

# The Road Ahead for Networking: A Survey on ICN-IP Coexistence Solutions

Mauro Conti, *Senior Member, IEEE*, Ankit Gangwal, Muhammad Hassan, Chhagan Lal, Eleonora Losiouk\*

**Abstract**—In recent years, the usage model of the Internet has changed, pushing researchers towards the design of the Information-Centric Networking (ICN) paradigm as a possible replacement of the existing architecture. Even though both Academia and Industry have investigated the feasibility and effectiveness of ICN, achieving the complete replacement of the Internet Protocol (IP) is a challenging task: (i) the process involves multiple parties, such as Internet Service Providers (ISPs), that need to coordinate among each other; (ii) it requires an indefinite amount of time to update hardware and software of network components; and (iii) it is a high risk goal that might introduce unexpected complications. Thus, the process of replacing the current Internet will inevitably lead towards a period of coexistence between the old and the new architectures. Given the urgency of the problem, this transition phase will happen very soon and people should address it in a smooth way.

Some research groups have already addressed the coexistence by designing their own architectures, but none of those is the final solution to move towards the future Internet considering the unaltered state of the networking. To design such architecture, the research community needs now a comprehensive overview of the existing solutions that have so far addressed the coexistence. The purpose of this paper is to reach this goal by providing the first comprehensive survey and classification of the coexistence architectures according to their features (i.e., deployment approach, deployment scenarios, addressed coexistence requirements and additional architecture or technology used) and evaluation parameters (i.e., challenges emerging during the deployment and the runtime behaviour of an architecture). We believe that this paper will finally fill the gap required for moving towards the design of the final coexistence architecture.

**Index Terms**—Coexistence Solutions, Future Internet Architectures, Information-Centric Networking, Internet Protocol, Secure Transition.

## I. INTRODUCTION

The current Internet architecture was designed for a small research community over three decades ago with the purpose of interconnecting multiple heterogeneous networks. At that time, nobody foresaw the popularity and longevity that the Internet architecture started gaining in late '80s and early '90s and that led towards the connection of over 3 billion of mobile and desktop devices. Today, people exploit networking devices for a variety of purposes, that go from simple web browsing to video conferencing and content distribution, with

the expectation of being always connected, regardless of their time and place. The misalignment between the original design and the current usage highlighted the limitations of the IP-based architecture and motivated researchers to explore new solutions to overcome them. Among those limitations, the primary concern is the performance of the current Internet, which has to cope with the huge number of connected devices all over the world and with the new pattern of use of the network. According to this study [1], currently there are around 23 billions of connected devices in the world, each one identified by a unique IP address and consuming the network bandwidth. With such a huge number of devices, the first issue is the availability of unique IP addresses to be assigned. Even though researchers originally chose to allocate 32 bits to compose an IP address through the IPv4 protocol, they had to introduce the IPv6 protocol to extend the number of allocated bits from 32 to 128. Network Address Translation (NAT) [2] is also another solution addressing the same problem, and it allows to assign the same public address to a set of devices belonging to the same private network. Thus, when using the private network each device has its own IP address, chosen within a range of private IP addresses, but, for an entry external to the network, all the devices have the same public IP address. To enable the communication between the private network and the Internet a firewall is responsible for intercepting a request, forwarding it to the Internet with the public IP and redirecting the incoming response to the appropriate device.

Another problem is given by the type of network traffic: most of it is made of HyperText Transfer Protocol (HTTP) requests, which means that users have changed the way they use Internet from a low-bandwidth interactive and store-and-forward approach towards a web and content dominated traffic. To support this, Cisco Visual Networking Index [3] shows that in recent years video traffic delivery has suddenly become very popular on the Internet, with an Internet traffic that will be 194 exabytes per month by 2021, and multimedia traffic up to 82%, from 70% in 2015. Furthermore, due to the technological advancements in hardware devices and an increasing deployment of pervasive computing application, it is indicated that the number of communicating devices (including smart devices) will be three-times more than the world's population [4]. Moreover, it has also been reported [5] that 86% of worldwide user traffic consists of only video data, which consists of Video on Demand (VoD), video streaming, Point to Point (P2P), and Television (TV).

Finally, from a security and privacy point of view the current Internet is not even able to guarantee some essential requirements, such as origin authentication, data integrity or

\* Corresponding author.

Mauro Conti, Ankit Gangwal, Muhammad Hassan, Eleonora Losiouk are with the Department of Mathematics, University of Padua, 35121, Italy. email: {conti, gangwal, hassan, elosiouk}@math.unipd.it

Chhagan Lal is with Simula Research Laboratory, Norway. Email: chhagan@simula.no

Manuscript received Month DD, YYYY; revised Month DD, YYYY, Month DD, YYYY, and Month DD, YYYY; accepted Month DD, YYYY.

Digital Object Identifier XX.XXXX/COMST.YYYY.XXXXXXX

data confidentiality, because of its lack of security by design. This is the motivation for the introduction of solutions, such as Internet Protocol Security (IPsec) suite [6] or Transport Layer Security (TLS) [7], that work on top of the current Internet and are aimed at overcoming its limitations.

For the above-mentioned reasons, researchers started designing new Internet architecture, such as Recursive INternet-network Architecture (RINA) [8] or ICN [9], that might replace the current one in the future. Among those, the most promising architectures adhere to the ICN paradigm: a new network communication model in which the traditional host-centric paradigm has been moved to the new information-centric networking. While in the current Internet two endpoints can start communicating only if they know the respective IP address, explicitly or by use of a Domain Name System (DNS), in ICN they can send requests specifying only content names, without being aware of contents location in the network. This decoupling between request sending and content transferring introduces several benefits: reduction of latency and network load due to in-network caching [10–13], inherent content integrity [14] and better support for mobility due to name-based routing [15, 16].

Due to its benefits and potential next-generation applications, ICN is gaining significant attention from both Industry and Academia, and several works surveyed and evaluated ICN architectures from different perspectives:

- *Architectural components* - G. Xylomenos et al. [17] summarize the core functionalities and discuss the differences, as well as the key weaknesses, of the existing ICN architectures. B. Ahlgren et al. [18] focus their discussion about ICN architectures on the undertaken design choices and features.
- *Network features* - A. Ioannou et al. [12] focus on the ICN caching problem and provide an analysis of the existing caching policies, along with the forwarding mechanisms that complement these policies. I. Abdullahi et al. in [19] span their survey through some selected ICN projects by investigating the used caching approaches. In their survey, G. Zhang et al. [20] focus on the state-of-art techniques for reducing cache redundancy and improving the availability of cached content. M. Zhang et al. in [21] survey the caching mechanisms in detail, illustrate how they work, and analyze their possible benefits and drawbacks. Finally, M. F. Bari et al. in [22] analyze and compare the naming and routing mechanisms proposed by the existing ICN research projects.
- *Security and privacy attacks* - E. G. AbdAllah et al. in [23] survey the existing literature in the direction of security and privacy in ICN, and discuss some open questions. R. Tourani et al. in [24] provide a summary and a taxonomy of security attacks on ICN architectures, along with their impact on ICN features (e.g., naming, routing, and caching).
- *Software and tools* - In [25], M. Tortelli et al. portray the composition of open source software tools available for ICN, such as ndnSIM, ccnSim, CCNPL-Sim, and Icarus.
- *Application to mobile networks* - C. Fang et al. in [26] present a brief survey of mobile ICN and discuss on the

research issues and challenges. X. Liu et al. in [27] highlight the advances in ICN and analyze its development trends, mainly focusing on Information-centric mobile.

However, none of those survey articles discuss the research issues and challenges affecting an ICN-IP coexistence scenario, as we aim to do in this paper. Only in [28], researchers from InterDigital Inc. and Huawei Technologies Co. Ltd. provided a comparison among the existing coexistence architectures, but they focused specifically on the different deployment approaches chosen by each solution.

ICN is also a suitable networking model for various emerging technologies, such as Internet of Things (IoT) [29, 30] and 5G [16, 31]. In the first scenario, ICN can help with establishing the connectivity among smart devices in an IoT environment, as well as in a smart city, in a smart e-health, and in a smart grid context. Also, the management of the huge amount of data generated by IoT devices (i.e., the IoT big data) is challenging in the existing IP architecture, while it is minimized by the in-network caching feature in ICN. This feature allows to reduce the traffic load on data producers by caching data on intermediate routers. Additionally, the receiver-driven communication in ICN allows IoT-receivers to ask for data without revealing their location information, thus being privacy supporting. Similarly, there are various advantages coming up from an ICN-based 5G architecture (i.e., 5G-ICN): (i) 5G-ICN provides a single protocol able to handle mobility and security, instead of using a diverse set of IP-based Third Generation Partnership Project (3GPP) protocols (such as in the case of existing mobile networks, e.g., Long Term Evolution (LTE), 3G, 4G), (ii) it provides a unifying platform with the same layer-3 Application Programming Interfaces (APIs) to integrate heterogeneous radios (e.g., Wifi, LTE, 3G) and wired interfaces in the same network, (iii) it converges services like computing, storage, and networking over a single platform, which enhances the flexibility of enabling virtualized service logic and caching functions anywhere in the network.

**Motivation.** The benefits of ICN can occur only in a full-ICN scenario, which implies a complete replacement of the current Internet. Despite its obvious need, this is a long and complex process, that requires the coordination among the different parties (i.e., ISPs), time, costs for updating hardware and software of the network components and ability to face all the new possible challenges. Previous attempts to replace a widely used technology, protocol or architecture (e.g., IPv4/IPv6 protocol, 3G/4G technology, 4G/5G technology) have always faced a long period of coexistence between the old and the new solution. In the same way, the replacement of the current Internet will involve a transition phase during which IP and ICN architectures will coexist. More specifically, we envision that in a coexistence scenario there will be ICN and IP “islands” surrounded by an IP or an ICN “ocean”, where an “island” will be a single device, a computer, an application or a server running either the ICN or the IP protocol, while an “ocean” will be a network containing components, that run different architectures.

Researchers working in this field have already addressed the coexistence of IP and ICN following two separate approaches. In the first, the research groups designed future Internet archi-

tures facing the coexistence only during the deployment of their testbeds and without considering it as part of the initial design. On the contrary, in the second case, the design of the future Internet architectures specifically addressed the coexistence of IP and ICN.

All the existing networking solutions that consider the coexistence are affected by a strong limitation: the lack of a comprehensive approach in addressing the coexistence. The purpose of those solutions is to improve a network performance indicator, without considering all the issues that arise in a coexistence scenarios, especially those regarding the security and privacy of the end users. To design the first complete coexistence architecture, it is necessary first to have a comprehensive overview of strengths and weaknesses of the existing solutions.

**Contribution.** The purpose of this paper is to provide the first complete survey and classification of the existing coexistence solutions. Details of ICN and of its working methodology are out of scope for this paper, since there are already several surveys addressing this aim [16, 24, 32]. Overall, the contributions of this paper are as follows:

- 1) We define a set of relevant features necessary for comprehensively analyze a coexistence architecture.
- 2) We provide the first comprehensive classification of all the main coexistence solutions.
- 3) We discuss the open issues and challenges affecting the existing coexistence architectures, by providing possible insights to design a more reliable future Internet architecture.

**Organization.** The paper is organized as follows: in Section II, we introduce the ICN concept, by comparing it with the current IP architecture and by illustrating its main benefits; Section III describes all the criteria we identified and used for the analysis and classification of the coexistence architectures; in Section IV we deeply illustrate each coexistence architecture and provide the motivation for our classification; in Section V, we discuss the main strengths and limitations of the current coexistence architectures, providing insights for improving the design of the future Internet; finally, in Section VI we conclude the paper.

## II. BACKGROUND

The purpose of this section is to provide an overview of the ICN paradigm (Section II-A), a comparison of the main features of IP and ICN architectures (Section II-B), the benefits of ICN (Section II-C) and, finally, the emerging technologies (Section II-D).

### A. ICN Overview

The ICN concept was first implemented in 2001 in the TRIAD project [33], by introducing a new *content layer* in the IP communication model. This layer provided several content-based features, among which: hierarchical content caching, content replication and content discovery, multicast-based content distribution, and name-based routing. Moreover, the layer supported end-to-end communication based on content name and Uniform Resource Locator (URL) by relying on

IP addresses only to reduce the role of transient routing tags. Although TRIAD routing mechanism used content names instead of IP addresses, the Transmission Control Protocol (TCP) and the IP protocols were still the backbone of the proposed architecture. In 2006, UC Berkeley and ICSI proposed the Data-Oriented Networking Architecture (DONA) [34], which improved TRIAD by incorporating data authenticity and persistence as key objectives of the architecture, but still having a strong dependency on the underlying TCP/IP. In 2009, the Palo Alto Research Center (PARC) revealed the Content-Centric Networking (CCN) [35] project. Soon after, the National Science Foundation (NSF) introduced its “Future Internet Architecture” program, which paved the way for Named-Data Networking (NDN) [36] - a branch of the CCN project. Both CCN and NDN significantly moved the TRIAD and DONA projects forward, by introducing a new network layer to definitely replace the existing TCP and IP ones. Thus, CCN and NDN are considered two key projects due to the considerable attention they brought to the ICN paradigm from both Academia and Industry, influencing also the design of the ICN architecture [37].

### B. Comparison Between IP-based and ICN-based Internet Architectures

Originally developed as part of the ARPANET project [38] during the 1960s, the current Internet is now often referred as TCP/IP architecture due to its most well-known protocols (i.e., TCP and IP). On the contrary, the ICN paradigm was first introduced in the TRIAD project [33] in 2001 and, then, followed by several architectures adhering to its new communication model. Since ICN is a paradigm, we will consider here the five main architectures to describe the technical features of the future Internet, while we will provide a comprehensive description of all the architectures addressing the ICN-IP coexistence in Section IV: (i) the DONA architecture [34], (ii) the CCN architecture [35], (iii) the NDN architecture [36], (iv) the Publish-Subscribe Internet Technologies (PURSUIT) architecture [39], and (v) the Network of Information (NetInf) architecture [40].

**Protocol Stack.** Both TCP/IP and ICN rely on a layered protocol stack, which is comparable to the Open Systems Interconnection (OSI) Reference Model [41], as shown in Fig. 1.

The TCP/IP stack includes the following four layers [42]:

- *Application* - it combines the functionality of the *Application*, *Presentation* and *Session* layers of the OSI model. It is responsible for sending and receiving data and it is specific for a particular type of application (e.g., DNS, HTTP).
- *Transport* - it targets the *Transport* layer of the OSI model and it is responsible for the end-to-end data transfer and data streams. Its most important protocols are TCP, which provides a reliable and connection-oriented service, and User Datagram Protocol (UDP), which offers an unreliable and connection-less service.
- *Internet* - equivalent to the *Network* layer of the OSI model, it provides addressing and routing functionalities

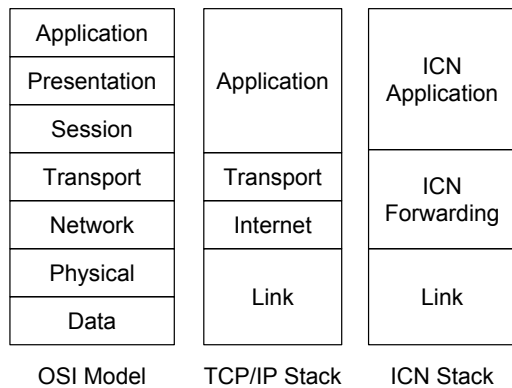


Fig. 1: Adaptation of the OSI seven layer model in the TCP/IP and ICN protocol stacks.

to ensure the delivery of messages to their destination. IP is the most important protocol, but it does not provide flow control or error handling.

- *Link* - equivalent to the *Data* and *Physical* layers of the OSI model, it manages the interaction among physical network components and it works as an interface with the network hardware.

Since the ICN stack is an evolution of the TCP/IP one [43–45], each layer is described with respect to the corresponding one in the Internet stack. More specifically, the layers of the ICN stack are the following ones:

- *ICN Application* - the protocols of this layer address content names instead of hosts locations. For example, the URL inside an HTTP request is replaced with the complete name of a content.
- *ICN Forwarding* - for any ICN-compliant architecture this layer offers routing functionalities for ICN interest and data packets equivalent to the TCP/IP *Network* layer in such a way that source and destination IP addresses are removed from the network packets and only the addressed content name is declared. According to the specific architecture, this layer can also provide the features of the TCP/IP *Transport* layer. In that case, the Interest/Data messages replace the TCP/IP segment/acknowledgement (ACK) messages and the content requester becomes responsible for the message sending rate in place of the content source (producer or intermediate router).
- *Link* - to be ICN-compliant, this layer introduces a mapping between Media Access Control (MAC) addresses and content names.

**Routing.** The purpose of the routing functionality is to route network packets from the source node till the destination node on one way and, then, from the destination to the source on the other.

Each TCP/IP packet specifies both source and destination nodes by including their IP addresses. An IP address is the unique identifier of each network component and it contains both the address of the network and the address of the specific component within that network. In the current Internet, routers are the main responsible for the routing functionality.

Equipped with at least two IP interfaces (i.e., an incoming and an outgoing one), each router receives IP packets in the incoming interface and checks whether there is a match, based on the longest prefix, in its Forwarding Information Base (FIB) internal data structure. The FIB contains a mapping between a network prefix and a router’s outgoing interface, together with the next-hop IP address. If there is a match in the FIB for the incoming packet, this is forwarded through the outgoing interface towards the next node in the network.

In ICN, the routing functionality differs according to the specific design of each architecture, but they all have a common design choice: the packets sent by a requester contain only the full name of the content and no IP addresses, neither the content requester’s one nor the content source’s one. In NDN and CCN architectures, contents are expressed through hierarchical names and routers use a longest-prefix match approach to find a possible entry in their FIB, which returns the name-prefix/prefixes of the next node/nodes in the network. On the contrary, DONA exploits a flat naming scheme to point to the contents available in the network and a name-based routing to redirect the packets until they reach the content source. A different approach is used by PURSUIT, which relies on a publish/subscribe model. Publishers publish their contents in the network and subscribers ask for a specific content by using a flat name scheme, made of two components: the Rendezvous Identifier (RI) and the Scope Identifier (SI). The first element addresses the component responsible to find the match between publisher and subscriber for a specific content, while the second is used to identify the sub-network where the rendezvous is. Once the subscriber obtains the location of the publisher from the rendezvous node, it sends its packet to the Topology Manager (TM) of the network where the content publisher is. The TM, then, identifies the path from the publisher to the subscriber and adds a series of Forwarding Identifiers (FIs) to the header of the packets. After that, the Forwarding Nodes (FNs) forward the packets only by using the FIs, without any routing table. Finally, the NetInf architecture adheres to both the approaches: name resolution, based on the publish/subscribe paradigm, and name-based routing.

**Name Resolution.** In the TCP/IP architecture there is a dedicated network component responsible for the name resolution, which is the DNS. This is a distributed service, which translates domain names, expressed in hierarchical URLs, into the corresponding IP addresses. The Internet is organized into separate DNS zones, each one under the direct control of an authoritative DNS server, and everytime a network device sends a request to its local DNS server, this might reply with a value saved in its cache or, otherwise, forward the same request to a remote server.

In ICN, the name resolution differs according to the chosen forwarding approach. In case of name-based routing, the requester specifies a content by providing its full name, which is the same analyzed by the routers to find the next hop in the network. On the other hand, in the name resolution approach, used by PURSUIT or NetInf, there is always a dedicated node in the network, which is responsible for the mapping between publishers and subscribers.

**Storing.** In the TCP/IP architecture, routers do not have

caching features, while in ICN, caching is fundamental and almost any node is able to cache contents and to serve the corresponding requests.

**Traffic Management.** In the current Internet, the traffic management, in terms of connection management, flow control and congestion control, is guaranteed by the TCP protocol. The establishment of a connection is regulated by the three-way handshake mechanism, through which the TCP protocol checks for the availability of the remote server, before exchanging any data with it. Only at the end of the handshake, the real communication starts, together with the data exchange, and it is regulated by the introduction of sequence numbers in the message blocks that enable the destination node to properly order all the received messages. The flow control is provided by the ACK messages received by the sender from the receiver every time a packet has been properly delivered. Thus, a sender never overflows the receiving host because the re-transmission of a packet is performed only after a timeout, which corresponds to either an ACK not received by the sender or three ACKs received. Finally, the congestion control refers to the prevention of the routers from becoming overflowed.

In ICN, some architectures, such as DONA, still rely on the existing transport protocols so that all the forwarding mechanisms and transport functionalities are guaranteed. However, other ICN solutions, such as NDN, do not provide the *Transport* layer functionalities and, instead, delegate them to the application itself or to the network packets. After a certain timeout, an application can transmit again a packet, which by design has a limited lifetime to prevent network congestion. Moreover, the availability of distributed caches, which means contents, all over the network should prevent losses due to congestion.

### C. Benefits of ICN-based architectures

The following ones are the key ICN benefits, which better motivate why this architecture is a potential candidate for the future Internet.

1) *Scalable and Cost-Efficient Content Distribution:* In a future world where the mobile video traffic will be dominant (e.g., video data will consume more than 80% of the IP traffic, wireless mobile devices will generate two-third of the Internet traffic [46], Netflix and YouTube together amount nearly 50% of Internet traffic), the current network operators will face challenges in meeting the bandwidth requirements from end users. Thus, the inherent ICN support for caching at the network layer [35], together with the receiver-driven mechanism, the inherent support for mobility and the multi-cast routing, make ICN fit the new network use in a multimedia streaming context [47–52].

2) *Mobility and Multihoming:* ICN also meets the requirements of the 5G network, such as global Internet access and user mobility over dense and heterogeneous networks by adapting to multiple radio access technologies (e.g., Wi-Fi and LTE). As a matter of fact, ICN supports the mobility at the network layer by decoupling time and space between request resolution and content transfer [16]. In particular, two

fundamental ICN features encourage seamless consumer mobility [15, 16]. The first is the receiver-driven communication model, where it is up to the consumer to request location-independent contents. The second is the connection-less request/response communication model between consumer and producer. Therefore, when a mobile consumer connects to a new Point of Attachment (PoA), the above two features allow the consumer to re-issue interests for the data that he has not received from the previous PoA. On the contrary, producer mobility is more challenging in ICN because of no distinction between routing locator and content identifier. Previous work have already proposed new solutions for an efficient management of producer mobility in ICN [15, 53].

3) *Disruption Tolerance:* Achieving an end-to-end communication through TCP/IP transport sessions in challenged networks is often difficult due to the sparse connectivity, high-speed mobility, and disruptions of such networks. Since the application protocol sessions are bound to transport sessions, the communication fails as soon as the transport session fails. In the current Internet, several applications do not require seamless communication with end-to-end paths [54]. As the primary objective is to access data objects, ICN is the perfect approach for Delay-Tolerant Networking (DTN) architectures [55, 56] due to the in-network caching with hop-by-hop transport functionality, which provides a store-and-forward mechanism and enables a better performance and reliability.

4) *Security:* Unlike the TCP/IP architecture, the ICN design comes with the security in mind. In particular, in ICN the security follows a data-centric model, which focuses on the importance of guaranteeing content integrity and source authentication. For a content-centric architecture, where contents can be located and provided in any point of the network, and not only by the original content producer, the above-mentioned features are particularly significant. To achieve this aim, ICN contents are always signed by the producer, thus allowing consumers to always verify content integrity and data-origin authentication [57].

### D. Emerging Technologies

Before thinking of redesigning the whole Internet architecture, researchers and companies have provided several solutions, which work on top of the current Internet, to overcome some of its limitations. Among those, the most successful attempts are the following emerging architectures: Software-Defined Networking (SDN), Network Functions Virtualization (NFV), Content Delivery Network (CDN) and DTN.

1) *Software-Defined Networking:* SDN [58] is an emerging networking paradigm that separates network control logic (i.e., the control plane) from the underlying switches and routers that forward the traffic (i.e., the data plane). By separating the control and data planes, the network switching/routing devices become simple forwarding devices and the control logic is incorporated in a logically centralized controller. This separation primarily helps in simplifying network (re)configuration, policy enforcement, and evolution [59]. The control plane and the data plane communicate via a well-defined programming

interface, i.e., the forwarding elements of the data plane request for instructions from the controller as well as the controller has direct control over the data plane elements using APIs. The most popular flavor of such APIs is OpenFlow [60]. An OpenFlow switch has one or more flow tables for handling packet-rules. When a rule matches with the incoming traffic, the OpenFlow switch performs certain actions (forwarding, modifying, dropping, etc.) on the traffic flow. The rules installed by the controller decide the role of an OpenFlow switch, i.e., it can behave as a switch, router, firewall, or middlebox (such as traffic-shaper, load-balancer).

2) *Network Functions Virtualization*: Diversity and dominance of proprietary appliances made service deployment, as well as testing, complex. NFV [61] was designed as a technology to leverage Information Technology (IT) virtualization by exporting network functions from the underlying dedicated hardware equipment to general software running on Commercial Off-The-Shelf (COTS) devices. Using NFV, the key network functions can be performed at various network locations, e.g., network nodes, data-centers, network edge, as required. NFV is different from SDN, and it only deals with the virtualization of network functions.

3) *Content Delivery Network*: The initial implementation of the Internet was designed to manage the traffic in a passive, end-to-end, and “best effort” approach [62]. With the explosion of user data and commercial content over the Internet, the “best effort” approach for traffic management became inefficient and unscalable. To handle this situation, CDN [62–64] was designed [46, 65, 66]. Nowadays, CDN appears as an integral and essential overlay network for the Internet [67–69] since it primarily aims to improve bandwidth availability, accessibility, and precise content delivery through content replication.

CDN architecture consists of several cache servers that are strategically located across the Internet. Typically, CDN holds a hierarchy of servers with multiple Points-of-Presence (PoP) that stores copies of identical content to satisfy user’s demand from most appropriate/closest site [70]. It also has back-end servers for intra-CDN content distribution. CDN categorically distributes web contents to the cache servers, which are positioned close to the users. As a result, CDN offers fast, efficient, and reliable web services to the users.

There are two fundamental approaches for the deployment of CDN: (i) overlay model, where content is replicated to thousand of servers worldwide, and (ii) network model, where routing configurations recognize the application services and forward them based on the predefined policies.

Even though CDNs improve content delivery, their performance is limited by the underlying ISPs. Usually, CDNs do not manage independent packet data services, rather they rely on the ISPs to make packet routing decisions. Moreover, both ISPs and CDN collectively provide end-to-end Quality of Experience (QoE)<sup>1</sup> for content delivery. Thus, coordination between ISPs and CDN providers causes a massive impact on the overall QoE [67].

4) *Delay-Tolerant Networking*: In the late 1990s, the widespread use of wireless protocols, together with an in-

creasing interest in vehicular communication, encouraged researchers to design the Interplanetary Internet (IPN) architecture. This was the first attempt to address the need of long distance communications that were inevitably affected by packets loss/corruption and delays. DTN [71] was first introduced as an adaptation of the IPN for terrestrial networks [72]: it is an overlay architecture that operates above the protocol stack of *ad-hoc* wireless networks and enables gateway functionality to interconnect them. To provide communication among networks having excessive delays due to highly repetitive link disruptions, DTN adopts the “store-carry-forward” routing scheme [73]: the main idea of this scheme is to have multiple nodes distributed over the network, each one able to receive a copy of the same message and then send it back to the destination node. This way, the delivery performance is improved and the destination node can receive the message from any location inside the network.

### III. COEXISTENCE ARCHITECTURES: FEATURES AND EVALUATION PARAMETERS

In order to classify the existing architectures, we identified the necessary features and evaluation parameters to have a complete overview of each coexistence solution. The former come with the design of a coexistence architecture, while the latter refer to the challenges introduced during its deployment in a real scenario. The features are as follows: *deployment approaches*, *deployment scenarios*, *addressed coexistence requirements* and *additional architecture or technology used*. On the other side, the evaluation parameters are: *traffic management*, *access control*, *scalability*, *dynamic network management* and *latency*. In the remaining part of this section, we will describe features (Section III-A) and evaluation parameters (Section III-B) used for analyzing each coexistence architecture.

#### A. Features

1) *Deployment Approaches*: The deployment of ICN into the TCP/IP architecture inevitably raises the following question: *How to introduce the ICN protocol into the TCP/IP protocol?* To achieve this aim, researchers identified three possible approaches, shown in Fig. 2: *overlay* in case of ICN running on top of the IP protocol, *underlay* in case of ICN running under the IP protocol and *hybrid* in case of a coexistence of both IP and ICN protocols [28]. In the *overlay* deployment approach, the aim is to enable the communication among several ICN “islands” in an IP “ocean” and is achieved through a tunnel over the Internet protocol. On the contrary, the *underlay* solution involves the introduction of proxies and protocol conversion gateways near to either ICN or IP “islands” to properly deliver and receive outgoing and incoming requests. As an example, an HTTP request sent to an ICN “island” is intercepted by a gateway, which is responsible for translating it into an ICN Interest. Then, the resulting ICN data packet is translated again into an HTTP reply sent back to the requester. Finally, the *hybrid* approach claims the coexistence of both ICN and IP, by adopting dual stack nodes able to handle the semantics of both IP and ICN packets. Given

<sup>1</sup>QoE is an all-inclusive model, which defines the quality perceived by a user when retrieving content or applications over the Internet.

the diversity of the two protocols, from a semantic and format point of view, a dual stack node can use various options to infer content names from an IP packet, such as performing deep packet inspection in the payload or looking into the content name in the IP option header.

2) *Deployment Scenarios*: The purpose of this feature is to analyze all the possible scenarios in which a coexistence architecture can be deployed among the others we identified and that are illustrated in Fig. 3.

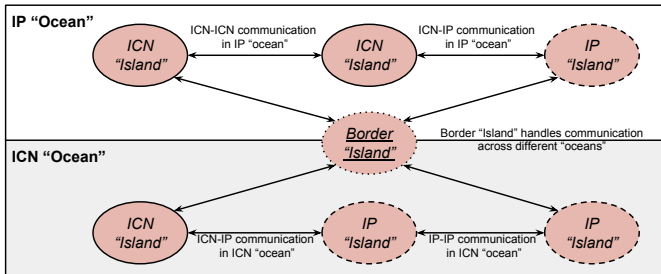


Fig. 3: Deployment scenarios for a coexistence architecture.

Each deployment scenario involves two “islands”, which run either the same networking architecture or two separate ones, surrounded by an ICN or an IP “ocean”. The possible different deployment scenarios are as follows:

- *ICN-ICN communication in IP “ocean”*.
- *ICN-IP communication in IP “ocean”*.
- *ICN-IP communication in ICN “ocean”*.
- *IP-IP communication in ICN “ocean”*.
- *Border Island* - communication between different “islands” in separate “oceans”.

3) *Addressed Coexistence Requirements*: In a coexistence scenario, the heterogeneity of the different networks might generate conflicts that prevent each individual architecture from guaranteeing its main features and properties. For example, since most of the ICN architectures do not preserve the native transport functionalities provided by the TCP protocol of the current Internet, one of their most significant limitations

is the traffic management. In a coexistence scenario, there would be a conflict between an IP “island” implementing its own logic for managing the traffic network and an ICN “island”, which does not support the same features.

Examining previous works [74], we consider the following requirements as the necessary ones to be supported in a coexistence scenario:

- *Forwarding* - the network forwarding devices should be able to handle packets with diverse routing identifiers (e.g., the variable-lengths of content names lead to dissimilar size of prefix-set and thus, different forwarding table look-ups).
- *Storage* - the network devices should support in-network caching to serve the content request and reduce bandwidth consumption. Nevertheless, the storage capacity of network devices also affects the size of the index table for the cached content and the time required to match the content name in the index table.
- *Security* - the network devices should preserve the security policies enforced in one (source) network to another (destination) network such as authenticating the digital signatures of content objects for content-based security or privacy policies.
- *Management* - the network devices should support management-related operations such as traffic-shaping/engineering, load-balancing, and explicit path steering.

4) *Additional architecture or Technology Used*: ICN and IP are not the only architectures that can coexist, and even the coexistence could be improved using other technologies. More specifically, ICN well fits with several different technologies that are already deployed in the current Internet infrastructure. Among those, there are SDN, NFV or CDN. The purpose of this feature is to collect all the architectures that the coexistence solutions involve.

## B. Evaluation Parameters

As evaluation parameters, we considered the following challenges arising during the deployment of a coexistence

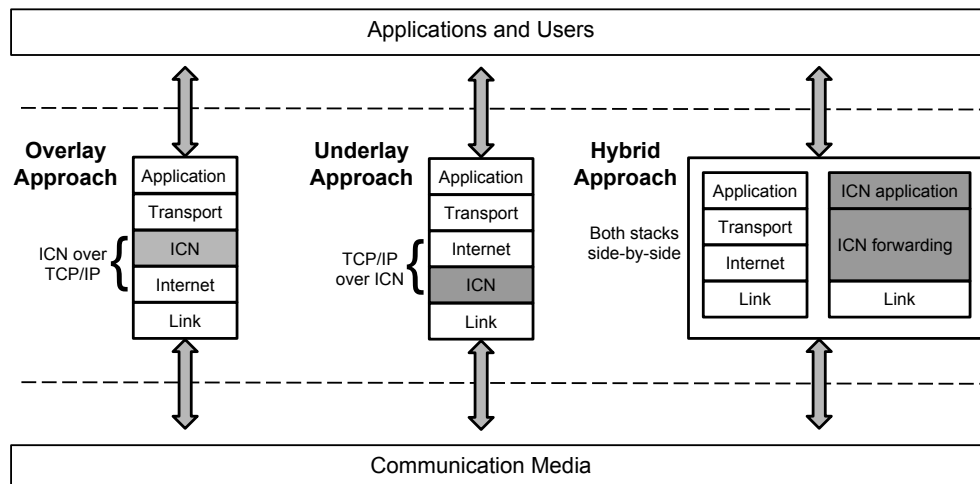


Fig. 2: Deployment approaches of ICN into the TCP/IP architecture.

architecture in a real scenario:

- *Access control* - in a networking context, access control uses a set of protocols to define, implement, and maintain policies that describe how the network nodes can be accessed by users/devices. Typically, it includes:
  - Authorization, authentication, and accounting of network connections.
  - Identity and access management.
  - Mitigation of non-zero-day attacks.
  - Policy lifecycle management.
  - Role-based controls of user, device, application.
  - Security posture check.
- *Scalability* - it ensures that the overall performance of a network will be not affected by the size of the network. In other words, scalability describes the ability of a network to grow and manage increasing demand.
- *Dynamic network management* - it is the process of administering and managing dynamic changes in computer networks, such as topology changes and handovers for seamless host mobility.
- *Latency* - it is defined as the amount of time a message takes to traverse a system. In a computer network, it is typically measured as the time required for a packet to be returned to its sender. The major factors for the network latency include propagation delays and delays due to routers, as well as storage devices.
- *Traffic management* - for a detailed description of the traffic management, we refer to Section II-B.

#### IV. CLASSIFICATION OF THE COEXISTENCE ARCHITECTURES

The purpose of this section is to illustrate the classification of the coexistence architectures according to the features and the evaluation parameters described in Section III. The summary of our findings is listed in Table I. Moreover, some of the architectures presented in this section has also shown their functional effectiveness and deployment feasibility with the help of real-world testbed deployments, we discuss the details about such those architectures in Section V-B.

##### A. PURSUIT

PURSUIT [75] was a European project financed by the Seventh Framework Program (FP7), started in September 2010 and ended in February 2013. PURSUIT is an evolution of the FP7 project Publish-Subscribe Internet Routing Paradigm (PSIRP) [39], proposing an ICN model based on a source node, that publishes an information, and on a client node, that subscribes to the content it desires. If the information is available, it will be delivered to the client. PURSUIT aims at improving PSIRP, meanwhile evaluating its performance, scalability, and coexistence with the current Internet network. Fig. 4 shows a simplified form of the architecture proposed in PURSUIT project.

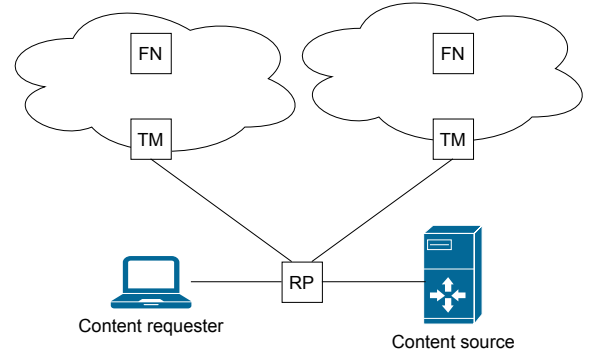


Fig. 4: Simplified view of the PURSUIT architecture.

PURSUIT architecture relies on the definition of a new data format and on the introduction of three new network components. PURSUIT addresses the data as information items, which consist of pair of identifiers, i.e., RI and SI. The former represents the real piece of information, while the latter specifies the group which the information belongs to. The three additional network functions addressed by PURSUIT are: Rendezvous Function (RF), Topology Function (TF), and Forwarding Function (FF). The RF plays a fundamental role in PURSUIT since it maps subscribers to publishers and supports names resolution. Moreover, it also initializes the delivery of information item to the client. The *Rendezvous Point (RP)* performs the RF and relies on a hierarchical distributed hash table internal data structure. The *TM* implements the TF by deploying a routing protocol to collect the topology of its domain and by exchanging routing information with other domains for global routing. The *Forwarding Node (FN)* implements the FF and it is also responsible for redirecting the information item to the client. In particular, the forwarding mechanism is label-based and uses a bloom filter [87] to speed up the information delivery. In addition, the *FN* offers also a caching facility.

As shown in Fig. 5, the PURSUIT node internal architecture encompasses several components, enabling the publish/subscribe communication model among the different stack layers. The *IPC Elements* implement a non-blocking inter-process mechanism, allowing users-space applications to issue publish/subscribe requests and communicate through the proposed prototype. The functionality of the *Local Proxy* element is to maintain a local record for all the pending subscriptions and, after receiving a request, dispatch it to the appropriate functions (i.e., *RF*, *FF*, *TF*). Finally, the *Communication Elements* are responsible for transmitting publications to the network. The design implementation of PURSUIT is based on Click elements [88]: it creates Ethernet frames and forwards them to the appropriate network interface. In addition, it provides the ability to utilize raw IP data packets as an alternative mechanism. This enables the prototype to be tested in Internet-wide scenarios.



TABLE I: Classification of the coexistence architectures (✓ Addressed ✗ Not addressed).

Parameter	Duration of the project/ Year of publication		PURSUIT [75]	NetInf [40]	NDN [76] & CCN [35]	O-ICN [77]	CONET [78]	GreenICN [79]	coCONET [80]	DOCTOR [81]	POINT [45]	RIFE [82]	CableLabs [83]	NDN-LAN [84]	hICN [85]	OPELIA [86]		
	2010 to 2013	2010 to today	2010 to 2013	2010 to 2013	2010 to today	2015	2010 to 2013	2013	2012	2014 to 2017	2015 to 2017	2015 to 2018	2016	2017	2018	2012		
Features	Deployment approaches	Overlay	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
		Underlay																
		Hybrid																
	Deployment scenarios	ICN-ICN communication in IP "ocean"	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		ICN-IP communication in IP "ocean"							✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		ICN-IP communication in ICN "ocean"									✓			✓	✓	✓	✓	✓
		IP-IP communication in ICN "ocean"									✓			✓	✓	✓	✓	✓
		Border Island									✓			✓	✓	✓	✓	✓
	Addressed coexistence requirements	Forwarding	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Storage	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Security		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Management		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Evaluation parameters	Additional architecture or technology used	PSIRP LAN	✓			SAIL SDN		SDN	SDN	NFV SDN	PURSUIT SDN	PURSUIT DTN	CDN	LAN	DNS	CONET SDN		
		Traffic management	✗		✗	✗								✗	✗			
		Access control																
		Scalability				✗		✗				✗			✗			
		Dynamic network management				✗						✗			✗			
		Latency									✗	✗			✗			
		NetInf transport functions Interaction.							SDN controller must manage every ICN request and rewrite several headers fields for every response packet.									
		New IP option overhead.																
		ICN capable OpenFlow-compliant network.																
		OpenFlow-compliant networking elements.																

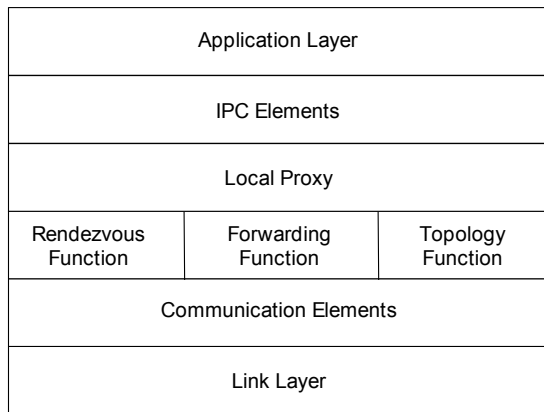


Fig. 5: Internal architecture of a PURSUIT node.

**Deployment Approach.** Trossen et al. [75] implemented a Layer-2 Virtual Private Network (VPN)-based *overlay* solution of PURSUIT among multiple nodes located in Europe, US and Asia. The prototype is established and verified on three different testbeds for experimental purposes, functioning as an overlay on LAN environment. To showcase a specimen of native ICN application, multimedia streaming services were hosted as a demonstration, showing a lossless transmission and comparable performance.

**Deployment Scenarios.** The *ICN-ICN communication in IP “ocean”* is the most suitable scenario for deploying PURSUIT, as it is also confirmed by the *overlay* approach adopted in the testbed.

**Addressed Coexistence Requirements.** PURSUIT guarantees the following three coexistence requirements:

- Forwarding - this is specifically provided by the *FN*, a software-based forwarder used for ICN messages exchange.
- Storage - the *FN*, which has the responsibility of redirecting information to the client, provides caching facility to furnish storage of information.
- Security - the security measures provided by PURSUIT refer to the access of information. Besides gathering information into groups, PURSUIT supports the information categorization into scopes, used for the definition of access privileges and policy implementations.

**Additional architecture or Technology Used.** PURSUIT is an evolution of the PSIRP project and its testbed has been realized as an *overlay* solution over a Local Area Network (LAN) environment.

**Evaluation Parameters.** The main issue introduced by the *overlay* deployment in the PURSUIT architecture is the traffic management. This is mainly due to the existing Internet applications and protocols, which are not completely compatible with the techniques implementing ICN over TCP/IP or UDP [9, 36, 89, 90] for traffic transport. Thus, many applications and protocols, such as HTTP based multimedia streaming protocols, might face false throughput estimations [91]. This is due to the TCP aggressiveness in presence of variations in content source location (e.g., dynamic caching and interest aggregation) [92].

## B. NetInf

The NetInf architecture [40] is the approach proposed by the European FP7 project SAIL [93], started in January 2010 and ended in February 2013. The key component of the NetInf architecture is the Convergence Layer (CL), which is able to map the information, expressed through any protocol (e.g., HTTP, TCP, IP, Ethernet), into specific messages compliant to a general communication paradigm. In particular, when two nodes communicate between each other, the functionality of a CL is to provide framing and message integrity to NetInf requests and responses.

Fig. 6 depicts the different CLs designed within the NetInf stack. In particular, CLs encompass an additional function (i.e., *Request Scheduling*) between the *NetInf Application* and the *NetInf Protocol*. The *CL1* functions over Ethernet, while *CL2* makes NetInf able to function over a variety of networks links and protocols such as HTTP, TCP/IP, Wireless Local Area Network (WLAN). The CLs also provide transport layer functions across different nodes such as flow control, congestion control and reliability.

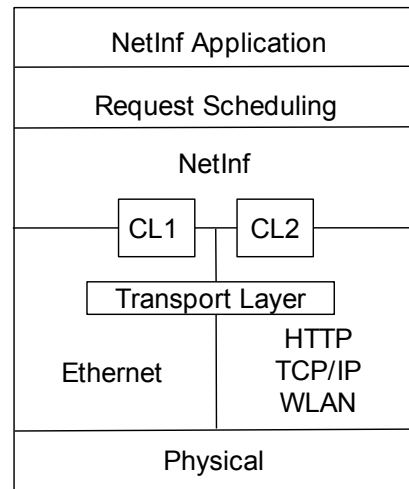


Fig. 6: Internal architecture of a NetInf node.

**Deployment Approach.** NetInf adheres to the *overlay* deployment approach, as it is confirmed by its first prototypes, deployed as an *overlay* strategy over TCP/UDP.

**Deployment Scenarios.** The NetInf architecture supports the *ICN-ICN communication in IP “ocean”* scenario.

**Addressed Coexistence Requirements.** The coexistence requirements provided by NetInf are as follows:

- Forwarding - NetInf guarantees both name-based forwarding and name resolution; NetInf message forwarding protocol relies on the lower-layer networking technology (e.g., TCP connection between two Internet hosts) and this communication is provided by the CLs.
- Storage - NetInf nodes support both on-path and off-path caching.
- Security - the CLs are responsible for the integrity of the messages exchanged in the architecture.

**Additional architecture or Technology Used.** Besides the standard TCP/UDP/IP tunneling, which is part of the *overlay* approach, NetInf does not rely on additional architectures.

**Evaluation Parameters.** The deployment of the NetInf architecture in a coexistence scenario introduces the following challenges: traffic management, due to the absence of interaction among the CLs transport functions and the NetInf transport functions, and access control. The first issue refers to the CLs, which are responsible for the interconnection of different types of networks into a single ICN network. For example, the interaction among the underlying protocols that provide really different communication services creates new challenges (e.g., from uni-directional, opportunistic message forwarding to flow- and congestion-controlled higher layer communication services; from delay-challenged to high-speed optical backbone networks). Concerning the access control limitation, in NetInf, it is not possible to apply controls over the accessibility levels of the information. Thus, anyone can access the published data without any restriction.

### C. NDN and CCN

Among the the existing implementations of the CCN paradigm [35], funded by the NSF [94] as part of the Future Internet Architectures program, there is the NDN research project [76]. From its first design late in 2010, the NDN main idea is to shift the existing IP host-to-host communication into a data oriented one by leveraging on an increased responsibility of the routers. Upon receiving a request for a content, the routers first check whether the content is already present in their cache (i.e., Content Store). If this is the case, they immediately return the content back, otherwise, they check the Pending Interest Table (PIT), searching for a pending request issued for the same content. If the PIT already contains an entry for the specific content, routers just collapse the current request into the PIT. If none of the previous cases verifies, routers forward the request to the next node in the network using the FIB, and keep waiting for the associated data to return back. Once the data packet arrives, all the pending interests for that content are satisfied just by sending the copy of data back to all the hosts which have requested it.

As shown in Fig. 7, NDN introduces some changes into the IP stack by adding the *Security* and *Strategy* novel layers: the first refers to the NDN design addressing the security of the content instead of the security of the communication channel between two nodes (which is how IP works); the second substitutes the network layer and provides the forwarding plane to forward *Content chunks* by giving the best choices to maintain multiple connectivities under varying conditions. In addition, the *Strategy* layer also supports security, scalability, efficiency and resiliency. Finally, NDN modifies the *Transport Layer* making it consumer-driven instead of producer-driven [95, 96], importing it into the NDN forwarding plane.

**Deployment Approach.** The common implementation of NDN and CCN includes *overlay* protocols, such as CCNx [9] and NDNLP [90], which are deployed over existing IP infrastructure. For instance, CCNx [89] showcases the explicit example of overlay by implementing CCN-over-UDP. In particular, it provides a method to transport CCNx messages between two nodes over UDP. Moreover, a concrete example of NDN

overlay architecture is provided by the `ndn-testbed`<sup>2</sup>, which connects multiple NDN nodes located in several continents over existing TCP/IP. The services provided in the trials of CCN/NDN include various projects, such as real-time video-conferencing [97], adaptive bit-rate streaming (not limited to end-to-end) [47, 49, 51] and `ndnSIM` (NDN simulator module on NS-3) [98].

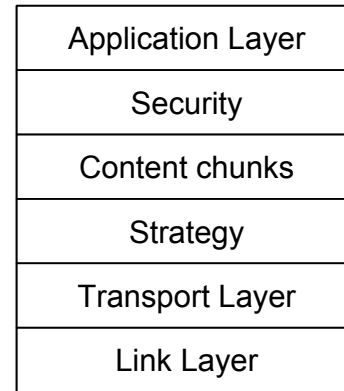


Fig. 7: NDN network stack [36].

**Deployment Scenarios.** NDN supports the *ICN-ICN communication in IP “ocean”* scenario, as it is confirmed by the `ndn-testbed`.

**Addressed Coexistence Requirements.** NDN guarantees the following three coexistence requirements:

- Forwarding - the router’s FIB is responsible for forwarding interests towards the content provider via one or more network interfaces based on the routes to the origin node(s). The requested data packet is then forwarded towards the requester by simply traversing, in reverse, the path of the preceding interest [36]. NDN supports also the multicast data routing, which improves receiver-driven multimedia delivery.
- Storage - NDN routers are enabled to cache contents.
- Security - NDN provides a data-centric security model where each data unit is uniquely signed by the data producer [99].

**Additional architecture or Technology Used.** Besides the standard TCP/UDP/IP tunneling, which is part of the *overlay* approach, the NDN project does not rely on additional architectures.

**Evaluation Parameters.** The tunneling approach, where NDN/CCN endpoints communicate over IP [100, 101], disowns the fundamental advantages of the content oriented networking (i.e., in-network caching and multicast forwarding) and the architectures implementing hop-to-hop connection-less (/oriented) connectivity (i.e., over TCP/UDP) suffer from a lack of traffic management [92]. In NDN/CCN networks, Congestion Avoidance (CA) is operated by the consumer rather than by the producer (server). This means that the Interests transmission rate is adapted in order to ensure that the delivery of a requested resource can make maximum fair use of the network. Existing NDN/CCN CA algorithms

<sup>2</sup><https://named-data.net/ndn-testbed/>

are largely based on the TCP CA algorithms, which assume that the bandwidth-delay product of the network fluctuates relatively slowly, as all the data packets traverse the same path from server to client. However, in NDN/CCN network content objects may be retrieved from various locations and may reach the consumer through different paths. Thus, the concept of a bandwidth-delay related to a single path and the use of TCP CA algorithms do not fit for NDN/CCN networks. In the NDN/CCN community, this is an active research area [102].

#### D. O-ICN

Overlay for Information-Centric Networking (O-ICN) [77] is a novel architecture, which leverages the SDN technology for separating data plane activities (i.e., forwarding and storing/caching of ICN contents) from control plane activities (i.e., naming, name resolution and routing). In particular, O-ICN introduces the ICN Manager as an extended version of a DNS server, which performs name resolution for both ICN and non-ICN requests. In case of an ICN request, the ICN Manager identifies the source of the content and sends to it the user's address, so that the source can route back the requested content to the user. In case of a non-ICN request, the standard routing mechanism of TCP/IP is followed. The naming scheme adopted by O-ICN is hybrid, i.e., both human readable and self-certifying as in the SAIL architecture [93]. Finally, the existing routers are modified to cache contents and communicate with the ICN Manager.

Fig 8a depicts the position of the novel *ICN-sublayer* proposed by O-ICN, which lies between the TCP/IP *Application Layer* and *Transport Layer*. More specifically, Fig. 8b describes the fields used by the new layer: the ICN flag bit (*F*), equal to 0 for an ICN request or to 1 for an ICN content; the three subsequent bits (1-4) reserved for additional purposes, and the remaining 28 bits for the total ICN header [103].

**Deployment Approach.** O-ICN relies on an *overlay* deployment solution by leveraging on the ICN Manager, which

performs dual tasks: name resolution, along with routing functionalities for ICN requests, and standard DNS resolution for the existing Internet requests. To evaluate the O-ICN architecture, authors in [103] present the Overlay ICN simulator (OICNSIM)<sup>3</sup>, an ns-3 based simulator where each O-ICN component is provided with helper classes and it is able to satisfy a wide variety of deployment scenarios. As an example, in [103], the authors studied the performance of OICNSIM for different ICN caching policies.

**Deployment Scenarios.** O-ICN supports the *ICN-ICN communication in IP "ocean"* scenario. Moreover, thanks to the ICN manager capability of manipulating both ICN and non-ICN requests, O-ICN can support also the *Border Island* deployment scenario.

**Addressed Coexistence Requirements.** The coexistence requirements addressed by O-ICN are as follows:

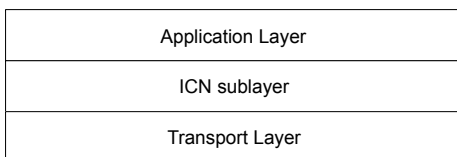
- Forwarding - the ICN Manager is responsible for the forwarding strategy.
- Storage - the data plane activities involve tactical storing/caching of ICN contents at different locations/routers/gateways and are managed by ICN routers.

**Additional architecture or Technology Used.** O-ICN exploits the SAIL solution for the naming scheme and the SDN technology for a separate management of data plane and control plane activities.

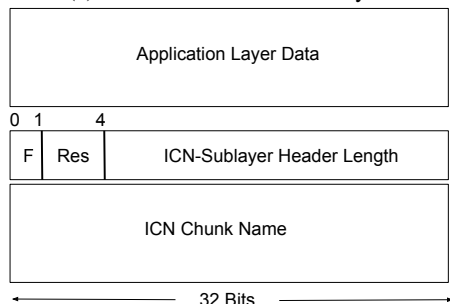
**Evaluation Parameters.** As for the previous *overlay* approaches, O-ICN is affected from a lack of traffic management. In addition, the overall solution suffers from scalability problems and the ICN manager is not able to guarantee its DNS functionalities in case of dynamic network conditions.

#### E. CONET

CONET [78] is an architecture designed for connecting several *CONET Sub System (CSS)*, which could be the whole Internet network, an IP autonomous system or a couple of network connected components. The main components of the CONET design, shown in Fig. 9, are as follows: *End-Node (EN)*, *Serving-Node (SN)*, *Border-Node (BN)*, *Internal-Node (IN)*, and *Name-System-Node (NSN)*. An *EN* requests some named-data by issuing an interest routed by the *BNs*, which are located at the border of *CSSs*. The route-by-name process identifies the *CSS* address of the next *BN*, which is closest to the *SN* as soon as the appropriate *CSS* is reached. Then, the *INs* forward the packet using the under-CONET routing engine. The *CSS* address of *EN* and the *CSS* addresses of the traversed nodes are appended to the packet. As soon as a CONET node is found to be able to provide the requested named-data, this is sent back on the reverse path to serve the requesting *EN*. All *BNs* and *INs* along the traversed path may cache the content.



(a) Position of the ICN sublayer.



(b) Detail of the ICN sublayer header format.

Fig. 8: Internal architecture of an O-ICN node.

<sup>3</sup>[https://www.nsnam.org/wiki/Contributed\\_Code](https://www.nsnam.org/wiki/Contributed_Code)

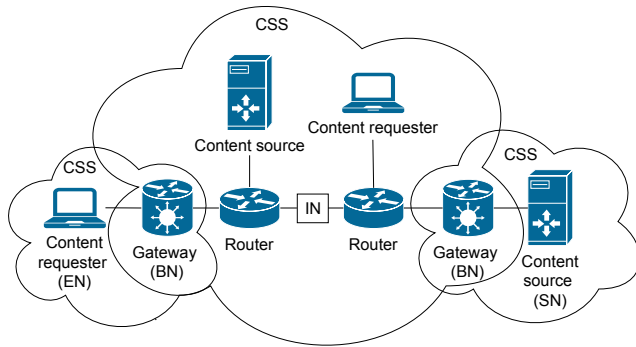


Fig. 9: Simplified view of the CONET architecture.

**Deployment Approach.** The CONET architecture can follow either an *overlay* or a *hybrid* deployment approach. In the first case, CONET works on top of the IP layer and the CSSs are nodes connected by overlay links (e.g., UDP/IP tunnels). In the second approach, the purpose is to make IP content-aware by introducing a novel IPv4 option or an IPv6 extension header. The network components will have then hybrid routing tables with both IP network addresses and names.

**Deployment Scenarios.** Considering the *overlay* solution, CONET supports the *ICN-ICN communication in IP “ocean”* scenario. On the contrary, the *hybrid* approach allows it to be deployed in the *Border Island* scenario as well.

**Addressed Coexistence Requirements.** CONET guarantees the following three coexistence requirements.

- Forwarding and Management - these are guaranteed by *BNs* and *NSNs*. In addition, *ENs* provide transport-level functionalities such as reliability and flow control. Since the logic for requesting a content involves sending separate interests, containing a small part of the named-data, the control of interest sending rate can be used as a TCP-like flow control mechanism.
- Storage - *BNs* are able to store contents.

**Additional architecture or Technology Used.** Besides the standard TCP/UDP/IP tunneling, which is part of the *overlay* approach, the CONET project does not rely on additional architectures.

**Evaluation Parameters.** The *hybrid* deployment solution is hard to be introduced since it requires a new IP option. However, with respect to the *clean-slate* approach, the *hybrid* one is less disruptive, and it allows the architecture deployment in different scenarios.

#### F. GreenICN

The SDN technology decouples control plane from data plane, and it provides a programmable, centrally managed network control that improves network performance and monitoring. SDN-based implementations of ICN exploit the centralized view available to SDN controller, which enables the SDN controller to install appropriate forwarding rules for ICN requests/responses in such a manner that the network elements only have to support IP forwarding. Vahlenkamp et al. in [79] proposed an implementation of ICN using SDN under their

GreenICN project. The proposal leverages ICN protocol’s Message IDs and features of SDN instantiations such as OpenFlow to rewrite packet header information. Fig. 10 presents a simplified view of this solution. Here, both the *Content requester* and the *Content source* are connected to *OpenFlow-enabled switches* that are managed by the *SDN controller*. Routing information for the content requests and responses, upon arriving on OpenFlow switches, is handled/rewritten by the instructions from the controller.

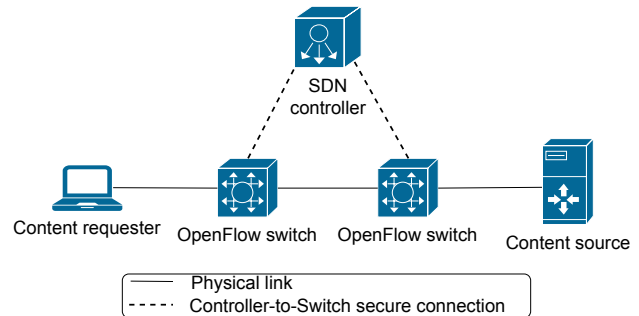


Fig. 10: Simplified view of the GreenICN architecture.

**Deployment Approach.** The proposed solution is an *overlay* ICN implementation as ICN data is sent over the SDN-managed IP packets.

**Deployment Scenarios.** Essentially, the authors in [79] propose ICN deployment over IP network, where an ICN-aware content source delivers the content to an ICN-aware requester over IP network. Hence, this solution supports both the *ICN-ICN communication in IP “ocean”* and the *ICN-IP communication in IP “ocean”* scenarios.

**Addressed Coexistence Requirements.** The architecture addresses the following coexistence requirements:

- Forwarding - network programmability offered by SDN enables forwarding and routing for ICN.
- Management - SDN centrally managed network control supports load-balancing, traffic engineering, and explicit path steering (e.g., through ICN caches).

**Additional architecture or Technology Used.** The authors argue that an ideal or native deployment of ICN, in which user devices, content sources, and intermediary network elements are ICN aware, may not be viable. Hence, the authors proposed to implement ICN-awareness in the SDN-enabled switches, where ICN packets are carried over the IP transport protocol. By using SDN, the authors target all the services/applications of the TCP/IP protocol stack.

**Evaluation Parameters.** In the proposed ICN implementation, SDN controller must manage every ICN request and rewrite several headers fields for every response packet, which might not scale with increased network size. Given that this solution is based on the widely accepted SDN technology - that supports agile deployment and rapid alternation in networking - the hardware modifications required for its deployment are low in those scenarios where SDN infrastructure already exists. Consequently, the time required for its deployment is also low. Nevertheless, the time and the hardware modifications required for its deployment would be higher if the SDN infrastructure does not already exist.

### G. coCONET

Similar to the work [79], Veltri et al. [80] proposed a CONET [78] inspired SDN-based implementation of ICN, called coCONET. Fig. 11 presents a simplified view of this solution. In this architecture, ICN nodes and user-terminals form the data plane and *Name Resolution Service (NRS)* nodes are placed in the control plane. Moreover, *ICN node* works as an OpenFlow switch, while *NRS node* works as an OpenFlow controller. To this end, the authors proposed to extend the OpenFlow protocol [60].

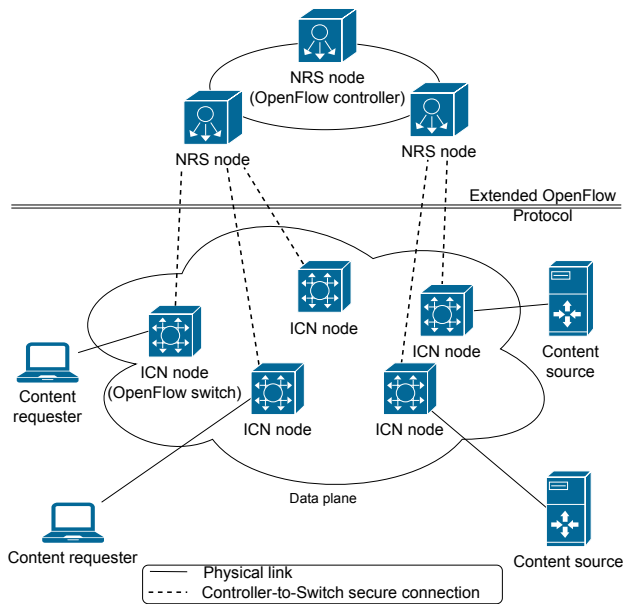


Fig. 11: Simplified view of the coCONET architecture.

**Deployment Approach.** Similar to the work [79], the proposed solution is an *overlay* ICN implementation as ICN data is encapsulated inside the SDN-based IP packets.

**Deployment Scenarios.** The proposed solution enables the *ICN-ICN communication in IP “ocean”* and the *ICN-IP communication in IP “ocean”* scenarios, where the underlying IP network is managed by OpenFlow-based SDN network.

**Addressed Coexistence Requirements.** The present architecture provides the following coexistence requirements:

- Forwarding and Management - SDN-based operations of the proposed approach support both forwarding and management of ICN traffic.
- Storage - ICN capable nodes cache the contents.
- Security - contents are cryptographically protected in order to assure content (and content generator) authentication and data integrity. This security service is provided through digital signature and can be verified through the public key associated to the private key of the content (or of the content generator). The proposed system enforces every ICN node to verify such signature before forwarding the content toward the interested end-nodes, to protect the network against Denial of Service (DoS) or other attacks.

**Additional architecture or Technology Used.** Here, the authors focus specifically on OpenFlow-based SDN implemen-

tations and target all the services/applications of the TCP/IP protocol stack. OpenFlow is a flavor of SDN.

**Evaluation Parameters.** The proposed solution requires ICN capable OpenFlow network devices for ICN operations. Due to such specific requirements, the hardware modifications and the time required for its deployment are high.

### H. DOCTOR

Deployment and securisation of new functionalities in virtualized networking environments (DOCTOR) [81] is an ongoing project funded by French Nation Research Agency. The project provides support towards the adoption of new standards by developing a secure use of virtualized network equipment. This leads to ease the deployment of novel networking architectures, thus enabling the coexistence of IP and emerging stacks, such as NDN, as well as the progressive migration of traffic from one stack to the other. DOCTOR proposes the use of NFV infrastructure to achieve the incremental deployment of NDN at a low cost. The project proposes an HTTP/NDN gateway to interconnect ICN “islands” to the IP world, and an experimental architecture able to process the web traffic passing through a virtualized NDN network.

In particular, DOCTOR first deploys a virtual network based on OpenvSwitch to provide an end-to-end network connectivity between the virtualized network services and to enable a software control of the networking infrastructure. Then, it selects NDN as an ICN protocol stack. More specifically, the NDNx software is *dockerized* to become a Virtualized Network Function (VNF), deployable in DOCTOR architecture. In DOCTOR, NDN is used both over IP and over Ethernet since most NFV tools are still IP-dependent. To test the functionality of the coexistence, the web is considered as an application layer service due to its high popularity and predominance in the global network shares. However, since the current web clients and servers do not yet implement NDN, dedicated gateways are used to perform an HTTP/NDN conversion. Since these gateways are conceived as VNFs, they can be deployed where and when required. In particular, two types of gateways are defined: (1) an ingress GateWay (iGW), aimed at converting HTTP requests into NDN Interest messages and NDN Data messages into HTTP replies; (2) an egress GateWay (eGW), aimed at converting NDN messages into HTTP requests, if the content is not available in the ICN network, and HTTP replies into NDN Data messages. Fig. 12 shows the high level architecture of a virtualized node in DOCTOR. The virtualized node is implemented on a single Linux server and it provides the required hardware resources for the VNFs, which can act as various components (e.g., NDN stack, IP stack, and HTTP/NDN gateway).

**Deployment Approach.** DOCTOR uses an *underlay* approach with the help of HTTP/NDN gateways, that can map the HTTP protocol with NDN messages and properly deliver the web content.

**Deployment Scenarios.** The iGW and eGW allow DOCTOR to support all the different deployment scenarios.

**Addressed Coexistence Requirements.** The DOCTOR architecture addresses the following coexistence requirements:

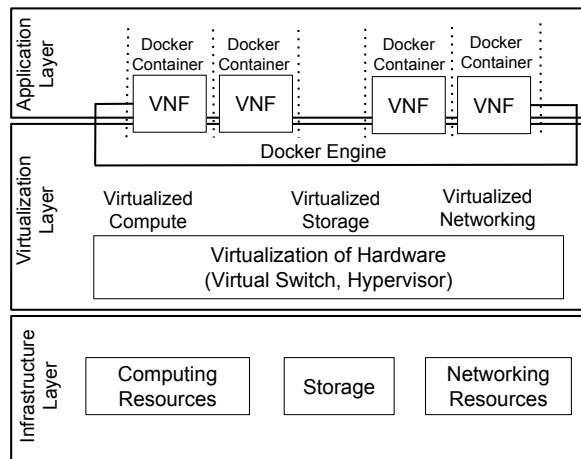


Fig. 12: Internal architecture of a DOCTOR virtualized node.

- Forwarding - explicit name based routing of NDN is performed at each router through the use of virtualized NDN stack.
- Storage - content stores perform the content caching.
- Security - DOCTOR supports the same content oriented security as NDN.
- Management - the control and management plane of VNFs in DOCTOR has been designed with respect to the recommendations of the ETSI NFV group, concerning the NFV MANagement and Orchestration (MANO) [104].

**Additional architecture or Technology Used.** The architecture of DOCTOR is flexible, as it is based on NFV and SDN principles. Its main component is the NFV infrastructure, which enables the resource virtualization to deploy the ICN protocol stack over the data plane and the MANO aspects over the control plane. As a computing virtualization framework, the architecture uses Docker, which relies on a lightweight virtualization principle

**Evaluation Parameters.** Among the key limitations of DOCTOR there is the latency, which occurs due to the repeated sending of requests to the ICN servers, acting as gateways and attached to the content source. Since content names are different among each other, each new content name represents a new routing identifier to be given to the gateways. This results in a continuous interaction between content publisher and gateways for each HTTP request.

### I. POINT

The H2020 project iP Over IcN- the betTer IP (POINT) [45] started in January 2015 and ended in December 2017. Its main purpose is to evaluate both quantitatively and qualitatively the improvements introduced by running ICN over an IP network. To achieve this aim, POINT designs an evolution of the PURSUIT architecture, which both leverages on the SDN technology and on additional network components that enable IP-based applications to run in the new setup without any modification. Those new elements are the Network Attachment Point (NAP) and the ICN Border GateWay (ICN BGW). The former directly interacts with the end user devices and is

responsible for the translation of all the IP protocol abstraction layers (e.g., HTTP, TCP and IP) into the ICN paradigm, while the latter controls the communication between ICN and IP networks. Furthermore, the NAP provides standard gateway functions such as NAT, firewall, and dynamic IP address assignment. The core ICN functionalities are provided by the PURSUIT components (i.e., TM, FN, and RP). Usually, content items are assigned a Routing Identifier (RID) and are stored on the publisher, which advertises the contents availability in the network. Then, a user device sends a request for a content item and the NAP transforms the interest into a subscription for a specific RID. The subscription is then sent to the RP, which triggers the TM towards the identification of a path between publisher and subscriber. The TM identifies all the nodes that need to be traversed and it calculates the associated FIs, which are placed in the packet header. At this point, the SDN switches are responsible for forwarding the packets by using only the FIs and not the routing tables. The SDN switches are not aware of the POINT architecture and are, instead, coordinated by an SDN controller, which communicates directly with the TM. This communication is bidirectional since the SDN controller informs the TM about any topology modification, and the TM notifies the SDN controller about the configuration to be placed on the SDN switches.

Fig. 13 shows the internal architecture of a POINT node. In the upper layer of the node, there are generic applications (i.e., *App1*, *App2*, *App3*, *App4*) which interact with a set of abstractions provided by POINT (i.e., *IP Abstraction*, *TCP Abstraction*, *HTTP Abstraction*, *CoAP Abstraction*). Those are aimed at enabling the communication between applications and ICN networks without requiring any modification from the application interface side. Each abstraction, then, cooperates with the *Pub/Sub (Information-centric) Service Abstraction* to adhere to a publish/subscribe paradigm, where information is delivered according to specific strategies (i.e., *LIPSIN*, *MSBF*, *POINT Alternative3*). Finally, POINT exploits also the SDN technology by introducing two new layers (i.e., *ICN-over-SDN shim layer* and *SDN*) just above the *L2 Transport Network layer*.

**Deployment Approach.** The POINT project falls under the *underlay* deployment approach due to the gateway components, which are responsible for the translation from the IP semantics into the ICN semantics.

**Deployment Scenarios.** The main purpose of the POINT architecture is to enable different subnetworks to communicate between each other. Thus, POINT supports the *Border Island* scenario.

**Addressed Coexistence Requirements.** Given that POINT is an evolution of PURSUIT, they both share the same coexistence requirements, i.e., forwarding, storage, and security.

**Additional architecture or Technology Used.** The POINT solution relies on both the PURSUIT architecture and the SDN technology.

**Evaluation Parameters.** The challenges introduced by the POINT project involve scalability, dynamic network management and latency of data transmission. The first two challenges refer to the appropriate configuration of SDN switches to face

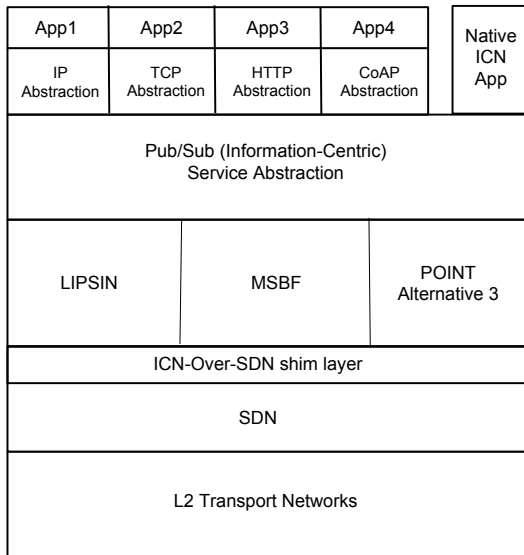


Fig. 13: Internal architecture of a POINT node.

an automatic update of the network topology (e.g., a new host being attached). On the contrary, the third challenge might be due to the high frequency of interaction between NAPs and RPs.

#### J. RIFE

The architecture for an Internet For Everybody (RIFE) [82] architecture is a Horizon2020 funded project, which started in February 2015 and ended in January 2018. Its aim is to develop a new network infrastructure that brings connectivity to communities living in remote locations or unable to afford the communication network costs. To achieve the purpose, the RIFE project focuses on three different challenges regarding the current end-to-end communication paradigm: reduction of capacity, energy, and redundant contents available in the network. The first can be achieved through a time-shifted access to network services and applications. The energy consumed by connected devices can be reduced by introducing a tolerance delay in the communication, so that devices can stay in an idle mode during the absence of network activity. Finally, the third aim is achievable by serving the same content to all the clients that require it, instead of releasing each time a new copy. The architecture addressing those objectives is a combination of IP, ICN, and DTN paradigms.

**Deployment Approach.** The RIFE architecture follows the *underlay* approach because of the gateway components, which are responsible for the translation from the IP semantics into the ICN semantics.

**Deployment Scenarios.** RIFE supports the *Border Island* scenario.

**Addressed Coexistence Requirements.** RIFE is an evolution of the PURSUIT architecture. Thus, the coexistence requirements addressed are the same, i.e. forwarding, storage, and security.

**Additional architecture or Technology Used.** The architecture proposed in the RIFE project is a modification of

the PURSUIT architecture and it relies on the coexistence of IP, ICN and DTN. This last architecture is responsible for introducing the delay and disruption tolerance required to enable the time-shift requirement.

**Evaluation Parameters.** No challenges have been found for the RIFE project.

#### K. CableLabs

Among the different *underlay* approaches, there is a solution designed by CableLabs, which is a non-profit Innovation and R&D lab focused on the introduction of fast and secure release of data, video, voice, and services to end users. CableLas proposes an incremental introduction of CCN/NDN in the existing CDNs to improve the overall content distribution without modifying IP routers [83]. The architecture designed by CableLabs requires first a migration of some services/applications to the ICN paradigm, and then the introduction of proxies. Those are able to manage the translation between HTTP and CCN. Once several ICN “islands” are deployed in the network, the communication among them is provided through IP tunneling.

**Deployment Approach.** The solution proposed by CableLabs adopts the *underlay* approach because of the gateway components, which are responsible for the translation from the IP semantics into the ICN semantics.

**Deployment Scenarios.** Except for the *Border Island*, the CableLabs architecture supports all the deployment scenarios.

**Addressed Coexistence Requirements.** The CableLabs architecture addresses the following coexistence requirements:

- Forwarding - the additional proxies introduced in the network to support the translations i.e., HTTP to CCN and CCN to HTTP, also work as CCN forwarder.
- Storage - as the architecture is an evolution of a CDN, by design the network nodes can cache contents.

**Additional architecture or Technology Used.** Throughout this project, CableLabs investigates how the CCN infrastructure is better in supporting a content-oriented network with respect to the current solutions, such as CDNs. Thus, CableLabs illustrates an incremental deployment of a CCN network over a CDN existing one.

**Evaluation Parameters.** The challenges identified by CableLabs with respect to their own architecture are as follows: traffic management, optimization of CCN router implementation (e.g., FIB/PIT sizing and memory bandwidth), optimization of CCN cache implementation, content object size and fragmentation (i.e., definition of the maximum content object size transmissible inside a network), CCN to HTTP and HTTP to CCN conversions (e.g, the computational complexity of the translation function).

#### L. NDN-LAN

The authors in [84] propose a *hybrid* ICN architecture in which content names are mapped to the MAC addresses. In particular, the authors present the design of a Dual-Stack switch (D-switch), which provides name-based forwarding for NDN traffic and address-based forwarding for conventional traffic such as IP. It can be seen from Fig. 14 that the key



component of D-switch architecture is the *Dispatcher*, which checks the *EtherType* field in the header of a received frame. When an IP frame is detected, the D-switch works like a traditional Ethernet switch and it forwards the frame using the MAC address. If an NDN frame (i.e., Interest or Data packet) is detected, the D-switch processes/forwards the frame based on the content name carried in the NDN header (i.e., Layer 3). In particular, the dispatcher either selects the *Process IP Traffic* or *Process NDN Traffic* module in the D-switch based on the value of *EtherType* field. In the *Process NDN Traffic* module, the PIT and FIB tables are modified to store the mapping between the content names and MAC addresses. For instance, when an Interest packet is received, the D-switch will forward it by searching the content name and its corresponding MAC in the FIB, and then fill the destination MAC address field in Ethernet header with the recorded MAC address.

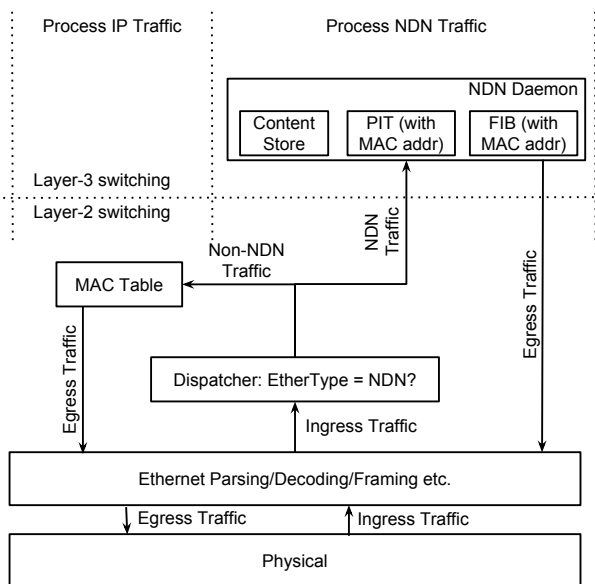


Fig. 14: Dual-stack switch internal architecture.

**Deployment Approach.** This coexistence approach falls under the *hybrid* approach because the D-switches are able to process both types of traffic (i.e., IP and NDN). In particular, a LAN consists (fully or partially) of D-switches that can process the data traffic received from NDN-enabled hosts, as well as IP hosts. However, a fully *hybrid* scenario needs to be consistent with D-switches only, else other techniques or polices/rules are required to perform the data forwarding.

**Deployment Scenarios.** Since the D-switches allow NDN traffic to run within the IP network, except for the *Border Island*, NDN-LAN supports all the deployment scenarios. As a matter of fact, due to the use of MAC-layer encapsulation only, the inter-network communications are not possible and the *Border island* scenario cannot be supported.

**Addressed Coexistence Requirements.** The present architecture provides the following coexistence requirements:

- Forwarding - full advantage of ICN features, such as in-network caching and native multicast, is supported when the underlying LAN consists of D-switches only. However, when the LAN has both D-switch and conventional

Ethernet switches, it has to be carefully designed to avoid conflict between name-based forwarding and address-based forwarding.

- Storage - in-network caching is only supported at D-switches, and it is responsibility of the network manager to prevent the conventional Ethernet switches from receiving ICN packets.
- Management - management of such a deployment is challenging due to limitations of topology creation and forwarding rules installation.

**Additional architecture or Technology Used.** NDN-LAN is mainly suitable for NDN applications that run in small and private networks such as university campus and within an organization. However, the proposed coexistence solution aims to support a variety of applications which includes NDN as well as IP applications. This is achieved through the following design goals: (i) coexistence with IP traffic, which ensures that the common mechanisms should run without any change or performance penalty, (ii) native NDN support, by not relying on tunnels or overlays, and (iii) incremental deployment and general applicability. The proposed solution does not make use of any specific technology to implement the D-switch logic. Minor hardware and software changes in the D-switches allow them to process the IP and NDN traffic in a controlled environment (i.e., LAN).

**Evaluation Parameters.** To implement the required logic and functionalities at D-switches so that it can support NDN-enabled traffic processing, some changes are required in the switch hardware, as well as software. Additional forwarding polices need to be installed in scenarios where D-switches coexist with conventional Ethernet switches. Without any standardization of these new software/hardware components, the applicability of the proposed solution in real-world coexistence applications is limited. Designing mechanisms that support the name-based forwarding, meanwhile coexisting with address-based forwarding within the same LAN, is a challenging task. Additionally, the process for D-switches to learn the forwarding table at Layer-2 and build name-based FIB at Layer-3 is an open problem that needs to be addressed. In LAN, the implementation of the proposed solution is simple and straightforward. However, as the LAN size increases and communication between different LANs is needed, the deployment cost will increase significantly, and the current solution needs to be extended to deal with new issues such as interoperability and scalability.

### M. hICN

Authors in [85] propose methods and systems to facilitate the integration of ICN into IP networks. The hybrid ICN (hICN) communication system claims to have the ability to preserve ICN features and advantages, while, at the same time, benefiting from exploiting an existing IP infrastructure. The major components of hICN communication system are as follows: (i) hICN-enabled IP router(s), capable of processing and forwarding both regular IP packets and IP packets enhanced with ICN semantics, (ii) IP router(s), capable of handling IP packets, and (iii) hICN router(s), being provisioned with a

consumer or producer application. The traditional IP packet headers have been modified to add the ICN semantics. As it is shown in Fig. 15, when a router receives an IP packet, then according to the IP header content, it can identify how to process it, i.e., using ICN or IP stack. The authors suggest two possible name mapping schemes for hICN content names to IP: (i) pure IP mapping, in which content name components can be directly encoded in the IP header, and (ii) optimized mapping, in which a subset of the content name component is encoded in the network header, while the remainder is encoded in the transport header.

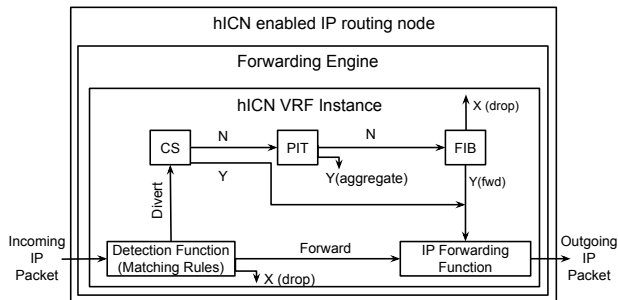


Fig. 15: Internal architecture of an hICN node.

**Deployment Approach.** As the hICN-enabled IP routers are able to process the IP, as well as the ICN traffic, hICN falls under the *hybrid* deployment approach. However, unlike NDN-LAN, in which MAC-to-content name mapping and conversely is performed, in hICN, the IP-to-content name and conversely is done.

**Deployment Scenarios.** Due to the presence of dual stack routers, the proposed architecture supports all the deployment scenarios.

**Addressed Coexistence Requirements.** hICN is among the best proposals supporting the coexistence because it retains most of the ICN basic features (e.g., layer-3 name-based routing, partial symmetric routing, object-based security, anchorless mobility, and in-network reactive caching). This is because hICN exploits the IPv4 and IPv6 header fields content semantic to identify whether the received packet is an IP Data packet or an IP Interest packet. The use of IPv4 or IPv6 RFC compliant packet formats guarantees the communication between an IPv4/IPv6 router and a hICN router. More specifically, the hICN router processes and forwards both the regular IP packets and the ICN-semantic-based packets. Hence, it preserves pure ICN behavior at Layer-3 and above by guaranteeing end-to-end service delivery between data producers and data consumers using ICN communication principles. The present architecture provides the following coexistence requirements:

- Forwarding - the hICN-enabled IP routers as well as IP routers use the same forwarding module.
- Storage - the cache stores are available on hICN-enabled IP routers, and the Interest packets could be satisfied by these routers if the requested content is available in the router cache.
- Management - for large scale usage of this architecture, the consumer and producer applications must have the

mapping of content-names with the corresponding IP addresses, so that the ICN packets can be processed seamlessly by the non-ICN enabled routers as well.

- Security - the architecture provides the same security features that are provided by ICN. However, the IP-only routers are not able to check the received data packets integrity and authentication, hence, at least one hICN-enabled IP router must be available in the route between the consumer and producer.

**Additional architecture or Technology Used.** The hICN proposal uses the IP packet header semantics to differentiate the ICN and IP packets, and the mapping table at hICN-enabled router or DNS is used for performing the mapping task. To support the interoperability among different networks, the edge router could translate the incoming packets to hICN compliant packets using a proxy. Therefore, hICN does not use any specific architecture (e.g., SDN) or technology (e.g., virtualization or tunnelling) to perform the coexistence.

**Evaluation Parameters.** The major challenges of hICN are similar to the other *hybrid* approaches and include a lack of support for heterogeneity, scalability, and standardization of the proposed changes in the traditional Internet protocols and network components. Moreover, the communication delay caused by the additional time used by hICN routers for the mapping could be an issue for delay sensitive applications. The hardware modifications are minimal because the hICN routers can be created by installing a software bundle in the existing IP routers. However, the memory requirements will increase due to the need of storage cache. The deployment effort will be considerable due to the need of the modifications in the consumers and producers applications.

## N. OFELIA

Blefari Melazzi et al. [86] proposed an SDN-based *hybrid* implementation of ICN under the OFELIA project. The proposed approach is an extension of the CONET architecture [78] for OpenFlow networks, where dedicated BNs perform name-to-location resolution, using an external system, for any requested Named Data Object (NDO). Fig. 16 presents a simplified view of this solution. The authors propose to include two different forwarding strategies in an *ICN node*: (1) to forward content requests; and (2) to deliver the data. *Forward-by-name* feature of an *ICN node* applies to Interest packets, while *Data Forwarding* is the mechanism that allows the content to be sent back to the device that issued a content request. *Content routing* is used to disseminate information about location of contents, and *Caching* is the ability of ICN nodes to cache data and to directly reply to incoming content requests. The OFELIA testbed was used in IRATI [8] project for experimental activities.

**Deployment Approach.** The proposed architecture adheres to a *hybrid* approach.

**Deployment Scenarios.** The proposed implementation of ICN is an extension of the CONET framework, in which BNs interconnect different CSSs. Hence, this solution supports the *Border Island* scenario.

**Addressed Coexistence Requirements.** The proposed system is based on CONET framework. Extending the primary

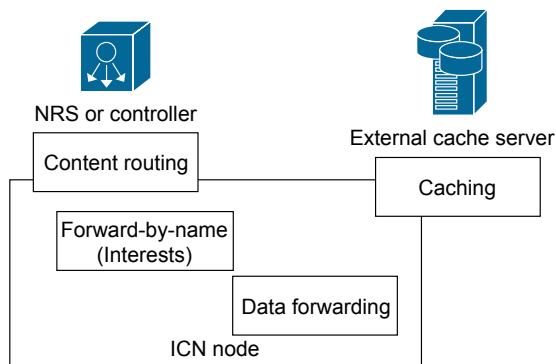


Fig. 16: Simplified view of the solution proposed by OFELIA.

goals of CONET framework, this architecture aims to support forwarding, storage, security and management for ICN deployment.

**Additional architecture or Technology Used.** The present solution strongly relies on the architecture proposed in the CONET project and, through SDN/OpenFlow, it targets all the services/applications of the TCP/IP protocol stack.

**Evaluation Parameters.** The architecture of the solution requires the networking elements to be OpenFlow compliant. Given that OpenFlow (SDN) has been widely adopted in the networking domain, the hardware modifications and the time required for its deployment are low in scenarios where OpenFlow-based network is already present. On another side, the hardware modifications and the time required for its deployment would be higher if OpenFlow-based network is not already present.

## V. DISCUSSION

The purpose of this section is to summarize the findings achieved through our systematic analysis of all the existing coexistence architectures (Section V-A), discuss their deployment in a real-world scenario (Section V-B) and the open challenges (Section V-C), along with some future directions concerning the coexistence between the current and the future Internet architectures (Section V-D).

### A. Summary of the survey

The main aim of this survey is to provide the necessary overview of the available solutions that already address the coexistence. We believe that it will help to move the research community towards the design of the most appropriate architecture for the future Internet. Thus, to guide the reader towards the interpretation of Table I, we add here two new tables, which are a summary of Table I. In particular, among all the features and evaluation parameters considered in this survey, the only ones that can be chosen by a network designer are the deployment approach and the possible additional architecture or technology used in the design of his solution. Thus, Table II and Table III are aimed at comparing each deployment approach and each additional architecture or technology used with respect to all the other features and evaluation parameters,

respectively. As a matter of fact, the deployment scenarios, as well as the addressed coexistence requirements, directly depend on the deployment approach or on the additional architecture or technology, while the evaluation parameters are dynamic properties evaluated during the runtime deployment of an architecture.

The content of the cells as well as their meaning is shared between Table II and Table III. More specifically, the content of each cell corresponds to the number of coexistence architectures addressing both the properties specified in the corresponding row and column (e.g., in the first cell of Table II the value equal to 7 means that there are 7 coexistence architectures adhering to the *overlay* approach and supporting the *forwarding* functionality). The meaning of the values in the cells is different throughout the table. In the upper part (i.e., rows referring to addressed coexistence requirements and deployment scenarios), the value in the cell refers to the number of architectures that guarantee a specific addressed coexistence requirement or a deployment scenario by adopting a deployment approach (listed in the columns). On the contrary, in the lower part of the table (i.e., rows referring to the evaluation parameters), the value in the cells refers to the number of limitations an architecture is affected from.

Table II shows on the columns the three different deployment approaches (i.e., *overlay*, *underlay* and *hybrid*), while on the rows there are all the other features, except for the architectures or technologies used, considered in Table III. Considering the deployment approaches, we found six architectures adopting the *overlay* solution, four the *underlay*, three the *hybrid* and one architecture (i.e., CONET) adhering to both *overlay* and *hybrid*. As it is shown in the table, a plausible reason for this greater adoption of the *overlay* approach might be the higher number of addressed coexistence requirements provided by it. As a matter of fact, almost all the *overlay* architectures guarantee the forwarding and storage features and the number of the architectures supporting security and management is higher than in the *underlay* and *hybrid* cases. While, adopting an *overlay* approach prevents architectures from being deployed in all the deployment scenarios: none of the *overlay* architectures covers either the *ICN-IP communication in ICN "ocean"* or the *IP-IP communication in ICN "ocean"* scenarios. Finally, considering the evaluation parameters, most *overlay* architectures are not able to properly manage the network traffic, but the other limitations are comparable with the ones affecting the *underlay* and *hybrid* solutions. Moreover, even if the number of challenges under the last class (i.e., *Other*) might be significant, we note that those limitations strongly depend on the design of each coexistence architecture.

Table III contains the same rows as Table II, while on the columns it shows all the additional architectures or technologies used in the analyzed coexistence solutions. Throughout this survey, we found the following results: one coexistence solution relying on the PSIRP architecture, two on LAN, one on SAIL, six on SDN, two on PURSUIT, one on CDN, one on DTN, one on CONET, and one on DNS. As it is clearly visible from the table, the reason for adopting the SDN technology in a coexistence scenario is given by its numerous benefits in

TABLE II: Comparison of all the deployment approaches for coexistence architectures - The value of each cell refers to the number of coexistence architectures addressing both the properties specified in the corresponding row and column.

		Deployment Approach		
		Overlay	Underlay	Hybrid
Addressed coexistence requirements	Forwarding	7	4	4
	Storage	6	4	4
	Security	4	3	2
	Management	3	1	3
Deployment scenarios	ICN-ICN communication in IP "ocean"	7	2	3
	ICN-IP communication in IP "ocean"	2	2	2
	ICN-IP communication in ICN "ocean"	0	2	2
	IP-IP communication in ICN "ocean"	0	2	2
	Border Island	2	3	3
Evaluation parameter	Traffic management	4	1	1
	Access control	1	0	0
	Scalability	2	1	2
	Dynamic network management	1	1	1
	Latency	0	2	2
	Other	4	4	2

terms of both features and evaluation parameters with respect to the other possible solutions.

### B. Deployment in a real-world scenario

A clean slate deployment of ICN requires overhauling the entire Internet infrastructure and changing all the host and producer applications. Thus, researchers have realized that it is difficult, as well as infeasible, to replace a greatly successful imperative architecture, such as the IP one, with a clean slate approach, and considered the three deployment configurations (e.g., overlay [75] [76], underlay [81] [45], and hybrid [85]). Moreover, moving from research testbeds to operational networks is very difficult and requires several trials on different large scale testbeds with different number of users.

The first ICN testbed, deployed within the framework of the NDN<sup>4</sup> project, is a shared research testbed that includes software routers, installed in several participating institutions, application host nodes, and other devices. In recent years, a significant number of trials have been conducted to evaluate the CCN/NDN-related software, which are the ones that mostly support the deployment on real networks by providing the set of specifications of the relative architectures (e.g., security, fragmentation, encapsulation, and packet format) [25]. At the same time, since improving the network capacity and minimizing the service latency, even at high network loads, are the key advantages of ICN, many real-world trials addressing the video streaming application scenario have been setup: Cisco and Verizon demonstrated the feasibility and possible advantages of hybrid-ICN in Verizon's labs, applying hICN to live video distribution over a mobile and multi-homed access network; Huawei and China Unicom recently started trials

for the ICN-as-a-Slice configuration, using video conferencing as application scenario to evaluate the security, mobility and bandwidth efficiency of ICN over a wired infrastructure [105]. Both such deployments plan to extend their prototypes to demonstrate the benefits of ICN over a 5G network. Considering the underlay deployment strategy, the Cisco, Internet2 and the U.S. Research and Education community funded the National Research and Education Network (NREN) ICN Testbed project. The purpose of the project is to advance the research in data-intensive science and network, by improving data movement, searchability, and accessibility. The project involves several Universities and US federal Government entities. The testbed has around 15 nodes connected through a nationwide VPN-based layer-2 underlay across the USA, relies on CCN implementation, and uses the Community Information-Centric Networking (CICN) [106] open-source software. In the ICN2020<sup>5</sup> EU project, a testbed has been created in an overlay deployment configuration over the public Internet. The testbed contains 37 nodes and measures the throughput of video applications under certain scenarios. ICN2020 also proposes the use of the GEANT Testbed Service<sup>6</sup> (GTS) to create an independent and isolated global-scale ICN testbed, and to extend the functionalities of the existing ones (e.g. NDN testbed). The above-mentioned deployments in a real-world scenario are the first efforts towards the adoption of ICN in a real network. However, many more evaluations and tests still need to be done.

### C. Open Challenges

According to our findings, the following challenges need to be addressed while designing an efficient and secure coexistence architecture.

- **Traffic management:** the existing Internet applications are not completely compatible with architectures implementing the *overlay* approach [9, 36, 89, 90] due to the issues that these applications introduce on the transport layer. Changing the addressing scheme from host-based to content-based, as well as changing network models from push to pull, are indeed the two obstacles in adapting the existing transport layer protocols to the NDN and CCN architectures. A vast number of existing applications and protocols, such as the HTTP based multimedia streaming protocols, might face false throughput estimations due to the aggressiveness of the underlying TCP in case of content source location variations [91, 92].
- **Latency:** one fundamental issue introduced by the solutions supporting the translation of IP and HTTP-level semantics into ICN [45, 82] is latency. This occurs due to the frequent requests sent to the NAP, that is attached to the source (also referred to as sNAP). Assuming a meaningful interaction between consumer and producer, the URIs are likely different for each content and for each new published content at sNAP, a new RID has to be added to the consumer NAP (cNAP) through the RF.

<sup>4</sup><http://named-data.net/ndn-testbed/>

<sup>5</sup><http://www.icn2020.org/>

<sup>6</sup>[https://www.geant.org/Services/Connectivity\\_and\\_network/GTS](https://www.geant.org/Services/Connectivity_and_network/GTS)

TABLE III: Comparison of all the additional architectures or technologies used in coexistence architectures - The value of each cell refers to the number of coexistence architectures addressing both the properties specified in the corresponding row and column.

		Additional architecture or technology used								
		PSIRP	LAN	SAIL	SDN	PURSUIT	CDN	DTN	CONET	DNS
<b>Addressed coexistence requirements</b>	Forwarding	1	2	1	6	2	1	1	1	1
	Storage	1	2	1	5	2	1	1	1	1
	Security	1	1	0	4	2	0	1	1	1
	Management	0	0	0	4	0	0	0	1	1
<b>Deployment scenarios</b>	ICN-ICN communication in IP "ocean"	1	2	1	4	0	1	0	0	1
	ICN-IP communication in IP "ocean"	0	1	0	3	0	1	0	0	1
	ICN-IP communication in ICN "ocean"	0	1	0	1	0	1	0	0	1
	IP-IP communication in ICN "ocean"	0	1	0	1	0	1	0	0	1
	Border Island	0	0	1	4	2	0	1	1	1
<b>Evaluation parameter</b>	Traffic management	1	2	1	1	0	1	0	0	0
	Access control	0	0	0	0	0	0	0	0	0
	Scalability	0	1	1	3	1	0	0	0	1
	Dynamic network management	0	1	1	2	1	0	0	0	0
	Latency	0	1	0	2	1	0	0	0	1
	Other	0	0	0	3	0	4	0	1	0

Thus, for each HTTP get request, sNAP and RF have to interact, causing an increasing network latency.

- **Topological limitations:** in *underlay* approaches, there might be several publishers for the same content that belong to the same network. In this case, whenever a consumer asks for a content released by different publishers, the RF should identify the best publisher and suggest the best content route. However, in the current architectures, the RF only announces which is the most appropriate publisher, leaving the other ones in a *silent* phase. This might lead to the generation of multi-point forwarding identifiers, which create unnecessarily long routing tables.
- **Routing and scalability:** the number of content objects, and its continuous growing in the current Internet, introduce a limitation in ICN solutions, which have to handle content names of a possibly indefinite length. Thus, the existing networking devices might not support the content-based routing and might have to face special requirements and optimizations.
- **Security issues in coexistence architectures:** below, we illustrate the security risks affecting the coexistence architectures.
  - **Attacks against NAP nodes:** in *underlay* approaches, an attack performed against a NAP node can cause much more damage than one performed against the rendezvous system. This is because a NAP is a node in an ICN network, which can be used by an attacker to launch prefix hijacking, replay attacks and many more attacks against the ICN core network.
  - **DoS attacks:** an external user sending a new IP address causes the introduction of a state into a NAP. The same action can cause the introduction

of states in centralized functions, such as the TF or the RF. Thus, if arbitrary users have a direct access to the centralized TF/RF, as it was the case in pure PURSUIT/PSIRP architectures [75], they could also easily generate a DoS attack.

- **Lack of authorization and access control:** for every new node added to a network, the entire topology needs to be updated to guarantee the proper link among the new and the old network nodes. Thus, an enhanced access control policy is required in ICN networks.
- **Attacks against the SDN controller:** there have been increasing concerns about the security of SDN-based networks. Many of these concerns are related to the fact that SDN controller may parse an arbitrary part of a packet's content, and use this information to set up states in the flow tables (and possibly in the controller). Moreover, systems that parse user generated packet input (e.g. Wireshark packet analyzer and Snort intrusion detection system) have been the frequent cause of security vulnerabilities due to the large permutation of potential cases. Since numerous ICN coexistence solutions propose to use SDN, they are potentially open to the inherent vulnerabilities of an SDN controller. Moreover, considering that an SDN controller is the logically centralized entity that affects the entire network, the risk is even higher.

#### D. Future Research Directions

As confirmed by the large number of coexistence projects (e.g., POINT, DOCTOR, and hICN) that we surveyed in this paper, Governments, Industry, and Academia are pushing towards the definition of a new Internet architecture (i.e.,

ICN) and its coexistence with the current one (i.e., IP). The significant effort put to assess the feasibility and effectiveness of ICN indicates that the ICN paradigm is being considered as a possible replacement for the current IP-based host-centric Internet infrastructure. Hence, we now present few research directions that need to be explored in this research field.

- **Secure transition phase:** from its start, ICN was purposefully designed to have certain inherent security properties such as authentication of delivered content and (optional) encryption of the content. Additionally, relevant advances in the ICN research community have occurred, promising to address each of the identified security gaps [107] [23]. However, due to the lack of real deployments, an array of security features in ICN networks are still under-investigated, including access control [108], security of in-network caches, protection against various network attacks (e.g., DDoS), and consumer privacy [24]. For instance, due to the distributed nature of content availability in ICN, securing the content itself is much more important than securing the infrastructure or the end points. This lack of addressing security goals in the final ICN paradigm is even more critical when considering the coexistence of TCP/IP and ICN, which could lead to the introduction of new attacks and security issues. One of the main limitations of existing projects is that all of them address only the existence of a transition phase without investigating the impact of coexistence on the security and privacy of the system. We believe that not only passing through this intermediate step is unavoidable, but also that it is important to assess the security and privacy vulnerabilities that might come up under the coexistence of both architectures.
- **Selection of an efficient coexistence approach:** in the literature, three main approaches (i.e., *underlay* [109], *overlay* [78], and *hybrid* [85]) have been used to deploy coexistence architectures. The *underlay* approach introduces communication latency due to the required mapping between IP and name addresses, which limits its usability for real-time and delay-sensitive applications. On the contrary, the *underlay* approach maintains an unaltered quality of service under both normal and exceptional conditions, such as failure, server and link congestion, which are common in operator networks. Considering the *overlay* approach, a major drawback is that it requires the definition and standardization of a new packet format, together with protocols that manage the mapping between ICN faces and IP addresses in the ICN routers FIB. Thus, *overlay* poses a significant challenge to network operators and developers. Additionally, upon new deployment, the tunnel configurations in *overlay* needs to be manually changed to include the newly deployed ICN nodes, and these point-to-point tunnels limit the ICN capability in utilizing the underlying broadcast media. Finally, the *hybrid* approach offers an interesting alternative as it allows ICN semantics to be embedded in standard IPv4 and IPv6 packets so that the packets can be routed through either IP routers or hybrid ICN routers. However, the detailed performance results for *hybrid* solutions are still incomplete, which limits its usage in real deployment scenarios.
- **Coexistence solutions that preserve inherent ICN advantages:** due to its inherent features such as in-network caching, interest aggregation, and content oriented security, ICN provides improved communication system and security by design. Therefore, these essential features of ICN should be protected while designing a coexistence architecture.
- **Optimized ICN-IP name-space mapping:** an important issue in the state-of-the-art solutions, that provide translation of IP/HTTP-level services into ICN (or vice versa), is to ensure that the communication latency is comparable with the one in the current network. In most of the coexistence solutions, that use some sort of translation at any networking layer (e.g., transport or network), the main problem is the repeated sending of newly published content information towards the translation server, which generates delay in the response path of requester and congestion in the network. The problem lies in the fact that the URL is likely different for every request (assuming some form of meaningful service interaction between IP client and ICN producer). Additionally, the existing channel semantics cannot be applied directly because the corresponding routing identifier at the ICN level is different for each publication, from the translation server to IP client. Also, realizing the rendezvous function approach, which is responsible for the response of new publications, requires continue interaction between server and content publisher. This causes an additional latency for the client requests, waiting for a fresh mapping of ICN-IP at each published event.
- **Data protection and confidentiality:** ensuring privacy for network entities (e.g., consumer and producer) in coexistence architecture is not a trivial task, mainly due to the poor privacy support provided in ICN [110]. Hence, it is important to investigate how the privacy issues were dealt in the current coexistence architectures. Ideally, names should reveal no more than what is currently revealed by an IP address and port. However, in ICN the name prefix reveals some information about the content, and the in-network caching and data in PIT might expose the consumer identity [111]. Therefore, the researchers should focus on the specific issues concerning the privacy and data protection in the coexistence scenarios. For instance, in a coexistence architecture, IP to name-prefix mapping is performed when an IP packet travels from IP to ICN network. In this scenario, the IP header does not reveal any information about the payload, but the prefix name does, thus, the data confidentiality is threatened when these data packets are traveling through the ICN “island”. In particular, since the use of name prefix for addressing the data in ICN reveals sufficient information to the passive eavesdropper, ensuring privacy means that names and payloads cannot be correlated. However, such privacy requirement would need an upper-layer service similar to the one that would resolve non-topological

identifiers (e.g., ICN name prefix) to topological names (e.g., IP network address).

- **SDN/NFV for efficient coexistence:** as mentioned earlier, the SDN technology separates the control plane from the data plane. The decoupled control plane is programmable and has a global view of the network that provides easier network management monitoring. SDN-based implementations of ICN exploit the centralized view available to the SDN controller, which enables the SDN controller to install appropriate rules in the data-plane to process ICN requests/responses. In the state-of-the-art, both *overlay* and *hybrid* ICN deployments have leveraged SDN to address different coexistence requirements, e.g., forwarding, storage, management, security, and interoperability. SDN has already been successfully adopted for network deployment; it makes SDN an appropriate choice for quick deployment of ICN with low hardware modifications. On the another side, NFV can help to virtualize several network functions that were previously implemented via physical devices.

## VI. CONCLUSION

In this paper, we survey various efforts done by researchers and industries in recent years to propose a design of ICN-IP coexistence architecture. All these architectures differ from each other according to their specific design, but they all adhere to the ICN paradigm, which means a content-oriented communication model in replacement of the current host-centric one. In our survey, we identify that all these architectures have important limitations: none of them has been designed through a comprehensive approach that considers all the new challenges introduced by a coexistence scenario. Instead, the main aim for most of them is to improve the current Internet by exploiting some of the core ICN features (i.e., forwarding, storage, management, and security). Even though security also belongs to that list of features, none of the existing architectures has considered it as the main purpose. In future, we believe appropriate coexistence architecture designs are needed to build a secure path towards the future Internet. This can be done by considering the limitations and necessary improvements of the existing coexistence solutions we have analyzed in this survey. With the set of future research directions and open questions that we have raised, our work will motivate researchers towards designing a complete solution for ICN-IP coexistence while tackling the key security and privacy issues.

## ACKNOWLEDGMENT

The work of M. Conti was supported by a Marie Curie Fellowship funded by the European Commission under the agreement PCIG11-GA-2012-321980. Ankit Gangwal is pursuing his Ph.D. with a fellowship for international students funded by Fondazione Cassa di Risparmio di Padova e Rovigo (CARIPARO). This work was supported by the European Commission under the Horizon 2020 Programme (H2020), as part of the LOCARD project (Grant Agreement no. 832735).

## REFERENCES

- [1] “Internet of Things (IoT) Connected Devices Installed Base Worldwide from 2015 to 2025 (in Billions).” [Online]. Available: <https://tinyurl.com/yyfuky2>
- [2] P. Srisuresh and K. Egevang, “Rfc3022: Traditional ip network address translator (traditional nat),” USA, 2001.
- [3] “Cisco visual networking index: Forecast and methodology: 20162021,” Rep., Sep. 2017, accessed: July 28, 2019. [Online]. Available: <https://tinyurl.com/y2jptcbn>
- [4] “The Zettabyte Era: Trends and Analysis,” Rep., Jun. 2017, accessed: July 28, 2019. [Online]. Available: <https://tinyurl.com/y32f4jwe>
- [5] “What Happens in an Internet Minute in 2016,” Intel, Santa Clara, CA, USA, accessed: August 2, 2019. [Online]. Available: <https://tinyurl.com/y5wrz8p6>
- [6] K. Seo and S. Kent, “RFC 4301: Security Architecture for the Internet Protocol,” BBN Technologies, RFC 4301, December 2005.
- [7] E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.3,” RFC 8446, Aug. 2018. [Online]. Available: <https://rfc-editor.org/rfc/rfc8446.txt>
- [8] E. Grasa, L. Bergesio, M. Tarzan, E. Trouva, B. Gaston, F. Salvestrini, G. Maffione, V. Carozzo, D. Staessens, S. Vrijders, D. Colle, A. Chappell, J. Day, and C. L., *Recursive InterNetwork Architecture (RINA), Investigating RINA as an Alternative to TCP/IP (IRATI)*, ser. Building the Future Internet through FIRE. Rivers Publishers, 2016. [Online]. Available: <https://tinyurl.com/y4kbyt5j>
- [9] M. Mosko, “CCNx 1.0 Protocol Specifications Roadmap.”
- [10] M. Diallo, S. Fdida, V. Sourlas, P. Flegkas, and L. Tassioulas, “Leveraging Caching for Internet-Scale Content-Based Publish/Subscribe Networks,” *IEEE International Conference on Communications (ICC)*, pp. 1–5, 2011.
- [11] Y. Tang, K. Guo, J. Ma, Y. Shen, and T. Chi, “A Smart Caching Mechanism for Mobile Multimedia in information Centric Networking with Edge Computing,” *Future Generation Computer Systems*, vol. 91, pp. 590 – 600, 2019. [Online]. Available: <https://tinyurl.com/y5qnozwp>
- [12] A. Ioannou and S. Weber, “A Survey of Caching Policies and Forwarding Mechanisms in Information-Centric Networking,” *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2847–2886, Fourthquarter 2016.
- [13] A. Seetharam, “On Caching and Routing in Information-Centric Networks,” *IEEE Communications Magazine*, vol. 56, no. 3, pp. 204–209, March 2018.
- [14] X. Fu, D. Kutscher, S. Misra, and R. Li, “Information-Centric Networking Security,” *IEEE Communications Magazine*, vol. 56, no. 11, pp. 60–61, 2018.
- [15] C. Anastasiades, T. Braun, and V. A. Siris, *Information-Centric Networking in Mobile and Opportunistic Networks*. Springer, 2014, pp. 14–30.
- [16] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, and F. R. Yu,

- “A survey of mobile information-centric networking: Research issues and challenges,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 2353–2371, 2018.
- [17] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, “A Survey of Information-Centric Networking Research,” *IEEE Communications Surveys Tutorials*, vol. 16, no. 2, pp. 1024–1049, 2014.
- [18] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, “A survey of information-centric networking,” *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, 2012.
- [19] I. Abdullahi, S. Arif, and S. Hassan, “Survey on caching approaches in information centric networking,” *Journal of Network and Computer Applications*, vol. 56, pp. 48 – 59, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804515001381>
- [20] G. Zhang, Y. Li, and T. Lin, “Caching in information centric networking: A survey,” *Computer Networks*, vol. 57, no. 16, pp. 3128 – 3141, 2013, information Centric Networking. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1389128613002235>
- [21] M. Zhang, H. Luo, and H. Zhang, “A survey of caching mechanisms in information-centric networking,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1473 – 1499, 2015.
- [22] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu, “A survey of naming and routing in information-centric networks,” *IEEE Communications Magazine*, vol. 50, no. 12, pp. 44–53, 2012.
- [23] E. G. AbdAllah, H. S. Hassanein, and M. Zulkernine, “A survey of security attacks in information-centric networking,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1441–1454, thirdquarter 2015.
- [24] R. Tourani, S. Misra, T. Mick, and G. Panwar, “Security, Privacy, and Access Control in Information-Centric Networking: A Survey,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 566–600, 2018.
- [25] M. Tortelli, D. Rossi, G. Boggia, and L. Grieco, “Icn software tools: Survey and cross-comparison,” *Simulation Modelling Practice and Theory*, vol. 63, pp. 23 – 46, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1569190X15300654>
- [26] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, and F. R. Yu, “A Survey of Mobile Information-centric Networking: Research Issues and Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2353–2371, 2018.
- [27] X. Liu, Z. Li, P. Yang, and Y. Dong, “Information-centric mobile ad hoc networks and content routing: A survey,” *Ad Hoc Networks*, vol. 58, pp. 255 – 268, 2017, hybrid Wireless Ad Hoc Networks. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870516301019>
- [28] A. Rahman, D. Trossen, D. Kutscher, and R. Ravindran, “Deployment Considerations for Information-Centric Networking (ICN),” *ICNRG draft*, 2019. [Online]. Available: <https://tinyurl.com/y24d4oly>
- [29] S. Arshad, M. A. Azam, M. H. Rehmani, and J. Loo, “Recent advances in information-centric networking-based internet of things (icn-iot),” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2128–2158, April 2019.
- [30] B. Nour, K. Sharif, F. Li, S. Biswas, H. Mounqila, M. Guizani, and Y. Wang, “A survey of internet of things communication using icn: A use case perspective,” *Computer Communications*, vol. 142–143, pp. 95 – 123, 2019.
- [31] C. Xu, M. Wang, X. Chen, L. Zhong, and L. A. Grieco, “Optimal information centric caching in 5g device-to-device communications,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2114–2126, Sep. 2018.
- [32] I. U. Din, S. Hassan, M. K. Khan, M. Guizani, O. Ghazali, and A. Habbal, “Caching in information-centric networking: Strategies, challenges, and future research directions,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 2, pp. 1443–1474, 2018.
- [33] D. Cheriton and M. Gritter, “TRIAD: A New Next-Generation Internet Architecture,” 2000.
- [34] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, “A Data-oriented (and Beyond) Network Architecture,” in *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, 2007, pp. 181–192.
- [35] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking Named Content,” in *5th ACM International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2009, pp. 1–12.
- [36] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, “Named Data Networking,” *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.
- [37] “Information-Centric Networking Research Group (ICNRG).” [Online]. Available: <https://irtf.org/icnrg>
- [38] J. McQuillan, I. Richer, and E. Rosen, “The New routing algorithm for the ARPANET,” *IEEE transactions on Communications*, vol. 28, no. 5, pp. 711–719, 1980.
- [39] V. Dimitrov and V. Koptchev, “PSIRP Project – Publish-subscribe Internet Routing Paradigm: New Ideas for Future Internet,” in *11th ACM International Conference on Computer Systems and Technologies (CompSysTech)*, 2010, pp. 167–171.
- [40] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl, “Network of Information (NetInf) - An Information-Centric Networking Architecture,” *Elsevier Computer Communication*, vol. 36, no. 7, pp. 721–735, 2013.
- [41] H. Zimmermann, “OSI Reference Model-The ISO Model of Architecture for Open Systems Interconnection,” *IEEE Transactions on Communications*, vol. 28, no. 4, pp. 425–432, 1980.
- [42] R. Braden, “Rfc1122: Requirements for internet hosts -



- communication layers,” USA, 1989.
- [43] C. Safitri, Y. Yamada, S. Baharun, S. Goudarzi, Q. Nguyen, and T. Sato, “An intelligent quality of service architecture for information-centric vehicular networking,” *Internetworking Indonesia Journal*, vol. 10, pp. 15–20, 01 2018.
- [44] F. Drijver, “Assessment of benefits and drawbacks of ICN for IoT applications,” 2018.
- [45] “POINT (iP Over IcN - the betTer IP).” [Online]. Available: <https://www.point-h2020.eu/>
- [46] “White paper: Cisco Visual Networking Index (VNI): Forecast and methodology, 2017–2022,” <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html>.
- [47] S. Lederer, C. Mueller, C. Timmerer, and H. Hellwagner, “Adaptive Multimedia Streaming in Information-Centric Networks,” *IEEE Network*, vol. 28, no. 6, pp. 91–96, 2014.
- [48] S. Lederer, C. Mueller, B. Rainer, C. Timmerer, and H. Hellwagner, “Adaptive streaming over Content Centric Networks in mobile networks using multiple links,” in *2013 IEEE International Conference on Communications Workshops (ICC)*, June 2013, pp. 677–681.
- [49] Y. Liu, J. Geurts, J. C. Point, S. Lederer, B. Rainer, C. Miller, C. Timmerer, and H. Hellwagner, “Dynamic Adaptive Streaming over CCN: A Caching and Overhead Analysis,” in *IEEE International Conference on Communications (ICC)*, 2013, pp. 3629–3633.
- [50] S. Petrangeli, N. Bouten, M. Claeys, and F. D. Turck, “Towards SVC-based Adaptive Streaming in information-Centric networks,” in *2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, June 2015, pp. 1–6.
- [51] B. Rainer, D. Posch, and H. Hellwagner, “Investigating the Performance of Pull-Based Dynamic Adaptive Streaming in NDN,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2130–2140, 2016.
- [52] J. Samain, G. Carofiglio, L. Muscariello, M. Papalini, M. Sardara, M. Tortelli, and D. Rossi, “Dynamic Adaptive Video Streaming: Towards a Systematic Comparison of ICN and TCP/IP,” *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2166–2181, Oct 2017.
- [53] Y. Zhang, A. Afanasyev, J. Burke, and L. Zhang, “A Survey of Mobility Support in Named Data Networking,” in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2016, pp. 83–88.
- [54] J. Ott and D. Kutscher, “Why Seamless? Towards Exploiting WLAN-Based Intermittent Connectivity on the Road,” in *TERENA Networking Conference*, 2004.
- [55] K. Fall, “A Delay-tolerant Network Architecture for Challenged Internets,” in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM ’03. New York, NY, USA: ACM, 2003, pp. 27–34. [Online]. Available: <http://doi.acm.org/10.1145/863955.863960>
- [56] L. Torgerson, S. C. Burleigh, H. Weiss, A. J. Hooke, K. Fall, D. V. G. Cerf, K. Scott, and R. C. Durst, “Delay-Tolerant Networking Architecture,” RFC 4838, Apr. 2007. [Online]. Available: <https://rfc-editor.org/rfc/rfc4838.txt>
- [57] A. Compagno, M. Conti, and M. Hassan, *An ICN-Based Authentication Protocol for a Simplified LTE Architecture*. Cham: Springer International Publishing, 2018.
- [58] H. Farhady, H. Lee, and A. Nakao, “Software Defined Networking: A Survey,” *Elsevier Computer Networks*, vol. 81, pp. 79–95, 2015.
- [59] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software Defined Networking: A Comprehensive Survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.
- [60] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: Enabling Innovation in Campus Networks,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.
- [61] Y. Li and M. Chen, “Software Defined Network Function Virtualization: A Survey,” *IEEE Access*, vol. 3, pp. 2542–2553, 2015.
- [62] A. Vakali and G. Pallis, “Content Delivery Networks: Status and Trends,” *IEEE Internet Computing*, vol. 7, no. 6, pp. 68–74, 2003.
- [63] A. Binder and I. Kotuliak, “Content delivery network interconnect: Practical experience,” in *11th IEEE International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2013, pp. 29–33.
- [64] M. A. Salahuddin, J. Sahoo, R. Glitho, H. Elbiaze, and W. Ajib, “A Survey on Content Placement Algorithms for Cloud-Based Content Delivery Networks,” *IEEE Access*, vol. 6, pp. 91–114, 2018.
- [65] Y. Nicolas, D. Wolff, D. Rossi, and A. Finamore, “I Tube, YouTube, P2PTube: Assessing ISP benefits of peer-assisted caching of YouTube content,” in *IEEE P2P*, 2013, pp. 1–2.
- [66] L. Sun, M. Ma, W. Hu, H. Pang, and Z. Wang, “Beyond 1 Million Nodes - Crowdsourced Video CDN: Architecture, Technology, and Economy,” *IEEE MultiMedia*, pp. 1–1, 2018.
- [67] V. Stocker, G. Smaragdakis, W. Lehr, and S. Bauer, “The growing complexity of content delivery networks: Challenges and implications for the Internet ecosystem,” *Telecommunications Policy*, vol. 41, no. 10, pp. 1003–1016, 2017.
- [68] D. Clark, Lehr, S. Bauer, P. Faratin, R. Sami, and J. Wroclawski, “The Growth of Internet Overlay Networks : Implications for Architecture , Industry Structure and Policy,” in *Proceedings of Telecommunications Policy Research Conference (TPRC-05)*, 2005.
- [69] P. Medagliani, S. Paris, J. Leguay, L. Maggi, C. Xue, and H. Zhou, “Overlay routing for fast video transfers in CDN,” *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 531–536, 2017.

- [70] B. Niven-Jenkins, F. L. Faucheur, and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement," *RFC*, vol. 6707, pp. 1–32, 2012.
- [71] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-Tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges," *IEEE Communications Surveys Tutorials*, vol. 14, no. 2, pp. 607–640, 2012.
- [72] A. V. Vasilakos, Y. Zhang, and T. Spyropoulos, *Delay Tolerant Networks: Protocols and Applications*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2011.
- [73] J. Liu, X. Jiang, H. Nishiyama, and N. Kato, "A general model for store-carry-forward routing schemes with multicast in delay tolerant networks," in *2011 6th International ICST Conference on Communications and Networking in China (CHINACOM)*, Aug 2011, pp. 494–500.
- [74] J. Ren, L. Li, H. Chen, S. Wang, S. Xu, G. Sun, J. Wang, and S. Liu, "On the Deployment of Information-centric Network: Programmability and Virtualization," in *IEEE International Conference on Computing, Networking and Communications (ICNC)*, 2015, pp. 690–694.
- [75] D. Trossen and G. Parisi, "Designing and Realizing an Information-Centric Internet," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 60–67, 2012.
- [76] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos *et al.*, "Named Data Networking (NDN) Project." [Online]. Available: <https://tinyurl.com/y3o5auhw>
- [77] S. Shailendra, B. Panigrahi, H. K. Rath, and A. Simha, "A novel overlay architecture for Information Centric Networking," in *21st National Conference on Communications (NCC)*, 2015, pp. 1–6.
- [78] A. Detti, N. Blefari Melazzi, S. Salsano, and M. Pomposini, "CONET: A Content Centric Inter-networking Architecture," in *ACM SIGCOMM workshop on Information-Centric Networking*, 2011, pp. 50–55.
- [79] M. Vahlenkamp, F. Schneider, D. Kutscher, and J. Seedorf, "Enabling ICN in IP Networks using SDN," in *IEEE International Conference on Network Protocols (ICNP)*, 2013, pp. 1–2.
- [80] L. Veltri, G. Morabito, S. Salsano, N. Blefari Melazzi, and A. Detti, "Supporting Information Centric Functionality in Software Defined Networks," in *IEEE International Conference on Communications (ICC)*, 2012, pp. 6645–6650.
- [81] "Deployment and securisation of new functionalities in virtualized networking environments." [Online]. Available: <http://www.doctor-project.org/>
- [82] "architecture for an Internet For Everybody (RIFE)." [Online]. Available: <https://rife-project.eu/>
- [83] G. White and G. Rutz, "Content Delivery With Content-Centric Networking." [Online]. Available: <https://tinyurl.com/y5328v4s>
- [84] H. Wu, J. Shi, Y. Wang, Y. Wang, G. Zhang, Y. Wang, B. Liu, and B. Zhang, "On Incremental Deployment of Named Data Networking in Local Area Networks," in *ACM/IEEE Architectures for Networking and Communications Systems (ANCS)*, 2017, pp. 82–94.
- [85] L. Muscariello, G. Carofiglio, and J. Augé, "System and Method to Facilitate Integration of Information-centric Networking into Internet Protocol Networks," in *CISCO Technology, Inc.*, 2018. [Online]. Available: <https://tinyurl.com/y46pc6w4>
- [86] N. Blefari Melazzi, A. Detti, G. Mazza, G. Morabito, S. Salsano, and L. Veltri, "An Openflow-based Testbed for Information Centric Networking," in *IEEE Future Network & Mobile Summit (FutureNetw)*, 2012, pp. 1–9.
- [87] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar, and P. Nikander, "LIPSIN: Line Speed Publish/Subscribe Inter-networking," in *SIGCOMM*, 2009.
- [88] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The Click Modular Router," *ACM Transaction on Computer Systems*, vol. 18, no. 3, pp. 263–297, 2000.
- [89] PARC, "CCNx Over UDP," 2015.
- [90] "NDNLP: A Link Protocol for NDN," 2012. [Online]. Available: <https://tinyurl.com/yx8ezmq>
- [91] M. Conti, R. Droms, M. Hassan, and S. Valle, "QoE Degradation Attack in Dynamic Adaptive Streaming Over ICN," in *19th IEEE International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2018, pp. 1–9.
- [92] M. Conti, R. Droms, M. Hassan, and C. Lal, "Fair-RTT-DAS: A robust and efficient dynamic adaptive streaming over ICN," *Computer Communications*, vol. 129, pp. 209 – 225, 2018.
- [93] "Scalable & Adaptive Internet soLutions (SAIL) European Commission's 7th Framework Program." [Online]. Available: <http://www.sail-project.eu/about-sail/index.html>
- [94] "NSF Future Internet Architecture Project." [Online]. Available: <http://www.nets-fia.net/>
- [95] G. Carofiglio, M. Gallo, and L. Muscariello, "ICP: Design and Evaluation of an Interest Control Protocol for Content-Centric Networking," in *IEEE INFOCOM Workshops*, March 2012, pp. 304–309.
- [96] G. Carofiglio, M. Gallo, L. Muscariello, and M. Papalini, "Optimal multipath Congestion Control and Request Forwarding in Information-Centric Networks," in *21st IEEE International Conference on Network Protocols (ICNP)*, 2013, pp. 1–10.
- [97] P. Gusev and J. Burke, "NDN-RTC: Real-Time Videoconferencing over Named Data Networking," in *2nd ACM Conference on Information-Centric Networking*, 2015, pp. 117–126.
- [98] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, "NDNSIM 2: An updated NDN simulator for NS-3," NDN, Technical Report NDN-0028, Revision 2, 2016.
- [99] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev, and L. Zhang, "An Overview of Security Support in Named Data Networking," *IEEE Communications Magazine*, vol. 56, no. 11, pp. 62–68,

- 2018.
- [100] I. Moiseenko and D. Oran, "TCP/ICN: Carrying TCP over Content Centric and Named Data Networks," in *3rd ACM Conference on Information-Centric Networking*, 2016, pp. 112–121.
- [101] A. Afanasyev, I. Moiseenko, and L. Zhang, "NDNSIM: NDN Simulator for NS-3," NDN, Technical Report NDN-0005, 2012. [Online]. Available: <https://tinyurl.com/y2ysqp8v>
- [102] K. Schneider, C. Yi, B. Zhang, and L. Zhang, "A Practical Congestion Control Scheme for Named Data Networking," in *3rd ACM Conference on Information-Centric Networking*, 2016, pp. 21–30.
- [103] A. Suvrat, S. Samar, P. Bighnaraj, R. Hemant, and S. Anantha, "O-icn simulator (oicnsim): An ns-3 based simulator for overlay information centric networking (o-icn)," in *Proceedings of the 1st Workshop on Complex Networked Systems for Smart Infrastructure*, ser. CNetSys 18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1315. [Online]. Available: <https://doi.org/10.1145/3265997.3266000>
- [104] "Network Functions Virtualisation (NFV); Management and Orchestration, ETSI GS NFV-MAN 001 V1.1.1 (2014-12)." [Online]. Available: <https://tinyurl.com/y966jvg4>
- [105] A. Chakraborti, S. O. Amin, A. Azgin, and R. Ravindran, "Design and evaluation of a multi-source multi-destination real-time application on content centric network," in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, Aug 2018, pp. 186–192.
- [106] "CICN, "Community Information-Centric Networking (CICN)"", 2017. [Online]. Available: <https://wiki.fd.io/view/Cicn>
- [107] F. Khan and H. Li, "Ensuring trust and confidentiality for adaptive video streaming in icn," *Journal of Communications and Networks*, vol. PP, no. 99, pp. 1–9, May 2019.
- [108] B. Li, D. Huang, Z. Wang, and Y. Zhu, "Attribute-based access control for icn naming scheme," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 2, pp. 194–206, March 2018.
- [109] G. Xylomenos, Y. Thomas, X. Vasilakos, M. Georgiades, A. Phinikarides, I. Doumanis, S. Porter, D. Trossen, S. Robitzsch, M. J. Reed, M. Al-Naday, G. Petropoulos, K. Katsaros, M. Xezonaki, and J. Riihijarvi, "Ip over icn goes live," in *2018 European Conference on Networks and Communications (EuCNC)*, June 2018, pp. 319–323.
- [110] C. Bernardini, S. Marchal, M. R. Asghar, and B. Crispo, "Privcn: Privacy-preserving content retrieval in information-centric networking," *Computer Networks*, vol. 149, pp. 13 – 28, 2019.
- [111] G. Acs, M. Conti, P. Gasti, C. Ghali, G. Tsudik, and C. A. Wood, "Privacy-aware caching in information-centric networking," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 2, pp. 313–328, March 2019.



**Mauro Conti** is Full Professor at the University of Padua, Italy, and Affiliate Professor at the University of Washington, Seattle, USA. He obtained his Ph.D. from Sapienza University of Rome, Italy, in 2009. After his Ph.D., he was a Postdoc Researcher at Vrije Universiteit Amsterdam, The Netherlands. In 2011 he joined as Assistant Professor the University of Padua, where he became Associate Professor in 2015, and Full Professor in 2018. He has been Visiting Researcher at GMU (2008, 2016), UCLA (2010), UCI (2012, 2013, 2014, 2017), TU Darmstadt (2013), UF (2015), and FIU (2015, 2016, 2018). He has been awarded with a Marie Curie Fellowship (2012) by the European Commission, and with a Fellowship by the German DAAD (2013). His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in the area of security and privacy. In this area, he published more than 250 papers in topmost international peer-reviewed journals and conference. He is Area Editor-in-Chief for IEEE Communications Surveys & Tutorials, and Associate Editor for several journals, including IEEE Communications Surveys & Tutorials, IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, and IEEE Transactions on Network and Service Management. He was Program Chair for TRUST 2015, ICISS 2016, WiSec 2017, and General Chair for SecureComm 2012 and ACM SACMAT 2013. He is Senior Member of the IEEE.



**Ankit Gangwal** received his BTech degree in Information Technology from RTU Kota, India in 2011 and his MTech degree in Computer Engineering from MNIT Jaipur, India in 2016. Currently, he is a Ph.D. student in the Department of Mathematics, University of Padua, Italy with a fellowship for international students funded by Fondazione Cassa di Risparmio di Padova e Rovigo (CARIPARO). His current research interest is in the area of security and privacy of the blockchain technology and novel network architectures.



**Muhammad Hassan** completed the Bachelors in Electrical (Telecommunication) Engineering from COMSATS Institute of Information Technology, Pakistan in 2008 and the Masters in Computer Network Engineering from HALMSTAD University, Sweden in 2013. Currently, he is a PhD student in Brain, Mind and Computer Science at the University of Padua, Italy. He is also a part of the SPRITZ Security and Privacy research group, Padua. His research interests are in the area of securing ICN architecture and related studies such as secure integration of existing technologies in future Internet architecture.



**Chhagan Lal** is currently working as a postdoctoral research fellow at Simula Research Laboratory, Norway. Previously, he was a postdoctoral fellow at Department of Mathematics, University of Padua, Italy, where he was part of the SPRITZ research group. He received his PhD in Computer Science and Engineering from the Malaviya National Institute of Technology, Jaipur, India, in 2014. During his PhD, he has been awarded with the Canadian Commonwealth Scholarship under the Canadian Commonwealth Scholarship Program to work in University of Saskatchewan, Saskatoon, SK, Canada. His current research areas include applications of blockchain technologies, security in software-defined networking, Information-Centric Networks, and Internet of Things networks.



**Eleonora Losiouk** Eleonora Losiouk is a Postdoc Fellow working in the SPRITZ Group of the University of Padova, Italy. In 2018, she obtained her Ph.D. in Bioengineering and Bioinformatics from the University of Pavia, Italy. She has been a Visiting Fellow at the Ecole Polytechnique Federale de Lausanne in 2017. Her main research interests regard the security and privacy evaluation of the Android Operating System and the Information-Centric Networking. During her Ph.D. she published several papers in peer-reviewed journals and IEEE conferences.