumption optimization, and approaches to enhance Satellite–Terrestrial and 5G networks. Moreover, our work reviews many newer works that have been published after the publication of [5]. However, we did not discuss the impact of merging SDN and ICN on securing each paradigm since it is well discussed in [5] and in a survey about ICN security [6];
- We summarize the existing approaches in ICN-SDN and elaborate on their pros and cons;
- We spotlight the concerns and challenges encountered by combining and merging ICN with SDN networks and provide some open research areas that researchers can work on.

The rest of this paper is organized as follows. Section 2 presents a brief introduction to ICN. Section 3 describes the fundamental concepts of SDN. Section 4 introduces the motivation behind merging ICN and SDN. Section 5 summarizes ICN implementation based on SDN, and the enhancements gained due to the merging of ICN-SDN. Section 6 discusses several open issues and future work directions. Section 7 concludes the paper.

2. Information-Centric Networking (ICN)

Once content/data become famous and popular, especially multimedia content, the demand for these data increases dramatically. Consequently, the content producers will face load balancing and bandwidth problems. To overcome these limitations, the idea of a Content Delivery Network (CDN) was introduced [7]. CDN consists of a group of servers that are geographically distributed.

These servers work together to provide fast delivery of Internet content. CDN reduces the bandwidth cost, balances the load on websites, increases websites' security, and increases content availability. Akamai CDN [8] is one of the popular CDNs that content producers are using to deliver their web content. Despite the advantages provided by CDN, it suffers from many limitations, such as: scalability, reliability, quality, and performance accuracy/efficiency [9].

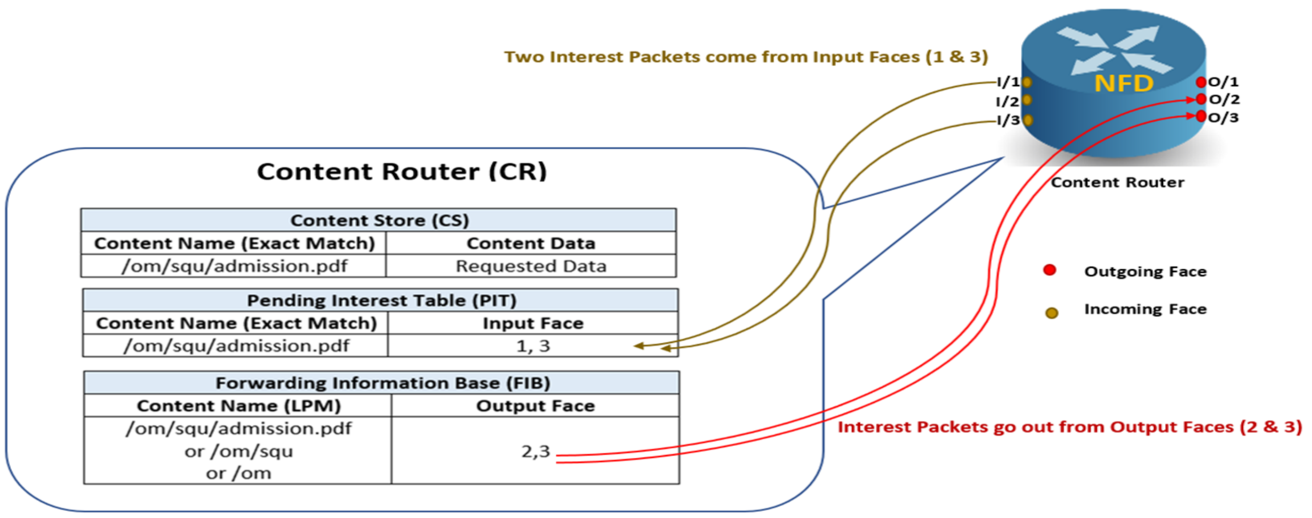
Due to the growing need for efficient and scalable content distribution, and to overcome the limitations mentioned above, there is an increased focus by researchers on Information-Centric Networking (ICN) [10]. ICN was introduced as a Future Internet architecture, which mainly: (1) decouples the information from its location and defines it by a content name, and (2) enables in-network caching [11].

Many research projects have a remarkable interest in content-centric network architectures and come under the broad ICN umbrella: COntent Mediator architecture for content-aware nETworks (COMET) [12], CONVERGENCE Project [13], Data-Oriented Network Architecture (DONA) [14], Publish-Subscribe Internet Technology (PURSUIT) [15], Scalable and Adaptive Internet Solutions (SAIL) [16], Content-Centric Internetworking Architecture (CONET) [17], and Named Data Networking (NDN) [18] which is based on CCNx [19].

Different ICN models can be broadly categorized into two architecture models:

1. Consumer-driven models—such as NDN [20], where the communication between the consumer and the publisher is initialized by the consumer request. In this type, the consumer sends a request to the network asking for a specific content. Once the content publisher receives the request, it will send the requested content to the requesting consumer;
2. Publisher-Subscriber models—such as DONA [14] and SAIL [16], where the publisher sends an announcement of the data it has. Then, the subscriber asks the publisher for the needed data.

NDN node is an example of ICN data exchange [20], since NDN is one of the prominent ICN architectures that many researchers are focusing on as a Future Internet architecture. NDN node is called the Content Router (CR) [21]. CR contains three main components, as shown in Figure 1:


Two Interest Packets come from Input Faces (1 & 3)
I/1
I/2
I/3
NFD
O/1
O/2
O/3
Content Router
Outgoing Face
Incoming Face
Interest Packets go out from Output Faces (2 & 3)
Content Router (CR)

Content Store (CS)	
Content Name (Exact Match)	Content Data
/om/squ/admission.pdf	Requested Data

Pending Interest Table (PIT)	
Content Name (Exact Match)	Input Face
/om/squ/admission.pdf	1, 3

Forwarding Information Base (FIB)	
Content Name (LPM)	Output Face
/om/squ/admission.pdf or /om/squ or /om	2,3

Figure 1. NDN Content Router Structure.

Tables: each CR has three main tables:

- Content Store (CS): CS is used in NDN networks to cache a copy of the Data packet in the traversed CRs. Each entry in CS contains cached data that are identified by the content name. CS has a vital role in NDN networks since it helps in: (1) network load balancing, (2) reducing the load on the producer, (3) saving bandwidth, (4) reducing content retrieval time, and (5) improving mobility and loss recovery;
- Pending Interest Table (PIT): this is responsible for adding entries to each received Interest packet until its requested data arrive or the entry's lifetime expires. Each PIT entry is identified by a name prefix and has a concatenated list of input Faces of the received Interest packets. PIT enables many capabilities in NDN: (1) multicast data delivery, (2) load balancing and control flow, (3) security, and (4) loop-free;
- Forwarding Information Base (FIB): this is a name-based lookup table. Each FIB entry is identified by a name prefix and has an ordered list of output Faces that reflects the next hop. FIB's role can be summarized as follows: (1) supports NDN in Multipath Forwarding, since each FIB entry has multiple next-hops [20], and (2) saves Round Trip Time (RTT) per interface that is taken and refreshed every time FIB receives a Data packet, which helps in estimating path performance;
- Faces: a face is a general name for a network interface. Interest and Data are sent and received through these faces. The face could be [21]: (1) direct connection between local network nodes via Ethernet, (2) overlay communication channel to remote nodes using TCP, UDP, or Websocket, or (3) inter-process communication channel to a local application on the same node via Unix sockets;
- Forwarder: this is called Named Data Networking Forwarding Daemon (NFD) and has been implemented and developed by the NDN team. NFD has many responsibilities, the main ones are to: (1) implement NDN face abstraction, (2) implement CR tables, (3) implement the forwarding plane that supports multiple forwarding strategies, and (4) manage Routing Information Base (RIB) table and synchronize routes in the RIB with FIB table.

NDN relies on two types of packets: Interest packets and Data packets. The consumer sends an Interest packet with the name of the requested content to the CR. Initially, the CR checks its CS for the requested content. If the needed data are not available in CS, the PIT is checked to see if another consumer requested the same content before, or if this is the first time. If the received Interest requests the same data as another Interest, it is aggregated into the existing PIT entry with the new consumer face number. Otherwise, a new entry is added to the PIT table that holds the content name in the Interest packet and the consumer's face number. Then, the Interest packet is forwarded to the FIB table, which will be forwarded to the associated face/s, multicast, or dropped according to NFD's strategy Once the Interest packet arrives at the content producer, the producer replies with a Data packet that contains the requested data with the same content name of the Interest packet, using symmetric communication (reverse-path) to reach the consumer. Once the Data packet reaches the NDN node, it caches a copy of the content in CS and forwards the Data packet to the requested consumer. The forwarding process between a consumer and a publisher at the NDN node is shown in Figure 2.

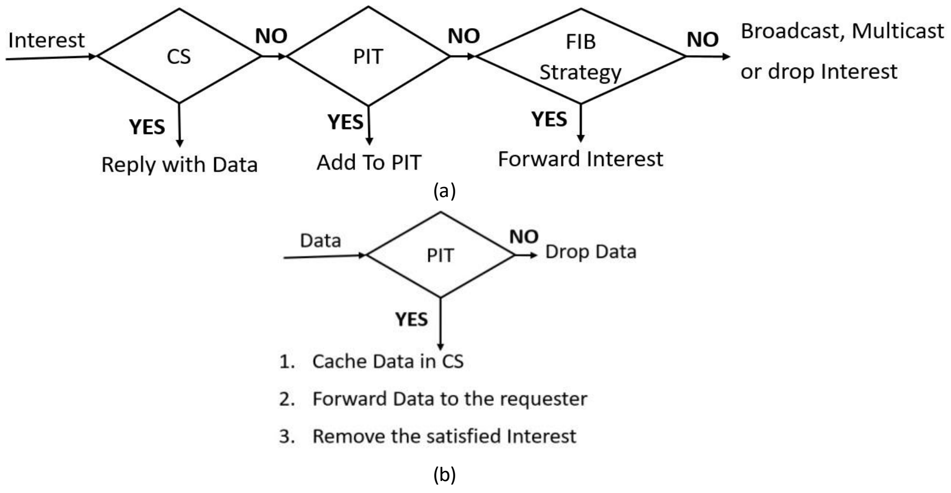


Figure 2. NDN Packet Forwarding Process, (**a**) Interest, (**b**) Data.

Naming, routing, caching, security, and mobility are common characteristics that distinguish ICN models from today's networks. Brief descriptions of these characteristics are as follows:

- Naming: requesting and retrieving data in ICN is based on the information name instead of its location. Most ICN models replace URL hierarchy names with flat names [14–17,22]. Conversely, NDN names are hierarchically structured and human-readable [20]. The user may either hash the whole content name or part of the content name with the SHA-256 cryptographic hash algorithm [23].
- Data Routing: since ICN is location-independent, retrieving the requested data is based on matching data with the requested name. In some ICN models such as NDN [20], data from producer to consumer traverse the same path that the request took from the consumer to the producer, i.e., the same path with a reverse direction. In other models, the requested data do not need to traverse request paths such as for DONA [14].
- Caching: In-network caching is one of the ICN characteristics that becomes available since the requested content is decoupled from its location. On-path caching is a default case, while off-path caching in many ICN models requires extra settings and registrations [24,25]. In-network caching is achieved by caching data on the ICN nodes between the consumer and the producer. Consequently, this in-network caching reduces the load on the producer and increases data availability. Moreover, caching plays a significant role in network load balancing, reducing the speed of data fetching, and retransmitting lost data packets. On the other hand, when the ICN node is full, a critical decision of which data packet(s) must be expelled has to be taken. Hence, this eviction must not be of popular data packet(s) and must not be performed frequently because this reduces CS performance. Replacement is mainly executed based on content popularity and priority. Many replacement algorithms can be used, such as RaNdom replacement policy (RND), First In-First Out (FIFO), Least Frequently Used (LFU), and Least Recently Used (LRU) [25–28].

- Security: Host-centric architecture suffers from many security issues that can be avoided in ICN models. Spams, Distributed Denial of Service (DDoS), and fake data are avoided in ICN implementations since each Interest is served with one data packet forwarded from aggregation points, i.e., CR, Resource Handler (RH), Resource Manager (RM), etc. Moreover, instead of securing the channel between the producer and the consumer in ICN, the content producer protects and signs the content itself. Hence, the consumer and the aggregation points verify content validity by designated keys that have been published at the initialization stage. However, new attack types that are targeted by ICN models have appeared, such as cache pollution, Interest flooding, and others [29].
- Mobility: the data can be received using different interfaces: Wi-Fi, 4/5G, Ethernet, Bluetooth, and others. Interest packets of the unreceived data must be resent when the consumer moves in some cases. For example, if a movable consumer receives the requested data and then changes its place, it needs to resend the request again to find a new publisher. Conversely, when the publisher changes its location, the intermediate devices of the previous and new networks must update their tables with this change, i.e., FIB tables, RM, and RH [24,30].

Table 2 summarizes a comparison between some of the most popular ICN models in terms of the ICN model's main components, naming, forwarding, and Interest/Data routing. The survey in [31] includes detailed information on different ICN models.

Table 2. Popular ICNs Platforms.

Abbreviation	DONA [14]	PURSUIT [15]	SAIL [16]	CONET [17]	NDN [18,20]
Full Name	Data-Oriented Network Architecture	Publish/Subscribe Internet Technology	Scalable and Adaptive Internet Solutions	Content-Centric Inter networking Architecture	Named Domain Networking
Components	Resource Handler (RH) per Autonomous System (AS): Caching Establish a routing path Publisher data registered in it	Per (AS) 1. Rendezvous Node (RN): receives Data and Interest packets 2. Topology Manager (TM): establishes routing paths 3. Forwarding Node (FN): Forwarding and caching	1. Two types of Name Resolution System (NRS): Local name resolution and Global name resolution 2. Content Router (CR): establish routing path and caching	1. Serving Node (SN): caching 2. Name System Node (NSN): name resolution	Content Router (CR): Content Store (CS) Pending Interest Table (PIT) Forwarding Information Base (FIB)
Naming	Flat naming	Flat naming	Flat naming	Flat naming	Hierarchical naming
Forwarding	Uses IP-based Forwarding	Uses Name-based Forwarding	Uses Name-based Forwarding	Uses both IP-based and Name-based Forwarding	Uses both IP-based and Name-based Forwarding
Interest Routing	From consumer to the publisher through RH	From consumer to the publisher through RN	From consumer to the publisher through local and global NRS	From consumer to SN through NSN	From consumer to the publisher through CR
Data Routing	From publisher to consumer through routers using the path established by RH	From publisher to consumer through FN using the path established by TM	From SN to consumer through content routers	From SN to consumer through routers	From publisher to consumer through CR using the same path Interest went through

3. Software-Defined Networking (SDN)

The number of network protocol standards is increasing and becoming more complex because of the increase in the number of network-connected devices and network-based services, which in turn mandates the upgrade of current protocols or the introduction of new ones. Consequently, the network cost increases, solidity decreases, and network administrators need to manage these enormous networks efficiently and cost-effectively. To avoid all the aforementioned, administrators switch from distributed control in traditional networks to centralized control using Software-Defined Networking (SDN) controller. Figure 3 demonstrates the differences between traditional and SDN networks. In traditional networks, both the data plane and control plane are coupled with forwarding devices, switches, and routers, i.e., forwarding devices are the decision-makers. On the other hand, in SDN networks, the data plane and control plane are decoupled, where the SDN controller has the control plane and is the decision-maker, i.e., it takes the actions of adding and removing flow entries in flow tables in the forwarding devices [32].

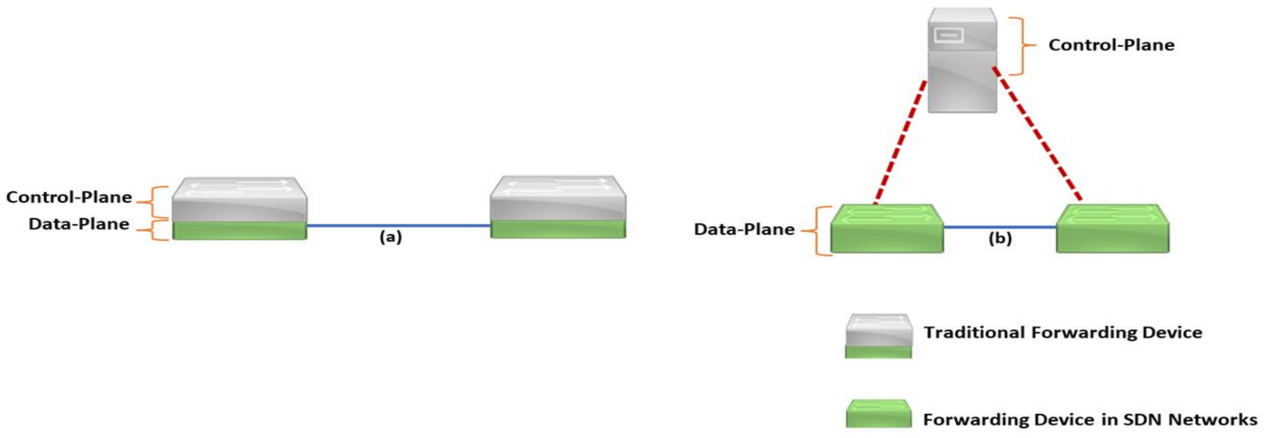

Control-Plane
Data-Plane
(a)
Control-Plane
Data-Plane
(b)
Traditional Forwarding Device
Forwarding Device in SDN Networks

Figure 3. (**a**) Traditional Network architecture, (**b**) SDN Network architecture.

The SDN network consists of three layers: the data plane layer, the control layer, and the application layer, as shown in Figure 4 [33–35]. The data plane layer contains the forwarding and switching devices. Southbound Application Programming Interface (API) connects data plane devices to the SDN controller(s) in the control layer by exchanging network information and packet forwarding rules. Although there are some Southbound protocols, such as Forwarding and Control Element Separation (ForCES) and Network Configuration Protocol (NetConf), which are developed by the Internet Engineering Task Force [36,37], and OpFlex, which is developed by Cisco [38], OpenFlow Protocol (OF Protocol), which is maintained by the Open Networking Foundation (ONF), is the most famous southbound API [39]. The control plane layer is a software logic that has a global view over the network and based on this view, it takes decisions that control the traffic. This layer may contain a cluster of controllers that communicate via eastbound and westbound APIs with each other. Moreover, this layer works as a bridge between the lower layer "Data plane" and the upper layer "Application layer." The application layer contains applications and network services such as traffic load balancing, QoS policies, security applications, and Traffic Engineering (TE) [40].

SDN depends on the programmable separation of the control plane from the data plane, which enables software control of the network forwarding. SDN has four essential features [41], as highlighted in Figure 4: dynamic flow control, network-wide visibility with centralized control, network programmability, and a simplified data plane.

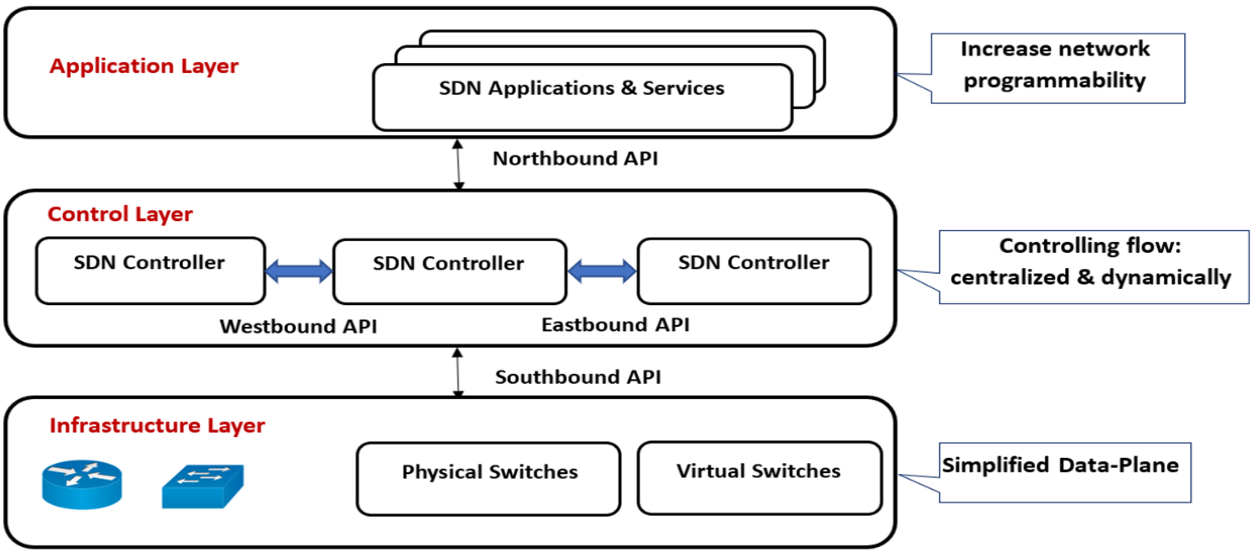


Figure 4. SDN Layers.

Data plane devices, that are connected to the SDN controller, use a table called the flow table. Each flow table consists of entries where each entry specifies how to handle the received traffic. The SDN controller takes decisions about adding/deleting flow entries to/from the flow tables of the switches based on the program or the protocol running on the controller. SDN architecture can run one or multiple controllers. A large enterprise network would need multiple controllers to maintain a greater size of the infrastructure. To increase the system availability, multiple controllers communicate and synchronize via the east/westbound interface. Various SDN controllers have been developed and emerged, such as NOX, POX, Ryu, ONOS, Floodlight, and ODL.

4. The Motivation of Hybrid Paradigm ICN-SDN

The main motivation for having a hybrid paradigm of ICN-SDN is to benefit from each architecture. This hybrid architecture is created by making ICN networks exploit the SDN approach of centralized control of network resources to optimize resource utilization and enhance network security, scalability, and large-scale deployment. This is achieved by utilizing an SDN controller in monitoring and controlling: (1) network policies, (2) forwarding and routing requested data to consumers, (3) data name and flow matching, and (4) in-network caching.

Furthermore, the SDN controller exploits ICN characteristics: (1) to secure the SDN controller from some attacks such as DDoS and spoofing, and (2) to enhance SDN controller performance by reducing the traffic passing through the controller because of in-network caching.

Despite the benefits of this hybrid architecture, ICN-SDN merging faces many obstacles including (1) Southbound protocols are by default not compatible with ICN since their matching fields are not based on name matching, and (2) SDN controllers do not support ICN functionality. To utilize the SDN controller in the hybrid architecture, it must be modified to (1) support ICN core functionalities such as forwarding and caching, (2) provide QoS and resource monitoring, (3) optimize the best route selection between consumer and publisher, and (4) optimize caching strategies and caching replacement strategies.

5. Approaches to Enabling ICN Using SDN

Much of the work is directed at utilizing the centralization and a global network view of the SDN controller in ICN networks. The following subsections highlight the modifications that occurred to the SDN controller from several points of view, as summarized in Table 3.

Table 3. ICN-SDN Merging Areas and the State-of-the-Art.

ICN-SDN Merging Areas	Related Works
Modifying OpenFlow	[42–45]
Intermediate Layer	[46–51]
Satellite-Terrestrial	[52–56]
Enhance 5G Networks	[57–61]
Traffic Engineering	[62–68]
Routing and Forwarding	[69–88]
P4	[46,83,84]
QoS	[52,58–68,81,85]
POF	[54,76,86]
Caching	[89–103]
Scalability & Mobility	[104–110]
Miscellaneous	[111–115]

5.1. Modifying OpenFlow Protocol and Switches

Some approaches are based on extending the OF Protocol with extra metadata and modifying the forwarding devices. The OF Protocol was extended in [42] with extra metadata to match with the extended Berkeley Packet Filter (eBPF) filters of NDN content names. The OF-switches were enhanced to deal with these changes, and the Ryu controller communicates with the NDN domain via IP tunnels.

Rajendran [43] extended and modified OpenFlow to embed ICN. This was achieved by modifying the OF-switch to include CS, PIT, and FIB tables. In addition, new messages and actions were added to OpenFlow to deal with switch modifications. The used controller, POX, was also extended to have a routing database. The data were also cached in the controller to reduce RTT, where the switch declares the content availability, aggregates the Interests, and sends them in one message to the controller to reduce the load on the controller.

Guesmi et al. [44] implemented NDN based on the SDN paradigm. They modified the SDN OF switches by implementing three different tables: CS, PIT, and Management Information Base (MIB). Moreover, they implemented four tables in the SDN controller: CS, FIB, MIB, and Data Information Base (DIB). Initially, switches populate their MIB tables by advertising Link Layer Discovery Protocol (LLDP) packets to all connected interfaces. Then, they send their MIB tables to the controller to fill its MIB table. Moreover, any producer in the network sends the name prefix of the produced content using "AddPrefix" packet to its switch, which will inform the controller of the prefix name to add it to its DIB table. Finally, the controller constructs its FIB table based on the information in the MIB and DIB tables. Based on their results, SDN in the NDN network reduces the memory and CPU consumption and the utilized bandwidth and increases the cache hit ratio. However, their implementation lacks much information and details in terms of the role of CSes in the controller and switches, and the structure of the "AddPrefix" packet and how to exchange it.

Moreover, Liu et al. [45] modified OF switches with CS and PIT and fused FIB tables with the OF flow table. Additionally, they extended OF the protocol with three Proto pipelines: Register, Interest, and Data. Furthermore, they encapsulate the NDN packets in Multi-Protocol Label Switching (MPLS), where the content name and chunk number are mapped to the MPLS label, and the Interest and Data contents are uploaded in the IP packet payload. They tested their design using Mininet and Ryu controllers. They noticed that the processing time in the controller and the lookup time in the CS and PIT is 2% more than needed to retrieve content in NDN networks without SDN. They also compared downloading data with different popularity and sizes, and they noticed that the performance is better for high-popularity elephant flow. However, the design was not tested on bandwidth, memory, and CPU utilization.

5.2. Adding Intermediate Layer

In this approach, an intermediate layer is used to resolve names to IP addresses. This intermediate layer can be implemented as a proxy and overlay. Nowadays, most networks have Proxy servers. Some studies utilize the Proxy server to deal with ICN networks. Feng et al. [46] used Programming Protocol-Independent Packet Processors (P4) to improve the HTTP request process by leveraging ICN, where the SDN controller is just used normally to inject rules in forwarding devices using P4. This was achieved by: (1) proposing a packet conversion agent in the proxy application in both client and server to convert HTTP requests from IP to ICN-P4 format and vice versa for the HTTP response, (2) designing ICN forwarding action in the forwarding device using P4 language, and (3) constructing PIT and CS tables in the forwarding device using P4. Therefore, once the P4 switch receives the converted HTTP request, i.e., ICN-P4, it checks its CS database. If it finds a match, the request is forwarded to the application that provides the cache service, i.e., the P4 switch does not store the data but records the state. If the requested data are not cached, the implemented forwarder checks the PIT table. If the name is already in PIT, the P4 switch updates PIT and drops the packet. Otherwise, the P4 switch forwards the packet. To make P4 switches capable of adding multiple requests in the PIT table, they use many one-dimensional arrays to implement multidimensional where the indices are the hashed values of the requested names. Even though their solution improves the transmission efficiency of the HTTP traffic, it may increase the total overhead, especially on the forwarding device, because of the conversion and hashing processes.

The network presented in [47] consists of multiple proxies. Each proxy includes one of the Distributed Hash Tables (DHT) that is shared with other proxies. This DHT is used as the content index. The controller forwards the received Interests to the closest proxy, which handles the retrieval of the requested data. If the proxy finds a match in its DHT, it forwards the Interest to the cache node that holds the requested data. Otherwise, it forwards the Interest to any cache node that, in turn, forwards it to the closest proxy server that may have requested data. This approach may overload the proxies since they resolve and forward all received Interests to the controller.

The work in [48,49] used a proxy to wrap and hash messages between the CCNx daemon and the OF-switch. The wrapper module in the proxy hashes the content name prefix in the Interest packet and puts it in the existing IP packet. However, the authors in [49] took off-path caching into consideration. This was achieved by implementing three modules on the controller to handle the ICN flows: (1) Manager to check the most popular content, (2) Optimizer to optimize and minimize the delay for cache calling and (3) Deflector to map the optimized results of the cached content with interfaces toward the node cached this content. Their solution solves the problem of cache replication and increases the hit ratio. On the other hand, the wrapper introduces a small delay and slightly decreases the forwarding efficiency by just 5%.

Overlaying ICN on the existing IP network may increase the complexity of IP routing. The authors in [50] discuss an overlay ICN approach, where the weighted graph theory was used—Topology Graph (TG)—to implement their topology where nodes are gateways, one per network, and links are established as overlay links between gateways. The controller is connected to multiple gateways in multiple NDN networks to dynamically construct and manage the ICN overlay. It also manages and updates FIB entries in the ICN gateway in the following cases: (1) add a new gateway: a request message is sent from the new gateway to the controller—with an assumption that the new gateway knows the controller IP to be connected. The controller informs the nearby gateway and updates their FIB, tables (2) add a new publisher: a request message with a new prefix is sent from the gateway—the new publisher is connected- to the controller to define the next hop by calculating the shortest path. Then, the controller adds a new FIB entry to this gateway with a new prefix, and (3) detects failure: the gateway sends a failure message to the controller, which checks connectivity between the consumer gateway and the publisher gateway. If any failure is detected, the controller updates the failure value in TG, recalculates the shortest paths for all publisher nodes, and updates FIB tables. If the failure value of a link exceeds a threshold value, the controller modifies the TG to balance the load, and then it updates FIB tables. According to the simulation results, the delay and content retrieval time is less. On the other hand, choosing the proper threshold value to modify TG depends on many parameters, which may negatively affect the approach performance.

The authors in [51] also adopted DHT for name resolution where they separated the Edge Service Network (ESN) and core transmission network. SDN technology is used in ESN to implement intelligent data routing and caching functions. IP functions are used in the core transmission network to provide high-speed and wide-area transmission. The global name resolution is performed by the controller utilizing DHTs and OF gateways between the domains. The main drawback of their work is not taking CCN routing specifications when making routing decisions. Table 4 summarizes the approaches that targeted ICN over OpenFlow and P4 as a southbound protocol in SDN.

Table 4. ICN over OpenFlow or P4.

Approach	Related Work	ICN Model	SDN Controller	Advantages	Limitations
1. Extending OF Protocol	[42]	NDN	Ryu	Provides ICN functionalities	OF switches and controllers must be enhanced to deal with changes in OF protocol
	[43]	NDN	POX		
	[45]	NDN	Ryu		Insufficiently tested
2. Extending OF Switch	[44]	NDN	SDN-Like (OF)	Enables content name processing in OF network	Standardizes OF switches from different vendors
	[45]	NDN	Ryu		
3. Using P4	[46]	NDN-Like (HTTP)	SDN-Like (P4)	Improves the transmission efficiency of HTTP traffic	Increases overhead especially on the forwarding device because of the conversion and hashing processes
4. Using Proxy	[47]	ICN-Like (HTTP)	FloodLight	Enables content name processing in OF network	1. Expensive because it requires extra devices such as proxies; 2. Overloads the proxies and forms delay and latency
5. Using Overlay	[50]	NDN	SDN-Like	Enables content name processing in OF network	Overlaying ICN over IP may not utilize all ICN benefits
	[51]	CCN	POX		

5.3. Satellite–Terrestrial Networks

Some works focus on merging ICN and SDN networks to enhance Satellite–Terrestrial networks. The authors in [52] proposed SDN-based ICN architecture for Satellite–Terrestrial networks. Their proposed architecture provides QoS, flexible resource utilization, and seamless communication that increases the quality of backhaul services and decreases the core network burden. On the other hand, some challenges arise because of this integration, such as: signaling, energy efficiency, caching mechanism, and security.

Liu et al. [53] proposed a user-driven cache replacement strategy for ICN–Satellite–Terrestrial Networks. This proposed strategy is based on the content popularity and cache capacity of the nodes to enhance and optimize the intra-network cache. They used the high-orbit satellite as an SDN controller that is responsible for controlling, managing, and updating routing, cache policies, and mobility.

The authors in [54] proposed an SDN-based ICN architecture for Satellite–Terrestrial Integrated Networks (STINs) that provides flexible management and efficient content retrieval for STIN. Their architecture consists of a three-layer satellite network: (1) a control plane (GEO) that contains a master controller, (2) a forwarding plane (MEO), and (3) Access-plane (LEO). Each plane has satellites that have different functionalities. GEO satellites are considered master controllers that are responsible for: (1) routing strategy calculation, (2) caching updates, and (3) mobility management. MEO satellites are considered assistant controllers that are responsible for: (1) routing policies between master and assistant controllers, and (2) some of the master controller responsibilities of routing strategy calculation and caching updates for the access satellites. LEO satellites are used to achieve better signal quality and a lower round-trip delay. They also contain an on-board switch based on POF to achieve compatibility with ICN. Each switch is equipped with a Cache Region that contains the cached content and the content's popularity. Moreover, to achieve better efficiency in caching and content retrieval, a neighbor cache sharing and a coded caching scheme are also proposed. Their work achieves significant traffic-load reduction. However, the performance is enhanced when the neighbor cache sharing is one-hop away and needs more improvement if the neighbor cache sharing is multi-hops away.

In addition, the authors in [55] extended the work in [56] and focused on implementing ICN-SDN architecture for Satellite–Terrestrial networks to improve the routing of Big Data. This is executed by implementing a modified NDN router with modified FIB and PIT tables. The proposed routing algorithm, called the Virtual Node Matrix Routing algorithm (VNMR), is placed in the main controller, and it is responsible for generating forwarding paths according to FIB and PIT tables. VNMR obtains the routing matrix based on the relative orientation of the source and destination nodes. Using VNMR helps in reducing the spatial complexity and the time complexity of the routing algorithm. Their architecture improves the routing efficiency in terms of request delay, request aggregation, FIB update speed, and avoiding congested and failed links. Even though this architecture uses LRU cache replacement, it leads to filling cache space. In other words, to achieve better results for the whole architecture, a new satellite cache replacement strategy is needed. We notice that by combining both approaches [54,55], the ICN-SDN for Satellite–Terrestrial network performance may be improved. Table 5 summarizes the approaches of ICN and SDN in Satellite–Terrestrial networks.

Table 5. ICN and SDN in Satellite-Terrestrial Networks.

Approach	Related Work	ICN Model	SDN Controller	Advantages	Limitations
1. QoS	[52]	ICN-Like (with FIB)	SDN-Like	Increases services' quality and decreases the burden	Initiate new problems: signaling, energy, caching, and security
2. Caching	[53]	ICN-Like	SDN-Like	Better efficiency of caching and content retrieval in STINs	1. Problems with data security 2. Complex system
	[54]	ICN-Like	SDN-Like (POF)		
3. Routing and Forwarding	[55]	NDN	SDN-Like (OF)	1. Decreases delay 2. Improves routing efficiency by avoiding congested and failed links	Fill up the cache space

5.4. Enhancing 5G Networks

Now, 5G is the fifth-generation technology standard for broadband cellular networks, which evolved from 4G cellular networks by providing a very high data rate "up to 10 Gbps", much lower latency, better capacity, and QoS to accommodate the increasing wireless traffic [116].

Some works include centralized control, SDN, and Network Function Virtualization (NFV) to enhance content delivery such as in [117,118]. ICN is an appropriate solution for wireless networks since it supports name-based routing, in-network caching, and mobility. Therefore, 5G can benefit from ICN and SDN paradigms to improve the QoS of 5G wireless networks and reduce latency, where ICN is used to enhance content delivery to mobile users by utilizing its caching characteristics, while SDN and NFV are used to improve network programmability and management. The work in [57] utilizes Mobile Edge Computing (MEC) 5G technology to achieve the fastest data communication capability in the Internet of Things (IoT).

The authors in [58] proposed a solution to enhance the QoS of retrieving Big Data multimedia over 5G. This was achieved by integrating: (1) ICN technology to choose the best cache station, and (2) NFV-based SDN architecture which monitors the wireless information dynamically to choose the shortest path with the lowest noise.

Li, Ota, and Dong [59] proposed an SDN-based orchestration scheme. Their scheme has both traditional, i.e., IP-based, and CCN flows. CCN is used to increase content availability and security by utilizing its characteristics and capabilities, such as: caching and location independence. At the same time, the SDN controller is used to manage and control CCN flow, traditional flows, and computation services at the edge of the network. They implement an SDN-based forwarding strategy that supports both ICN-based and IP-based forwarding. Their results outperformed the traditional structure for small-scale networks.

Vakilinia and Elbiaze [60] utilized ICN caching and proposed two algorithms to minimize latency in 5G backhaul networks: (1) a distributed algorithm that is located in backhaul switches to detect and handle congestion locally and temporarily by using threshold blacklists for IP-addresses and content-names, and (2) centralized algorithm that is located in the SDN controller to treat congestion by selecting the optimal routes dynamically, balancing the traffic load, and orchestrating Radio Base Stations (RBS) all over the backhaul networks. NFV is merged with the SDN controller to manage mobile traffic. Their approach shows better results in terms of latency and application utilization in comparison with other computing-based architectures used for Radio Access Networks (RAN) such as Cloud-RAN and MEC.

Since multimedia video/audio streaming traffic comprises 70 percent of mobile traffic, Zhang, and Zhu [61] proposed an architecture that aims to solve traffic delay issues for multimedia Big Data transmission in 5G networks and to fit in with multimedia time-sensitive and bandwidth-intensive applications. This was achieved by integrating: (1) ICN to utilize in-network caching, (2) NFV to encapsulate the physical layer into multiple virtualized networks to select the best path, and (3) SDN to dynamically reconfigure 5G wireless and radio access resources. To optimize this integration, they also proposed three virtual networks and transmit power allocation schemes to: (1) maximize the effective capacity for a single requester, (2) optimize the aggregate effective capacity and power allocation fairness for multiple requesters, and (3) coordinate noncooperative gaming strategy among all requesters, respectively, that shows a better convergence in comparison with Nash equilibrium.

5.5. Traffic Engineering (TE)

Some researchers employ TE in ICN-SD networks to provide QoS or Quality of Experience (QoE) to service [62,63]. The authors in [63] attempted to achieve QoS for queueing and managing traffic on the chosen route. They proposed two map lists: (1) map and aggregate NDN flows with various types/classes/priorities, and (2) map different types/classes/priorities to different queues using multiple management and scheduling algorithms. NOX controller monitors and analyses flow information. It also regulates map lists by: (1) ranking flows based on priority, (2) choosing a queue scheduling algorithm, and (3) testing the impact on the Interests and contents.

Zhang et al. [64] utilized deep learning (DL) and reinforcement learning to overcome complex issues that occur by having ICN in the network. Moreover, DL is used to optimize traffic bandwidth utilization and load balancing in SDN networks. They implemented an intelligent TE (iTE) in the SDN controller by applying: (1) DL technique to take advantage of the correlation between bandwidth demand and content names, i.e., the needed bandwidth can be estimated because the requested data are related to the content name, (2) bloom filter to collect in-network cache information, and (3) reinforcement learning in an implemented module, Parallel Decision-Making (PDM), to make TE decisions adaptively. However, iTE faces data security and privacy problems. Moreover, the authors noticed that deep learning is influenced by data quantity and quality.

Liu et al. [65] proposed a centralized control edge computing (EC) architecture for video streaming in NDN networks. Videos are divided according to the content and user's interest and then distributed to different EC nodes to avoid redundant cache and save cache resources. Since EC nodes suffer from limited caching and computing resources, the need for an SDN controller appears, which is used to offer a solution for content scheduling problems under resource constraints. According to the results in [65], the proposed architecture enhances QoE and reduces both transmission delay and bandwidth utilization. However, the name rules provided by data providers and users must be taken into consideration in the moving process.

Abar et al. [66] proposed a fog computing approach to improve the QoE on the Internet of vehicles for users requesting video multimedia. They proposed a cache-node selection algorithm that uses SDN strategy and cloud-based technology, Fog Computing, in the ICN network.

Hayamizu et al. [67] proposed an elastic Function Offloading Network (FON) that provides QoE for streaming users and service reliability for Service Function Chaining (SFC). However, the implemented design does not take into consideration the resource availability of service functions, multipath routing, and load balancing.

The work in [68] is a sample of managing IoT networks by combining ICN-SDN networks to enhance users' QoE and to increase network performance and security. The authors in [68] proposed multiple controlling planes: Operational, Tactical, and Strategic. The operational plane is an ICN architecture smart-data plane that contains controllers which control local clusters. The tactical plane manages clusters in the operational plane, and it does preprocess tasks before sending data to the cloud. A strategic plane is a cloud-controlled environment that manages the entire network. Moreover, each network cluster in each plane has three controlling units: Model, View, and Control (MVW). MVW units help in separating and controlling functionalities. They found that separating controller functionalities over MVW units improves the controllers' performance and allows concurrent task processing. While having in-network caching in ICN architecture decreases the traffic of actions distributed from the controller to the switch. Table 6 summarizes the approaches targeting QoS when ICN and SDN are combined.

Table 6. QoS in ICN and SDN.

Approach	Related Work	ICN Model	SDN Controller	Advantages	Limitations
1. Enhancing 5G Networks	[58]	ICN	SDN-Like	Optimizes QoS of 5G networks in terms of latency and content delivery	May face instability problems in large-scale wireless networks
	[59]	CCN	Ryu		
	[60]	NDN-Like	SDN-Like (OF)		
	[61]	ICN	SDN-Like (OF)		
2. Having Map-lists	[63]	NDN	NOX	Load balancing, prioritizing flows, and optimizing path BW utilization	It may increase the system's complexity and produce a delay
3. Artificial Intelligence (AI)	[64]	NDN-Like	SDN-Like	Optimizes the whole network performance and load balancing	May increase system complexity

Table 6. *Cont.*

Approach	Related Work	ICN Model	SDN Controller	Advantages	Limitations
4. Video Streaming	[65]	NDN	SDN-Like	Express the user's interest preferences in real-time for video streaming	Video names are not the same for the data provider and the requester
	[66]	NDN	SDN-Like (OF)		
5. Service Function Chaining	[67]	CCN	SDN-Like (OF)	Provide service reliability and video streaming QoE	Not support multipathing, and load balancing

5.6. Routing and Forwarding

Many research areas focus on having a controller to optimize ICN routing. In [69,70], an SRCS architecture was proposed. SRSC is an NDN clean-slate implementation that uses NDN messaging. First, the controller knows all nodes in its domain and the edge nodes of the adjacent domains. Then, it fills its content table, which combines the content name with the nodes that store this content. Each CS in SRSC informs the controller if any content is added or removed. Hence, the controller updates its content table. When a node receives a request, and its CS does not have the content, or its FIB table does not have the forwarding entry for this request, it will send this request to the controller. Based on the content table, if the content is within the controller's domain, the controller computes the shortest path between the requesting node and the node having content. Otherwise, the controller computes the shortest path between the requesting node and the edge nodes of adjacent domains. Then, the controller replies with a forwarding rule to nodes along the reverse path. Accordingly, large networks must be divided into subdomains and have multiple distributed controllers to avoid overloading one controller. According to the results in [69,70], the number of network messages is reduced, and the cache hit ratio is increased.

The Controller-based Routing Scheme for Named Data Network (CRoS over NDN) was proposed in [71–74]. CRoS provides a control layer above NDN. This approach limits the Interest flooding, which reduces the routing overhead. The controller's responsibility is to make routing and forward decisions based on signaling information on content names in the received packets. First, the controller identifies the consumer node and the producer node. Then, it calculates the shortest path between them based on the number of sequenced nodes. Then, a new FIB entry is added to each FIB table in each node between the consumer and the publisher. One drawback of this approach is that FIB entries need to be updated if the network topology is changed. Furthermore, this approach produces congestion in the network because the controller must parse the signaling information. The suggested solutions in [69,71–74] add a burden on the controller since the routers consult the controller on each Interest packet.

Son et al. [75] used the concept of Content-Centric, NFD, and SDN to implement a forward strategy. This was achieved by having a Forwarding Table (FT) (instead of PIT and FIB in the original NDN), Content Update Table, and Content List Table in multiple distributed controllers. The proposed FT has the following components: (1) content name, (2) router ID, (3) cluster-ID, (4) requesting node, and (5) requesting cluster. Once a router receives a request, it initially asks the local controller about the requested content location. If the controller finds the requested content name in its Content List Table, it will notify the router, which will modify the Interest packet to the content location. Otherwise, the router modifies the Interest packet with a default value to route the Interest to the next cluster. Once the Interest reaches its destination, the publisher asks its local controller about the requester location that is saved in the controller's Content Update Table. Finally, the values in the Content Update Table are recorded in the Content List Table when the requester receives the Data packet. This approach overloads the controller with many processes: (1) process each Interest, (2) process each Data packet, and (3) update its tables.

The authors in [76] increased the CCN routers' flexibility by proposing an SDCCN architecture that supports a programmable forwarding strategy and caching policies. The controller and CNN-switch in SDCCN architecture communicate through a Protocol Oblivious Forwarding (POF) instead of an OF protocol. POF was chosen because (1) it supports OpenFlow messages and load balancing, (2) it can process variable-length content names, and (3) it eliminates the need for hashing names since it can process the bytes of packets. The POF flow table is used to implement the FIB table, which contains three fields: rules, actions, and statistics. The rule field contains a content name and a bit mask that will indicate whether the corresponding bit of the content name will be considered during the forwarding lookup or not. This lookup is performed by comparing the bytes presented in rules in the rule field and the bytes of a field, defined by the tuple search keys {offset, size}. Moreover, the authors extend the switches with a new Cache Rules Table (CRT), which is responsible for storing cache rules that determine which content should be stored in the cache. The controller in their prototype is in charge of: (1) installing a forwarding strategy to switches, (2) managing cache policies and CSs, and (3) updating and modifying SDCCN tables, i.e., PIT, FIB, CS, and CRT. However, using bit masks and bit-wise search keys increases the complexity of the proposed architecture.

The main concept of decoupling in SDN was also used in [77]. The controller checks and monitors the proposed routing strategy protocol to solve some NDN issues such as scalability, bandwidth consumption, and latency especially caused by FIB updating, i.e., NDN uses a flooding mechanism to update FIB. This is mainly performed by implementing multiple tables in network switches and controller. The implemented main table is Flow-FIB. The controller used its global view of the network to implement this table for each switch, which includes one or more actions for each prefix to forward the Interest to its request. Then, it sends these actions to the Flow-FIB table in the network switch. The switch does not have to communicate with the controller for each Interest, since it has the corresponding action of the received prefix in its Flow-FIB table. This reduces wasted time and decreases the load on the controller. Moreover, this work proposed a cache replacement policy within the switch itself. This decision is taken based on the content popularity retrieved from statistical information from the Flow-FIB table.

An SDN-like intradomain routing (SDAR) is proposed in [78], where the controller uses its holistic view of the network to collect network information from nodes using NDN packets. Then, it utilizes this information to calculate the best path for each node by using either single-path or multi-path routing algorithms. As a result of having all computations in the controller instead of taking place in nodes, multi-path routing in SDAR enhances routing performance because it reduces traffic overhead and dynamically responds to network topology changes and link failure.

Luo et al. [79] targeted the Couple Service Location and Inter-Domain Routing (CoLoR) ICN model. CoLoR has a node called Resource Management (RM) in each domain that peers with another RM of another domain. Content reachability in its domain is RM's responsibility. They introduced a logical controller called Network Controller (NC) that has the following responsibilities: (1) managing one or more domains, (2) taking decisions on where to cache contents, and (3) routing computation by utilizing two identifiers: (a) Service Identifiers (SID) that identify services and (b) Path Identifiers (PID) that identify flows. Their work tried to utilize the concept of having a logical controller in the network that does not have an OF-Controller.

A multipath forwarding strategy of Interest pipeline distribution that is controlled by a centralized controller was proposed in [80]. The strategy enhances parallel data retrieval from multiple on-path or off-path routers. The controller utilizes its global view on the network to (1) compute the forwarding strategy, (2) create and manage a map of the current state of NDN on-path and off-path routers, and (3) specify the pipeline depth per router. The controller communicates with routers via two types of messages: (1) a forwarding message, where the router informs the controller that it does not have the requested data, and (2) an update message, where the router informs the controller of the newly cached data. This approach has a one-time cost with a little latency overhead that depends on the closeness of the controller to the routers, not on the number of routers. This latency occurs because the consumer must wait for the controller update configuration before connecting to the appropriate router. This can be resolved by placing the controller closer to the router.

The authors in [81] introduced a virtual point between content sources to optimize many-to-many multicasting. They implement Multicast-based Traffic Optimization (MTO) as an SDN-aid. SDN controller has seven modules: three for collecting network statistics, topology, and caching information; and four modules for MTO that are used to implement a tunnel-based multicast tree to aggregate traffic with the same source, destination, and QoS requirements. This tunnel improves network scalability since bandwidth is allocated based on tunnels. The results show that MTO has almost the same performance in terms of bandwidth utilization and load balancing as for normal NDN forwarding and other multicast approach proposed in [82] in small networks with a small number of content sources. In contrast, it shows a better performance in large and complex networks. However, the performance of MTO is not studied in terms of link failure and fault tolerance. Moreover, MTO parameters must be accurately chosen to avoid metric insensitivity.

Madureira et al. [83] proposed NDN-Fabric network architecture that combines SDN and NDN. NDN-Fabric was implemented in P4 language. The proposed architecture combines content-centric and path-based models. NDN Fabric has one logic centralized controller in the core network and distributed controllers in the edge network. This architecture: (1) enhances the NDN scalability of NDN, and (2) improves packet forwarding efficiency by using multipath forwarding, route segmentation, route redundancy, and load balancing.

In [84], the authors tried to enhance the main NDN intradomain routing protocol, Named-data Link State Routing (NLSR). They introduced a routing mechanism that combines NLSR with SDN by utilizing the P4 language. The SDN controller uses its global view of the network to establish resource-location mapping, and it is responsible for processing NLSR packets. Leveraging the SDN controller reduces NLSR's convergence time and the number of exchanged routing packets.

The approach in [85] has multiple distributed controllers, where each controller has two tables to maintain network information: (1) a local content forwarding table and (2) an adjacent content forwarding table. The controller decides the optimal route based on the tables' information and by using the Genetic Algorithm (GA). The algorithm takes into consideration QoS constraints and aims to optimize the whole network performance. This approach also has content gateways between NDN and IP networks that are responsible for protocol conversion. The Interest packet is forwarded to the controller if there is no matching in PIT, CS, and FIB, respectively. The controller chooses the optimal path in its local NDN domain. If it does not find the content that Interest requested, it will forward the Interest to an adjacent NDN domain. Otherwise, the Interest is posted to the internet through the gateway.

A prototype called SPARC was proposed in [86]. SPARC integrates ICN into the POF network by adding ICN_Protocol filed to the Ethernet frame header to distinguish ICN packets from other packets such as IP packets. SPARC has a cluster of controllers that are implemented in a hierarchical structure, global and regional controllers, and work in hybrid mode, i.e., combining both reactive and proactive modes. Each controller has five modules: (1) device discovery module to store information about network devices, (2) link discovery module to detect interconnected links between POF switches, (3) topology management module to create a holistic view of the network, (4) message communication module to create a synchronous horizontal channel between global controllers and asynchronous vertical channel between a global controller and its associated regional controllers, and (5) ICN routing and mobility module to compute the best path to requested content in a stable and mobile network. In SPARC, all ICN hosts must first inform their regional controllers of their content resources. Then, when any host requests some content, the first POF switch replies with the requested data if it caches these data. Otherwise, the

POF switch forwards the request to the regional controller, which in turn asks the global controller. If the global controller finds the requested data in its domain, it will choose the best provider, the best path, and where to cache these data. If the data are not in their domain, the global controller will communicate with another global controller via the horizontal channel to calculate the best path and route data from the neighbor domain. Once a host changes its place, one of the two scenarios may occur. In the first scenario, the host changes its location but within the same domain. In this case, the detached POF device informs the regional controller, which in turn informs the global controller. The latter will dynamically redirect the requesters to the new position. In the second scenario, the host changes its location to another domain. In this case, the global controller of the host's previous location will forward the requests to the global controller of the new domain, which will recalculate the best path and inform the other global controllers.

The authors in [87] divided ICN into two communities based on the Interests' similarities. They implemented a clean-slate network using NDN messages and multiple SDN controllers. Each controller is responsible for its cluster Interests, community division, and routing decisions. There are two types of routing: (1) intracommunity routing based on the same community information, and (2) intercommunity routing based on the social relationship among communities.

Zhang et al. [88] also partitioned the domain into communities using eigenvalues. The main difference between the work in [87,88] is the routing methods. The latter uses the following routing methods: intracommunity routing and intercommunity routing. The intracommunity routing provides the optimal route between the consumer and the last forwarding point in the same community based on Ant Colony Optimization (ACO) with the help of the information in a Content Index Table (CIT) that includes content index information. The intercommunity routing suggests a path from the consumer in the local community to the next community based on interaction frequency with the help of the information in the Community Topology Table (CTT) that includes community topology information. Solutions suggested in [86–88] increase the scalability of routing and data retrieval in ICN networks because of using inter/intra clusters of SDN controllers, where many critical tasks are offloaded from the top-layer controllers to the internal controllers. On the other hand, and because of the same reasons, both approaches are vulnerable to malware dissemination. Table 7 summarizes the approaches that try to optimize routing and forwarding when ICN and SDN are combined.

Table 7. Routing and Forwarding in ICN and SDN.

Approach	Related Work	ICN Model	SDN Controller	Advantages	Limitations
1. New SDN-based routing mechanism	[69,70]	NDN	SDN-LikeS	Reduces the routing overhead	1. Agnostic 2. Adds burden on the controller
	[71–74]	NDN	SDN-Like		
	[79]	CoLoR	SDN-Like		
2. Implement a new FW strategy	[75]	CCN-Like	SDN-Like	Eliminates Interest/Data packets flooding	Overloads the controller with many processes
3.POF FIB flow table and cache replacement algorithms to implement CRT	[76]	NDN	SDN-Like (POF)	1. Eliminates name to a hash mapping 2. Increases CCN routers' flexibility and forwarding strategy performance	Increases complexity
4. Adding tables to the forwarding devices	[77]	NDN	SDN-Like	Reduces the wasted time and the BW consumption	Violates one of NDN specifications

Table 7. *Cont.*

Approach	Related Work	ICN Model	SDN Controller	Advantages	Limitations
5. Multipathing Forwarding Strategy	[78]	NDN	SDN-Like	Reduces traffic overhead and handles link failures	Latency may occur
	[80]	NDN	SDN-Like		
6. Multicast Forwarding	[81]	NDN	SDN-Like	Optimizes scalability, load balancing, and bandwidth utilization	Does not study link failure
7. Using P4	[83]	NDN	SDN-Like (P4)	Improves routing, forwarding, and scalability	Small hashing process overhead
	[84]	NDN	ONOS (P4)		
8. Cluster of controllers and Metaheuristic Algorithms	[85]	NDN	SDN-Like	Optimizes the scalability of routing and data retrieving	May increase network vulnerability to the virus spreading
	[86]	ICN-Like	SDN-Like (POF)		
	[87,88]	NDN	SDN-Like		

5.7. Caching

Since caching is one of the main concepts in the ICN network, many works have investigated how to benefit from having an SDN controller in the network to optimize caching data in ICN. In [89], the authors claimed that the Interest forwarding process can be accelerated when data caching is performed on chosen routers rather than on all traverse routers. This is executed by implementing a greedy algorithm implemented in a centralized controller.

A hybrid caching strategy (HCache) was proposed in [90]. HCache contains two caching algorithms: (1) a deterministic caching algorithm at the edge routers to reduce the delay at the end-user, where the SDN controller global view is leveraged to collect the network information that is needed to calculate the caching policy and to inform edge routers how to cache, and (2) a probabilistic caching algorithm at the core routers to enhance the cache variety and reduce inter-domain traffic, where the probability takes into consideration content popularity and access delay reduction. The authors compare HCache to other deterministic and probabilistic caching algorithms such as Leave Copy Down (LCD) and ProbCache, and the results show that HCache outperforms other algorithms in terms of access delay, hit-ratio, and link traffic.

The controller in [91] has a strategy management module that: (1) decides which content to be cached based on the content popularity using the Linear Network Coding (LNC) algorithm, and (2) decides the route to the content to the closest cached data. This caching strategy and content routing help in minimizing bandwidth utilization.

The SDICN prototype was proposed in [92], where OF-switch downloads requested data from data stores instead of caching it on the switch to overcome memory issues. The OF-switch resolves the content name as an IP option. Then, the OF-switch encapsulates the original Interest in a new packet since the SDN controller cannot deal with the new IP option. SDN controller in this approach is used to balance data traffic by optimizing the data distribution in the network cache. This is achieved by implementing the following algorithms in the controller: (1) Content Locating (CL) for redirecting consumer requests to the data center with minimum cost, (2) Content Optimal Deployment (COD) for executing content optimal deployment such as which data must be cached and where to cache data, and (3) Path Optimizer (PO) for balancing network. According to the results, this approach surpasses ICN.

Zhang et al. [93] proposed an SDN-based caching decision policy that enhances multimedia delivery by reducing video delivery latency and computational complexity using Integer Linear Programming (ILP). The proposed policy considers network dynamics, such as cache location and content properties.

In addition, the authors in [94] suggested caching in the controller, which may result in controller flooding and inefficient in-network caching. This approach uses hierarchical hashing to map contents' names to values that match with the "Longest Prefix Match". However, the names are limited to eight components which can be considered another drawback [94].

The approach in [95] has both CCN switches and IP switches in its network topology. SDN controller makes cache decisions based on content popularity, which minimizes the routing delay. This was achieved by applying new caching algorithms: (1) exponential-time exact Integer Linear Problem for small networks, and (2) polynomial-time heuristic algorithms for large-scale networks. The results of this approach were better than the results of the three famous algorithms, Cache Everything Everywhere (CEE), Leave Copy Down (LCD), and PROB, in terms of delay and hits.

Predicting content popularity using deep learning was presented in [96]. Deep-Learning-based Content Popularity Prediction (DLCPP) is distributed among the forwarding devices. DLCPP uses Stacked Auto-Encoder (SAE) and Softmax deep learning methods to extract and classify popularity Spatio-temporal features to predict the content popularity. These predicted popularities are sent to the controller. Accordingly, the controller helps the connected forwarding devices in content caching. The nodes in this work are randomly chosen, and the authors suggested using some well-known methods to choose the input nodes, such as betweenness and coloration, to overcome this drawback. Both of the works in [95,96] do not mention how the use OF controller works with the CCN switches.

Yang et al. [97] proposed a caching strategy based on GA in ICN networks, called Multiple-Round Parallel Genetic Algorithm (MRPGA). They claimed that MRPGA guarantees a feasible solution by decomposing the problem and having multiple round iterations. Because of the holistic view that the SDN controller has, MRPGA relies on it to collect the popularity information of CRs. According to their simulation results, MRPGA has a higher cache hit ratio and needs much less running time than normal GA.

Jmal and Fourati implemented Cache Management (CM) in the NDN network to store the popular contents instead of caching each requested content in nodes in the path [98]. However, the content name is mapped to the IP address to be handled by OF-controller. SDN controller manages CM, which chooses the caching strategy and takes popularity decisions based on thresholds. Interest is forwarded from the consumer to the border node with OF-controller. OF-Controller adds a new entry in the flow table and sends a request to CM to check if it has the content and updates the content popularity counter. If CM has the requested content, it informs the controller and sends the content to the consumer through the controller. Otherwise, the controller installs the new rules in the OF-switch, and CCN ordinary process occurs. If OF-controller does not find the requested content in its network, it asks the adjacent OF-controllers and takes the response to the shortest path routing. As per the results, the cache resources are optimized, and the hit ratio is high. However, there is an extra load on the controller and the bandwidth utilization increases since all Interests and Data packets are going through the controller, which slows down management of the cache replacement.

The authors in [99] used stateful SDN instead of stateless ones. Stateful SDN refers to an SDN where switches have intelligence and make some decisions. In their design, the controller is responsible for network-wide decisions, while switches are responsible for local decisions. This was achieved by using an OpenState Switch equipped with a cache agent and actual storage to implement the NDN node. Each NDN node contains PIT, FIB, CS, and Cache Lookup Table (CLT). Moreover, Interest and Data packets are implemented as UDP payloads, and they use the methods mentioned in [48,119–121], to process NDN packets and parse data fields in UDP packets in the OF-switch. The controller has no role after the OpenState Switch configuration is completed, which violates the main concept of SDN.

The work in [100] targeted the NDN network with some changes to the FIB table fields. SDN controller has two tables: (1) the Global Topology table that has information about the network topology, and (2) the Global Data table that has information about data in the network. Based on this information, the controller builds Flow-FIB tables in switches. Flow-FIB has four extra fields for caching purposes: counter, rank, popularity, and action. If the Interest content name is available in Flow-FIB, then the action will serve it with the requested data from CS. Otherwise, the PIT will be checked. If it exists in PIT from another port, the new requester port will be added to PIT. Otherwise, the switch will ask the controller to look for the requested content. Every time Flow-FIB receives an Interest, the counter field is incremented by one. The switch increasingly orders the requested data and puts the order in the rank column. It will then use the ranked information to calculate content popularity. Finally, the content in the CS will be reordered based on content popularity. Interest processing in this work is faster than the normal NDN since the switch immediately consults the Flow-FIB table instead of going through the normal NDN process for dealing with an Interest packet.

In [101], the authors proposed a solution for large data distribution and retrieval across multiple NDN routers' caches. When an NDN router receives an Interest in the requested large data, and it does not find the requested data in its CS, it will forward it to the controller. Then, the controller will sort the routers in its domain based on the available space in each router's CS, where the router with the largest space comes first. The controller will decide the number of routers needed to cache the file, splitting the received data among the number of routers. After that, the controller will send configuration instructions on how to forward the interest to the router. The router will use these instructions to forward multiple interests to request these large data in segments. In addition, the controller will send more instructions to the chosen routers that will store chunks of data. This work shows a good transfer performance, but it suffers from small latency due to the communication overhead with the controller, especially when the NDN routers are more than one hop away from the controller. Hence, the work can be improved if the controller considers the routers' locations.

Zhang et al. [102] proposed a cached content-locating mechanism for ICN content that uses: (1) Bloom Filtering (BF) to represent cache information, (2) Compress Sensing to compress the BF, which in turn reduces the cache announcing overhead, and (3) SDN controller to manage and maintain cache information. Initially, the SDN controller sends a request to ICN nodes asking for their compressed cache information. After that, the ICN nodes use BF to map the popular information, compress the resultant BF, and send it to the controller. Then, the SDN controller recovers this compressed information and updates its cache database. Finally, when the controller receives a new content request, it looks inside its database to locate the ICN node that caches the needed content. Their mechanism can deal with the enormous amount of cached data and the frequently updated cached data. On the other hand, the error ratio in the mechanism differs according to the application and must be carefully chosen to avoid depreciation in accuracy and performance.

Wireless networks were taken into consideration in [103], where the authors modified the OpenFlow protocol to support matching and forwarding based on names instead of IP addresses. Moreover, they used the SDN controller to improve the efficiency of content delivery of Wireless Mesh Network (WMN) in ICN networks. Instead of having all computations in wireless mesh nodes, the controller collects important network information and statistics. Then, it uses these statistics in: (1) route decision, and (2) cache management, which is performed by: (i) choosing the cache location, (ii) providing a cache identifier, and (iii) distributing cached data among the network layer. Their results show that the proposed approach is better when users are locally converged, and the average response delay is reduced since the content is cached in a close place to users. Table 8 summarizes the approaches that focus on caching when ICN and SDN are combined.

Table 8. Caching in ICN and SDN.

Approach	Related Work	ICN Model	SDN Controller	Advantages	Limitations
1. New caching algorithms	[91]	NDN-Like	SDN-Like	1. Reduces BW utilization and delay 2. Increases hit ratio	May violate the content confidentiality and privacy
	[95]	CCN	SDN-Like (OF)		
	[100]	NDN	SDN-Like		
	[90]	NDN	SDN-Like		
	[93]	ICN-Like	SDN-Like		
2. External data stores	[92]	ICN-Like	NOX	Solves the low storage capacity of OF switches	Increases overhead on controller
3. Hierarchical hashing	[94]	CCN	SDN-Like (OF)	Enables LPM and name prefix aggregation	1. Floods Controller 2. Limited prefix length
4. Deep Learning	[96]	NDN	Floodlight	Good Performance	1. Produces small overhead 2. Choosing nodes randomly
5. Metaheuristic Algorithm	[97]	NDN	SDN-Like	1. Increase hit ratio 2. Accelerated convergence	Increase overhead on controller
6. New controller's management module	[98]	NDN	Floodlight	1. Optimizes utilizing cache resources 2. Increases hit ratio	Increases load on controller and BW utilization
7. Stateful SDN	[99]	NDN-Like	SDN-Like (OF)	Zero control traffic and short latency after the initial configuration	Violates the SDN concept
8. Cache large data	[101]	NDN	SDN-Like	Better performance than default NDN	1. It adds a delay 2. Just applicable to Big Data
9. Compressed Sensing	[102]	ICN-Like	SDN-Like	Overcomes main cache issues	Lack of accuracy affects performance
10. WMN utilizes cache points, and the controller optimizes cache management	[103]	ICN-Like (HTTP)	KulCloud OpenMUL v4.0.1	Improves Name-based Wireless efficiency, especially for local coverage	Performance and delay depend on the cache location

5.8. Scalability and Mobility

Because of caching and content-based networks, ICN faces a scalability problem. Several works [104–108] have proposed approaches to solve this scalability issue. Gao et al. [104,105] designed a cluster of controllers that are arranged hierarchically. Each area/domain has a controller to manage: (1) local network resources such as devices, link bandwidth, and storage usage, and (2) local content resources such as Interest matching, route decision, and cache replacement. These area controllers are connected to a root controller that optimizes resource utilization and ensures scalability since it has a centralized global view of the network.

The authors in [106] proposed a novel architecture caching and forwarding strategy for NDN based on SDN called (FCR-NS). The architecture improves the routing by using separate routing within the zones (intra-zone) and between the zones (inter-zone). Each zone is divided into two planes: the data plane and the control plane. The Control-plane contains a controller that contains four tables: (1) Global-Top, (2) Global-Data, (3) Global-

Data-Bloom Filter (BF), and (4) Routing Information Base (RIB). The data plane, on the other hand, contains the switches, which contain five tables each. The SDN controller manages the fast intra-zone routing process by using the first two tables in the controller to build a FIB table for all switches in the controller's domain. It also manages the fast interzone routing process by consulting the Global-Data-BF table if the requested data exist in its table. If the requested content is available within the controller's domain, the requested data will be forwarded to the requested node. Otherwise, the controller forwards the content to another controller. Using BF speeds up the name searching and forwarding processes. Moreover, the architecture avoids caching content redundancy and resource wastage due to its usage of an efficient cache replacement strategy that is based on the real-time calculation of the popularity of the data by the switches.

The approach in [107] used off-line information of the network topology to proactively calculate the number of cache servers and their optimal location based on: (1) higher closeness centrality, (2) minimum path stretch values, (3) higher betweenness centrality, and (4) load balancing in the network. After that, the controller installs the flow rules in the switches that map content names to cache servers. When a switch receives a request, and it matches with one of the entries in its flow table, the content will be delivered to the requesting node from the switch's closest cache server. Otherwise, the request is forwarded to the controller to calculate the best path between the content provider and the cache server. Then, the requested content is delivered to the requested node after saving a copy of the content in the cache server. This approach shows remarkable results in terms of traffic overhead and latency compared with other approaches. However, if the number of cache servers increases, the computation time for the location of cache servers using the values of closeness centrality, path-stretch, and betweenness centrality increase drastically.

The authors in [108] focused on reducing the computation time of the approach proposed in [107]. The computation overhead occurs because the traffic matrices are sparse since most of the traffic passes through only a few switches called important switches. The approach in [108] solves the sparse matrices problem by connecting each important switch to a caching server. The location of each important switch is calculated by using Singular Value Decomposition (SVD) and QR factorization with column pivoting technique of linear algebra. Their approach shows a reduction in computation time, traffic overhead, and power consumption.

In [109], NDN overlays IP. The authors tried to leverage a centralized SDN controller to solve the problems that occur because of moving consumers. In their approach, when the consumer detaches from a network, the SDN controller dynamically updates FIB tables. Then, when the consumer connects to a new network, the SDN controller controls the synchronization between the consumer and the producer. Based on their observations, when a consumer detached from one network and joined another, it did not lose the content updates during the handover. However, their approach suffers from control overheads. These overheads increase as consumer speed, network size, and window size increase. Moreover, the same authors tried to evaluate their approach in terms of energy consumption [110]. The results show that the proposed approach is better and more effective for bigger contents' size and larger intervals between requests. However, their results in both works [109,110] did not reflect the real world since their scenario included two routers and one controller, consumer, and publisher. Table 9 summarizes the approaches targeting scalability and mobility when both ICN and SDN are combined.

5.9. Miscellaneous

Different approaches try to combine both paradigms in different ways. The authors in [111,112] proposed a Function-Centric Service Chaining (FCSC) mechanism that extended content naming to function naming where names are used to identify, forward, and retrieve services in addition to data chunks. It has a naming layer that separates the policy module, which determines the needed function to handle the received flow from the routing module. The controller is responsible for deciding which policy the income flow needs and then sends the result to the ingress. Ingress tags the function list in the packet header as hierarchy naming (/DPI/Firewall), as in NDN. Once the switch or the middle box receives the packet, it removes the name from the header. After that, the packet is forwarded to the next function based on Longest Prefix Matching (LPM) in switch FIB. FIB is controlled by the routing module, which can be distributed using Open Shortest Path First (OSPF) or centralized using SDN.

Table 9. Scalability and Mobility in ICN and SDN.

Approach	Related Work	ICN Model	SDN Controller	Advantages	Limitations
1. Hierarchy cluster of controllers	[104,105]	PURSUIT and NDN	SDN-Like	Optimizes resource utilization and ensures scalability	A problem in burden trade-off between domain controllers and root controllers and setup delay
	[106]	NDN	SDN-Like		
	[107]	NDN	SDN-Like		
	[108]	NDN	SDN-Like		
2. Multiple protocols	[109,110]	NDN	SDN-Like	Overcomes consumer mobility by losing content updates during the handover	Suffers from control overheads

Popular multimedia content retrieval was taken into consideration in the work suggested in [113]. The controller was used to: (1) improve ICN scalability and reliability, and (2) reduce network congestion. This helps in multimedia content retrieving by caching content as close as possible to the consumer. This was achieved by analyzing the gathered statistics of multimedia data requested by users of certain geographical regions in the controller.

Chen et al. [114,115] proposed a joint resource allocation for caching, computing, and networking to balance energy consumption and network usage cost. The data-plane in their approach included caching and computing devices in addition to forwarding devices. The controller contains a management module that manages the aforementioned devices. Moreover, the controller dynamically guides different interest request types (content name request or service request) to their devices. It orchestrates networking, caching, and computing to allocate optimal resources by reducing complexity, computational signaling overhead, and energy consumption.

Table 10 includes a summary of a performance analysis of some of the related works, in terms of bandwidth utilization, throughput, latency, cache hit ratio, and resources consumption.

Table 10. Related Work Performance.

Paper	Performance
[42]	The 'cost' of header manipulation is 2000 PPS.
[44]	• NDN–SDN provides a 19% reduction in BW usage compared to the NDN-Best-route strategy. • CPU switch consumption is low in comparison with NDNS [77]. • NDN–SDN reduces the memory overhead by 9% compared to NDN-Flooding. • With 400 nodes, NDN–SDN reduces the delay time by 27% compared to NDN-Flooding. • With 300 nodes, NDN–SDN increases cache hits by 29% compared to NDN-Flooding.
[45]	• The percentage of download time reduction in SDN-NDN will be stable at about 30%. • The time for table lookup of CS and PIT, with the processing time of the Controller, is about 2%.
[47]	• At a hit rate of 80%, the server's throughput decreases by 52%. • The improvement in the delay is nearly 75%.
[48,49]	Wrapper slightly degrades forwarding performance but no more than 5%.

Table 10. *Cont.*

Paper	Performance
[50]	• The controller adds links to the overlay in order to keep the target failure below the selected threshold. • The delay in the topology with SDN is 0.25 ms less than without SDN. • Data retrieval time is on average less than 500 s for topology with SDN.
[53]	Hit cache ratio of the proposed UDPAM is around 3.5%, while for LCE it is 2%, and LRU is 1.25%.
[54]	12.5% of traffic is saved compared with the baseline.
[55,56]	Average delay of the proposed approach is 50 ms and it is about 80 ms in traditional SDN networks.
[59]	• Because of caching, the download traffic from cloud servers with the proposed solution (ECCN) is reduced nearly to 0 after 20,000 s. • The download traffic with the mobile edge computing method is more than 3 Mb/s after 35,000 s and for the original mode, it is 50 Mb/s all the time. • The hit ratio with ECCN is near 100% after 20,000 s, while the mobile edge computing method only has a 60% hit ratio at the same time.
[60]	• Average response delay for the proposed dynamic location increases from 20–60 ms while the uplink traffic rate increases. • Proposed scheme has the highest application utilization both for TCP and ICN in comparison with Cloud-RAN (CRAN) and MEC.
[64]	Proposed mechanism leads to consistently higher throughput (around 12.5% more) in comparison with Load balancing, shortest path first, and Deep Reinforcement Learning mechanisms.
[66]	• Proposed approach (FellowMecache) always takes the highest throughput (from 0.3 to 0.41) in comparison with other approaches that mostly have throughput <0.22. • Delay of FellowMecache is 1.02×10^{-3} ms while for other approaches, it is between 5.67×10^{-3} ms to 2.66×10^{-3} ms. • Overhead ratio of FellowMecache decreases with the increase in the number of nodes while it increases for other mechanisms, i.e., the overhead ratio is 32% and 17% of FellowMecache when the number of nodes increases from 50 to 500 nodes.
[67]	• Proposed SFC/FON throughput is higher and better than SFC. • The normalized traffic load for SFC/FON is around 1 and for SFC is about 6 because SFC/FON can reduce such redundant traffic by leveraging multicast and cached video segments.
[69,70]	The proposed SRSC overhead represents 18.0% of the overall traffic that comes from the control messages
[71–74]	• Signaling efficiency is 75% better in the proposed scheme than in NLSR. • Convergence time is 1000 s faster in comparison with NLSR.
[75]	• Signaling efficiency in the proposed forwarding strategy is more efficient than other strategies since it takes less time for content delivery, especially for a high number of requests. • As the number of requests increases, the number of interests in the proposed strategy slightly increases because it does not broadcast interests to all neighbors as some other strategies.
[77]	• With 1500 nodes, the use of the proposed NDNS architecture minimizes the bandwidth utilization by 26% compared to NDN-flooding strategy. • With 2500 nodes, the use of NDNS architecture minimizes the delay time by 27% compared to NDN-flooding strategy. • With 3000 nodes, the use of NDNS architecture minimizes the bootstrap times by 26% compared to NDN-flooding strategy. • With 3000 nodes, the use of NDNS architecture raises the cache hits by 9% compared to NDN-flooding strategy.
[78]	• In the proposed approach (SDAR), when a link failure occurred and lasted more than the threshold, 64 Interests interrupted the controller under the single-path settings and only 16 Interests in the multi-path settings. • SDAR multi-path enhances response speed for traffic redirecting upon link changes and failures.
[80]	For transferring 1000×8 KB file • Distributed Multipath Forwarding Strategy (D-MP) proposed strategy performs 10.4× and 12.5× better than the default NDN implementation with in-network caching disabled and enabled, respectively. • Centralized SDN control for the Multipath Forwarding Strategy (S-MP) proposed strategy performs performance gains of 10.6× and 12.6× with in-network caching disabled and enabled, respectively. • S-MP performs 1.92% and 0.8% for transferring 1000×8 KB files.
[81]	MTO outperforms NDN in terms of overall network cost on all topologies, especially for larger topologies since MTO can efficiently utilize multiple content sources while others cannot, e.g., for random topologies MTO cost is 50% less than normal NDN.

Table 10. *Cont.*

Paper	Performance
[83]	• When the number of FIB entries is fixed, PPS is almost the same for NDN-FAB and UDP/IP • When the number of FIB entries is variable, the NDN-FAB switch can process 21% more PPS than UDP/IP switch. • NDN-FAB outperforms UDP/IP in terms of delay in all cases (fixed or variable number of hops and number of FIB entries), e.g., when the number of FIB entries is fixed average delay in UDP/IP switch 100x more than for NDN-FAB.
[84]	• In the proposed NLSR-P4, the number of Root Advise (RA) Interest packets is 42.9% less than the traditional NLSR. • In the proposed NLSR-P4, the number of Link-State Advertisement (LSAs) Interest packets is 33.4% less than the traditional NLSR.
[85]	• In the proposed routing algorithm, BW utilization is 22% less than Yang's and Hou's algorithms. • The proposed routing algorithm can realize lower resource consumption than Yang's and Hou's algorithms e.g., for 50 nodes network, power consumption is 10^4, 1.1×10^4, and 1.2×10^4, respectively.
[86]	• Throughput in the proposed architecture (SPARC) increases at a near-linear trend when the number of global and regional controllers increases. • SPARC suffers from 80 MB memory consumption.
[87]	For Deltacom network topology: • Average throughput is 2.7× more in the proposed RISC than normal NDNF. • RISC is more stable where the fluctuation coefficient is 50% less than NDNF. • RISC's overall delay is about 4.3 ms. • Average controller processing time in RISC is 0.15 ms.
[88]	• Proposed CRSCD delay is about 13 ms, while it is around 8 ms for RISC [87] • The average routing count is approximately 4.5 and 5 for CRSCD and RISC, respectively. • Routing success rate is approximately 0.997 and 0.973 for CRSCD and RISC, respectively.
[89]	For proposed centralized in-network caching is double the traditional ICN caching.
[90]	• Link traffic reduces from 1600 kb/s (in LCD) and 1500 kb/s (in ProbCache) to 1200 kb/s (in the proposed HCache). • Delay is 30% less in HCache than LCD and 22% less than ProbCache. • Hit ratio is 50% better in HCache than in LCD and 40% less than ProbCache.
[92]	• The processing delay is 3.5 ms in the proposed framework SDICN. • SDICN storage overhead is 50% less than CCN. • SDICN hit ratio is 85% compared with CCN.
[93]	• The proposed SDN-based caching decision policy reduces the number of interest packets (from 18,987 to 17,279 when network cache size ranges from 10% to 100%), while the other policies must generate or broadcast at least 97,378 interest packets. • The proposed SDN-based caching decision policy enhances the hit ratio (from 25% to 65% when network cache size increases from 10% to 100%), while it is 55% for the other policies.
[97]	The SUR (the ratio of the total volume of caches to the total caching) of MRPGA is better than that of GA, i.e., SUR is 60% and 80% for MRPGA and GA, respectively.
[121]	Total delay caused by NDNFlow: • OVS adds an additional delay of 0.300 ms • Configuring a new content flow costs an additional 62.172 ms at the CCNx daemon. • The OVS daemon adds 240.624 ms.
[100]	• The proposed replacement policy (NC-SDN) minimizes the number of packets by 27%, compared to the conventional NDN. • With 1000 nodes, NC-SDN minimizes the reception time of a message by 38% compared to NDNS [77]. • Controller CPU usage for both NC-SDN and NDNS increases from 0 to 85%, then decreases to less than 50% and around 75% when the network is stable, respectively. • NC-SDN hit ratio is better than other NDN replacement policies, e.g., it is 23%, better than the classic NDN-LFU, and 13.5% better than NDNS [77].
[101]	• For proposed system architecture, CMS dataset can be retrieved 28% faster from the NDN routers caches compared to Default NDN architecture. • For proposed system architecture with Prefetch, CMS dataset can be retrieved 38% faster from the NDN routers caches compared to Default NDN architecture.

Table 10. *Cont.*

Paper	Performance
[102]	• The proposed scheme requires only about 3.6 bits to announce cached content, while FNR [122] and SCAN [123] require 8 bits and 18.8 bits, respectively
[105]	Average cache hit ratio changes from 0.74% (at the beginning of the simulation) to 0.93% (after 4000 s simulation) at the area controllers.
[106]	• With a CS of capacity = 600, the proposed FCR-NS minimizes BW consumption by 6% compared to NDNS, by 12% compared to NLSR, and by 18% compared to OSPFN. • With a CS of capacity = 800, FCR-NS reduced interest overhead BW consumption by 14% compared to NDNS, by 27% compared to NLSR, and by 33% relative to OSPFN • With a CS of capacity = 120, FCR-NS has a latency time that is 47% faster than Best-route and 61% faster than flooding. • With a CS of capacity = 120, FCR-NS increased the cache hit ratio relative to LFU by 25%, 33% compared to LRU, and 42% compared to Random.
[110]	For content $\leq$5 KB energy consumption is almost the same for the proposed architecture and NDN core network, while the conceived protocol is able effectively to reduce the amount of energy consumed in case of bigger content.
[111]	• Packet loss rate for the proposed FCSC is around 4% while for the traditional SDN is around 22%. • Latency of FCSC is 75 ms and for the traditional SDN is 80 ms.
[113]	• Interest Satisfaction Rate for the proposed SDN-Cache is around 12% more than SDN networks and 23% better than the best route strategy. • RTT of SDN-Cache is 14% less than SDN and 9.5% less than the best route strategy.

6. Open Research Areas

Since both architectures, ICN and SDN, are considered new topics, many works focus on developing them individually and integrating them. In this section, we highlight some issues and challenges that deserve the attention of the SDN and ICN research communities.

6.1. Name Resolution and Name Look-Up

Some ICN models have name resolvers as the main component, such as RH in DONA [14], Resource Manager (RM) in CoLoR [22], and two types of Name Resolution Systems (NRSs) in SAIL [16]. As mentioned in [124,125], name resolution is executed by having an NRS, which may utilize Distributed Hash Table (DHT), such as for Pastry [126,127], Chord [128], and multilevel DHT [129]. Having NRS in the SDN controller is one of the areas that deserves attention.

ICN models face a scalability issue that needs name look-up solutions, especially for reaching domain names on the Internet. This is because the number of registered domains is around 370 million [130], each has subdomains and pages, and each of these subdomains and pages should be represented as a name inside the FIB table and routing table. Some name-lookup solutions to solve this issue are summarized in [131]. Moreover, NDN DNS (NDNS) mentioned in [132] leverages the look-up functionality learned from DNS and its security extensions. However, leveraging the SDN controller for name look-up has not received enough attention from the research community. SDN controller may play an effective role in name lookup especially if FIB tables of routers connected to the SDN controller are implemented as one FIB table in the controller. Moreover, a flow latency issue will appear as a result of this scalability issue and the name look-up process, which must also be taken into consideration.

6.2. Name-Based Applications

Most networks and Internet applications are based on IP addresses. With the proposed Name-based networks, the need to implement applications and prototypes that are compatible with names appears. These applications will make the switching from location-centric networks to name-centric networks more reliable and accepted. The authors in [133], presented some of the Name-based applications that are implemented to be consistent with CCN networks. Applications that need centralized management, such as IoT applications and QoS, can be implemented in the SDN application layer and controlled by the SDN controller.

6.3. Artificial Intelligence, Federated Learning, and Metaheuristic Algorithms

Nowadays, Artificial Intelligence (AI) attracts significant attention in the research community. Because of its incredible precision, accuracy, and speed, AI has been used in many applications, such as predicting, classifying, fraud detection systems, speech, image recognition, 5G network enhancement [118,134,135], and many more. SDN controller in [96] employs Stacked Auto-Encoders (SAE) and Softmax for obtaining Spatio-temporal features of content popularity.

Federated learning is a decentralized machine learning (ML that is based on training an algorithm across multiple decentralized servers instead of uploading all datasets to one server as in centralized ML. There are two surveys [136,137] that discuss approaches that apply Federated Learning in ICN and SDN, respectively. However, and to the best of our knowledge, there are no work targets for Federated Learning with integrated ICN and SDN.

Metaheuristic algorithms are modern numerical optimization tools that are inspired by processes usually found in biology and used to solve sophisticated optimization problems. This can be seen in the Particle Swarm Optimization Algorithm (PSO) [138], BrainStorm Optimization Algorithm [139], and the Genetic Algorithm (GA) [140]. SDN controller in [85,97] utilizes GA to choose the optimal route ly and to enhance caching, respectively.

As noticed and to the best of our knowledge, there is little work on the SDN controller utilizing AI, Federated Learning, or metaheuristic algorithms to optimize ICN features and network resources. This makes it an open research area for researchers to conduct more studies that take into consideration AI or metaheuristic algorithms in the hybrid ICN-SDN paradigm to provide QoS and network performance.

6.4. Blockchain

Many works took blockchain in ICN networks into consideration, such as [141–147]. Moreover, surveys in [148,149] review works that couple ICN networks with blockchain technology—Blockchain-Based Information-Centric Networking (BICN)—to improve security against different malicious behaviors, such as hijacking, cache pollution, and Denial of Service (DoS) attacks.

Huang et al. [150] used blockchain in the network slice of trust over their proposed computing framework. The proposed framework utilizes Intelligent Eco Networking (IEN) as a shared in-network computing infrastructure and it contains two logical layers: (1) the element layer where Name-based computing functions occurred, and (2) the control layer where SDN-Like controller schedules the running tasks to balance the efficiency and global optimization.

Nevertheless, BICN suffers from some privacy concerns, broadcast transactions, and distributed in-network caching, which can be solved by having a centralized controller. However, the research area lacks work related to the blockchain when ICN and SDN are combined.

6.5. Big Data Management

IoT, Cloud, and fog computing are hot topics with many challenges that remain unsolved. Controlling and supervising IoT devices and sensors face many challenges such as how to secure and architect the connected devices, since the IoT environment is considered complex and inhomogeneous, i.e., devices may have different requirements of bandwidth, power, and latency. Moreover, the large amount of streaming data generated by IoT devices need almost real-time processing and analysis, which requires fog computing in distributed clouds.

Din et al. in [151] claimed that since the content is an essential feature for both ICN and IoT, and because the number of connected IoT devices is increasing and the current Internet cannot handle this enormous number of devices, integration between both ICN and IoT is needed.

Combining the SDN controller with ICN can be utilized to improve the monitoring, controlling, and modifying of the management of big data. Having SDN enhances IoT application management in ICN networks and helps in minimizing: (1) packet loss, (2) error rate, and (3) latency. Moreover, using AI and ML by the SDN controller in collecting and classifying data from IoT devices will enhance the controller's real-time decisions that must be disseminated. However, some significant problems must be taken into consideration such as data sharing, storage, retrieval, QoS prediction, and performance scalability.

6.6. Enhancing 6G Networks

The fifth generation of cellular networks appears to accommodate changes in people's lives and handle more connected devices. Therefore, 6G is intended to overcome the 5G limitations that will occur because of the anticipated increase in human demands in terms of quality and quantity. To overcome these limitations, 6G must have better capacity, bandwidth, and less latency [152]. The 6G-enabled tactile internet (TI) is characterized by low latency and high availability and reliability that is suitable for applications such as augmented reality (AR), and virtual reality (VR). To the best of our knowledge, [153] is the only work that is targeting ICN and 6G networks. Liao et al. [153] proposed new technologies for AR/VR services in ICN networks. The AR/VR services of Information-Centric Massive IoT devices utilize ICN in-network caching, blockchain, and edge computing to ensure the required quality of service in 6G networks.

We believe that utilizing ICN in-network caching and performing some computing processes in the SDN controller will enhance 6G performance.

Table 11 summarizes all open research areas of merging ICN and SDN paradigms, their related issues, and the predicted results of solving these issues.

Table 11. ICN-SDN Open Research Areas.

Open Research Area	Related Issues	Results of ICN-SDN
Name Resolution and Name Look-up	Use SDN to solve: 1. ICN scalability issue because of the size of the FIB table. 2. ICN name lookup.	1. Reduce flow latency. 2. Enhance scalability.
Name-based Applications	Implement ICN applications that need centralized control, especially for IoT and vehicle ad-hoc networks.	1. Choose the best policy. 2. Reduce transmission delay and loss rate. 3. Increase data delivery. 4. Enhance QoS.
Artificial Intelligence	Use learning abilities to optimize the utilization of ICN-SDN resources.	1. Enhance cache hit ratio. 2. Smart routing, which enhances data delivery and reduces congestion. 3. Better decision-making in SDN.
Blockchain	1. Privacy concerns. 2. Broadcast transactions. 3. Distributed in-network caching.	1. Enhance security and authentication. 2. Enhance caching. 3. Reduce transactions.
Big Data Management	1. Handling enormous data and a number of devices. 2. Data sharing and storage. 3. QoS prediction.	1. Reduce packet loss and latency. 2. Enhance scalability. 3. Enhance QoS performance.

Table 11. *Cont.*

Open Research Area	Related Issues	Results of ICN-SDN
6G Enhancement	1. Control flow and computing. 2. Centralized control of data caching. 3. Securing transmitted data. 4. Control and manage large-scale wireless networks.	1. Enhance QoS of 6G networks. 2. Decrease latency. 3. Enhance data delivery. 4. Enhance large network stability.

7. Conclusions

The integration of ICN-SDN is a substantial, promising, and forthcoming approach for the future internet, which leverages the strengths of both architectures to improve network manageability, stability, reliability, mobility, and security. On the other hand, it reveals some new relevant issues and questions that must be addressed to achieve the optimal benefits of this hybrid paradigm.

This paper presents a thorough and comprehensive overview of ICN, SDN, and the motivation for the integration of ICN and SDN paradigms. The paper has considered ICN-SDN integration from different perspectives, such as: modifying SDN networks from multiple points of view to make them compatible with names, transforming ICN network characteristics by adding centralized controllers, e.g., caching scalability and routing, traffic engineering and QoS, and Satellite–Terrestrial networks. We provide an in-depth comparative analysis of the different state-of-the-art works in this area by highlighting their strengths and limitations.

This paper also points out the challenges and open research areas that have not been addressed sufficiently by the research community, such as name lookup, AI, and IoT management.

Author Contributions: Conceptualization, M.A. (Manar Aldaoud); methodology, M.A. (Manar Aldaoud); validation, M.A. (Manar Aldaoud); formal analysis, M.A. (Manar Aldaoud); investigation M.A. (Manar Aldaoud); resources, M.A. (Manar Aldaoud); data curation, M.A. (Manar Aldaoud); writing—original draft preparation, M.A. (Manar Aldaoud); writing—review and editing, M.A. (Manar Aldaoud), D.A.-A., M.A. (Medhat Awadalla) and F.K.; visualization, M.A. (Manar Aldaoud); project administration, M.A. (Manar Aldaoud) and D.A.-A.; funding acquisition, M.A. (Manar Aldaoud) All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Acknowledgments: The authors would like to thank Sultan Qaboos University for supporting this work finically and providing the facilities to conduct this research.

Conflicts of Interest: The authors declare that they have no competing interests.

References

1. Rowshanrad, S.; Parsaei, M.; Keshtgari, M. Implementing NDN using SDN: A review of methods and applications. *IIUM Eng. J.* **2016**, *17*, 11–20. [CrossRef]
2. Jmal, R.; Fourati, L.C. Content-Centric Networking Management Based on Software Defined Networks: Survey. *IEEE Trans. Netw. Serv. Manag.* **2017**, *14*, 1128–1142. [CrossRef]
3. Fazea, Y.; Mohammed, F. Software Defined Networking based Information Centric Networking: An Overview of Approaches and Challenges. In Proceedings of the 2021 International Congress of Advanced Technology and Engineering (ICOTEN), Taiz, Yemen, 4–5 July 2021; pp. 1–8.
4. Jmal, R.; Fourati, L.C. Emerging applications for future internet approach based-on SDN and ICN. In Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications (AICCSA), Hammamet, Tunisia, 30 October 2017–3 November 2017; pp. 208–213.
5. Zhang, Q.; Wang, X.; Huang, M.; Li, K.; Das, S.K. Software Defined Networking Meets Information Centric Networking: A Survey. *IEEE Access* **2018**, *6*, 39547–39563. [CrossRef]

6. Tourani, R.; Misra, S.; Mick, T.; Panwar, G. Security, Privacy, and Access Control in Information-Centric Networking: A Survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 566–600. [CrossRef]
7. Binder, A.; Kotuliak, I. Content Delivery Network Interconnect: Practical experience. In Proceedings of the 2013 IEEE 11th International Conference on Emerging eLearning Technologies and Applications (ICETA), Stara Lesna, Slovakia, 24–25 October 2013; pp. 29–33.
8. Akamai-Team. Akamai. Available online: https://www.akamai.com/ (accessed on 26 November 2022).
9. Pathan, M.; Buyya, R.; Vakali, A. *Content Delivery Networks: State of the Art, Insights, and Imperatives*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 9, pp. 3–32.
10. Yu, K.; Eum, S.; Kurita, T.; Hua, Q.; Sato, T.; Nakazato, H.; Asami, T.; Kafle, V.P. Information-Centric Networking: Research and Standardization Status. *IEEE Access* **2019**, *7*, 126164–126176. [CrossRef]
11. Ahlgren, B.; Dannewitz, C.; Imbrenda, C.; Kutscher, D.; Ohlman, B. A survey of information-centric networking. *IEEE Commun. Mag.* **2012**, *50*, 26–36. [CrossRef]
12. COMET-Team. Content Mediator Architecture for Content-Aware Networks (COMET). Available online: http://www.comet-project.org/ (accessed on 26 November 2022).
13. CONVERGENCE-Team. The Convergence Project. Available online: http://www.ict-convergence.eu/ (accessed on 26 November 2022).
14. Koponen, T.; Chawla, M.; Chun, B.-G.; Ermolinskiy, A.; Kim, K.; Shenker, S.; Stoica, I. A data-oriented (and Beyond) network architecture. *ACM SIGCOMM Comput. Commun. Rev.* **2007**, *37*, 181–192. [CrossRef]
15. PURSUIT-Team. Publish-Subscribe Internet Technology (PURSUIT). Available online: https://www.fp7-pursuit.eu/ (accessed on 26 November 2022).
16. SAIL-Team. Scalable and Adaptive Internet Solutions (SAIL). Available online: https://sail-project.eu/ (accessed on 26 November 2022).
17. Detti, A.; Melazzi, N.; Salsano, S.; Pomposini, M. CONET: A Content Centric Inter-Networking Architecture. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking, Macau, China, 24–26 September 2011. [CrossRef]
18. NDN-Team. Named Data Networking. Available online: https://named-data.net/ (accessed on 20 November 2022).
19. Mosko, M.; Solis, I.; Wood, C. *Content-Centric Networking (CCNx) Semantics*; No. rfc8569; Internet Request for Comments; RFC Editor: Phoenix, AZ, USA, 2019.
20. Afanasyev, A.; Burke, J.; Refaei, T.; Wang, L.; Zhang, B.; Zhang, L. A Brief Introduction to Named Data Networking. In Proceedings of the MILCOM 2018—2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, USA, 29–31 October 2018; pp. 1–6.
21. Afanasyev, A.; Shi, J.; Zhang, B.; Zhang, L.; Moiseenko, I.; Yu, Y.; Shang, W.; Li, Y.; Mastorakis, S.; Huang, Y.; et al. *NFD Developer's Guide*; Florida International University: Miami, FL, USA, 2018. [CrossRef]
22. Luo, H.; Chen, Z.; Cui, J.; Zhang, H.; Zukerman, M.; Qiao, C. CoLoR: An information-centric internet architecture for innovations. *IEEE Netw.* **2014**, *28*, 4–10. [CrossRef]
23. Bari, M.F.; Chowdhury, S.R.; Ahmed, R.; Boutaba, R.; Mathieu, B. A survey of naming and routing in information-centric networks. *IEEE Commun. Mag.* **2012**, *50*, 44–53. [CrossRef]
24. Xylomenos, G.; Ververidis, C.N.; Siris, V.A.; Fotiou, N.; Tsilopoulos, C.; Vasilakos, X.; Katsaros, K.V.; Polyzos, G.C. A Survey of Information-Centric Networking Research. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1024–1049. [CrossRef]
25. Zhang, M.; Luo, H.; Zhang, H. A Survey of Caching Mechanisms in Information-Centric Networking. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 1473–1499. [CrossRef]
26. Din, I.U.; Hassan, S.; Khan, M.K.; Guizani, M.; Ghazali, O.; Habbal, A. Caching in Information-Centric Networking: Strategies, Challenges, and Future Research Directions. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 1443–1474. [CrossRef]
27. Chand, M. A Comparative Survey On Different Caching Mechanisms In Named Data Networking (NDN) Architecture. *JETIR* **2019**, *6*, 264–271. [CrossRef]
28. Ioannou, A.; Weber, S. A Survey of Caching Policies and Forwarding Mechanisms in Information-Centric Networking. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2847–2886. [CrossRef]
29. AbdAllah, E.G.; Hassanein, H.S.; Zulkernine, M. A Survey of Security Attacks in Information-Centric Networking. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 1441–1454. [CrossRef]
30. Zhang, Y.; Afanasyev, A.; Burke, J.; Zhang, L. A survey of mobility support in Named Data Networking. In Proceedings of the 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, CA, USA, 10–14 April 2016; pp. 83–88.
31. Hassan, S.; Habbal, A.; Alubady, R.; Salman, M. A Taxonomy of Information-Centric Networking Architectures based on Data Routing and Name Resolution Approaches. *J. Telecommun. Electron. Comput. Eng.* **2016**, *8*, 99–107.
32. Xia, W.; Wen, Y.; Foh, C.H.; Niyato, D.; Xie, H. A Survey on Software-Defined Networking. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 27–51. [CrossRef]
33. Azodolmolky, S. *Software Defined Networking with OpenFlow*; Packt Publishing Ltd.: Olton, UK, 2013.
34. Goransson, P.; Black, C.; Culver, T. *Software Defined Networks: A Comprehensive Approach*, 2nd ed.; Morgan Kaufmann: Burlington, MA, USA, 2017; Volume 1.

35. Jarraya, Y.; Madi, T.; Debbabi, M. A Survey and a Layered Taxonomy of Software-Defined Networking. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1955–1980. [CrossRef]
36. IETF. *Forwarding and Control Element Separation (ForCES)*; No. rfc6041; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2010.
37. IETF. *Network Configuration Protocol (NETCONF)*; No. rfc 6241; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2011.
38. CISCO. *OpFlex: An Open Policy Protocol White Paper*; CISCO: San Jose, CA, USA, 2014.
39. ONF. *OpenFlow Switch Specification*; Open Networking Foundation: Palo Alto, CA, USA, 2015.
40. Trois, C.; Del Fabro, M.D.; de Bona, L.C.E.; Martinello, M. A Survey on SDN Programming Languages: Toward a Taxonomy. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2687–2712. [CrossRef]
41. Shin, S.; Xu, L.; Hong, S.; Gu, G. Enhancing Network Security through Software Defined Networking (SDN). In Proceedings of the 25th International Conference on Computer Communication and Networks (ICCCN), Waikoloa, HI, USA, 1–4 August 2016; pp. 1–9.
42. Zuraniewski, P.; Adrichem, N.; Ravesteijn, D.; Ijntema, W.; Papadopoulos, C.; Fan, C. Facilitating ICN Deployment with an Extended openflow Protocol. In Proceedings of the 4th ACM Conference on Information-Centric Networking, 26 September 2017; pp. 123–133.
43. Rajendran Jeeva, B.E. *OF-ICN: OpenFlow-Based Control Plane for Information-Centric Networking*; University of Dublin: Dublin, Ireland, 2016.
44. Guesmi, T.; Kalghoum, A.; Alshammari, B.M.; Alsaif, H.; Alzamil, A. Leveraging Software-Defined Networking Approach for Future Information-Centric Networking Enhancement. *Symmetry* **2021**, *13*, 441. [CrossRef]
45. Liu, Y.; Li, C.; Li, T.; Song, J. A Novel IP-ICN Coexistence Deployable Frame-work Based SDN. In Proceedings of the 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 12–14 March 2021; pp. 2017–2022.
46. Feng, W.; Tan, X.; Jin, Y. Implementing ICN over P4 in HTTP Scenario. In Proceedings of the 2019 2nd International Conference on Hot Information-Centric Networking (HotICN), Chongqing, China, 13–15 December 2019; pp. 37–43.
47. Trajano, A.F.R.; Fernandez, M.P. ContentSDN: A Content-Based Transparent Proxy Architecture in Software-Defined Networking. In Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), Crans-Montana, Switzerland, 23–25 March 2016; pp. 532–539.
48. Nguyen, X.; Saucez, D.; Turletti, T. Providing CCN Functionalities over OpenFlow Switches; hal-00920554, Version 1; 2013. Available online: https://hal.inria.fr/hal-00920554/document (accessed on 26 November 2022).
49. Xuan Nam, N.; Saucez, D.; Turletti, T. Efficient caching in Content-Centric Networks using OpenFlow. In Proceedings of the 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Turin, Italy, 14–19 April 2013; pp. 67–68.
50. Zali, Z.; Hashemi, M.R.; Cianci, I.; Grieco, A.; Boggia, G. A controller-based architecture for information centric network construction and topology management. *China Commun.* **2018**, *15*, 131–145. [CrossRef]
51. Xing, C.; Ding, K.; Hu, C.; Chen, M.; Xu, B. SD-ICN: Toward Wide Area Deployable Software Defined Information Centric Networking. *KSII Trans. Internet Inf. Syst.* **2016**, *10*, 2267–2285. [CrossRef]
52. Zhang, Y.; Wang, Y. SDN based ICN architecture for the future integration network. In Proceedings of the 2016 16th International Symposium on Communications and Information Technologies (ISCIT), Qingdao, China, 26–28 September 2016; pp. 474–478.
53. Liu, Z.; Li, Y.; Zhu, J.; Yao, Q.; Ren, X. User-Driven Cache Replacement Strategy for Satellite-Terrestrial Networks Based on SDN. In Proceedings of the 2020 IEEE 6th International Conference on Computer and Communications (ICCC), Chengdu, China, 11–14 December 2020; pp. 680–688.
54. Li, J.; Xue, K.; Liu, J.; Zhang, Y.; Fang, Y. An ICN/SDN-Based Network Architecture and Efficient Content Retrieval for Future Satellite-Terrestrial Integrated Networks. *IEEE Netw.* **2020**, *34*, 188–195. [CrossRef]
55. Liu, Z.; Zhu, J.; Zhang, J.; Liu, Q. Routing algorithm design of satellite network architecture based on SDN and ICN. *Int. J. Satell. Commun. Netw.* **2020**, *38*, 1–15. [CrossRef]
56. Liu, Z.; Zhu, J.; Pan, C.; Song, G. Satellite Network Architecture Design Based on SDN and ICN Technology. In Proceedings of the 2018 8th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 15–17 June 2018; pp. 124–131.
57. Liyanage, M.; Porambage, P.; Ding, A.Y.; Kalla, A. Driving forces for Multi-Access Edge Computing (MEC) IoT integration in 5G. *ICT Express* **2021**, *7*, 127–137. [CrossRef]
58. Radhika, K.; Murali Mohan Babu, Y.; Periasamy, J.K.; Saravanan, T.R. Service Oriented Virtual Machine for Maximising Quality of Service in Wireless Networks. *J. Phys. Conf. Ser.* **2021**, *1964*, 042086. [CrossRef]
59. Li, H.; Ota, K.; Dong, M. ECCN: Orchestration of Edge-Centric Computing and Content-Centric Networking in the 5G Radio Access Network. *IEEE Wirel. Commun.* **2018**, *25*, 88–93. [CrossRef]
60. Vakilinia, S.; Elbiaze, H. Latency Control of ICN Enabled 5G Networks. *J. Netw. Syst. Manag.* **2019**, *28*, 81–107. [CrossRef]
61. Zhang, X.; Zhu, Q. Information-Centric Virtualization for Software-Defined Statistical QoS Provisioning Over 5G Multimedia Big Data Wireless Networks. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1721–1738. [CrossRef]
62. Chanda, A.; Westphal, C.; Raychaudhuri, D. Content based traffic engineering in software defined information centric networks. In Proceedings of the 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Turin, Italy, 14–19 April 2013; pp. 357–362.

63. Sun, Q.; Wendong, W.; Hu, Y.; Que, X.; Xiangyang, G. SDN-based autonomic CCN traffic management. In Proceedings of the 2014 IEEE Globecom Workshops (GC Wkshps), Austin, TX, USA, 8–12 December 2014; pp. 183–187.
64. Zhang, Q.; Wang, X.; Lv, J.; Huang, M. Intelligent Content-Aware Traffic Engineering for SDN: An AI-Driven Approach. *IEEE Netw.* **2020**, *34*, 186–193. [CrossRef]
65. Liu, D.; Wang, Z.; Zhang, J. Video Stream Distribution Scheme Based on Edge Computing Network and User Interest Content Model. *IEEE Access* **2020**, *8*, 30734–30744. [CrossRef]
66. Abar, T.; Rachedi, A.; ben Letaifa, A.; Fabian, P.; el Asmi, S. FellowMe Cache: Fog Computing approach to enhance (QoE) in Internet of Vehicles. *Future Gener. Comput. Syst.* **2020**, *113*, 170–182. [CrossRef]
67. Hayamizu, Y.; Matsuzono, K.; Hirayama, T.; Asaeda, H. Design and Implementation of ICN-Based Elastic Function Offloading Network for SFC. In Proceedings of the 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), Tokyo, Japan, 28 June–2 July 2021; pp. 278–282.
68. Saadeh, H.; Almobaideen, W.; Sabri, K.E.; Saadeh, M. Hybrid SDN-ICN Architecture Design for the Internet of Things. In Proceedings of the 2019 Sixth International Conference on Software Defined Systems (SDS), Rome, Italy, 10–13 June 2019; pp. 96–101.
69. Aubry, E.; Silverston, T.; Chrisment, I. SRSC: SDN-based routing scheme for CCN. In Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft), London, UK, 13–17 April 2015; pp. 1–5.
70. Aubry, E.; Silverston, T.; Chrisment, I. Implementation and Evaluation of a Controller-Based Forwarding Scheme for NDN. In Proceedings of the 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), Taipei, Taiwan, 27–29 March 2017; pp. 144–151.
71. Torres, J.V.; Ferraz, L.H.G.; Duarte, O.C.M.B. *Controller-Based Routing Scheme for Named Data Network*; Electrical Engineering Program; COPPE/UFRJ: Rio de Janeiro, Brazil, 2012.
72. Torres, J.V.; Duarte, O.C.M.B. CRoS-NDN: Controller-based Routing Strategy for Named Data Networking. Available online: https://pdfs.semanticscholar.org/f34a/eeea9c7aa58ae6994f3cc9bb33cb0a197ee6.pdf (accessed on 14 December 2022).
73. Torres, J.V.; Alvarenga, I.D.; Pedroza, A.d.C.P.; Duarte, O.C.M.B. Proposing, specifying, and validating a controller-based routing protocol for a clean-slate Named-Data Networking. In Proceedings of the 2016 7th International Conference on the Network of the Future (NOF), Buzios, Brazil, 16–18 November 2016; pp. 1–5.
74. Torres, J.V.; Alvarenga, I.D.; Boutaba, R.; Duarte, O.C.M.B. An autonomous and efficient controller-based routing scheme for networking Named-Data mobility. *Comput. Commun.* **2017**, *103*, 94–103. [CrossRef]
75. Son, J.; Kim, D.; Kang, H.S.; Hong, C.S. Forwarding strategy on SDN-based content centric network for efficient content delivery. In Proceedings of the 2016 International Conference on Information Networking (ICOIN), Kota Kinabalu, Malaysia, 13–15 January 2016; pp. 220–225.
76. Charpinel, S.; Santos, C.A.S.; Vieira, A.B.; Villaca, R.; Martinello, M. SDCCN: A Novel Software Defined Content-Centric Networking Approach. In Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), Crans-Montana, Switzerland, 23–25 March 2016; pp. 87–94.
77. Kalghoum, A.; Gammar, S.M. Towards New Information Centric Networking Strategy Based on Software Defined Networking. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017; pp. 1–6.
78. Liu, Y.; Wadekar, H. SDAR: Software Defined Intra-Domain Routing in Named Data Networks. In Proceedings of the 2016 IEEE 15th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 31 October–2 November 2016; pp. 158–161.
79. Luo, H.; Cui, J.; Chen, Z.; Jin, M.; Zhang, H. Efficient integration of software defined networking and information-centric networking with CoLoR. In Proceedings of the 2014 IEEE Global Communications Conference, Austin, TX, USA, 8–12 December 2014; pp. 1962–1967.
80. Alhowaidi, M.; Nadig, D.; Ramamurthy, B.; Bockelman, B.; Swanson, D. Multipath Forwarding Strategies and SDN Control for Named Data Networking. In Proceedings of the 2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Indore, India, 16–19 December 2018; pp. 1–6.
81. Zhang, Q.; Wang, X.; Lv, J.; Huang, M. MTO: Multicast-Based Traffic Optimization for Information Centric Networks. In Proceedings of the 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS), Tianjin, China, 4–6 December 2019; pp. 259–266.
82. Azgin, A.; Ravindran, R.; Wang, G. Scalable Multicast for Content Delivery in Information Centric Networks. In Proceedings of the 2018 International Conference on Computing, Networking and Communications (ICNC), Maui, HI, USA, 5–8 March 2018; pp. 105–111.
83. Madureira, A.L.R.; Araújo, F.R.C.; Araújo, G.B.; Sampaio, L.N. NDN Fabric: Where the Software-Defined Networking Meets the Content-Centric Model. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 374–387. [CrossRef]
84. Guo, X.; Liu, N.; Hou, X.; Gao, S.; Zhou, H. An Efficient NDN Routing Mechanism Design in P4 Environment. In Proceedings of the 2021 2nd Information Communication Technologies Conference (ICTC), Nanjing, China, 7–9 May 2021; pp. 28–33.
85. Li, J.; Xie, R.-c.; Huang, T.; Sun, L. A novel forwarding and routing mechanism design in SDN-based NDN architecture. *Front. Inf. Technol. Electron. Eng.* **2018**, *19*, 1135–1150. [CrossRef]

86. Li, M.; Wang, X.; Tong, H.; Liu, T.; Tian, Y. SPARC: Towards a Scalable Distributed Control Plane Architecture for Protocol-Oblivious SDN Networks. In Proceedings of the 2019 28th International Conference on Computer Communication and Networks (ICCCN), Valencia, Spain, 29 July–1 August 2019; pp. 1–9.
87. Lv, J.; Wang, X.; Huang, M.; Shi, J.; Li, K.; Li, J. RISC: ICN routing mechanism incorporating SDN and community division. *Comput. Netw.* **2017**, *123*, 88–103. [CrossRef]
88. Zhang, S.; Wang, X.; Qian, X.; Huang, M. An intelligent SDN-enabled CCN routing mechanism with community division. *Trans. Emerg. Telecommun. Technol.* **2019**, *31*, e3698. [CrossRef]
89. Wang, S.; Zhang, B. Centralized In-network Caching for Information Centric Networking with Decoupling Data and Control Planes. In Proceedings of the 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC), Orlando, FL, USA, 17–19 November 2018; pp. 1–2.
90. Wu, H.; Li, J.; Zhi, J.; Ren, Y.; Li, L. A Hybrid ICN Caching Strategy Based on Region Division. In Proceedings of the 15th International Conference on emerging Networking EXperiments and Technologies, Orlando, FL, USA, 9 December 2019; pp. 78–79.
91. Wang, J.; Ren, J.; Lu, K.; Wang, J.; Liu, S.; Westphal, C. An optimal Cache management framework for information-centric networks with network coding. In Proceedings of the 2014 IFIP Networking Conference, Trondheim, Norway, 2–4 June 2014; pp. 1–9.
92. Xiulei, W.; Ming, C.; Chao, H.; Xi, W.; Changyou, X. SDICN: A software defined deployable framework of information centric networking. *China Commun.* **2016**, *13*, 53–65. [CrossRef]
93. Zhang, Z.; Lung, C.; St-Hilaire, M.; Lambadaris, I. An SDN-Based Caching Decision Policy for Video Caching in Information-Centric Networking. *IEEE Trans. Multimed.* **2020**, *22*, 1069–1083. [CrossRef]
94. Ooka, A.; Ata, S.; Koide, T.; Shimonishi, H.; Murata, M. OpenFlow-based content-centric networking architecture and router implementation. In Proceedings of the 2013 Future Network & Mobile Summit, Lisboa, Portugal, 3–5 July 2013; pp. 1–10.
95. Muhui, S.; Bing, C.; Xiaojun, Z.; Yanchao, Z. Towards optimal cache decision for campus networks with content-centric network routers. In Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, 27–30 June 2016; pp. 810–815.
96. Liu, W.; Zhang, J.; Liang, Z.; Peng, L.; Cai, J. Content Popularity Prediction and Caching for ICN: A Deep Learning Approach With SDN. *IEEE Access* **2018**, *6*, 5075–5089. [CrossRef]
97. Yang, F.; Tian, Z. MRPGA: A Genetic-Algorithm-based In-network Caching for Information-Centric Networking. In Proceedings of the 2021 IEEE 29th International Conference on Network Protocols (ICNP), Dallas, TX, USA, 1–5 November 2021; pp. 1–6.
98. Jmal, R.; Fourati, L.C. An OpenFlow Architecture for Managing Content-Centric-Network (OFAM-CCN) based on popularity caching strategy. *Comput. Stand. Interfaces* **2017**, *51*, 22–29. [CrossRef]
99. Mahmood, A.; Casetti, C.; Chiasserini, C.-F.; Giaccone, P.; Härri, J. Efficient caching through stateful SDN in named data networking. *Trans. Emerg. Telecommun. Technol.* **2017**, *29*, e3271. [CrossRef]
100. Kalghoum, A.; Gammar, S.M.; Saidane, L.A. Towards a novel cache replacement strategy for Named Data Networking based on Software Defined Networking. *Comput. Electr. Eng.* **2018**, *66*, 98–113. [CrossRef]
101. Alhowaidi, M.; Nadig, D.; Ramamurthy, B. Cache Management for Large Data Transfers in Named Data Networking using SDN. In Proceedings of the 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Goa, India, 16–19 December 2019; pp. 1–6.
102. Zhang, Q.; Wang, X.; Huang, M.; Lv, J. Compressed Sensing-Based Cached Content Locating for ICN. *IEEE Commun. Lett.* **2018**, *22*, 2020–2023. [CrossRef]
103. Kim, W.-S.; Chung, S.-H.; Moon, J.-W. Improved content management for information-centric networking in SDN-based wireless mesh network. *Comput. Netw.* **2015**, *92*, 316–329. [CrossRef]
104. Gao, S.; Zeng, Y.; Luo, H.; Zhang, H. Scalable area-based hierarchical control plane for software defined information centric networking. In Proceedings of the 2014 23rd International Conference on Computer Communication and Networks (ICCCN), Shanghai, China, 4–7 August 2014; pp. 1–7.
105. Gao, S.; Zeng, Y.; Luo, H.; Zhang, H. Scalable control plane for intra-domain communication in software defined information centric networking. *Future Gener. Comput. Syst.* **2016**, *56*, 110–120. [CrossRef]
106. Kalghoum, A.; Saidane, L. FCR-NS: A novel caching and forwarding strategy for Named Data Networking based on Software Defined Networking. *Clust. Comput.* **2019**, *22*, 981–994. [CrossRef]
107. Badshah, J.; Kamran, M.; Shah, N.; Abid, S.A. An Improved Method to Deploy Cache Servers in Software Defined Network-based Information Centric Networking for Big Data. *J. Grid Comput.* **2019**, *17*, 255–277. [CrossRef]
108. Badshah, J.; Alhaisoni, M.M.; Shah, N.; Kamran, M. Cache servers placement based on important switches for SDN-based ICN. *Electronics* **2019**, *9*, 39. [CrossRef]
109. Benedetti, P.; Ventrella, A.V.; Piro, G.; Grieco, L.A. An SDN-aided Information Centric Networking Approach to Publish-Subscribe with Mobile Consumers. In Proceedings of the 2019 Sixth International Conference on Software Defined Systems (SDS), Rome, Italy, 10–13 June 2019; pp. 130–137.
110. Benedetti, P.; Piro, G.; Grieco, L.A. An Energy Efficient and Software-Defined Information-Centric Networking Approach to Consumer Mobility. In Proceedings of the 2020 22nd International Conference on Transparent Optical Networks (ICTON), Bari, Italy, 19–23 July 2020; pp. 1–4.

111. Arumaithurai, M.; Chen, J.; Monticelli, E.; Fu, X.; Ramakrishnan, K. Exploiting ICN for Flexible Management of Software-Defined Networks. 2014. Available online: http://conferences2.sigcomm.org/acm-icn/2014/papers/p107.pdf (accessed on 26 November 2022).
112. Arumaithurai, M.; Chen, J.; Maiti, E.; Fu, X.; Ramakrishnan, K.K. Prototype of an ICN based approach for flexible service chaining in SDN. In Proceedings of the 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hong Kong, China, 26 April–1 May 2015; pp. 5–6.
113. Bacher, F.; Rainer, B.; Hellwagner, H. Towards controller-aided multimedia dissemination in Named Data Networking. In Proceedings of the 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Turin, Italy, 29 June–3 July 2015; pp. 1–6.
114. Chen, Q.; Yu, F.R.; Huang, T.; Xie, R.; Liu, J.; Liu, Y. Joint Resource Allocation for Software-Defined Networking, Caching, and Computing. *IEEE/ACM Trans. Netw.* **2018**, *26*, 274–287. [CrossRef]
115. Chen, Q.; Yu, F.R.; Huang, T.; Xie, R.; Liu, J.; Liu, Y. An Integrated Framework for Software Defined Networking, Caching, and Computing. *IEEE Netw.* **2017**, *31*, 46–55. [CrossRef]
116. Agiwal, M.; Roy, A.; Saxena, N. Next Generation 5G Wireless Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1617–1655. [CrossRef]
117. Camps-Mur, D.; Gavras, A.; Ghoraishi, M.; Hrasnica, H.; Kaloxylos, A.; Anastasopoulos, M.; Tzanakaki, A.; Srinivasan, G.; Antevski, K.; Baranda, J.; et al. AI and ML Enablers for Beyond 5G Networks. 2021. Available online: https://5g-ppp.eu/wp-content/uploads/2021/05/AI-MLforNetworks-v1-0.pdf (accessed on 26 November 2022).
118. Karamplias, T.; Spantideas, S.T.; Giannopoulos, A.E.; Gkonis, P.; Kapsalis, N.; Trakadas, P. Towards Closed-loop Automation in 5G Open RAN: Coupling an Open-Source Simulator with xApps. In Proceedings of the 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit), Grenoble, France, 7–10 June 2022; pp. 232–237.
119. Salsano, S.; Blefari-Melazzi, N.; Detti, A.; Morabito, G.; Veltri, L. Information centric networking over SDN and OpenFlow: Architectural aspects and experiments on the OFELIA testbed. *Comput. Netw.* **2013**, *57*, 3207–3221. [CrossRef]
120. Eum, S.; Jibiki, M.; Murata, M.; Asaeda, H.; Nishinaga, N. A design of an ICN architecture within the framework of SDN. In Proceedings of the 2015 Seventh International Conference on Ubiquitous and Future Networks, Sapporo, Japan, 7–10 July 2015; pp. 141–146.
121. Adrichem, N.L.M.v.; Kuipers, F.A. NDNFlow: Software-defined Named Data Networking. In Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft), London, UK, 13–17 April 2015; pp. 1–5.
122. Cao, J.; Pei, D.; Zhang, X.; Zhang, B.; Zhao, Y. Fetching Popular Data from the Nearest Replica in NDN. In Proceedings of the 2016 25th International Conference on Computer Communication and Networks (ICCCN), Waikoloa, HI, USA, 1–4 August 2016; pp. 1–9.
123. Lee, M.; Song, J.; Cho, K.; Pack, S.; Kwon, T.T.; Kangasharju, J.; Choi, Y. Content discovery for information-centric networking. *Comput. Netw.* **2015**, *83*, 1–14. [CrossRef]
124. Katsaros, K.; Fotiou, N.; Vasilakos, X.; Ververidis, C.; Tsilopoulos, C.; Xylomenos, G.; Polyzos, G. On Inter-Domain Name Resolution for Information-Centric Networks. In *International Conference on Research in Networking*; Springer: Berlin/Heidelberg, Germany, 2012. [CrossRef]
125. Liu, H.; Azhandeh, K.; de Foy, X.; Gazda, R. A comparative study of name resolution and routing mechanisms in information-centric networks. *Digit. Commun. Netw.* **2019**, *5*, 69–75. [CrossRef]
126. Fotiou, N.; Katsaros, K.V.; Xylomenos, G.; Polyzos, G.C. H-Pastry: An inter-domain topology aware overlay for the support of name-resolution services in the future Internet. *Comput. Commun.* **2015**, *62*, 13–22. [CrossRef]
127. Rowstron, A.; Druschel, P. *Pastry: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-Peer Systems*; Cornell University: Ithaca, NY, USA, 2001; Volume 2218, pp. 329–350.
128. Stoica, I.; Morris, R.; Liben-Nowell, D.; Karger, D.; Kaashoek, F.; Dabek, F.; Balakrishnan, H. Chord: A scalable peer-to-peer lookup protocol for Internet applications. *IEEE Trans. Netw.* **2003**, *11*, 17–32. [CrossRef]
129. D'Ambrosio, M.; Dannewitz, C.; Karl, H.; Vercellone, V. MDHT: A Hierarchical Name Resolution Service for Information-Centric Networks. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking, Toronto, ON, Canada, 19 August 2011; pp. 7–12. [CrossRef]
130. Verisign. Verisign Q1 2022 Domain Name Industry Brief. Available online: https://www.verisign.com/en_US/domain-names/dnib/index.xhtml (accessed on 6 September 2022).
131. Majed, A.-q.; Wang, X.; Yi, B. Name Lookup in Named Data Networking: A Review. *Information* **2019**, *10*, 85. [CrossRef]
132. Afanasyev, A.; Jiang, X.; Yu, Y.; Tan, J.; Xia, Y.; Mankin, A.; Zhang, L. NDNS: A DNS-Like Name Service for NDN. In Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, Canada, 31 July–3 August 2017; pp. 1–9.
133. Qiao, X.; Wang, H.; Tan, W.; Vasilakos, A.V.; Chen, J.; Blake, M.B. A survey of applications research on content-centric networking. *China Commun.* **2019**, *16*, 122–140. [CrossRef]
134. Xiong, Z.; Zhang, Y.; Niyato, D.; Deng, R.; Wang, P.; Wang, L.C. Deep Reinforcement Learning for Mobile 5G and Beyond: Fundamentals, Applications, and Challenges. *IEEE Veh. Technol. Mag.* **2019**, *14*, 44–52. [CrossRef]
135. Li, J.; Zhang, X. Deep Reinforcement Learning-Based Joint Scheduling of eMBB and URLLC in 5G Networks. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 1543–1546. [CrossRef]

136. Rahman, A.; Hasan, K.; Kundu, D.; Islam, M.J.; Debnath, T.; Band, S.S.; Kumar, N. On the ICN-IoT with federated learning integration of communication: Concepts, security-privacy issues, applications, and future perspectives. *Future Gener. Comput. Syst.* **2023**, *138*, 61–88. [CrossRef]
137. Ma, X.; Liao, L.; Li, Z.; Lai, R.X.; Zhang, M. Applying Federated Learning in Software-Defined Networks: A Survey. *Symmetry* **2022**, *14*, 195. [CrossRef]
138. Marini, F.; Walczak, B. Particle swarm optimization (PSO). A tutorial. *Chemom. Intell. Lab. Syst.* **2015**, *149*, 153–165. [CrossRef]
139. El-Abd, M. Global-best brain storm optimization algorithm. *Swarm Evol. Comput.* **2017**, *37*, 27–44. [CrossRef]
140. Abdel-Basset, M.; Abdel-Fatah, L.; Sangaiah, A.K. Chapter 10—Metaheuristic Algorithms: A Comprehensive Review. In *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*; Sangaiah, A.K., Sheng, M., Zhang, Z., Eds.; Academic Press: Cambridge, MA, USA, 2018; pp. 185–231. [CrossRef]
141. Khelifi, H.; Luo, S.; Nour, B.; Moungla, H.; Ahmed, S.H.; Guizani, M. A blockchain-based architecture for secure vehicular Named Data Networks. *Comput. Electr. Eng.* **2020**, *86*, 106715. [CrossRef]
142. Li, H.; Wang, K.; Miyazaki, T.; Xu, C.; Guo, S.; Sun, Y. Trust-Enhanced Content Delivery in Blockchain-Based Information-Centric Networking. *IEEE Netw.* **2019**, *33*, 183–189. [CrossRef]
143. Pan, Q.; Wu, J.; Li, J.; Yang, W.; Guan, Z. Blockchain and AI Empowered Trust-Information-Centric Network for Beyond 5G. *IEEE Netw.* **2020**, *34*, 38–45. [CrossRef]
144. Shi, J.; Zeng, X.; Han, R. A Blockchain-Based Decentralized Public Key Infrastructure for Information-Centric Networks. *Information* **2022**, *13*, 264. [CrossRef]
145. Abdellah, A.S.; Saif, S.; ElDeeb, H.E.; Abd-Elrahman, E.; Taher, M. A Secured Blockchain-based Information-Centric Network. *J. Comput. Sci.* **2022**, *18*, 266–280. [CrossRef]
146. Lyu, Q.; Qi, Y.; Zhang, X.; Liu, H.; Wang, Q.; Zheng, N. SBAC: A secure blockchain-based access control framework for information-centric networking. *J. Netw. Comput. Appl.* **2020**, *149*, 102444. [CrossRef]
147. Ali, A.; Iqbal, M.M.; Jabbar, S.; Asghar, M.N.; Raza, U.; Al-Turjman, F. VABLOCK: A blockchain-based secure communication in V2V network using icn network support technology. *Microprocess. Microsyst.* **2022**, *93*, 104569. [CrossRef]
148. Abdellah, A.; Saif, S.M.; ElDeeb, H.E.; Abd-Elrahman, E.; Taher, M. A Survey of Using Blockchain Aspects in Information Centric Networks. In Proceedings of the International Conference on Advanced Intelligent Systems and Informatics, Cairo, Egypt, 19–21 October 2020; Springer: Cham, Switzerlan, 2021; pp. 292–301.
149. Asaf, K.; Rehman, R.A.; Kim, B.-S. Blockchain technology in Named Data Networks: A detailed survey. *J. Netw. Comput. Appl.* **2020**, *171*, 102840. [CrossRef]
150. Huang, S.; Chen, R.; Li, Y.; Zhang, M.; Lei, K.; Xu, T.; Yu, X. Intelligent Eco Networking (IEN) III: A Shared In-network Computing Infrastructure towards Future Internet. In Proceedings of the 2020 3rd International Conference on Hot Information-Centric Networking (HotICN), Hefei, China, 12–14 December 2020; pp. 47–52.
151. Din, I.U.; Asmat, H.; Guizani, M. A review of information centric network-based internet of things: Communication architectures, design issues, and research opportunities. *Multimed. Tools Appl.* **2019**, *78*, 30241–30256. [CrossRef]
152. Routray, S.; Mohanty, S. Why 6G? 2019. Available online: https://www.researchgate.net/publication/331700779_Why_6G (accessed on 26 November 2022).
153. Liao, S.; Wu, J.; Li, J.; Konstantin, K. Information-Centric Massive IoT based Ubiquitous Connected VR/AR in 6G: A Proposed Caching Consensus Approach. *IEEE Internet Things J.* **2020**, *8*, 5172–5184. [CrossRef]