NNCP: A Named Data Network Control Protocol for IoT Applications

Boubakr Nour*, Kashif Sharif*, Fan Li*, Hakima Khelifi‡, and Hassine Moungla§¶
*School of Computer Science, Beijing Institute of Technology, Beijing, China
‡School of Information and Electronics, Beijing Institute of Technology, Beijing, China
§LIPADE, University of Paris Descartes, Sorbonne Paris Cite, Paris, France
¶UMR 5157, CNRS, Institute Mines Telecom, Telecom SudParis, Nano-Innov CEA Saclay, France
Email: {n.boubakr, kashif, fli, hakima}@bit.edu.cn, hassine.moungla@parisdescartes.fr

***Abstract*—Named Data Networking is a promising architecture that aims to realize Information-Centric Network design. The current NDN design & implementation only details interest and data packets which ensures ubiquitous data dissemination. Currently, there is no support of control messages similar to Internet control messaging protocol of IP networks. The use of content name instead of host addresses, combined with interest-data exchange model, and interest aggregations makes the design of such a protocol a challenging task. In this paper, we present a control protocol for named data networks, namely NNCP, that can relay different network error, information, notification, and service messages. The designed protocol may improve the network performance especially in the Internet of Things environment, and can easily be extended to support different information centric platforms.**

I. Introduction

The design of current Internet model was primarily motivated by the need to allow communication between physical nodes using well-known IP addresses. Each requester should know or be able to discover, the IP address of content provider. However, today's Internet has a large diversity of applications and services e.g., video streaming, social media, on-line games, etc. Moreover, numerous sensor and smart objects have been connected to the Internet, making human surroundings more smarter and independent, which has become the Internet of Things (IoT) [1]. Smart phone, wireless sensors, smart vehicles are an example of today's Internet users. These entities may interconnect with each other, provide services, and take decisions. IoT is applied, and can be extended, to almost any application domain [2], including smart city services, health-care monitoring, smart transportation, and vehicular management & control, etc [3], [4].

Users are not interested in who offers the content or services, rather they are more concerned with content consumption. The Internet infrastructure is witnessing a shift from host-connectivity towards content-oriented communication. Thus, Information-Centric Networking (ICN) [5] has gained traction as a future Internet paradigm that may replace the current host-centric model by using the name of content as the main network element instead of host addresses. This approach decouples the content from its original location, and intermediate nodes may cache the content and serve it for future requests [6]. Instead of securing the communication channel, ICN leverages content-based security [7] by applying all security mechanisms to the content itself. Hence, ICN may enhance content dissemination & access, and improve overall network performance. Due to the nature of IoT applications, and content consumption, ICN is considered as the most suitable future architecture of IoT [8].

Under the concept of ICN, various architectures have been designed such as Content-Centric Networking (CCN) [9], and Named Data Networking (NDN) [10]. Both NDN and CCN use the same name-based concept and are similar to each other in most working details. NDN is an enhanced version of CCN funded by National Science Foundation, that implements an Interest-Data exchange model using a pull-based mechanism. Interest packet is triggered by a consumer requesting a specific content. This Interest is forwarded across the network based on its name. When the Interest reaches a node which has the designated content (either the owner or a replica node), a data packet is sent back to the consumer following the reverse path of Interest.

Information-Centric Networking Research Group (ICNRG), under Internet Research Task Force, has also highlighted ICN as a future Internet architecture. Various drafts and research efforts have been proposed, including ICN challenges [11] and ICN requirements for IoT [12]. Other aspects such as naming [13] and routing [14] have gained more interest by the research community. Despite the simple and clean design of NDN, the network behavior may change during the communication, where errors and other anomalies can occur. Hence, mechanisms to know the network status & health, react to different errors, and notify the IoT network and elements are required.

In traditional IP networks, the management and control protocols have been designed as separate add-ons on top of IP layer, aiming to measure the network performance and troubleshooting. To the best of our knowledge, there is no prior work which addresses a control protocol in ICN/NDN, let alone, specialized for IoT. This article contributes to fill this gap, by presenting a built-in *Control Protocol for Named Data Network*, namely NNCP with emphasis on IoT. To treat real-world issues in NDN deployment and control, we divide the protocol plane into three main classes: *Standard Errors*, *Notification Messages*, and *Service Messages* class. Each class

has its own packet format with different control codes without breaking NDN communication logic.

The rest of the paper is organized as follows: in the next section, we discuss the existing ICN/NDN efforts and the challenges that are present when designing a control protocol for NDN. Section III presents NNCP design principles and packet format. Section IV provides a detailed presentation of different control messages. The implementation and discussion of other ICN implementations are shown in Section V. Finally, we conclude the paper in Section VI along with future works.

II. Background & Challenges of a Control Protocol for NDN

Several publications have appeared in recent years documenting ICN and NDN, and merging it with different technologies as a new Internet architecture. Amadeo et al. [8] discussed the applicability of ICN on top of IoT, by highlighting various challenges and guidelines. Similarly, work in [15] discusses ICN as communication model for vehicular networks, and proposes a framework based on content-centric network. Ahmed et al. [16] focused on software defined-named vehicular networks, and the work aims to combine both NDN and SDN to improve VANET networks. Work in [17] uses ICN as the main network architecture to realize 5G, where authors propose an application-driven framework taking the benefits of network functions virtualization and software defined networking on top of ICN. Most of these works highlight the applicability of NDN to IoT, but are limited to basic communication model. Hence, the need to properly define a control protocol is left out. Below, we list the fundamental challenges, which are present in designing such a protocol.

- Request-Response Model: From NDN communication perspective, data transfer is *always solicited*, where a consumer initiates communication using an Interest packet that carries the name of the requested content, using a receiver-driven hop-by-hop forwarding mechanism. The data packet follows the reverse path from any node which has the data. Moreover, without pending interest table (PIT) entries, the packet flow is not possible. Hence, packet exchange outside of interest-data model is not defined. Unlike IP networks, where IP can identify a node and messages can be encapsulated in an IP-packet can be sent to a layer in stack, NDN does not have any node identification.
- Interest Aggregation: NDN natively supports multi-path Interest forwarding towards multiple destinations using a *stateful forwarding plane*. Each NDN node is considered as a forwarder node. When receiving multiple Interest requesting the same content, Interest Aggregation is performed at the network level, and only one Interest is forwarded upstream. This improves the network performance and reduce the overhead. A copy of the requested data may be forwarded downstream (multicast), taking benefit of the information recorded by Pending Interest Table (PIT) during Interest forwarding. However, the producer or intermediate node have no information as to individual requester. Hence, a traditional error message cannot be sent to a specific node at all.
- Layered Process: In IP-based network each node is identifiable and every protocol is running as a process, and encapsulation can be used to communicate with a single protocol on a specific node. In contrast, NDN system does not conform to this behavior. The only identity is the content name [18]. Hence, it becomes a challenge to implement the same IP concept in NDN framework. The content requesting and data delivery in NDN are done using content name, without any explicit node addresses. Content consumers, producers, and intermediate nodes have no information about who requests the content, who served it (either original producer or cache store), or who forwarded the packet.

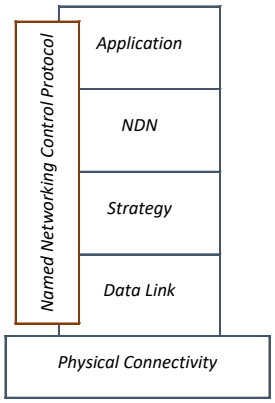


Fig. 1: Vertical protocol implementation in NDN stack.

In essence, traditional Internet control message protocol cannot be directly used in NDN, unless there are major modification to the forwarding plane [19]. It is important to note, that the performance of ICN has been demonstrated to be better than of IP networks [20], due its design. Hence, there is a need to capitalize on this improvement and have a control protocol specially designed for NDN. This forms the prime motivation behind this work. We present a control protocol, aiming to notify concerned nodes/service points with network information and errors, enabling an efficient state transfer between different network entities, without introducing new routing logic or violating NDN communication principles.

III. Named Network Control Protocol: Basics

The main objective of a robust network control protocol is to help realizing a reliable and efficient network. Thus, NNCP inherently uses the NDN forwarding strategy to ensure an efficient control message dissemination between nodes. The protocol uses vertical implementation, as shown in Figure 1, and is able to deliver the message to appropriate layer of the NDN stack. Although there are fundamental differences between NNCP and ICMP, but some of the workings and messages are adopted from it. ICMP messages over the years have proven their need and benefits, which cannot be

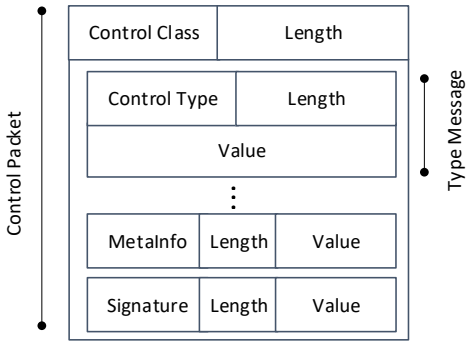


Fig. 2: General packet format.

disregarded altogether. Below, we describe NNCP protocol general packet format and the processing rules.

A. General Packet Format

In addition to the Interest/Data packets of NDN, we define a new packet type named *Control* packet, depicted in Figure 2, which is based on Type-Length-Value (TLV) [21]. Each NNCP message is defined with *ControlClass* and *ControlType*. The *ControlType* is unique and defines the type of message along with the control message itself, while *ControlClass* identifies the class that the control message belongs to. Here, we define three main classes: Standard Error, Notification, and Service messages, with each class having a collection of control type messages. The corresponding numeric values are defined according to NDN project TLV-TYPE number assignment [22]. Under any Control Class, multiple control type TLV messages can be nested inside the packet. On the contrary, a control packet may only belong to a single class.

In most cases, the content of value field in Control-Type TLV is the name of content, which triggered the creation of this message. Additional information can be added in future works.

B. Packet Processing Rules

Fundamentally, an error message should not be generated in reply to another NNCP packet (unless it is a query, which has to be responded). Upon receiving any NNCP packet, the node extracts the *Control Class* of message and then the *Control Type* TLV messages. Each message is processed according to the value it contains by the appropriate layer of NDN stack. The forwarding actions are summarized in Table I.

IV. NNCP: Control Messages

Based on the network usage behavior, we classify NNCP messages mainly into three classes, as listed in Table I. It is important to note that this is not an exhaustive list of messages. Most of these have been adopted from ICMP, and discussed in the light of NDN. More messages can be easily added to the protocol by defining the appropriate Type value and processing rules.

A. Standard Error Messages

This class is used to represent standard NDN errors. They can be represented as shown below, where MetaInfo contains information regarding freshness of this packet, and the signature is that of the creator of packet.

```
Error ::= CLASS TLV-LENGTH
      ERROR-TYPE TLV List
      MetaInfo
      Signature
```

• *Content Unreachable:* This message is generated by content provider or a replica node, in response to the received Interest request for content that can not be reached. Either the content has been removed by the provider or has been moved to another place (due to mobility). The error message will be forwarded towards the consumer(s) using the reverse path created by the received Interest packet. The intermediate nodes remove PIT entries on forwarding this message in reverse direction.

• *Prohibited Access:* This message is generated by content provider or replica node, in response to Interest request for content that has been administratively prohibited using any access-control filter or the node has been configured to reject all the traffic for a specific prefix. The error message will be forwarded toward the requesters using the reverse path created by the received Interest packet. The intermediate nodes remove PIT entries on forwarding it. This error is specific to network level where all interests are treated similarly. For selective interest filtering application level exchange should be used.

• *Invalid Path:* This message is generated by an intermediate node when processing an Interest packet that can not be forwarded as there is no matching entry toward the content name in Forwarding Information Base (FIB) table in routing/forwarding strategy or link failure. The error message is sent only one hop toward the interface where the Interest packet has been received. The error message will be forwarded to the routing module, where a decision of further forwarding is done based on routing strategy.

• *Packet Too Big:* This message has been inherited from ICMP, as the MAC layer for pure NDN has not been defined or standardized yet. In a traditional MAC, size of frame is limited, depending on physical technology. Hence, we use the same mechanisms of ICMP to inform the next hop only, that the data packet desired is too big to be forwarded on this link. The forwarding layer may find another path.

• *Unsolicited Data:* This message is generated by an intermediate node in response to received unsolicited Data packet (no PIT entry matches the name on that packet). The error should be sent back using the same interface from which the unsolicited packet has been received, and for only one hop. Technically, an NDN node should not receive an unsolicited Data packet under normal circumstance. However, this message can be used to support Persistent Interest or Long Live Interest for pull-based communication [19].

• *Parameter Problem:* The Parameter Problem Error Message could be generated by any NDN node, in response to finding a problem in packet parameters. The error message is

TABLE I: List of Messages

Control Class	Class Value	Control Type	Control Value	Processing Rules				Responding Layer/Module
				Generated By	Sending Layer	Towards	Forward	
Standard Error	128	Content Unreachable	131	Provider Replica	NDN	Consumer	Yes	Application: Depends on the application design.
		Prohibited Access	132	Provider Replica	Strategy	Consumer	Yes	
		Invalid Path	133	Intermediate	NDN	Received Face	No	NDN: Depends on the routing module.
		Packet too Big	134	Intermediate	Strategy	Received Face	No	Strategy: Fragment any future packet.
		Unsolicited Data	135	Intermediate	NDN	Received Face	No	NDN: Expunge Persistent PIT entry.
		Parameter Problem	136	Intermediate	NDN	Received Face	No	Strategy: Retransmit if possible.
		Fragmentation Error	137	Intermediate	Link Layer	Received Face	No	Link Layer: Fragments retransmit.
Notification Message	129	Bottleneck Notification	141	Intermediate	Strategy	Neighbors	No	Application: Reduce Interest Sending Rate.
		Explicit PIT Entry Remover	142	Intermediate	NDN	Upstream Nodes	Depends	NDN: Update PIT Table.
Service Message	130	Content Discovery & Meta Extraction	151	Consumer	App.	Provider CDS Server	Yes	Application: Send content name, meta file.
		Link & Network Health	152	Any node	NDN	Any Node	Yes	NDN: Depends on the service point.
		Route Reservation	153	Consumer	NDN	Provider Intermediate	Yes	NDN: Reserve route/path.
		Mobility Handover	154	Intermediate	NDN	Intermediate	No	Strategy: PIT updating routine.
		Device Grouping	155	Provider	NDN	Consumer	No	Application: Group name assignment.

sent only one hop to the neighbor, from which the erroneous packet is received.

• *Fragmentation Error:* Mapping MAC address to NDN faces is not well investigated in the literature, and all packets are broadcasted in the network and treated based on content name. The Fragmentation Error Message is associated to NDNLP protocol [23] that functions between NDN layer and physical/link layer [24], and will be generated if the reassembly timer exceeds. In such case we provide a mechanism to recover the missing fragment rather than dropping the message entirely.

B. Notification Messages

The Notification Messages are generated automatically by any NDN node based on the behavior of the networking process. The generated notification will reach only its neighbors and can not be further routed in the network. We define the associated packet to this class as follows:

```
Notification ::= CLASS TLV-LENGTH
                 NOTIFICATION-TYPE TLV List
                 MetaInfo
                 Signature
```

• *Bottleneck Notification:* The notification message is generated by an intermediate node when a forward link utilization reaches some configurable threshold, and becomes a bottleneck for communication. Information is disseminated only to those neighbors which are using that link. Nodes that receive the notification can appropriately take counter measures, in collaboration with forwarding strategy.

• *Explicit PIT Entry Remover:* An Explicit PIT Entry Remover Notification Message is generated by an intermediate node where a PIT timeout entry has expired. The notification will be sent only one hop toward upstream nodes listed on the PIT face's entry. When a node receives this notification, it will check the PIT to find a match, if there is only one face associated to this entry then NNCP will remove the entry and forward the notification to the list of faces on the entry, otherwise (and because of the interest aggregation), it will remove only the face from the entry and discard the notification packet.

C. Service Messages

The service messages are similar to request/reply packets, and initialized by any NDN node toward another node in order to fetch information, apply policy, or actions. Packets under this class are structured as follows:

```
QueryMsg ::= CLASS TLV-LENGTH
             QUERY TLV
             Nonce
             Signature
RespMsg  ::= CLASS TLV-LENGTH
             RESPONSE TLV
             MetaInfo
             Signature
```

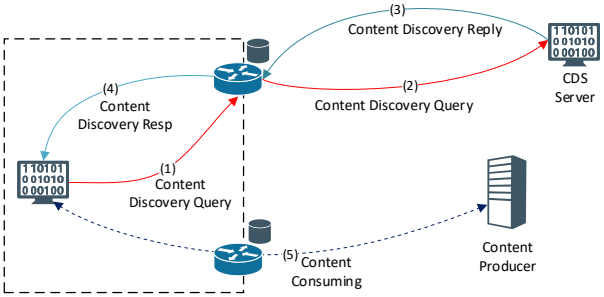


Fig. 3: Content Discovery & Meta Extraction

• *Content Discovery & Meta Extraction:* A consumer can send a Content Discovery & Meta Extraction Service Message asking to discover the existence of content based on keywords. In case of existence, the consumer will receive meta-data file for the requested content. The meta file is a text file with information related to content, fragmentation, number of chunks, etc. As shown in Figure 3, a consumer sends a *Content Discovery & Meta Extraction* toward *Content Discovery Service* (CDS), a DNS-like service [25]. Based on this information, content can be better extracted from the producers. Similar to any content, the meta file can be cached by intermediate nodes, and served for future requests to reduce load on CDS.

• *Link & Network Health:* This message is used to query information about network health. We assume that each node has a service point which has a name and can respond accordingly. For example, it can be used to query information about an NDN component status (PIT, CS or FIB) using the name */NDN/Control/Query/NodeName/PIT* and as a response to this query, the status of the PIT table can be returned.

• *Route Reservation:* This message is used to reserve a route for a publish/subscribe communication system. A node before sending an Interest for periodic data delivery, sends Route Reservation Message. Each node in the path toward the content checks the FIB table to forward the Interest and records the requested name in the PIT. At the same time, it tags this entry with a lifetime (number of data packets or a time limit for communication). The publisher sends Data packet correspondingly, and forwarded based on the reserved path recorded by NNCP.

• *Mobility Handover:* The native NDN design has no explicit handover support. After a node moves and joins a new network, it resends an Interest packets for pending Interest on its PIT. This control message can be used to indicate the node movement, and redirect any pending data. Also, it can be used to register the new node for pending data that may arrive after its mobility (subscription packets). We leave this control as future use in MAC layer mobility.

• *Device Grouping:* In various management cases, network administrators need to group different devices for application and configuration needs. This service message allows nodes and services to join a group under the same name. A unique group-name is assigned to a collection of services/devices. All generated information will be forwarded to this set of nodes similar to a multicast system.

V. Implementation & Co-existence Discussion

The main objectives of NNCP is to enhance the network functionalities. In interest of page limit, we carry experiments only on congestion control to evaluate its performance and efficiency. Furthermore, we discuss the applicability of NNCP on top of other ICN implementations and co-existence with IP-based networks.

A. Implementation & Evaluation

We have implemented the proposed protocol on top of ndnSIM [26]. We use NNCP to send a *Bottleneck Notification* when link utilization reaches some threshold value, in order to avoid the congestion. Therefore, the consumers reduce their *Interest Sending Rate* for an amount of time (t). Using this mechanism, we maximize the link bandwidth, ensure efficient data transmission, and minimize the packet drop rate. A bottleneck topology is used, including four consumers characterized with links capacity of 10Mbps, and 1Mbps for the bottleneck link.

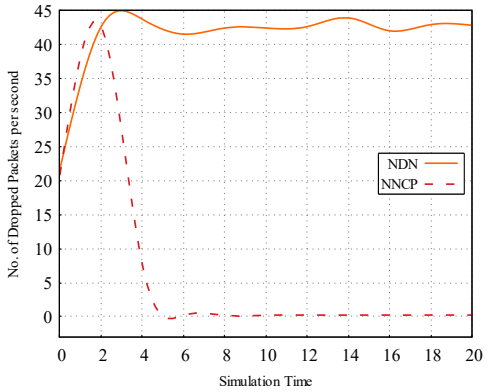
The result of drop packets on bottleneck link is shown in Figure 4a, and Figure 4b shows the satisfied interest ratio. As can be seen, that due to the use of this notification message, the number of drop packet is minimized and becomes approximately null, as compared to normal NDN behavior, where large number of packets are dropped. Similarly, the use of NNCP to avoid congestion maximizes interest satisfaction, by ensuring that maximum requests are satisfied by producer.

B. Discussion on ICN Implementations

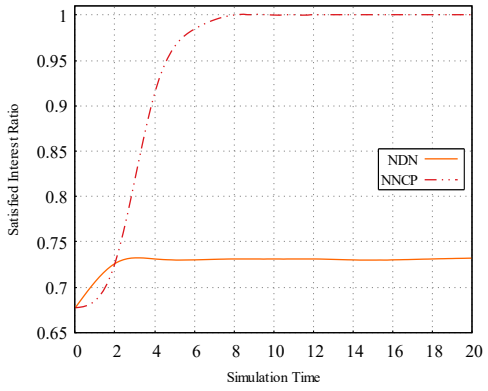
NDN deployment (experimental or real world) has been done in two different ways. 1) Above L3: where NDN runs on top of IP, and 2) Pure NDN: where IP is replaced by NDN layer. In the first case, performance of NDN is affected by the performance of underlaying network. However, this is a fast way to deploy NDN. The second type is underlay, which means that NDN replace IP layer totally, and runs directly on top of Ethernet. In this case, NDN fully exploits the underlaying network and achieves better performance than overlay method. However, it is quite difficult to change the current stack on all the current routers and node.

In a hybrid environment, NNCP can easily be translated to ICMP, as some of the messages are intended for similar purposes. The actual mapping requires a translation mechanism which is left as future work of this paper.

Moreover, NNCP can also be easily extended to run on CCN architecture, with minor modification due to the fact that both NDN & CCN have very similar design principles. Recent Cisco Hybrid ICN (hICN) [27] maps ICN names to IPv6 addresses. NNCP control messages can be easily extended to run with hICN, which means the translation from ICN-based networks to IP networks can also be done using mapping mechanics.



(a) Droped Packets



(b) Satisfied Interests Ratio

Fig. 4: Bottleneck Link Simulation Results

VI. Conclusion & Future Work

In this paper we have presented NNCP, a control protocol for named data networks. NNCP is designed to control the network behavior by disseminating information of different errors and network changes. Moreover, many services are offered by NNCP such as content discovery and meta file extraction that can be used to get a global view about the requested content. Also network health monitoring and reservation of route can be done using NNCP. All these features combined with the adaptive forwarding of NDN can produce a reliable network and efficient data delivery. As future work, real world implementation is planned with a large scale network.

Acknowledgements

The work of Fan Li is partially supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61772077, 61370192 and 61432015. Drs. Sharif & Li are co-corresponding authors.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] D. Bendouda, A. Rachedi, and H. Haffaf, "Programmable architecture based on software defined network for internet of things: connected dominated sets approach," *Future Generation Computer Systems*, vol. 80, pp. 188–197, 2018.

[3] B. Hammi, R. Khatoun, S. Zeadally, A. Fayad, and L. Khoukhi, "Internet of Things (IoT) Technologies for Smart Cities," *IET Networks*, vol. 7, no. 1, 2018.

[4] T. Mekki, I. Jabri, A. Rachedi, and M. ben Jemaa, "Vehicular cloud networks: Challenges, architectures, and future directions," *Vehicular Communications*, vol. 9, pp. 268–280, 2017.

[5] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, 2012.

[6] B. Nour, K. Sharif, F. Li, H. Moungla, A. E. Kamal, and H. Afifi, "NCP: A Near ICN Cache Placement Scheme for IoT-based Traffic Class," in *IEEE GLOBECOM Conference*, 2018.

[7] R. Tourani, S. Misra, T. Mick, and G. Panwar, "Security, privacy, and access control in information-centric networking: A survey," *IEEE Communications Surveys & Tutorials*, 2017.

[8] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos, "Information-centric networking for the internet of things: challenges and opportunities," *IEEE Network*, vol. 30, no. 2, pp. 92–100, 2016.

[9] 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 *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, 2007, pp. 181–192.

[10] 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," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[11] D. Kutscher, S. Eum, K. Pentikousis, I. Psaras, D. Corujo, D. Saucez, T. C. Schmidt, and M. Waehlisch, "Information-Centric Networking (ICN) Research Challenges," RFC 7927, 2016.

[12] K. Pentikousis, B. Ohlman, D. Corujo, G. Boggia, G. Tyson, E. B. Davies, A. Molinaro, and S. Eum, "Information-Centric Networking: Baseline Scenarios," RFC 7476, 2015.

[13] H.-K. Zhang, F. Song, W. Quan, J. Guan, and C. Xu, "Uniform information with a hybrid naming (hn) scheme," Internet Engineering Task Force, Internet-Draft, 2017, work in Progress.

[14] P. Mendes, R. Sofia, V. Tsaoussidis, S. Diamantopoulos, and C.-A. Sarros, "Information-centric Routing for Opportunistic Wireless Networks," Internet Engineering Task Force, Internet-Draft, 2018, work in Progress.

[15] Z. Su, Y. Hui, and Q. Yang, "The next generation vehicular networks: A content-centric framework," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 60–66, 2017.

[16] S. H. Ahmed, S. H. Bouk, D. Kim, D. B. Rawat, and H. Song, "Named data networking for software defined vehicular networks," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 60–66, 2017.

[17] R. Ravindran, A. Chakraborti, S. O. Amin, A. Azgin, and G. Wang, "5G-ICN: delivering ICN services over 5G using network slicing," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 101–107, 2017.

[18] B. Nour, K. Sharif, F. Li, H. Moungla, and Y. Liu, "M2HAV: A Standardized ICN Naming Scheme for Wireless Devices in Internet of Things," in *WASA Conference*. Springer, 2017, pp. 289–301.

[19] B. Nour, K. Sharif, F. Li, and H. Moungla, "A Distributed ICN-based IoT Network Architecture: An Ambient Assisted Living Application Case Study," in *IEEE GLOBECOM Conference*, 2017.

[20] I. Moiseenko and D. Oran, "TCP/ICN: carrying TCP over content centric and named data networks," in *ACM ICN Conference*, 2016, pp. 112–121.

[21] A. Afanasyev, J. Shi, L. Wang, B. Zhang, and L. Zhang, "Packet Fragmentation in NDN: Why NDN Uses Hop-By-Hop Fragmentation," *NDN Technical Report NDN-0032*, 2015.

[22] "NDN TLV-TYPE number assignment," Online: www.named-data.net/doc/NDN-packet-spec/current/types.html.

[23] J. Shi and B. Zhang, "NDNLP : A Link Protocol for NDN," *NDN Technical Report NDN-0006*, pp. 1–15, 2012.

[24] P. Kietzmann, C. Gündoğan, T. C. Schmidt, O. Hahm, and M. Wählisch, "The need for a name to MAC address mapping in NDN: towards quantifying the resource gain," in *ACM ICN Conference*, 2017.

[25] A. Afanasyev, X. Jiang, Y. Yu, J. Tan, Y. Xia, A. Mankin, and L. Zhang, "NDNS: A DNS-Like Name Service for NDN," in *IEEE ICCCN Conference*, 2017, pp. 1–9.

[26] A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," *NDN Technical Report NDN-0005*, pp. 1–7, 2012.

[27] "Mobile Video Delivery with Hybrid ICN: IP-integrated ICN solution for 5G," *Cisco*, 2017.