논문 20-45-06-06 The Journal of Korean Institute of Communications and Information Sciences '20-06 Vol.45 No.06
https://doi.org/10.7840/kics.2020.45.6.979

Information-Centric Networking Platforms: A Survey

Huhnkuk Lim◆°, Gwangcheon Sin*

ABSTRACT

The existing IP-based network exchanges various types of data based on the data's location in the network. However, general users have no concern about data location and interested in data itself. At the same time, modern development in networking require better data management solutions for scalability, security and mobility. This highlights the need for a new architecture. Information-centric networking (ICN) has been proposed to overcome the existing problems of current IP-based networking. The initial architectural idea of ICN has several approaches that follow the main idea of ICN but use different implementation solutions. In this article, we describe the main ICN platforms and their implementation specifications. We also address the main features, strong/weak points, and applications for ICN platforms. To the best of our knowledge, this article is the first attempt to describe and compare various existing ICN platforms in the form of a single survey. By presenting detailed implementation specifications, features, strong/weak points, and applications for each ICN platform, we provide an overview of the various platforms existing in ICN. A broad understanding of the types of platforms in ICN will be valuable to users and engineers who are interested specifically in utilizing and customizing current ICN platforms.

Key Words : Information-Centric Networking, ICN Platform, Implementation Specifications, Features, Strong/Weak Points, Applications

Ⅰ. Introduction

IP-based architecture has several issues that have created bottlenecks in current network architecture. Currently, the Internet is used to exchange tremendous amounts of data between different points of the globe. The data sources include various types of manually operated and automated devices (sensors, cameras, etc.). Moreover, in the foreseeable future, the number of these data sources will increase swiftly. The way how the modern network is utilized makes physical location of those points irrelevant for the general user. On the other hand in IP-based architecture, a connection between two points is host centric with known endpoint locations. Furthermore, in host-to-host connection establishment of a secured channel is critical to prevent various security threats. However, such connections continue to remain vulnerable to external unauthorized access. In-network trust system also based on the security of this connection and could also be affected by this issue. To support mobile devices, IP-based network is developing by increasing their architectural complexity. Moreover, mobility is still not a well-supported feature of the current Internet. The increasing scale of data transmission through the network, existing potential security threats, increasing number of mobile data users, demonstrate that the existing architecture cannot satisfy the future requirements of the global network.

※ 본 연구는 과학기술정보통신부와 정보통신기획평가원의 SW중심대학사업의 연구결과로 수행되었음 (과제번호: 2019-0-01834)
◆° First and Corresponding Author : Division of Computer and Information Engineering, Hoseo University, hklim@hoseo.edu, 조교수, 정회원
* Information and Security Track, Division of Computer and Information Engineering, Hoseo University, 학생(학부생)
논문번호 : 202003-040-B-RU, Received February 27, 2020; Revised April 10, 2020; Accepted April 17, 2020

To address the issues existing in IP architecture, a new network architecture was introduced. Information-centric networking (ICN) is a future network architecture that places information (data) at the center of interest rather than the actual source of this data (host). Information in ICN is named and requested by names from the network that delivers the request to the host. ICN also supports multicasting and temporary storage in the form of in-network cache. There are also several approaches to solving security and mobility issues. Those issues cannot be addressed within the architectural limitations of IP-based architecture.

In Named Data Networking (NDN), which is one of the representative ICN architectures, the data is retrieved by data names rather than the IP addresses of the machines that host the data[1]. Two data structures are used by NDN routers to facilitate packets forwarding, i.e., Forwarding Information Base (FIB) and Pending Interest Table (PIT). FIB is used to forward Interest packets towards the data. PIT keeps the records of interfaces on which the Interest packets are received and the data names requested by those Interests. PIT ensures that the Data packets follow the reverse path of the Interest packets. A third data structure called Content Store (CS) is used at routers to cache Data packet for satisfying future Interests. Upon receiving an Interest in the router if the requested Data is available in the router cache, they will satisfy the Interest without forwarding it to the next hop based on FIB[1].

The Interest packet is forwarded to the next hop based on FIB entry if the data is not available in router cache. Before forwarding the Interest packets, its incoming interface and data name is stored in the PIT for data forwarding. When a Data packet is received at a router, it will be forwarded to the interface in the PIT entry against this data name. The data will be dropped if there is no entry against this data name in the PIT[1]. The Interest/Data packet forwarding procedure in the NDN router is explained in Figure 1.

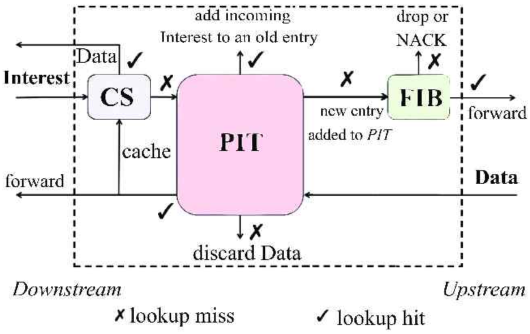

Fig. 1. Interest/Data packet forwarding procedure in NDN.

Currently, ICN architecture has several approaches that sharing the same concept, but are different in their realization and design. Among those projects, we discuss Named-Data Networking (NDN)[1], Publish-Subscribe Internet Technology (PURSUIT)[2], Hybrid Information-Centric Networking (hICN), iP Over IcN- the betTer IP (POINT)[3]. Several architectures like SAIL, PSIRP, DONA are not mentioned in this survey. The communities of those approaches are not active and lack a prototype platform that represents the architecture. On the other side, we have provided a short description of GreenICN and UMOBILE in a separate section. Even though those two platforms are separate approaches in ICN, all of them are targeting to solve a specific problem of networking based on one of the previously mentioned platforms.

This survey aims to present and discuss the implementation specifications, the features, strong/weak points, and applications for the ICN platforms. Details of ICN architectural features have already been studied in various articles, and it has been found that all ICN architectures share various similar aspects[4,5]. ICN platforms have differences and similarities in their implementation structure, together with a set of functionalities that divide or group them in a different manner. By presenting detailed implementation specifications, the features, advantages/disadvantages, and applications for each ICN platform, we provide an overview of the various platforms existing in ICN. This information can be useful for users and engineers who are interested specifically in utilizing and customizing ICN platforms.

The remainder of this article consists of the following sections: In Section II, we provide a short overview of the ICN. In Section III, we discuss the several ICN platforms and describe their detailed implementation specifications. Section IV discusses features, advantages/disadvantages, and applications for each ICN platform. Finally, in Section V, we conclude this article.

Ⅱ. ICN Features and Approaches

2.1 ICN Features

To find information/data in current IP-based networking, a user needs to know the name (location) of the exact host that has this information. However, for current Internet users, location is no longer important, and only the information itself is what matters. In this case, it is more efficient to assign a name to the information (data) rather than the host. Each piece of content (data packet) in ICN architecture is named with relatable and sometimes even human-readable name. This named data can be retrieved from the network using its name and disregarding the location of the information source.

In host-centric architecture, data transmitted through an established path as a flow of bytes that are not recognized by the network as an individual file. In contrast, the primary goal of ICN is to protect the information using a name self-certification function and content hash that is attached to the packet. Features such as name aggregation and in-network caching reduce the chances of a successful Denial of Service (DoS) attacks by reducing the number of requests that reach the data source. Moreover, ICN is a receiver-driven architecture, and the content sender does not have full control over the information that flows between two points.

Mobility becoming an increasingly critical issue for current IP-based networking, due to the growing number of mobile devices and mobile traffic. An end-to-end communication reduces the efficiency of mobile protocols in IP networks. In Mobile IP protocol when a handoff occurs, data must be transferred first to the previous connection point and then to the current point. In contrast, ICN is based on its own publish/subscribe model where a user requests the data from the network that forwards the request to the host. On the host side, the host publishes the data for any request that reaches it. However, data publication in ICN is just an announcement of existing data and not its actual transmission. These two points make it possible to request data again by satisfying the request from a nearby cache rather than going directly to the host.

2.2 ICN Approaches

NDN is an information-centric approach with hierarchical content naming[6,7]. The hierarchical structure of names is highly utilized in its packet forwarding, security, and trust systems. Each packet that arrives to the NDN router is processed by three data structures. By looking into these data structures, it will be decided whether to forward, cache or discard the packet. To request content, a user sends an Interest packet to a content source, which answers back with a Data packet, as shown in Figure 1.

The PURSUIT architecture uses flat naming, where names are called IDs[8]. However, the name consists of a pair of two IDs, one of which may be hierarchical if needed. Moreover, with the use of hierarchically structured rendezvous nodes (RNs), which form a group called a REndezvous NEtwork (RENE), the network can route the packets through it. Multiple RENEs are assigned to different domain levels to ensure communication success. PURSUIT was designed to replace the existing IP-based system with the publish-subscribe approach completely.

h-ICN is the protocol that integrates ICN and IP based on CCNx architecture[9]. At the same time, the usual names of ICN are inserted inside of IPv6 addresses and make possible routing/forwarding in both name-based and location-based over the same IP infrastructure.

In the POINT architecture, the core functionalities are based on the PURSUIT architecture with the target for better integration with IP based network[10]. It creates bidirectional communication between Topology Manager of PURSUIT and SDN switches through the SDN controller[11].

Ⅲ. ICN Platforms

3.1 NDN

The main part of the NDN platform consists of the NDN C++ library with eXperimental eXtensions (ndn-cxx) and the NDN Forwarding Daemon (NFD)[12]. Ndn-cxx contains all the main elements needed to implement different NDN applications including the NFD network forwarder, which is a core component of the NDN platform. NFD consist of several modules that are interconnected with each other. Figure 2 shows the existing NFD modules and their relationships. Some of them are part of other modules, and some are not directly a part of the NDN forwarder. Each module has its own set of key functionalities and responsibilities that are utilized by the platform. We describe these modules in detail below.

First, we describe the ndn-cxx library, which is not a direct module of NFD but is highly connected with its modules. The library is in charge of packet format encoding and decoding as well as data structure management and the global event scheduler. Ndn-cxx also contains NDN's security section (ndnsec), which is used to generate a hierarchical key-chain for a basic trust system and packet signing.

The Core module is a set of common services like basic logging, hash computation routines, and configuration files that support forwarding and management operations. Using the configuration files, it is possible to control various features in NFD. The basic logger provides the ability to access several different logging levels with different levels of information. To support fast name-based operations, the Core module uses different hash functions. It includes versions of the functions for 16-bit, 32-bit, 64-bit, and 128-bit hashing, which are automatically selected based on the platform hash size.

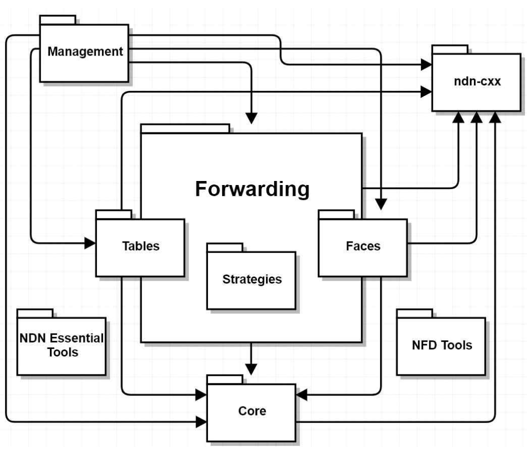

Fig. 2. NFD Platform.

To process the packets on the path, NFD utilizes the Forwarding module, which interacts with three other modules: Face, Tables, and Strategies. Moreover, Strategies is the main part of the Forwarding module, which implements a framework to support various forwarding strategies using forwarding pipelines. A pipeline can be triggered by a specific event on the path, after which it performs a set of steps on the packets or Pending Interest Table (PIT) entries. On the Strategies module side, the strategy returns a decision about whether, when, and where to forward the Interest. It also decides whether this happens before or after the pipelines. Currently, NFD operates several strategies: best route, multicast, client control, NCC (i.e., CCN backwards), access router, and adaptive Smooth Round Trip Time (SRTT) based forwarding (ASF) strategies[13]. The best route strategy is a default strategy in NFD and sends a packet based on the path cost, where lower path costs are preferable.

Through the Face module, the NFD generalizes the existing physical network interfaces and overlays communication and inter-process channels using a single face system. NFD sends Interest, Data, and Nacks through the faces. However, faces compensate for the difference between underlying protocols during forwarding by handling the underlying communication mechanisms. An NFD face consists of two parts: a link service (upper part) that provide a high-quality network-layer packet delivery service and a transport (lower part), which is in charge of the link-layer packet (Type-Length-Value (TLV) packet) delivery service.

The Tables module uses several tables, supplementary lists, and temporary data storage to provide the main data structure mechanism of NFD. Tables are data structures needed for FIB, CS, PIT, and Strategy Choice in order to serve modules in the NFD.

The Management module is a part of the NFD management protocol and provides NFD control and monitoring through the configuration files and Interest-based application programming interface (API). The NFD management is divided into several modules, each of which is in charge of an NFD subsystem.

3.2 CCNx

Content-Centric Networking (CCN) is the approach on which NDN is based, but there are a few differences between them[14,15]. CCN utilizes the CCNx platform, which is implemented in a set of modules and installed with the use of the Distillery application, which pulls the following required modules together:

(1) LongBow: a software framework based on the C language;
(2) Libparc: PARC's C library, which provides a set of features and capabilities for C-based programs;
(3) Libccnx-common: a common library with a set of functions and data structures;
(4) Libccnx-transport-rta: the networking base stack that is for utilization under a forwarder;
(5) Libccnx-portal: an API for Interest and Content Object-based communication;
(6) Metis and Athena: CCNx has two forwarding modules. Methis[20] is a standard forwarder, and Athena is utilized as a framework for the testing and development of the CCNx forwarding agents.

Figure 3 represents Methis forwarder that is used for interconnection of various applications and network by receiving and forwarding the packets from various directions[16]. Its purpose is very similar to NFD in NDN and can be divided into several major modules. The two fundamental modules are IO Module and Message Processor Module. The IO Module consists of Listeners and Connection that are responsible for bringing a proper protocol with certain type of connection (TCP, UDP, Ethernet etc) and attaching a listener to them.

Message module receives the packets and convert

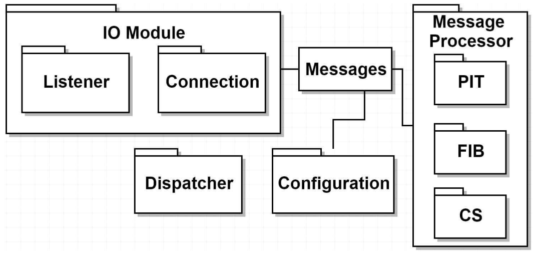

Fig. 3. CCNx Methis Platform

them into the MethisMessage format that can be used by Message Processor Module. The MethisMessage is an extent map of important TLV fields to their location inside a packet and also carries information about the ingress Connection.

After receiving the MethisMessage, Message Processor will direct the Interest and Content Objects to the processing path. Message Processor consists of three tables PIT, FIB and CS which functionality is identical to tables of NFD.

3.3 PURSUIT

To provide ICN capabilities, PURSUIT uses Blackadder as a platform on the nodes[17]. Blackadder that is show in Figure 4, supports various types of Unix-based operating systems like Linux, Mac OS X, and FreeBSD. The majority of Blackadder is created as a set of elements using the Click modular router platform. Click is a software framework that is used to implement packet processing configurations that divide the platform into connected elements. The PURSUIT platform consists of the following elements.

InterProcess Communication (IPC) elements (Netlink, ToNetlink, and FromNetlink): These

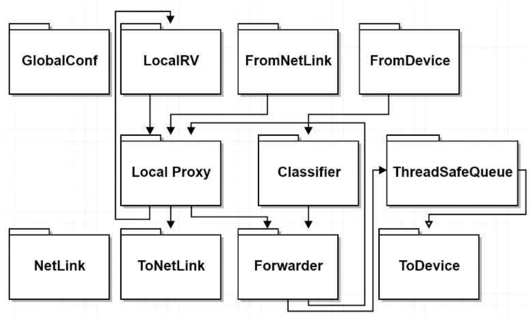

Fig. 4. PURSUIT Platform

elements comprise a socket-based interface for passing messages (corresponding to the functions in the service model) between Blackadder and applications. This interface is implemented as a Netlink socket in Linux and as a Unix domain socket in FreeBSD and Mac OS X.

Local proxy (LocalProxy): This is the core element that dispatches messages (e.g., pub/sub requests as well as published data and other events) between applications and Blackadder elements (depending on the selected dissemination strategies). It also maintains a local state for applications.

Local rendezvous (LocalRV): This element stores information about the publishers and subscribers of scopes and information items and performs interest matching. Each Blackadder node runs the separate local rendezvous instance. Within a domain, one node is responsible for the domain's local rendezvous, which uses the same service model as the applications.

Forwarding element (Forwarder): This element forwards packets (currently based on Line Speed Publish/Subscribe Inter-Networking (LIPSIN) in-packet Bloom filters) between the communication elements as well as to/from the local node (i.e., the local proxy element).

Network interface elements (e.g., ToDevice and FromDevice): These elements are click elements for sending and receiving packets to and from network interfaces or raw IP sockets.

3.4 hICN

Hybrid ICN (hICN) is a project of Cisco which is a unique network architecture based on the Internet protocol that rethinks communications around information itself and not on the location of the data. Cisco has already announced to deploy an open-source version of hICN.

hICN is based on already existing protocol CCNx, and create a protocol stack together with other components like packet forwarder, socket API and supporting libraries[18]. Figure 5 represents all mentioned components of hICN platform.

The Vector Packet Processing (VPP) plugin is implemented as a main forwarder for hICN and enabled the ICN functions. The plugin can be run on various types of hardware that is currently existing and do not create a bottleneck in Network I/O. However, some devices like end devices and low resource equipment unable to support VPP, therefore platform includes a lightweight purely socket based forwarder application.

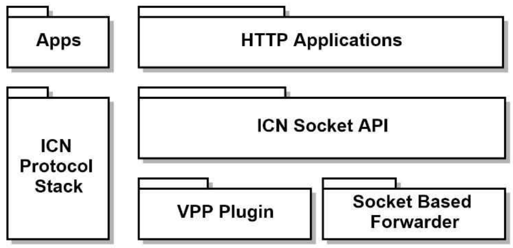

Fig. 5. hICN Platform

The ICN socket API was developed to provide reliable and unreliable stream and datagram ICN socket types. The socket provides functionalities like data segmentation and reassembly as well as naming and signing. On the other side, reliable stream socket is handling the flow, congestion control, loss detection and recovery.

3.5 POINT

POINT, presented in the Figure 6, supports SDN by design, hence the integration with an SDN controller is one of the key project achievements [19,20]. POINT specified a mechanism to support ICN over SDN networks, utilizing arbitrary masks and without disrupting legacy IP/SDN services and

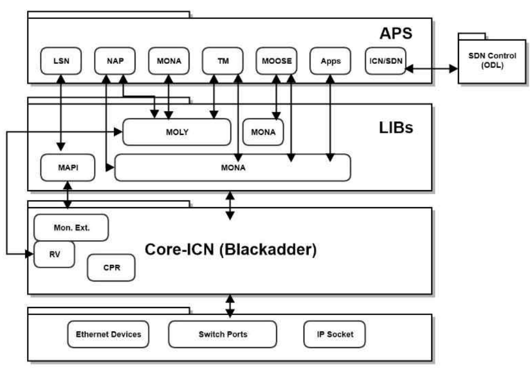

Fig. 6. POINT Platform

protocols. This mechanism has been supported in OpenFlow switches since OpenFlow v1.2. However certain SDN controllers, including OpenDaylight, did not support the required features to configure the SDN switches accordingly. OpenDaylight application was implemented to utilize this feature and provides the required ICN functionality: like interface to the ICN topology; resource management elements; the bootstrapping of ICN topologies over SDN networks; the monitoring of the link status and statistics of ICN nodes; and a user interface for the network administrator to perform ICN functionalities.

3.6 Other Platforms

There are several existing additional ICN platforms that we do not discuss individually, mainly because these platforms utilize already mentioned platforms (e.g., NDN or PURSUIT) and adapt them for the specific requirements and specifications of their approach.

GreenICN is an NDN/CCN based platform that combines a modified name-based platform with DTN-like architecture for disaster response and video-sharing applications[21,22]. The primary objective of GreenICN is to bring ICN techniques to cellular networks because of the high use of mobile devices (e.g., cell phones) in disaster scenarios as the main source of information.

UMOBILE is a mobile-centric approach that implements a combined DTN/ICN common platform [23,24]. Because of the extensive growth of various IoT devices, UMOBILE project aims to bring the features of architectures like ICN and DTN to the mobile world. It utilizes the NFD platform to provide name-based capabilities for mobile devices based on Android and Raspberry Pi platforms.

Ⅳ. Discussions on ICN Platforms

4.1 Features of ICN Platforms

All ICN platforms are implemented based on similar information-centric approach. However, even if the core approach of the implementation are similar, their realizations are notably different, which makes those platforms unique features. However, there are still some drawbacks in the designs of those platforms. Some of the drawbacks are caused by their architectural specifications and they come directly from implementation decisions.

In this subsection, we discuss features and strong/weak points of ICN platforms with respect to their implementation ideas. Table 1 represents the features of ICN platforms, and it shows if they were implemented in each of the platforms or not. Some platforms have not fully implemented certain features and those were marked as partially implemented features.

The NDN platform provides basic functionalities collected inside of the NDN C++ library and forwarding daemon. This makes a node capable of processing and sending Interest and Data, which are the duties of a router on the path to the data source. With these two software elements, it is also possible to maintain the security and trust systems of NDN by participating in a hierarchical key-based trust system. To enable a node to publish, serve, and retrieve data, the operator can install different types of application on top of it. For example, there is the ChronoChat application, which is the NDN based message exchange application that requires ndn-cxx and NFD as well as some additional supplementary NDN applications to be installed before it will work. All nodes in the NDN include a basic platform regardless of their role, which creates simplicity and unification among the nodes. This also simplifies the implementation of new applications, regardless of by whom they were created.

The NDN platform is decentralized and does not implement DNS-like capabilities on platform level compared to the other existing approaches. However, it does not mean that it cannot support such a method at all. There is an existing NDN-based application that mimics the structure of the DNS system[25].

PURSUIT's Blackadder, compared to other existing platforms, is modelled at a native level in Unix based OSs. Hence, when other approaches are implemented as user-level applications, the core elements of the PURSUIT platform are integrated with the OS kernel (especially with the virtual

Table 1. Features of ICN platforms.

ICN Platform	NDN	PURSUIT	hICN	POINT
Common Platform (all nodes share a basic platform)	Yes	No	Yes	No
Centralized or Distributed Operation	Decentralized	Centralized	Decentralized	Centralized
IP Network Integration	Partial (Only Coupled)	Full (Two-Phase Forwarding)	Full (Integrated with IP)	Full (Through SDN)
Supports Non-ICN Applications (applications implemented on top of other platforms or architectures)	No	Yes	Yes	Yes
Self-Certified Names	Yes	No	Yes	No
Hierarchical Names (can support hierarchical naming)	Yes	No	Yes	No
SDN Support	No	No	No	Yes

memory and file system). By this overall performance of content caching and processing are substantially improved.

Blackadder can support both on and off-path caching between host and client. Off-path caching node start to act as a publisher and announce the stored content by advertising it to the one the mediation nodes (PURSUIT RH nodes)[23]. Off-path caching is supported by several ICN platforms while the rest are using on-path caching only.

Even though the core of the POINT platform is based on the Blackadder, in this platform developers merged the well-known ICN platform with SDN based controller which the main target is to provide better control and management of ICN network and provide a better correlation between IP and ICN networks. That was also considered as the main project achievement with an aspect that OpenFlow switches have supported it. However, there are still some switches like OpenDaylight that are not supported.

hICN, the core platform feature, is simultaneous support of both IP address and ICN names through mapping and routing/forwarding based on both types of communication. hICN simplifies mobility, low-latency edge caching and processing in 5G heterogeneous network and service providers environment. hICN builds upon VPP and provides a high-performance host-stack as well as an end-client stack for any desktop and mobile OS (Windows, iOS, macOS, Android, Linux) which makes it available to the customers who intended to develop services using hICN. hICN does not support SDN currently.

4.2 Applications for Each Platform

Within the developing of ICN platforms, research communities also develop different applications based on those platforms. In this subsection, we address existing applications supported by each ICN platform. However, due to the immaturity of platforms or smaller sizes of their communities, some platforms don't have representative applications. That is why we will only cover those that have known applications and will omit one that does not. The simulator-based application won't be covered either.

The NDN platform has several applications for data storing and sharing, including real-time communication applications. Among them, we can highlight ChronoChat, ChronoShare (decentralized file sharing), ndnrtc (video/audio conferencing), repo-ng (NDN based file system) and nTorrent (BitTorrent-like NDN app). Also, several of scientific data management applications exist in NDN, and those are ndn-atmos and kisti-sci. Edge-based augmented reality for NDN exists through ICE-AR app and IoT in NDN represented by NDNFit (NDN Heath app) and NDN-HOME (smart home app) applications. Last but not least, NDN has its simulator (ndnSim) and emulator (mini-ndn) applications.

The CCNx platform has experienced only two major versions and didn't have a variety of applications as one of its ancestors NDN. But as have a couple of applications that were developed within the platform. Those applications are ccnChat (text message exchange app) and ccnFileProxy (for files transmission).

The representative applications for POINT are Edge-ICN which is a Constrained Application Protocol CoAP based IoT application and IPTV application for ICN.

hICN provides better integration with HTTP based application with the use of standard ICN multicasting, caching and mobility. The application that hICN community has highlighted is a Mobile Video Delivery application for 5G networks. hICN offers a variety of services including video streaming, real time communication, edge computing for low latency services and also benefits cloud environment by offering better delivery services than TCP.

Finally, PURSUIT Blackadder platform has legacy applications for video and voice streaming.

V. Conclusion

In this article, we surveyed the platforms of several well-known ICN approaches. We investigated the methods used to implement them. Although ICN platforms share many key architectural functionalities, their design, implementation, and structures were different from each other. We also identified and summarized features, strong/weak points, applications for each ICN platform through the discussions on ICN platforms. Among the existing ICN platforms, some have evolved significantly (e.g., NDN) while others have not yet made strong progress in comparison (e.g., hICN). The NDN platform had an increased variety of NDN-based applications developed on top of the NFD platform, compared to other platforms.

Through the understanding of detailed implementation specifications, features, strong/weak points, and applications for ICN platforms, we expect that this survey can be helpful for users and engineers who are interested specifically in utilizing and customizing current ICN platforms.

References

[1] *NSF Named Data Networking project*, [Online]. Available: http://www.named-data.net/.

[2] *FP7 PURSUIT project*, [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/.

[3] *H2020 POINT project*, [Online]. Available: https://www.point-h2020.eu/.

[4] 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 Commun. Surv. & Tuts.*, vol. 16, no. 2, pp. 1024-1049, 2014.

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

[6] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66-73, 2014.

[7] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. 5th Int. Conf. Emerging Networking Experiments and Technologies*, pp. 1-12, Rome, Italy, Dec. 2009.

[8] S. Tarkoma, M. Ain, and K. Visala, "The publish/subscribe internet routing paradigm (PSIRP): Designing the future internet architecture," in *Future Internet Assembly*, pp. 102-111, Prague, Czech Republic, Apr. 2009.

[9] *H2020 UMOBILE project*, [Online]. Available: http://www.umobile-project.eu/.

[10] G. Xylomenos, Y. Thomas, X. Vasilakos, M. Georgiades, A. Phinikarides, I. Doumanis, S. Porter, D. Trossen, S. Robitzsch, M. J. Reed, and M. Al-Naday, "IP Over ICN Goes Live," in *2018 EuCNC*, pp. 319-323, Ljubljana, Slovenia, Jun. 2018.

[11] M. Vahlenkamp, F. Schneider, D. Kutscher, and J. Seedorf, "Enabling information centric networking in IP networks using SDN," in *2013 IEEE SDN4FNS*, pp. 1-6, Trento, Italy, Nov. 2013.

[12] A. Afanasyev, J. Shi, B. Zhang, L. Zhang, I. Moiseenko, Y. Yu, W. Shang, Y. Huang, J. P. Abraham, S. DiBenedetto, and C. Fan, "*NFD developer's guide*," Dept. Comput. Sci., Univ. California, Los Angeles, CA, USA, Tech. Rep. NDN-0021, 2014.

[13] V. Lehman, A. Gawande, B. Zhang, L. Zhang, R. Aldecoa, D. Krioukov, and L. Wang, "An experimental investigation of hyperbolic routing with a smart forwarding plane in ndn," in *2016 IEEE/ACM 24th IWQoS*, pp. 1-10, Beijing, China, Jun. 2016.

[14] M. Mosko, *Metis CCNx 1.0 Forwarder*, arXiv preprint arXiv:1707.04832, 2017.

[15] *NSF Content Centric Networking project*, [On-line]. Available: https://blogs.parc.com/.

[16] A. Afanasyev, X. Jiang, Y. Yu, J. Tan, Y. Xia, A. Mankin, and L. Zhang, "NDNS: A DNS-like name service for NDN," in *2017 26th IEEE ICCCN*, pp. 1-9, Vancouver, Canada, Jul. 2017.

[17] J. Kjallman and N. Fotiou, *Progress report of component implementations*, PURSUIT Deliverable D3, 2012.

[18] G. Carofiglio, *Mobile video delivery with Hybrid ICN*, Cisco Tech. Rep., 2016, https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/ultra-services-platform/mwc17-hicn-video-wp.pdf

[19] A. N. Mays, A. K. Mohammed, and G. Xylomenos, *Interim Platform Implementation Documentation*, POINT Deliverable D3.3, 2017.

[20] M. AL-Naday, M. AL-Khalidi, M. Reed, *Final Platform Implementation Documentation*, H2020 POINT, 2018.

[21] M. Arumaithurai, A. Detti, T. Hasegawa, J. Seedorf, S. S. Adhatarao, *Final Green ICN Architecture*, Green ICN Deliverable D1.1.3c, 2016.

[22] *FP7 GreenICN project*, [Online]. Available: http://www.greenicn.org/.

[23] O. Ascigil, V. Sourlas, I. Psaras, and G. Pavlou, "Opportunistic off-path content discovery in information-centric networks," in *2016 IEEE Int. Symp. LANMAN*, pp. 1-7, Rome, Italy, Jun. 2016.

[24] P. Mendes, W. Moreira Jr., S. Diamantopoulos, N. Berzirgiadinnis, I. Komnios, I. Psaras, S. Rene, A. Lertsinsrubtavee, A. Sathiaseelan, S. Perez Sanchez, and F. Almeida, *System and Network Deployability Design*, UMOBILE Deliverable D2.4, 2017.

[25] L. Muscariello, G. Carofiglio, J. Auge, and M. Papalini, *Hybrid Information Centric Networking Internet-Draft*, Internet Engineering Task Force, 2018. Avaliable: ietf.org/doc/html/draft-muscariello-intarea-hicn-00(https://datatracker)

임 헌 국 (Huhnkuk Lim)

1999년 2월 : 항공대학교 전자공학과 학사

2001년 2월 : 광주과학기술원 정보통신공학과 석사

2006년 2월 : 광주과학기술원 정보통신공학과 박사

2006년 3월~2020 2월 : 한국과학기술정보연구원 슈퍼컴퓨팅본부 책임연구원

2015년 3월~2020년 2월 : 과학기술연합대학원대학교 HPC 및 데이터과학 전공 교수

2020년 3월~현재 : 호서대학교 컴퓨터정보공학부 조교수

<관심분야> 사물인터넷, 엣지 컴퓨팅, Vehicular NDN, 통신 컴퓨팅 융합, 정보중심네트워킹(ICN), 임베디드 소프트웨어

[ORCID:0000-0002-8032-1597]

신 광 천 (Gwangcheon Sin)
2016년 3월~현재 : 호서대학교 컴퓨터정보공학부 정보보호학 전공
[ORCID:0000-0002-0949-5930]