الجمهوريــة الجزائريــة الديمقراطيــة الشعبيــة

République Algérienne Démocratique et Populaire

وزارة الـتعلـيـم الـعالي و الـبحـث الـعلمـي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

---

Université MUSTAPHA Stambouli
Mascara
Faculté des Sciences Exactes
Département d'Informatique

جامعة مصطفى أسطمبولي
معسكر

THESE de DOCTORAT de 3ème cycle

Spécialité : Informatique

Option : Technologie de l'information et de la communication

Intitulée

Proactive Mobility, Naming and Caching in the context of Future 5G Networks

Présentée par : SERHANE OUSSAMA

Le..31 - 05 - 2022

Devant le jury :

Président	Debbat Fatima	Professeur	Université Mustapha Stambouli, Mascara
Examinateur	Benslimane Sidi Mohammed	Professeur	Ecole supérieure en Informatique de Sidi Bel Abbes
Examinateur	Meddber Meriem	MCA	Université Mustapha Stambouli, Mascara
Encadreur	Yahyaoui Khadidja	MCA	Université Mustapha Stambouli, Mascara
Co-encadreur	Hassine Mongla	Professeur Associé	Université Paris Descarte, France

Année Universitaire : 2021-2022

List of Publications

JOURNALS:

[1] **Ousssama Serhane**, K. Yahyaoui, B. Nour, and H. Moungla, "ICN Content Naming and In-Network Data Caching in Next-Generation 5G and Beyond Networks", *IEEE Internet of Things Journal*, Sep 2020.

[2] **Ousssama Serhane**, K. Yahyaoui, B. Nour, R. Hussain, S. M. Ahsan Kazmi, and H. Moungla, "PbCP: A Profit-based Cache Placement Scheme for Next-Generation IoT-based ICN Networks", *Transactions on Emerging Telecommunications Technologies*, Under Review.

CONFERENCES:

[1] **Ousssama Serhane**, K. Yahyaoui, B. Nour, and H. Moungla, "CnS: A Cache and Split Scheme for 5G-enabled ICN Networks", in *IEEE International Conference on Communications (ICC)*, June 2020, Dublin, Ireland.

[2] **Ousssama Serhane**, K. Yahyaoui, B. Nour, and H. Moungla, "A Label-based Producer Mobility Support in 5G-enabled ICN Networks", in *International Wireless Communications & Mobile Computing Conference (IWCMC)*, June 2020, Limassol, Cyprus.

[3] **Ousssama Serhane**, K. Yahyaoui, B. Nour, and H. Moungla, "Energy-aware Cache Placement Scheme for IoT-based ICN Networks", in *IEEE International Conference on Communications (ICC)*, June 2021, Canada.

Contents

List of Figures

List of Tables

List of Algorithms

Abstract of thesis entitled

Proactive Mobility, Naming, and Caching in the Context of Future 5G Networks

Submitted by **Oussama Serhane**

for the degree of Doctor of Philosophy

at University of Mustapha Stambouli Mascara

The explosive growth of today's connecting devices, and massive traffic demands, increased the load on current Internet infrastructure, promotes rethinking in a novel design of networks paradigm. The 5G and beyond networks ambitions are not restricted to support such momentum, but also it will provide massive data exchange with high reliability. Nevertheless, IP-based communication is still the predominant Internet communication method, it has never been designed to address such a new requirements, and most patches are complex and may fail over time. Information-Centric Networking (ICN), on the other hand, is a new candidate paradigm, which aims at addressing different inherent problems in the current host-centric model. ICN improves network performance by providing ubiquitous caching capabilities at the network layer. Although numerous efforts have been made to integrate ICN with the 5G network, most of them are either violate the ICN native design or do not consider the nature of the 5G architecture as Core/Edge separation. In this thesis, we study and investigate the possibilities to use ICN as a communication enabler for the 5G network. Inspired by the extensive research results in the 5G-based ICN network area, we provide a detailed and systematic review of this merger. This dissertation proposes a series of enhancements for different ICN aspects (i.e., naming, mobility, and In-network caching) to provide seamless and efficient 5G-based ICN networks. Qualitative analyses and extensive experiments prove the efficiency of our schemes that outperform the existing solutions and exhibit better system performance with less overhead.

Keywords: 5G Networks, Content Centric Network (CCN), Internet of Thing (IoT), Information Centric Network (ICN), Complex Network, Named Data Networking (NDN).

مُلَخّــص

أَدَّى النُّمُوّ الْهَائِل لمختلف لأجهزة الِاتِّصَال مَصْحُوبًا بالطلبات الْهَائِلَة عَلَى الْبيَانَات، إِلَى زِيَادَةِ الْحِمْلِ عَلَى الْبِنْيَة الْحَالِيَّة للإنترنت . مِمَّا يَدْفَعُ إِلَى إِعَادَةِ التَّفْكِيرُ فِي تَصْمِيم جَدِيد لنموذج الشبكات مِنْ أَجْلِ مَرَافِقِه الْمُتَغَيِّرَات الْحَالِيَّة وَالْمُسْتَجِدَّة . لَا تَقْتَصِرُ طموحات شَبَكَات الجِيل الْخَامِس وَمَا بَعْدَهَا عَلَى دَعْم مِثْلِ هَذَا التَّغْيِيرِ، وَلَكِنَّهَا ستوفر أيضًا تبادلًا هائلاً للبيانات بموثوقية عَالِيَة . في حين لَا يَزَالُ الِاتِّصَال السَّائِد عَبر الإنترنت مُسْتَنِدًا إِلَى الايبي (IP) بُرُوتُوكُول ، وَلَمْ يَتِمَّ تَصْمِيمِه أبدًا لتلبية مِثْل هَكَذَا متطلبات ، كَمَا أَنَّ مُعْظَمَ التصحيحات مُعَقَّدَة وَقَد تفشل بِمُرُور الْوَقْت . مِنْ جِهَةِ أُخْرَى ، تُعَدّ الشبكات المتمحورة حَوْل الْمَعْلُومَات (ICN) نموذجًا جديدًا مرشحًا ، حَيْث يَهْدِفُ إِلَى مُعَالَجَة الْمَشَاكِل الْمُخْتَلِفَة المتأصلة فِي نَمُوذَج الْحَالِيّ الَّذِي يَرْكُز عَلَى الْمُضِيف . يَعْمَل ICN عَلَى تَحْسِينِ أَدَاء الشَّبَكَة مِنْ خِلَالِ تَوْفِير التخزين الْمُؤَقَّتِ فِي كُلِّ عُقْدَةٍ عَلَى مُسْتَوَى طَبَقَة الشَّبَكَة . عَلَى الرَّغْمِ مِنْ الْجُهُود المبذولة الَّتِي تعْمَلُ عَلَى دَمْج ICN مَع شَبَكَة الجِيل الْخَامِس ، إِلَّا أَن مُعْظَمُهَا امَّا يُنْتَهك الْبِنْيَة الْأَصْلِيَّة لـ ICN أَوْ قَدْ لَا يَأْخُذُ فِي عَيْنِ الِاعْتِبَار هَنْدَسَةٌ الجِيل الْخَامِس كَالْفَصْلِ بَيْنَ الْجَوْهَرِ و حافَة . فِي هَذِهِ الْأُطْرُوحَة ، ندرس إمكانيات اسْتِخْدَام ICN كأداة الِاتِّصَال لشبكة الجِيل الْخَامِس . مِنْ خِلَالِ نَتَائِج بحثية مكثفة فِي مَجَالِ شَبَكَة ICN الْقَائِمَةُ عَلَى الجِيل الْخَامِس ، نُقَدِّم مُرَاجَعَة مُفَصَّلَة ومنهجية لِقَابِلِيَّة هاذا الدَّمج . تقترح هذه الرسالة سلسلة من التحسينات لجوانب ICN المختلفة (مثل التسمية والتنقل والتخزين المؤقت داخل الشبكة) لتوفير شبكات ICN قائمة على الجيل الخامس تتسم بالسلاسة والفعالية . تثبت التحليلات النوعية والتجارب المركزة كفاءة مخططاتنا التي تتفوق في الأداء على الحلول الحالية وتُظهر أداءً أفضل للنظام مع تخفيض في التكاليف .

كلمات مفتاحية : شبكات الجيل الخامس، شبكة متمحورة حول المحتوى، انترنيت الأشياء، الشبكات المتمحورة حول المعلومات، الشبكات المعقدة، شبكة البيانات المسماة.

Abstract

La croissance explosive des dispositifs de connexion d'aujourd'hui et les demandes de trafic massives ont augmenté la charge sur l'infrastructure Internet actuelle et incitent à repenser une nouvelle conception du paradigme des réseaux. Les ambitions des réseaux 5G et au-delà ne se limitent pas à soutenir un tel élan, mais elles fourniront également un échange de données massif avec une grande fiabilité. Néanmoins, la communication basée sur IP reste la méthode de communication Internet prédominante, elle n'a jamais été conçue pour répondre à ces nouvelles exigences, et la plupart des correctifs sont complexes et peuvent échouer avec le temps. Information-Centric Networking (ICN), d'autre part, est un nouveau paradigme candidat, qui vise à résoudre différents problèmes inhérents au modèle actuel centré sur l'hôte. ICN améliore les performances du réseau en fournissant des capacités de mise en cache omniprésentes au niveau de la couche réseau. Bien que de nombreux efforts aient été faits pour intégrer ICN au réseau 5G, la plupart d'entre eux violent la conception native ICN ou ne considèrent pas la nature de l'architecture 5G comme une séparation Core/Edge. Dans cette thèse, nous étudions et étudions les possibilités d'utiliser ICN comme catalyseur de communication pour le réseau 5G. Inspirés par les résultats de recherche approfondis dans le domaine du réseau ICN basé sur la 5G, nous fournissons un examen détaillé et systématique de cette fusion. Cette thèse propose une série d'améliorations pour différents aspects ICN (nommage, la mobilité et la mise en cache dans le réseau) afin de fournir des réseaux ICN transparents et efficaces basés sur la 5G. Des analyses qualitatives et des expériences approfondies prouvent l'efficacité de nos systèmes qui surpassent les solutions existantes et présentent de meilleures performances système avec moins de frais généraux.

Mots Clés: 5G Networks,Content Centric Network (CCN), Internet of Thing (IoT), Information Centric Network (ICN), Complex Network, Named Data Networking (NDN).

Chapter 1

Introduction

In the coming decade, there will be tens of billions of connected devices including smartphones, laptops, and IoT, etc. A recent statistic from Cisco [1] predicts that more than 25 devices will be connected in the near future, while this number is expected to rise exponentially. Such tremendous numbers of connected devices more users' demands for content and compute-intensive services. In parallel, this explosive growth will considerably increase the traffic in mobile networks, leading to a significant paradigm shift in future cellular networking design. On the other hand, the current problems of Internet are a natural consequence of its architecture, it was intended to address simple communication demands as audio and file transmission, in a period when a network was required to share expensive and rare resources, such as long-distance communication links, and mainframe computers [2].

In fact, Internet model was developed many years ago to allow connectivity between specific end-points or devices using unique Internet Protocol (IP) addresses. Hence, most Internet communication happens between a client and a well-known server host based on their predefined IP addresses. However, the current user's interests have since changed, where they are more linked to data sharing rather than end-to-end connectivity. Moreover, the tremendous growth of Internet users with the appearance of new applications has given rise to new requirements from the architecture perspective, such as support for scalable content distribution, security, mobility, energy efficiency, etc. Although numerous add-ons have been developed to support these new functionalities, most of those patches increased the complexity of the overall architecture while others proved to be only temporal solutions.

1.1 Motivation

Today's Internet is invading all life's areas. It has become impossible to dispense with the services it affords to the various sectors of society, e.g., the economic sector, business, education, etc. Wireless communication, notably cellular communication, has established its meriting due to its ease of use. It is witnessing an exponential growth of connected devices that acts as the main factor that drives and pushes the world to new communication platforms where zero-latency communication is required.

The 5G network appears as a robust technology solution that aims to fulfill these requirements. Ubiquitous wireless access with high-speed wireless communications is the prominent feature of the 5G in order to address the high traffic demands. Such a new technology aims to provide a nearly zero round trip latency, low delay, and broad wireless coverage. However, the traditional host-centric model does not, at least in part, fulfill the requirements of today's usage and user demands. This model has been refined to be adapted to the current complex communication system where security, mobility, and flexibility are challenging requirements.

Information-Centric Networking emerged as a powerful enabling technology for the provisioning of scalable and efficient real-time services. It represents a revolutionary communication paradigm for the Future Internet [3]. Different from the current IP-based communication model, ICN poses the attention on content to be exchanged (i.e., information or exchanged data), instead of the need to establish an end-to-end connection between involved peers (i.e., host or server). As content is independent from the location, in-network caching can be applied during communication by caching the content more closer to consumers, which improves data retrieval and reduce network traffic. Moreover, ICN natively supports the users' mobility, simply by re-issuing any unsatisfied requests. Furthermore, ICN provides easy data retrieval based on a request-reply exchange model, content-based security by attaching all security-related information within the data itself, and native support of multicast.

On the other hand, most of the communication patterns, in current applications and usage, inherently follow a content oriented paradigm [4]. By analyzing the change in users' behavior, we can easily distinguish between the earlier and the current communication usage. Users are interested in consuming the requested data from the network irrespective of knowing who is providing such data. For instance, use cases such as watching video content or update mobile applications with recently published content (e.g., Push notification), are mainly focusing on the content as the main element instead of building an entire communication session with the content provider.

Indeed, ICN design and features can facilitate 5G realization [5]. Specifically, ICN switches the address-based Internet architecture to a named-content-based. Hence, name abstraction enables seamless integration and interaction among the different network entities (e.g routers, access points, users' devices, etc), in-network content caching improves the content availability in the network that enhances the quality of service, data-embedded security mechanism enforces security and privacy by focusing on the content and not the communication channel, finally, the clean ICN design improves network performance by reducing network congestion. Thus, we believe that ICN is a suitable communication paradigm that can be used in 5G networks. Although ICN attracts various researchers, it is still in its earlier stage before being widely deployed.

1.2 Problem Statement

Regardless of the enormous efforts in this area; that strive to provide ICN as a communication enabler for the future 5G network, by merging the ICN paradigm with the 5G ecosystem, it still has had various aspects that need to be studied and considered. In fact, the performance and the effectiveness of such an integration stringently relied on the underlying ICN architecture. To date, various ICN research projects have been proposed in the literature such as Data Oriented Network (DONA) [6], Publish Subscribe Internet (PURSUIT) [7], Network of Information (NetInf) [8], Content Centric Networking (CCN) [9], and Named Data Networking (NDN) [10, 11]. Although these projects have different designs and architecture, they share an identical mechanism; addressing the data content by its name rather than the network address of the hosting node.

ICN aims at decoupling the content from its original location. Hence, the network layer becomes responsible for content delivery by applying in-network caching [12]. This feature enhances content distribution and availability, decreases communication latency, overcomes a single point of failure issues, and enhances the overall network performance. However, it is essential to ensure that the caching scheme: (1) does not affect the ICN principle design, (2) pushes popular content near to consumers with less resource usage and lower cost, (3) and does not cache redundant copies in the 5G network.

Besides, 5G deployment will rely on establishing numerous Small Base Stations (SBS) [13], which may rises mobility management issues due to incurring huge signaling overhead, notably for the IP-based model. Although the ICN receiver-driven mechanism enhances the users' mobility and reduces the handover time, seamless mobility for real-time applications still demands a control plane. Designing a mobility control mechanism for the 5G-ICN network is extremely required.

1.3 Contributions

The main goal of this thesis is to study the applicability of ICN as a communication model for 5G and beyond networks. Our purpose at designing suitable supports to achieve 5G-ICN adaptation, intending to achieve a full benefit from ICN features.

We start by covering the built-in 5G ecosystem [14], then highlight the research gap that brings the utility of using ICN as a communication paradigm. We identify the existing key challenges and classify them into two main classes: The naming-related issues and the caching management problems. we propose four contributions detailed as follows:

The first contribution addresses distributed caching management. We propose Cache and Splite (CnS) [15] a distributed cache strategy. In this strategy, each node decides to keep/split the content to their connecting nodes neighbors. The algorithm considered content popularity as a value of content freshness, distance factor, and demand rate in order to decide whether to cache or push the content downstream to their neighbor nodes. This solution aims to improve the content distribution over the network as well as maintain cache utilization. Our simulation results, considering multiple scenarios, reviled that CnS overpass the compared strategies.

The second contribution addresses future massive growth devices, notably IoTs. In such a case, a management plan of large-scale caching is fundamental and required. We have proposed a Profit-based Cache Placement Scheme (PbCP) as a centralized cache scheme [Under Review]. In our scheme, we formulated content placement as an optimization problem. The objective is to maximize the utility function, formulated considering multiple factors that influence caching efficiency. Therefore, we adapted the Tabu search algorithm in order to achieve optimal solutions. The simulation results show a considerable improvement, especially in terms of resource utilization, data transmission time, content diversity ratio, and the number of cache replacement operations.

The third contribution addresses the energy efficiency issue, in future 5G, that plays a significant role on both sides, network operator by increasing their profit; client by extending device energy, e.g., enhancing their battery life. In this context, we propose an Energy Efficiency-based Caching scheme (EaCP) [16]. In order to achieve the best energy resource utilization, our strategy aims to make a trade-off between transport energy and caching energy. The extensive simulation shows that our proposed scheme records a noteworthy energy-saving with low data replication while increasing the cache hit ratio.

The fourth contribution addresses solving an inherent 5G-ICN shortcoming, the producer mobility issue. In this work, we first identify the mobility-related challenges, then we propose a Label based Producer mobility support (LbPM) [17]. In this design, the purpose is to achieve a location free for the producer and its generated content. We proposed a labeling solution in order to prevent the widespread, in case of producer handover occurs, of up-date the routing tables. Our solution not only reduces the network congestion but also enhances end-to-end content delivery, therefore significantly reducing latency.

1.4 Thesis Outline

In this chapter, we have started the research problem, highlighted the motivations, and discussed our contributions. The remainder of this thesis is organized into six chapters as follows:

Chapter 2 provides the background of this thesis. We first start by providing an overview of the 5G ecosystem, then presenting the ICN paradigm and its features. We also present the working principle of NDN, discuss the feasibility of using ICN in the 5G network, and state different research efforts. This chapter also presents in-depth state of the art of ICN solutions in three main parts. The first part deals with ICN solutions from the aspect of naming. The second part deals with the existing 5G-ICN efforts for caching schemes. The last part, states the 5G-ICN mobility supports.

Chapter 3 introduces the first contribution, which is related to the distributed caching management for the 5G-ICN network. This chapter is organized as follows; we first highlight the requirements and challenges for content cache placement in ICN. Then we present the system model and formulate the data caching placement problem. Next, we detail the proposed scheme from both theoretical and practical aspects. Finally, we describe the implementation and discuss the obtained results.

Chapter 4 provides a centralized content caching scheme. It first presents the requirements of an efficient caching strategy and highlights the underling challenges. Then, it exhibits in detail the proposed design, the formulation of the profit model, and the adaptation of the optimization algorithm. Then, it discusses the implementation and evaluation against other strategies.

Chapter 5 introduces the third contribution, which is related to the caching energy improvement for the 5G-ICN network by considering IoT application as the main use case. This chapter is organized as follows; we first highlight the requirements and challenges for an energy efficiency caching placement in ICN. Then we present the system model and formulate the IoT placement problem. Next, we detail the proposed scheme. Finally, we describe the implementation and discuss the obtained results.

Chapter 6 presents A Label-based Producer Mobility Support for the 5G-ICN network. It starts by highlighting mobility requirements and challenges in 5G-based ICN networks. Then, it presents the specification, for producer mobility issues, and discusses the label-based naming scheme in detail. Later, it discusses the implementation and evaluation.

The last chapter concludes the thesis and all the work carried. It recapitulates the challenges, contributions, and summaries our findings. It also presents future perspectives that can be followedup by this thesis.

Chapter 2

Information-Centric Networks for 5G: State of the Art

2.1 Introduction

Historically, and almost every ten years, mobile access technology goes through a revolutionary development resulting in significant enhancement. This rapid change is a response to the enormous demands generated by the pervasive growth of data. 5G network, has emerged as the next phase of mobile telecommunications standards, resulted from an evolution of various ancestor generations of the wireless network. This new standard provides advance wireless communications leveraging technologies to connect people-to-people, people-to-data, people-to-everything, and everything-to-everything in order to achieve unified connectivity. Indeed, like all new technologies, the 5G is pushed with specific qualifications. According to The 3rd Generation Partnership Project (3GPP) [18], 5G consists of a set of classes (Detailed in Section 2.2.1) including enhanced mobile broadband to connect everyone to everything with high coverage area. Besides, the Ultra-Reliable Low Latency Communication (URLLR) provides not only a high data rate with low latency links but also an extremely ultra-reliability and high level of security.

The 5G network is an application-driven architecture [19], aims at supporting the network functionalities by providing content caching and computation capabilities at the network near to consumers. In order to enable the next-generation 5G networks, several technologies have been proposed to fill the deficit and reach the needed requirements by offering easier management and low-cost deployment meanwhile ensuring both Quality of Service (QoS) and Quality of Experience (QoE). One of the most essential standards in the 5G ecosystem, the Mobile Edge Computing (MEC) paradigm [20], describes the platform that has the ability to perform computation and executing services close to consumers. This is achieved by enabling computation offloading from the cloud's data centers to the edge servers. On the other hand, Software Defined Networking (SDN) technology [21] introduces agile management by leveraging a global network's view. The data plane and the control plane in SDN are separated. Therefore, a centralized hardware entity is used to represent the central controller, which is directly programmable. SDN is mostly associated with Network Function Virtualization (NFV) [22]. The latter aims at reducing the deployment cost by decoupling the networks' hardware entities and their dedicated software, then running software on a single platform aiming to provide a single softwarization of network function entity. Each of these technologies tends to provide ultra-reliability communication for the 5G network.

The main focus of the researchers is to find an alternative solution by pushing the communication to a new dimension with high reliability and high data rate aiming at satisfying all the necessities. 5G network is not only a new wireless network, but a global ecosystem that has the ability to provide all the requirements and needs, starting from high coverage tends to connect everything to everything, to high bandwidth and low latency achievement.

The forthcoming 5G era will significantly reshape today's information society, it is obvious that 5G seeks to meet the exponential growth in the demand for communications and data exchange in a wide range of areas, including healthcare, manufacture, agriculture, transportation, etc. However, there are still challenges to support the stringent quality-of-service and quality-of-experience requirements from emerging use cases and applications [23]. Moreover, the current communication model is failing to reply to the current users' needs and requirements and it needs some tuning. This motivates the exploration of new network designs in order to overcome the inherent limitations of conventional IP-based technologies [24]. Recent data-driven approaches such as Content-Oriented Networks and Information-Centric Networking [25] may fix the above issues.

The communication is now shifting towards the Information-Centric Networking (ICN) paradigm [26]. ICN, uses simple content names to derive communication. It decouples the content from its original location, adopts content-based security that secures the content and not the communication channel, and hence applies in-network content caching at the network level. These features make ICN a simple yet efficient communication paradigm for today's Internet [2]. The primary impulse of ICN architecture is to tackle the shortcomings of IP-based networks [27]. ICN changes the concept of host addressing to content naming and enables naming abstraction in order to facilitate the integration of new network functions [28, 29]. ICN also adopts a clean communication model where the content is considered the first-class citizen rather than the host or the owner, which brings new opportunities and features by allowing a simple and ubiquitous in-network content caching, mobility, multicast, etc.

In the current communication model, the consumer should beforehand know the provider's Internet Protocol address (IP) to maintain a session and start the communication. Most of the communication features (e.g., security, mobility, management, etc.) are not built-in [24]. Adding new features and services ends up by designing a complex communication protocol suite that may fail at time and overhead the network resources, which literally affects the network performance and scalability. Thus, different benefits will be achieved in 5G networks if ICN is adopted as a communication enabler. ICN paradigm has attracted intensive interests and been considered as a promising direction for realizing 5G networks. Moreover, 5G solution should support mobility on demand for many scenarios, ranging from very high mobility to low mobility or stationary devices. ICN provides a set of features such as abstraction of content from the location through names [18, 30], consumer mobility support [20], in-network content caching, and content-based security [21]. In this chapter, we provide the background of this thesis Figure 2.1 provides a general organization and taxonomy used.

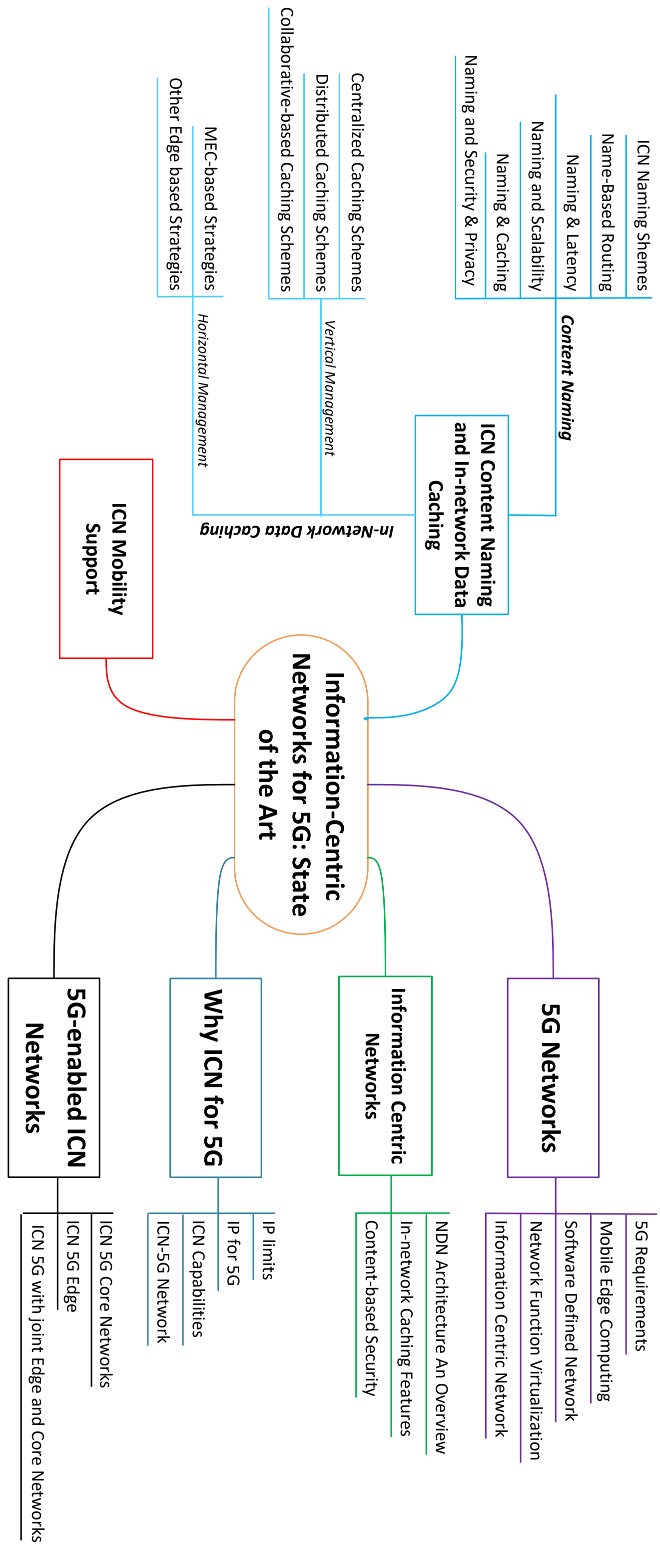


Figure 2.1: State of Art: Organization and taxonomy.

2.2 5G Networks

Driven by each era's requirements, the wireless telecommunications world has evolved from one generation to another. In the beginning, the human tended to communicate without wire. The first footprint on wireless communication was the 1G that intended to provide voice communication based on the analogical mode. However, the poor quality of audio transmission pushed to change the transmission system, hence, paved the way for a second generation. The 2G came as a successor generation that digitized communication to provide better voice quality. This evolution has contributed to deliver not only a good voice quality but also Internet data. In that era, the massive evolution of the received devices, consuming more data streams, added a burden to this technology. The 3G helped keep pace with the evolution of users' devices, including smartphones and tablets, while the ecosystem is witnessing a massive growth in the number of connected devices, new exchange content type as videos, rising demands and bandwidth [31]. These challenges led to the development of 4G, which took it upon itself to provide new services and uses cases.

With regard to the technological evolution behind wireless communications, it is strongly associated with the growth of connecting devices. Nowadays, the massive explosion of the devices and their usage so far has coincided with the increased cravings of bandwidth consumption accompanied by the massive data generation, result in poor QoE and QoS. On the other hand, the rapid development of the communication pattern reviles to new communication usage, e.g., M2M communication, which requires low latency access. The main focus of the researchers is to find an alternative solution by pushing the communication to a new dimension with high reliability and high data rate aiming at satisfying all the necessities. Unlike the 4G cellular network, the new 5G network is not only a new wireless network, but a global ecosystem that has the ability to provide all the requirements and needs, starting from high coverage tends to connect everything to everything, to high bandwidth and low latency achievement.

2.2.1 5G Requirements

According to International Telecommunication Union (ITU) [32], the fundamental use cases of 5G networks, as presented in Table 2.1, are summarized in three essential points: Enhanced Mobile broadband (EMb), Machine to Machine connectivity (M2M), and UltraReliable and Low-Latency Communications (URLLC).

The EMb describes the different use cases that fall into the massive data delivery that requires a high data rate, especially in vast coverage areas. The 5G network highly dependents on different radio access technologies including the higher density of Base Stations (BS), Small Base Station (SBS), heterogeneous networks, and WiFi access in order to ensure high area coverage as well as increasing the network throughput [33, 34, 35]. This allows the 5G wireless network to be promised as a technology to replace the current wire access[36]

M2M communication contains the state of communication between multiple machines without human intervention. These use cases introduce a new communication model among different devices equipment, including IoT sensors and actuators, that communicate with each other [37]. Therefore, the 5G network depends on new communication techniques, e.g., Millimeter Waves and Massive MIMO, taking into account the sensitivity of communication between these machines. Finally, the URLLC category is the fundamental element of 5G networks. It may contain more use cases from other classes.

Table 2.1: Use cases covered in 5G networks.

Use Case	**Example**	**Enabled Technologies**
EMb	• High speed video delivery • Massive coverage • Huge device connection	• Small Base Station • Heterogeneous networks • Wi-Fi access
M2M	• IoT device interaction • Intelligent sensors • Vehicle-to-Vehicle	• Millimeter waves • Massive MIMO • Beamforming
URLLC	• Robotics controlling • Tactile Internet • self-driving car	• Core & Edge Network

Indeed, this class aims at providing low latency and extremely high reliability. The network task lies to provide low latency for various time-sensitive applications, and hence, leveraging with new radio access technologies and the MEC paradigm and robust core management technologies. The impact of the 5G networks (compared to the previous technologies) appears in providing better service at a lower cost and supporting the new emerging applications, e.g., Augmented Reality, Virtual Reality, Gaming, as well as enhancing multimedia services such as Ultra-High Definition display, Multi-View High Definition display. The 5G may rely on a promising solution in order to fulfill these requirements. That solution can be represented in bringing the computation and caching more closer, this means integrating them inside the network itself rather than the far Cloud servers. Hence, a set of new standards and paradigms are promising to achieve these goals. In the rest of this part, we present the key enabling technologies that contribute to the success of the 5G ecosystem.

2.2.2 Mobile Edge Computing (MEC)

Mobile Edge Computing [38], also known as Multi-access Edge Computing, refers to the platform that performs computation and caching services, close to end-users, at the edge of the network [39]. Figure 2.2, depicts a high-level design of MEC functionality within a 5G network, including different interactions between the MEC/Could and MEC/Devices. MEC empowers the 5G network to support the ever-increasing amount of mobile applications and the pursuit of endeavors URLLC by offloading some services from the cloud the edge server [40].

In early 2012, cisco present the Fog computing standard [41]. Fog computing is considered as an extension of the cloud computing paradigm along the delivery path from the core of the network to the edge of the network [42]. It is a virtualized platform that aims to provide computation, storage, between end devices and traditional cloud servers. In contrast to Fog computing [43, 44], MEC is limited to the mobile cellular network and may not provide computing along the entire communication path. In MEC, the radio access network may share its resources to provide fast data delivery, reduce communication latency, and decrease network congestion. MEC is a suitable paradigm for a range of today's Internet applications [45], such as low latency content delivery, data analytics, and computation offloading [46, 47, 48]

The Emergence of MEC: Cloudlet [49] is considered as the initial design of edge computing. The University of Carnegie Mellon has proposed it in 2009. The idea behind Cloudlet is to install computer hardware at the edge of the network that provides computing capacity. Thus, the connecting devices use the Cloudlet platform to perform computation rather than reaching the cloud. Although Cloudlet proves its worth, it is not designed for completely mobile infrastructure. In particular, Couldlet is supposed to be mostly accessed by connecting devices through WiFi connection, which impacts the mobility of users. Industry Specification Group (ISG) within the European Telecommunications Standards Institute (ETSI), on the other hand, has developed the MEC concept, in 2014 [50]. MEC describes the framework that has the ability to offer

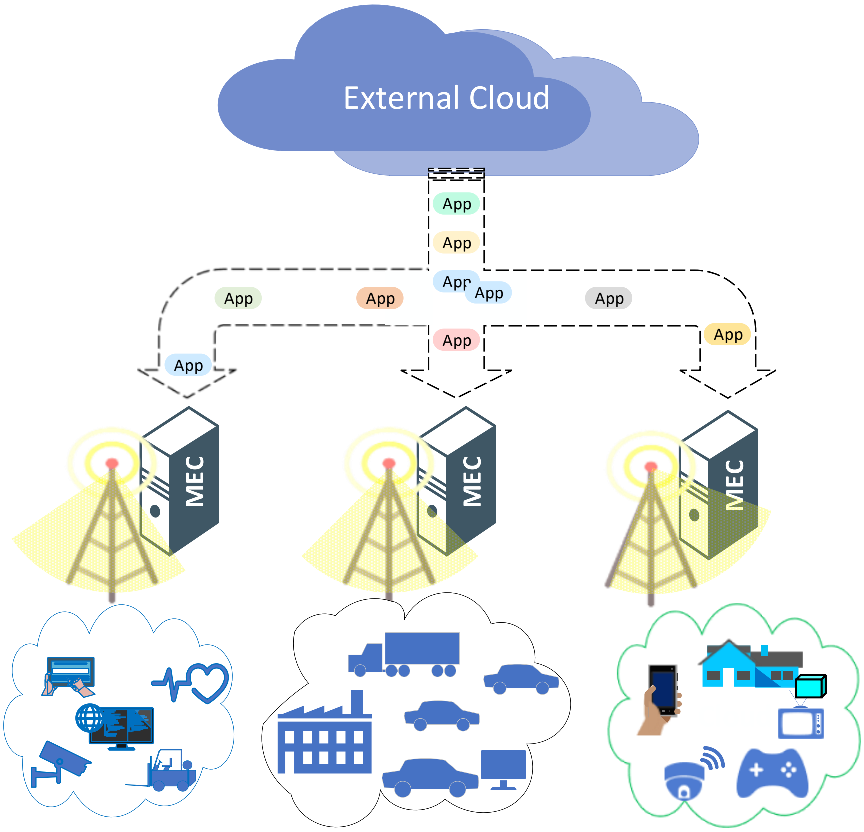


Figure 2.2: An overview of MEC paradigm.

cloud computing capability and an Information technology services environment at the edge of the 5G network, by offloading computation tasks from connecting devices to the MEC platform.

Features of MEC In multi access edge computing paradigm, cloud computing services are expanded to the edge of the network. The computational tasks, from mobile devices to the MEC server, do not only enhance QoE and maximize battery life on the user side, but also reflect on the whole network. The unique advantages that are potentially offered by MEC can be summarized as follows:

- **Efficiency:** MEC server will be closely integrated with connecting devices systems to enhance overall efficiency and performance. Based on real testbed by the computation offloading to the MEC platform, Dolezal *et al.* [51] demonstrate that the energy consumption of the user device's reach up to 93%, besides a decrease of the latency up to 88%.
- **Low-Latency**: MEC provides powerful computing from the nearest location to user devices, thus reducing the need to transfer processing tasks to the remote cloud through the core network, especially for real-time applications requirement. Thus, MEC reduces communication latency and decreases network congestion
- **Security:** The proximity of MEC to users' devices leads to a reduction in the distance that information needs to traverse, which helps in diminishing the chance of eavesdropping.
- **Cognition:** In MEC, the radio access network may share its resources and pool resources along the edge. MEC can distribute computing power and control functions anywhere in the 5G network and connect devices to take full advantage of the resource's availability.

Benefits of MEC MEC technology is one of the keys that tends to reduce the backhaul link's congestion while achieving the latency requirement for the 5G network. Thus, there will be diverse prospects and features for the MEC platform. From the end-users' point of view, MEC provides offloading services for different use cases (e.g., Virtual Reality, M2M, driving assistance system) to enhance QoE by improving the energy efficiency, reducing the deployment cost and achieving the required data delivery deadlines. Moreover, the MEC platform enhances the network performance by providing optimization applications to reduce the load on the backhaul link, especially during peak hours. Furthermore, form the 3rd party services' perspective, MEC empowers the big data interpretation for different treatments. MEC can also contribute to the effectiveness of the network resources, for example, extending the caching capacity of the memory nodes size towards comprising some content [52].

2.2.3 Software Defined Network (SDN)

The fundamental equipment of today's network infrastructure are routers and switches. These devices are mainly responsible for packets switching, forwarding, and routing. Managing and controlling a large number of these devices may end up with different issues in terms of scalability and maintenance. SDN [53] is a new paradigm that aims to shifts the way of network management from the distributed controller to single entity decision making. To do so, SDN decouples between the control plane and the data plan. The data plan is simple forwarding switches, and the control plan consists of the controller entity hardware. The controller has a global view of the network, and its purpose is to inject the routing rules into the connecting switches. In other words, the device entity becomes simple data (packet) forwarding called SDN switch, where the forwarding decisions are flow-based. This controller provides an efficient control to the whole network via software-based programming interfaces (i.e., southbound interface and northbound interface). Network devices become simple forwarder switches while the centralized controller [54] decides all routing/forwarding rules. The use of SDN makes the network management easier and more efficient, improves the overall performance, and facilitates network evolution.

According to the Open Networking Foundation (ONF)'s white paper [55], a reference SDN architecture may consist of three layers, as shown in Figure 2.3 and explained below:

- **Management Plane:** It is the upper layer in the architecture and known as the application layer. It provides an access point in various forms, such as Application Programming Interfaces (API). An SDN application can conveniently access the global network view. Thus, SDN offers a Platform-as-a-Service (PaaS) model for networking [56].

- **Control Plane:** It is the middle layer in the architecture and known as the core layer. It serves as a bridge between the two other layers via the Southbound Interface and Northbound interface. The control layer consists of a centralized controller with the responsibility of downward flows, controlling and collect network status from the infrastructure layer, and provide a global view of the network to the application layer. In the upward flow, the controller takes requests from the application layer to manage the network devices in the infrastructure layer [53].
- **Data Plane:** It is the lowest layer in the architecture and known as the infrastructure layer. It consists of SDN switching devices. These elements are interconnected with each other to formulate a single network aiming at collecting the network status, storing this information temporally,

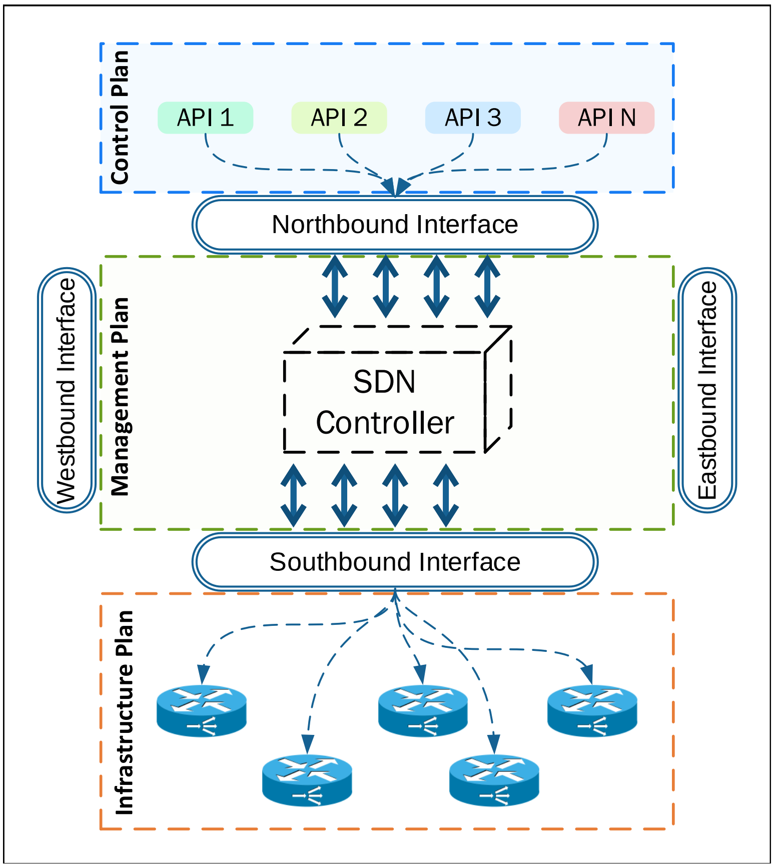


Figure 2.3: An overview on SDN technology.

sending them to the controllers, and processing packets based on rules provided by the controller [57].

One of the promises of the 5G network is to enhance mobile broadband. In doing so, the SDN must handle a large geographical area in a distributed fashion. Each SDN controller maintains two additional interfaces: East and Westbound Interface introduced to support multiple SDN controllers. By leveraging these interfaces, the SDN controllers inter-communicate to share network information and coordinate their decision-making process. Various implementations of distributed SDN controller has been proposed including Onix [58], Disco [59], ElastiCon [60], and ONOS [61]. Each architecture has a dedicated approach. Some of them use a distributed file system in order to maintain a logically centralized and distribute the control plane, while others apply a master/slave communication style.

SDN Features: SDN delivers numerous features to the 5G network since the network entities (switch/router) became a simple packet forwarding devices that flow the injected flow rules. This can help in:

- SDN not only provides the ability for networks to be programable but also unifies the multi-vendor and multi-technology environment. This enables network operators and service providers to innovate faster and allow software-based innovation [62].
- SDN provides high transmission speed since SDN switches do not need to perform any matching or packet inspection to make a forwarding decision for each packet. This simplifies their tasks and provides the efficient use of network resources [57].
- The network administrator has the ability to alter and manage the needed characteristics to nodes from a central location. This overrides the configuration for each router separately, especially for large-scale networks [56].

In SDN architecture, the northbound APIs connect the centralized controller with the services and applications running over the network. The main job of this component lies to transfer the needs for each application, e.g., storage, bandwidth, etc. where the network tries to satisfy the request of each application.

2.2.4 Network Function Virtualization (NFV)

Among the essential components of today's network are the embedded hardware appliances that afford essential network functionality. These appliances are also known as Middleboxes as they are integrated to provide network services such as Deep Packet Inspection (DPI), Firewalls, Network Address Translation (NAT), Load Balancers, etc. The network operators pay high costs to deploy and manage these Middleboxes in order to meet the network services requirements. However, the new services' innovation raises new challenging problems such as providing these services in the required dedicate period. Moreover, these appliances still face manageability problems that result in high expensive deployment costs and are time-consuming, in which, service providers consider it as a failure modes [63]. To overcome the aforementioned issues and problems, NFV has been introduced as an emerging paradigm that aims at extracting the software from its installed hardware and therefore providing flexibility and agility to the networks' services [64].

NFV, firstly introduced in October 2012 [65] authored by over twenty of the world's largest telecommunications operators, including Deutsche Telekom, American Telephone, British Telecom, etc. They formed an Industry Specification Group (ISG) within the European Telecommunications Standards Institute (ETSI) to bring the NFV's philosophy to reality. Figure 2.4 depicts the transformation model from the physical model towards the NFV system. In traditional mechanisms, the data packet must be handled by multiple dedicated hardware and servers that offer multiple services respecting sequential and logical chaining. The NFV-based system aims at accelerating this process based on the virtualization technology that improves the end-to-end interconnection.

NFV, technically, tends to decouple between the software and the dedicated hardware. Therefore, this software becomes a virtual function instance or Virtual Network Function (VNF) [66] , that runs on a virtual machine in a dedicated server, e.g., standard commercial off-the-shelf servers (COTs). Functions are chained together and orchestrated in order to provide a network service [67] . Hence, NFV technology brings vital features, such as facilitating network services provisioning and improving network scalability, flexibility, and cost-efficiency.

NFV Benefits The major benefits of NFV can be summarized as bellows:

- NFV contributes not only to reducing the middleboxes density in the network but also in facilitating the creation, deployment, management of services, and reduce the Capital Expenditure (CAPEX) and Operational Expenditure (OPEX).
- By adopting NFV, the physical radio resources can be abstracted and sliced into virtual network resources that hold certain corresponding functionalities. Therefore, multiple parties can share the same infrastructure with isolated network services.
- NFV tends to reduce end-to-end network interaction, which can reflect on reducing the 5G round-trip latency and decreasing the network delay.

The NFV aims at optimizing the network functions, which bring potential benefits to the network through enhancing service delivery and reduce cost. SDN has the capability to handle decisions of massive infrastructure devices (switches) from a single controller, which facilitates the network administration. Both SDN and NFV contribute to improve network flexibility and provide fast service deployment. The ONF published several white papers that

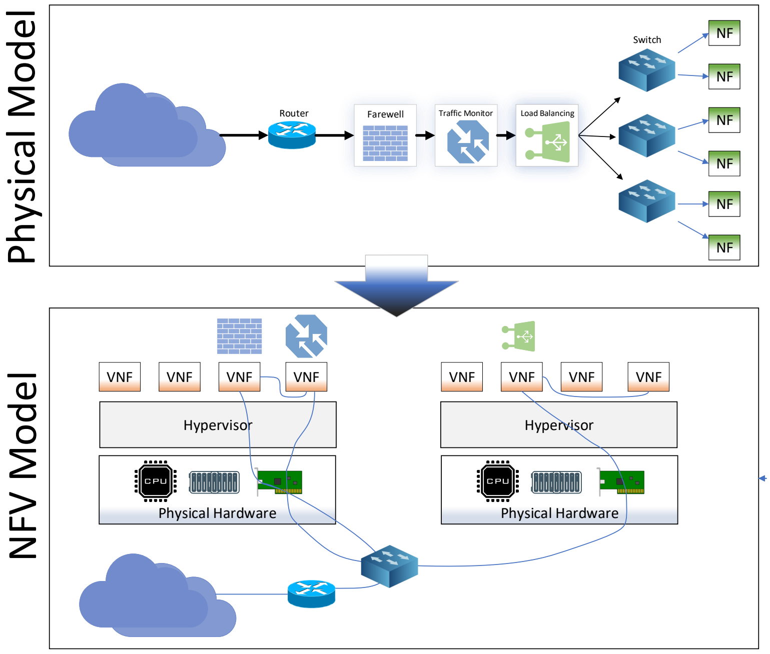


Figure 2.4: An overview on NFV.

describe the coexistence between the NFV and SDN. For instance, work in [68] illustrates the architecture design and highlights its advantages.

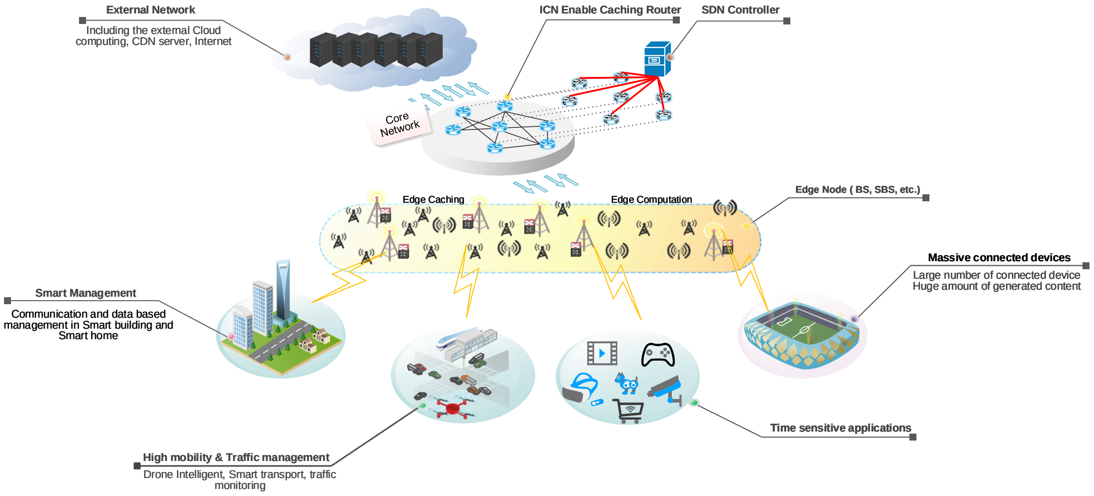


Figure 2.5: An Overview on next-generation network.

The 5G network promises on affording high-quality services with low latency requirements and the highest reliability. In this regard, various technologies, as depicted in Figure 2.5, have been proposed and developed. The application-driven nature of 5G networks contributes to sharing the same infrastructure between different services while achieving better resource exploitation and enough QoE and QoS while taking into consideration diminishing in the CAPEX and OPEX. A multitude of technologies lies behind the 5G revolution. The purpose is to invest in a new network architecture that implements both computing/caching capabilities. Hence, leveraging next-generation networking, including MEC, is mandatory to improve cloud computing capabilities and enhance the network edge. Besides, the merger of NFV and SDN may contribute to reducing network management, enhancing services provisioning [69], decreasing the complexity, and allowing for software-based innovation. However, the need for agile content distribution highlights the major limits of the IP based model. Moreover, total lean on the virtualization technologies to deploy the 5G network brings the traditional Virtual Machine (VM) issues such as VM migration [70] since traditional IP mobility management remains a hard challenge to address.

2.3 Information-Centric Networks

The Internet communication model is deeply routed in the Shannon-weaver communication model. This model is basically a point-to-point-based communication where there is no information regarding the exchanged content. All that we have is a unidirectional link: the sender address, the used channel, and the address of the receiver. This type of networking cannot model today's Internet and applications, where there is a huge number of connected devices, sensors, and routers that are forming the today's Internet. The current communication model is failing to reply to the current users' needs and requirements and it needs some tuning. Recent data-driven approaches such as Content-Oriented Networks and Information-Centric Networking [71] may fix the above issues.

ICN [71] is a new communication paradigm that promises to replace the current IP concept. ICN is built upon the use of the content name as the primary network element rather than the host address (IP) [72]. To meet the requirements of Future Internet Architecture (FIA) [73], various architectures have been proposed, as shown in Figure 2.6, under the umbrella of ICN, including DataOriented Network Architecture (DONA) [6], Content Mediator Architecture for Content-Aware Networks (COMET) [74], Convergence [75], MobilityFirst [76], Scalable and Adaptive Internet Solutions (SAIL) [77], Publish-Subscribe Internet Technology (PURSUIT) [78], Nebula [79], eXpressive Internet Architecture (XIA) [80], ChoiceNet [81], Content-Centric Network (CCN) [82], Named-Data Network (NDN) [10] and Green ICN [83]. Although these architectures have different design principles and implementions, all of them are based on the use of the content name as the main network element to drive the communication. Content-Centric Network and Named-Data Network, in particular, are the most active ICN projects that use hierarchical human-readable names to identify content. In the following, we will highlight the architecture design of NDN/ICN and highlight their working principle.

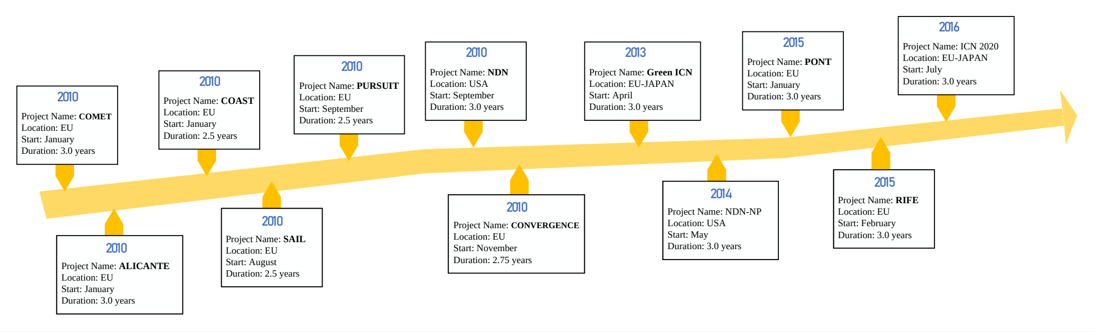


Figure 2.6: Historical overview of ICN projects.

2.3.1 ICN/NDN Architecture Overview

Content-Centric Network and Named-Data Network, in particular, are the most active ICN projects that use human-readable names to identify content. Next, we will illustrate the main components entities in ICN concept implementation. We will start from the packet design to the inside node structure tables until the entire request and reply process. Then, we will illustrate the ICN caching and naming features.

NDN Packet Design: The main focus on today's networking is hosts bind to IP addresses. Since in IP based networking that there is unnecessary layers [84], NDN idea to networking that changing this concept to content itself. NDN suggests a new architecture model aiming to fits today's networking needs. Technically, a user or client just states its own want that is named in protocol as Interest. Whenever a user needs data it sends an Interest packer as depicted in Figure 2.7 (a), and the network is responsible to brings the data back. In this fetching model, a user doesn't connect to another user or server. It is a defacto standard. On the other hand, the network actually pipes the data also. The network reply to the user by sending a Data packet Figure 2.7 (b). The design of such a simple communication model provides more manageability options (inside the network itself) as it allows content independent from the original producer and allows for data classification, etc.

NDN Node Architecture: An NDN node maintains three data structures: (a) Content-Store Table(CS): to temporary cache data and serve it for future requests, (b) Pending Interest Table (PIT): to keep track of the unsatisfied interest

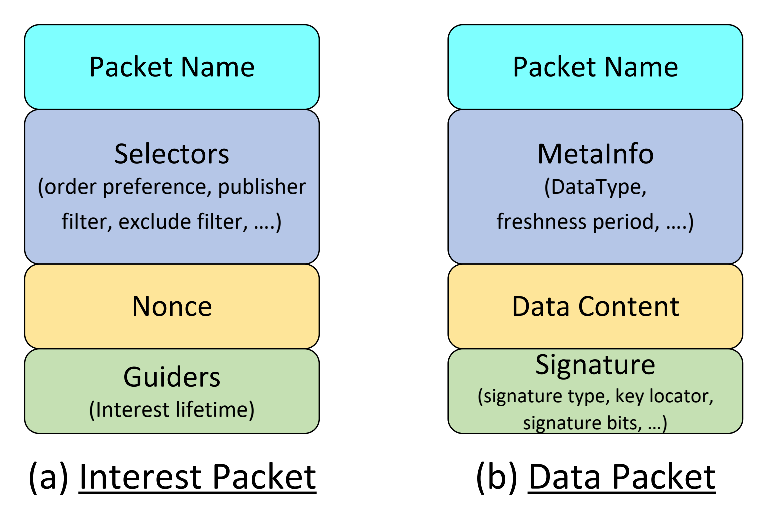


Figure 2.7: Packet Types In Named Data Network

and to deliver the data packets back to consumers, and (c) Forwarding Information Base (FIB): to record the suitable interfaces for each reachable name-prefix.

A. Interest Forwarding Process: After receiving an interest packet Figure 2.8, the forwarding node consults its local CS to check if a copy of the requested content exists. If a match is found in the CS, then Data will be returned from the cache and is sent out via the same received interface and the Interest packet is dropped. Otherwise, If cache miss, a PIT look-up is performed to verify if a similar interest has been already forwarded. If a PIT match is found, the node adds the received interface ID to the PIT entry (also referred to as interest aggregation), returned data will be pushed to that interface and discards the interest packet. If there is PIT misses, the Interest name will be added to the PIT table, then checked in FIB. If not, it will be dropped or returned by negative acknowledgment. Otherwise, if interest is suitable with FIB, it will be forwarded by the Forwarding Strategy.

B. Data Forwarding Process: In contrast to the interest forwarding process, only the PIT is involved in the Data packet forwarding Figure 2.9. When a node receives a data packet, it checks the PIT to verify whether the associated request has been already forwarded via the node that receives the interest packet or not. If no match is found in the PIT table, it means that no interest carrying this name has been forwarded and thus the data packet is considered an

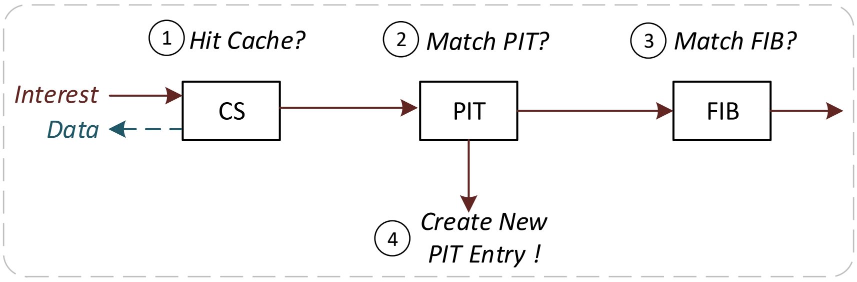


Figure 2.8: Interest Forwarding Plane.

unsolicited packet and dropped immediately. Otherwise, the node forwards the data to all listed interfaces in the PIT entry (multicast), and expunges the PIT entry. In the meantime (according to the current caching placement strategy), the data may or may not be cached in the local CS.

Name-based Forwarding: NDN is a receiver-driven architecture that implements the Interest-Data exchange model. A consumer to request content triggers an Interest message. This message carries out the requested content name and is forwarded upstream using name-based routing rules. When the Interest message reaches the original content producer or a replica node that has the requested content, a Data message is delivered back carries the requested content. By using the content name as the primary network element, ICN decouples the content from its original location, which enables the network layer to cache and serve content for future demands regardless of the availability of the original content publisher [85, 86]. Moreover, ICN implements content-based security [87], which means all security-related information is applied to the content and not the communication channel. These information travel and cache with the content in all its usage cycle.

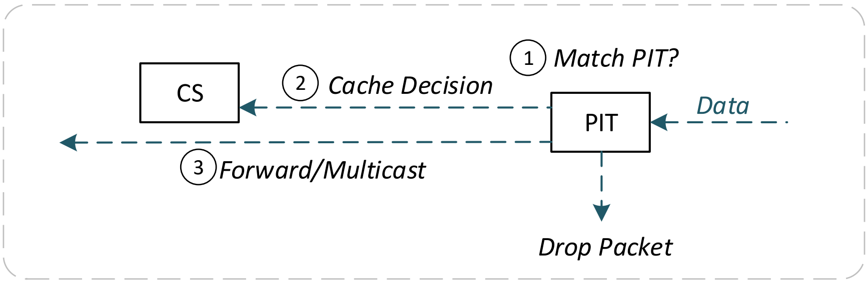


Figure 2.9: Data Forwarding Plane.

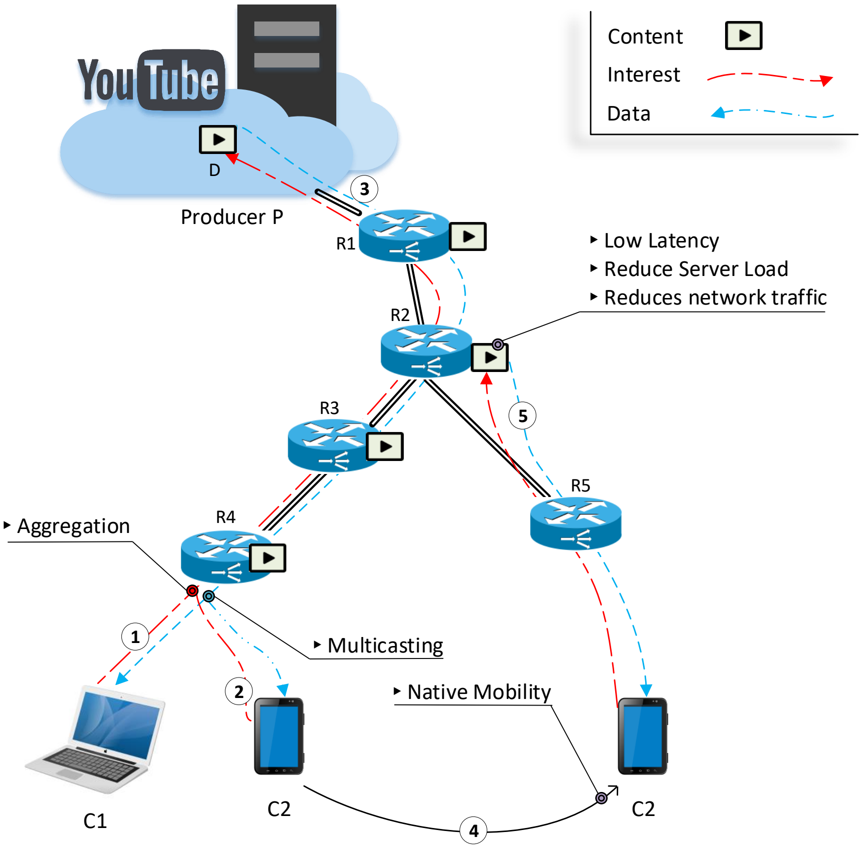


Figure 2.10: An overview on ICN communication.

Figure 2.10 illustrates the ICN communication concept. Assuming the node *C*1 is requesting content *D*, the ICN concept requires the consumer to specify only the name of the requested content. The content is fetched from the original producer *P*, and all intermediate nodes in the communication path are able to keep a copy of the content in their content stores. When another consumer, e.g., *C*2 requests the same content *D*, the demand will be satisfied by *R*2's content-store without the need to communicate with *P*. Besides, when there is a content transmission error, the re-transmission process is assigned to the closest node (replica node) that has cached the requested content. The consumer does not need to make another request to the original producer *P* [88]. Similarly, during node mobility, an ICN node is not required to request a new IP address after connecting to the new network. Instead, it sends only the unsatisfied requests using only the content name [89].

2.3.2 In-network Caching Features

In ICN, each node can cache the frequently used content leveraging their embedded memory [90]. ICN architecture built-in content caching management services that enhance the end-to-end delay. The Interest can be satisfied with intermediate nodes, which decreases the latency and reduces the network conjunction. The in-network content caching maintains two main operations: content placement and replacement policy [91]. The former consists of selecting which content to cache and at which node (placement). The latter aims at selecting which content to remove from the content-store to keep room for the newly arrived popular content (replacement).

2.3.3 Content-based Security

Unlike IP-based networks, ICN maintains security by applying a content-based security concept [92, 93, 94] . Since the content is the first citizen in ICN architecture, ICN secures the content itself rather than securing the communication channel. All security-related information is stuck with the content and travels with him during the communication.

ICN decouples the content from its original location. It relies on a set of built-in features such as location independent content naming and content-based security. These features enable the network layer to provide pervasive and transparent in-network caching. The caching feature tends to improve content delivery, reduce load at the original provider, and efficiently distribute content in the network [62].

2.4 Why ICN for 5G ?

The forthcoming 5G era will significantly reshape today's information society, it is obvious that 5G seeks to meet the exponential growth in the demand for communications and data exchange in a wide range of areas, including healthcare, manufacture, agriculture, transportation, etc. However, there are still challenges to support the stringent quality-of-service and quality-of-experience requirements from emerging use cases and applications. This motivates the exploration of new network designs in order to overcome the inherent limitations of conventional IP-based technologies [95]. In the following, we will list some limitations and describe why the IP-centric model is ill-fitted to the new application and use cases of the 5G network. Then, we will argue how the ICN model can handle such limitations.

2.4.1 IP limits

The Internet functionality is to deliver content (what) to serve the requested needs of different applications. IP protocol names the endpoints, for instance, server and client (where) in order to facilitate universal interconnectivity as the current way to deliver data. This end-to-end communication principle has a fundamental mismatch between data delivery and naming of the endpoint, which leads to many embedded limitations as:

Security failures: IP protocol was originally designed with no security support. Multiple solutions and patches were added later to secure the entire communication channel e.g., TLS, IPSec [24, 96]. However, the security is might be guaranteed only in the communication session that was established and not all the time. Besides, most add-on security patches may fail over time.

Dealing with heterogeneous networks: The IP protocol names each attachment point, which leads to binding with an underlying identifier, known as the layer-2 identifier [97]. Therefore, such an approach makes it hard for the IP-based network to support multi-homing and handle dynamic change in dynamic topologies, for instance, ad-hoc scenarios. Thus, it is so hard to build a steady end-to-end channel to ship bits, notably, in some heterogeneous networks scenarios, e.g., sensors network, vehicle network, delay- torrent network, etc.

New Functionalities Support: IP-based host-centric protocols work awkwardly in mobile environments. Hence, the IP model can not fit well the new applications and use cases, in which the device position or identity is not that important, such as IoT scenarios, sensor networks, vehicle networks, etc. For these applications, the importance is to the content rather than the producer or source devices. Hence, the data is what interests and not the device location or even device name. For example, a user may be interested in knowing the temperature level in his house, if we consider multiple sensors can provide such data, he doesn't care which sensor can provide him such information more than he cares about the information itself.

2.4.2 IP for 5G

Current research efforts and propositions on 5G network architecture adopt different views and propositions. One of these views propose a 5G design centering on the evolved radio access network (RAN) while preserving 4G's core network architecture, but over a flexible network virtualization networking (NFV/SDN)-based infrastructure. 5G also means to inherit the drawbacks of the current IP architecture, with respect to:

- Complex core networking based on tunneling technology e.g BGP [98], to support mobility and network heterogeneity.
- Complex techniques and solutions are deployed on top of IP protocol to avoid security threats, most of these solutions result in significant resource usage high operational costs.
- A networking infrastructure that can not leverage the agility of computing and storage resources in the transport infrastructure.
- Lack multihoming support [99, 24].

2.4.3 ICN Capabilities

The clean and straightforward design of ICN architecture helps in implementing a simple yet efficient communication platform for 5G by avoiding complex protocols to ensure security, mobility, and data management [17]. Besides, the use of content naming instead of host addressing helps in providing a unified platform for different applications and use cases within the same application programming interfaces. Moreover, employing in-network content caching helps in affording content at the network layer regardless of the original producer's reachability and enhancing the quality of service and user experience [15, 100]. Next, we will mention most of the features, which are built-in ICN by design, that promote a promising candidate, and can replace IP based in 5G and beyond networks.

- **Build-in data security:** In ICN network, a producer authenticates the content itself, e.g., its cryptographic signature, to each data piece that it sends over the network. Therefore, a data consumer or user can easily assess and checks the trustworthiness of the data. To this end, ICN secures the data rather than the entire communication channel as the IP-centric model, which makes communications much more secure and trustable.
- **Efficient content delivery:** ICN supports universal in-network caching, data can be cached on any network elements along their transmission paths. Each node including, router, SBS, gateway, user device, can cache the frequently used content leveraging their embedded memory. By leveraging the build-in ICN caching management services, requests can be fulfilled from the intermediate nodes, therefore, reducing the end-to-end delay, decreasing the latency, and reducing the network conjunction. Moreover, bringing the popular content toward the edge network leads to a decrease in the backhaul-link consumption, reduces the network core congestion, and reduces the Interest trips. Hence, the bandwidth used for delivering content, especially the popular ones, can be significantly reduced [101].
- **Built-in users mobility support:** mobility is one of the fundamental requirements for future networks. ICN decouples between content and its location. It allocates unique and location-irrelevant names to each network node and data (e.g., devices, and services, content). As the name is fully decoupled from the network location, an entity's name would not change when it moves.

2.4.4 ICN-5G

Each ICN-5G node (i.e, router, gateway, base station) will cache content traversing the network while routing back the content to the requester. Therefore, ICN architecture built-in caching management services will enhance the end-to-end delay; the Interest can be fulfilled from the intermediate node decreasing the latency and reducing the network congestion. Thus, 5G-ICN architecture based on the NFV/SDN framework can realize not only content distribution but also a top-down service-centric platform. Besides ICN-based service delivery platform becomes a natural extension of the cloud into the network infrastructure and benefits from MEC computing capabilities. Thus, open the possibility to:

- ICN-5G can integrate computing, storage, and network virtualization on the same resource platform.
- Possibility for an ICN-5G service delivery to orchestrate complex service logic execution leveraging SDN and NFV and content processing at the MEC.
- In a real-world testbed, ICN has proven its usefulness in ad hoc infrastructures and delivering unified IoT services over the 5G framework[102].

The 5G network promises on affording high-quality services with low latency requirements and the highest reliability. The application-driven nature of 5G networks contributes to sharing the same infrastructure between different services while achieving better resource exploitation and enough QoE and QoS taking into consideration diminishing in the CAPEX and OPEX. A multitude of technologies lies behind the 5G revolution. The purpose is to invest a new network architecture that implements both computing/caching capabilities. Hence, leveraging next-generation networking, including MEC, which is mandatory to improve cloud computing capabilities and enhance the network edge. Besides, SDN and NFV contribute to rendering the agile network management that facilitates innovation and network programmability. Finally, the role of ICN communication is obvious as a clean model that contributes to enhance content distribution and delivery and improve caching management.

2.5 5G-enabled ICN Networks

Different benefits will be achieved in 5G networks if ICN is adopted as a communication enabler. ICN paradigm has attracted intensive interests and been considered as a promising direction for realizing 5G networks. As of this writing, several research efforts are attempting to adopt ICN within different 5G entities. In this thesis, we classify these efforts based on the already existing boundaries in the 5G network specification, for instance, Core and Edge network.

2.5.1 ICN 5G Core Networks

5G-ICN core solutions aim to adapt ICN architecture to support 5G use cases and new functionalities as network programming, network slicing, etc. Besides the ICN features, such a new supported option will systematically reduce the in-network congestion. In the following, we will present an in-depth analysis of the most recent works that follow this curriculum.

Liang *et al.* [29] combine ICN, NFV, and SDN to virtualize wireless architecture in 5G networks. The main idea consists of taking advantage of the virtual network created and managed by the NFV and SDN in order to integrate the ICN-based communication model. The objective is to allow the network to share the same infrastructure (via virtualization) and embed content caching simultaneously, hence, reducing the redundant cached content among different network slice. The architecture enables virtual resource allocation (including content slicing, network slicing, and the flow slicing) with the caching strategies as an optimization problem. The objective is to maximize the gain of the virtual network operations to improve the end-to end network performances. Therefore, the authors suggest the interior point method to solve the problem. Compared with the traditional approaches, the simulation results show that the proposed scheme has a high ability to reduce pressure on backhaul-link. However, it does not take into account user behavior, e.g., mobility. Moreover, the caching strategy may need more investigation since it is based only on content popularity. Similarly, Tian *et al.* [103] propose a 5G core network architecture based on ICN communication standard, namely 5G@ICN. The authors start by analyzing some technical-related gaps and shortcomings, that inhibit a seamless integration of ICN in the 5G core network, including name resolution problem, multihoming, multipath data transmission, network slicing, and network programmability. Then, to address these technical limitations, 5G@ICN architecture has been designed based on the concept of network softwarization. The idea is to enable a programmable data plane and leveraging a centralized controller for ICN. The experimental results showed that the 5G@ICN prototype test can achieve QoS requirements of drone control and live video streaming while considering mobility circumstances. However, the caching ICN manageability was neglected in this work.

2.5.2 ICN 5G Edge

Enhancing Edge capabilities improves the user quality of experience. Most of the efforts in ICN-5G Edge-based solutions focus on leveraging ICN features for the Edge platform. In the following part, we will present most of the works in this area.

Li *et al.* [104] propose an SDN-based orchestration framework that merges MEC functionalities and ICN/CCN capabilities at the edge network. The authors use SDN framework to support both ICN and IP forwarding by decoupling the data plane and the forwarding plane. The management process is assigned to a centralized SDN controller. The authors also design an SDN-based forwarding strategy in order to support both ICN name-based forwarding and the conventional packet level flows. The simulation results show the effectiveness of the proposed architecture that outperforms the conventional structure. However, the proposed solution does not take the case of a large-scale network where the transaction from IP to ICN and verse-versa become substantial and challenging. Lertsinsrubtavee *et al.* [105] propose Picasso, an ICN-based MEC framework. Picasso is designed to adapt within the high network dynamic where service delivery can fail due to links' instability. Therefore, the authors exploit the ability of ICN to not only use for content delivery but also service delivery. The solution is based on the Docker platform to create and deploy services. Picasso takes benefits from ICN features such as caching the created services, fast service retrieval, and delivery. Compared with the traditional IP-based MEC frameworks, Picasso achieves high traffic accomplishments and low latency. However, the proposed solution may violate the simplicity of ICN architecture. It needs additional modifications in the design principle to support additional communication models such as push and pull traffic. Benkacem *et al.* [106] propose an ICN based Content Delivery Network (CDN) that aims at enhancing the content delivery in the 5G next-generation networks while reducing the core network conjunction. The proposed architecture leverages MEC and NFV to joint both ICN and CDN to provide an active content distribution service. The architecture consists of a double level of content cache/distribution system. At the same time, the consensus algorithm tends to connect between them (the CDN slice and the ICN slices) at a specific getaway. The simulation results show that the proposed platform outperforms the conventional IP-CDN based in terms of throughput, publishing time, and content retrieval. However, the major drawback can be the high overload at the ICN/CDN gateway regarding the size of the transferred content. The applicability of this architecture remains open questions.

2.5.3 ICN 5G with joint Edge and Core Networks

In the end-to-end data delivery, all the 5G parts were involved including the core network and the network edge. However, focusing only on the core or the edge will result in ignoring the entire end-to-end data delivery pipeline. In the following, we will analyze the most cited works that attempt to consider ICN as a communication protocol for both core and edge-based 5G networks.

Tourani *et al.* [107] propose an ICN-based Mobile Converged Network (ICN-MCN) architecture. The authors use ICN in order to enhance the device's multihoming when using different embedded interfaces in the users' device. Hence, the network provides fast content retrieval and seamless node mobility via both content and host multihoming. ICN-MCN aims at enhancing the user's quality of service and quality of experience. The authors also discuss some opening challenges to bring novel architecture in real-world applications, such as adopting the native support of multihoming in order to enable multi-interface and ensure fast transfer for the high volume content, the need for an efficient caching and cache replacement strategy, and addressing the producer mobility-related issues. However, such an architecture may need major modifications of heuristics algorithm to be adapted in high network heterogeneity and address the consumers' behavior such as mobility and the diversity in demand for better bandwidth utilization. Zhang *et al.* [108] focus on QoS provisioning for content transmission in 5G networks. The authors propose an architecture that aims at guaranteeing massive data delivery under delay-bounded. The architecture combines both NFV, SDN, and ICN in which the latter plays a vital role due to its receiver-driven mechanism and the pervasive in-network caching. NFV is mainly used to encapsulate the physical layer and SDN as global network management. The proposed architecture aims at jointing the needed resources to select the optimal delivery path that fulfills the QoS requirements. In doing so, three scenarios of virtual network selection and transmit power allocation schemes have been presented, including a single requester, multiple requesters, and under non-cooperative gaming theory among all requesters. The results prove the efficiency of the solution. The non-cooperative game strategy shows a high convergence over the nash equilibrium. However, the proposed schemes may be foolproof in the wired connection, but the instability of the wireless communication link may pose some issues.

Zhou*et al.* [109] propose an integrated framework to support caching, computing, and communication within the 5G next-generation networks. The essential components of the proposed architecture are MEC to provide computation, ICN to enable caching capability, and SDN to act as a global virtual network manager. The objective is to provide a virtual network on-demand service with specific requirements, e.g., virtual network with only computing, only caching capability, or with both. To do so, the authors consider all possible elements interaction use cases between MEC and ICN (the framework usability includes MEC-assists ICN and/or ICN-assists MEC).They then adopt a distributed Alternating Direction Method of Multipliers (ADMM) algorithm [95] to trade-off between caching, communication, and computing resources. The simulation results prove the hypotheses of the need to jointly computation and caching and using them to assist each other. Although the architecture allows better use of resources on the same infrastructure, the security and privacy aspects have not been covered.

Each work of the studies mentioned above attempts to provide a different solution that joins the ICN within the 5G network. Some of these efforts consider the content to be allocated as a network resource, and others tend to optimize the bandwidth based on the in-network content caching feature, etc. Besides, the majority of these works try to enhance the 5G network performance by leveraging ICN as a communication enabler. Therefore, this can be achieved based on its fundamental elements which are content naming, and in-network content caching. In the following two parts, we systematically review the existing content naming and in-network caching schemes in 5G-based ICN networks.

2.6 ICN Content Naming and In-network Data Caching

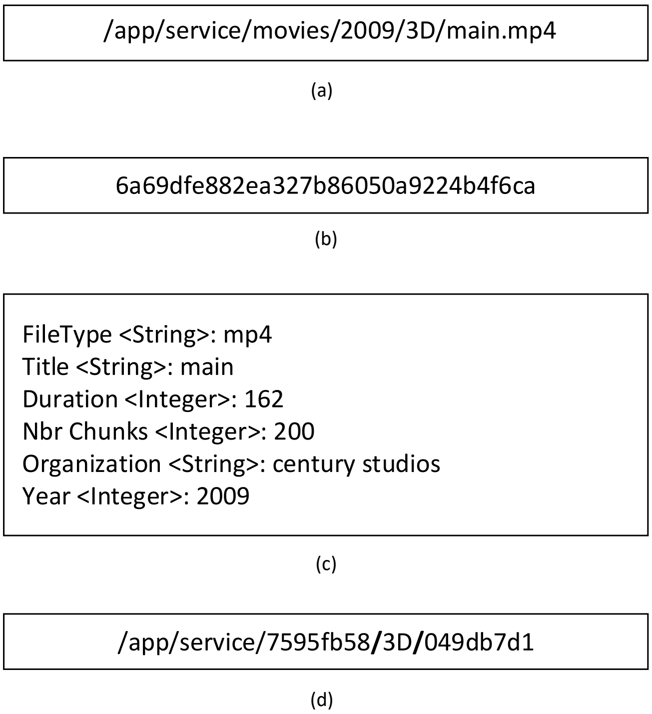
In this section, we will cover the most essential pieces of ICN architecture. Content naming and data caching, the former provides content identification and a routing mechanism, which leads to improving content retrieval and enhancing the latency, and the latter increases the content availability and ensures fair data distribution.

2.6.1 Content Naming

The content name is the key element in ICN, since the fundamental concept of the ICN is to switch the address-based Internet architecture to a named-content-based. An ICN name should be a unique identifier of each content object, which permits either the user or the intermediate nodes to locates the best content holder. It should also be persistent to validate the content. The used naming scheme must be scalable and allows name aggregation. Naming schemes in ICN introduces four main types of naming schemes:

- **Hierarchical Name:** consists of multiple components to identify the application and describe the services. It has a similar structure to current Uniform Resource Identifiers (URIs) and may originate user-friendly and convey meaning names to users. Hierarchical naming enhances scalability since name prefix can be aggregated. However, the main drawback appears in the small content size where the content name becomes much larger than the content itself. Figure 2.11(a) depicts an example for content name prefix in a hierarchical format.

- **Flat Name:** is mainly obtained through hash algorithms applied to content. There is neither a semantic nor a structure behind a flat name. The generated name is not human-friendly and can hardly be assigned to dynamic content that is not published yet. Flat naming has scalability issues since it does not support routing aggregation. Figure 2.11(b) shows a flat name for the hierarchical name in Figure 2.11(a). This result has been obtained by using the MD2 hashing algorithm.
- **Attribute-Value based Name:** has a collection of attributes; each attribute has a name, a type, and a set of possible values (creation date/time, content type and location, version, etc.). Collectively, they represent a unique content and its properties. Attribute-value based naming scheme supports an easy searching process by using known content keywords, which means it is quite possible to find many contents against one search request. Hence, it is hard to ensure naming uniqueness and find unique content in a short period.Figure 2.11(c) illustrates attribute naming scheme's



/app/service/movies/2009/3D/main.mp4

(a)

6a69dfe882ea327b86050a9224b4f6ca

(b)

FileType <String>: mp4
Title <String>: main
Duration <Integer>: 162
Nbr Chunks <Integer>: 200
Organization <String>: century studios
Year <Integer>: 2009

(c)

/app/service/7595fb58/3D/049db7d1

(d)

Figure 2.11: ICN naming schemes for the same content.

example.

- **Hybrid Names:** It combines at least two of the previously discussed schemes or all of them. The idea is to take advantage of the base scheme's best features to improve network scalability, performance, and enhance security and privacy. For example, taking advantage of name aggregation to enhance the lookup process, and the fixed length of flat names to save space and attribute values to provides keyword searching and security/privacy. Since each content has a unique name identification, the generation of the hybrid names may become a challenging issue, notably for non-static and real-time generated content. Figure 2.11(d) depicts an example of hybrid name.

Each ICN architecture uses different naming scheme that addresses specific issues. For instance, MobilityFirst [76] adopts flat names and combines Broadcast and Content Based (CBCB) [110] while NDN [11] uses hierarchical names. Zheng *et al.* [111] presents notes about differences and similarities between the naming in both IP and NDN architecture. The paper provides clarifications regarding the prefix components for ICN structure including location, address, locator, and route. The authors conclude that the ICN naming and routing communication model fits the appropriate communication abstraction than that in the IP model. Adhatarao *et al.* [112] present a qualitative and quantitative comparison for both hierarchical and flat names. The authors use different metrics in their study, including name lookup efficiency, name aggregate feature, semantic of produced names, and namespace manageability. The study proves that the forwarding performance is directly impacted by the lookup time. Hence, flat name records a fast lookup, but the explosion of content in real-world scenarios can create a real dilemma since each content is identified with a unique name. On the other hand, the forwarding performance on top of hierarchical names is mainly affected by PIT and FIB's size. However, hierarchical names support the name aggregation feature, which reduces the size of the tables and the lookup time. Hierarchical names also improve multicasting that can send one packet to multi-users. Finally, the study concludes that hierarchical names achieve a higher lookup complexity than flat names since the lookup process requires parse and lookup for each name component to determine the next interface.

Name-based Routing Name-based Routing: The use of names to identify the content introduces a name based routing concept to discover and deliver the content to one or more requester without using any host identification [113]. When a consumer triggers a request asking for specific content. A discovery process will be started to find the content based on its name. The request packet is forwarded hop-by-hop using name-based rules to specify the next forwarding interface until reaching the original or a replica node that has the requested content, hence the content will be delivered back to the requester[111].

Figure 2.12 describes an overview of ICN content discovery and delivery mechanisms. The consumers $C1$, $C2$, and $C3$ send requests to get different content located at the producer P. First, the $C1$ starts requesting content (e.g., video) from P with interest name /P/Videos/V1.mp4. Then, the C2 sends an Interest to get content (e.g., image) using the Interest name /P/Images/Img1.jpg. Both Interest packets are routed towards the producer using the FIB table. Indeed, the Interest carries the same name-prefix, which allows for the aggregation at router $R5$ and, thus, reduces the network traffic. The $C3$ needs a document file by issuing an Interest with the name of /P/Documents/Book.pdf. The Interest traverses the network hop-by-hop until the content-store of the

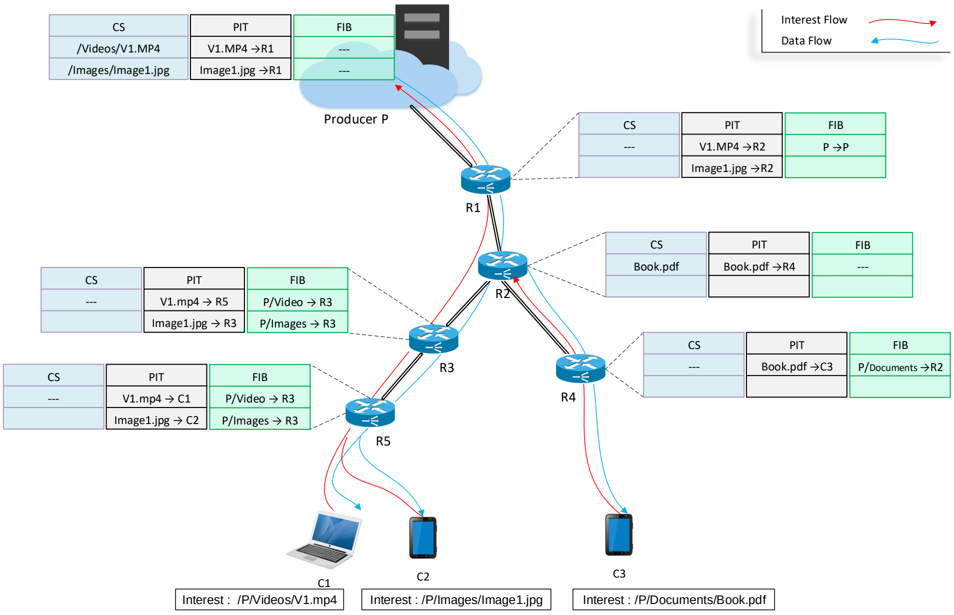


Figure 2.12: Content discovery and delivery in ICN.

router $R2$ and finds the desired content. Content caching reduces the distance to travel and hence reduces the delay.

In a nutshell, the name-prefix is responsible for driving Interest packets to content provider. Any changes in the naming scheme design can directly affect the lookup performance, forwarding, and network performance in general. In the following sections, we review the existing naming schemes by focusing on their aim objectives, Table 2.2 provides a summary of this discussion.

Table 2.2: Summary of ICN naming schemes.

Ref.	Type	Features	Comparison	Limitations	Year
Latency					
[114]	• Hybrid	• Short name length • Lookup time improvement	• Hierarchical • NDN-HNS [115]	• Not human-readable • Ignore nodes mobility	2020
[116]	• Hybrid	• Avoid communication loop • Support push and pull model	• Flat & Hierarchical	• Complex Interest forwarding process	2018
[117]	• Hybrid	• Introduce Query system	• ISC [118] • HFHN [116]	• Extra Interest processing time	2019
[119]	• Hierarchical	• Introduce naming encoding • Reduce lookup time	• NCT [120] • LNPM [121]	• Lack of push and pull model. • Impact of Interest decoding	2019
Scalability					
[122]	• Hierarchical	• Flexible naming scheme	• Hierarchical	• Content representativity • Semantic issue	2016
Caching					
[123]	• Hierarchical	• Caching improvement	• Focus on caching	• Discard naming scheme evaluation	2017
[124]	• Hierarchical	• Content distribution enhancement	• Focus on caching	• No evaluation	2017
Mobility					
[125]	• Hierarchical	• Flexible naming scheme • Producer's mobility-aware	• Hierarchical	• No evaluation	2014
[126]	• Hybrid	• Producer mobility support	• Hierarchical	• No evaluation • Additional signaling overhead	2018
Security					
[127]	• Hierarchical	• Security reinforcement	• CCVPN [128]	• Affect on mobility and latency	2019

Naming and Latency

During the Interest forwarding process, each node checks whether it has the name-prefix on its PIT and FIB tables. This verification mechanism is called the lookup process. The required time to perform lookup directly impacts the Interest delivery, which increases the network delay due to the verification time. In other words, design an appropriate naming scheme can reduce the lookup time, and hence reduce the network delay and the overall latency. Nour *et al.* [114, 129] design a Multilayer Multi-component Hierarchical AttributeValue naming scheme that combines prefix-labeling with variable-length encoding methods. The authors use the Decimal Classification and Fibonacci encoding schemes to identify the hierarchical location of the content. Besides, they incorporate different attributes in each naming level with a set of properties. The proposed scheme consumes less memory and provides a short lookup time and routing process. However, the obtained names are not friendly and use only numbers which are not human-readable.

Arshad *et al.* [116] propose a hybrid naming scheme that combines both hierarchical and flat structure. The overall goal is to reduce the lookup time by minimizing the name length. The scheme also tends to support the Push and Pull communication models for specific uses cases. The hierarchical naming structure is used to describe the domain name, location, and task. The flat name describes both device-name and data. The authors also design specific algorithms for both Interest and Data packets forwarding based on hash functions. The results prove the efficiency of the proposed scheme in different uses cases (mobile and static nodes) in terms of loop avoidance and improve the latency and the Interest aggregation while reducing the lookup delay. However, when an Interest packet reaches a node, the forwarding process time may raise extra upstream delay. Therefore, it affects content retrieval time and QoS. Similarly, Rehman *et al.* [117] propose a hybrid naming scheme based on hierarchical and flat structure. The hierarchical prefix is mainly used to route the Interest and Data packets. The core idea is to use a query system based on flat names. The authors use a hash function and define specific keywords to interact with smart building scenario. The proposed scheme may fill the gap of smart interaction between different devices under different requirements and communication models. The simulation results prove the efficiency of the proposed scheme in terms of energy consumption, average delay, and the number of packets processed per node. However, the applicability of a query system as a naming scheme in general usage is an open question. Khelifi *et al.* [130] focus on the name lookup process by designing a Name-to-Hash Encoding scheme. The authors tend to reduce the length of the hierarchical name by encoding each component separately into a fixed hash using the CRC32 algorithm and hence reducing the name length. The lookup is performed using heuristic name matching based on the WuManber algorithm [131]. Their solution's performance is based on CRC32 (e.g., encoding, collision, etc.) and the heuristic matching process of the Wu-Manber algorithm. Although the scheme reduces the size of the name, it does not support producer mobility. Moreover, the upstream Interest decoding process may affect retrieval delay.

Naming and Scalability the original content producer or an entity that controls names generates the application namespace. Any changes or updates in the application namespace can directly affect the performance of content delivery. Jung *et al.* [122] suggest adding an alias name to the application namespace in NDN to enhance the namespace diversity and management, such as updating, modification, joining, and adding. Since the content name is strongly related to the routing in the hierarchical prefix, any management or update operation in application namespace can affect the function of the routing/forwarding. The alias name has a global view of the application's namespaces. When a consumer requests content, the alias name is used as a translator to the corresponding namespace. The proposed scheme eliminates the barrier in the NDN routing, which can appear in multi-producer and multi-routing for the same content in the specific application. At the same time, the namespace management becomes more natural under the alias name controlling, hence improving the application scalability and content retrieval. However, the main problem with this alias is that it is not representative of the content and may not carry any semantics.

Naming and Content Caching The content naming in ICN is very flexible, and the design and convention of its semantic return to the application developer to choose what is suitable for its usage. A name does not include only features related to routing and mobility support, but it can also be used to enhance the caching capabilities. Shan *et al.* [123] propose extending the naming scheme to comply with caching schemes. The authors propose to extract a group of preferences related to the content name from the Interest packet. The content item is modified to contain the content's format attribute (e.g., text, video, photo, etc.) and the theme attribute (e.g., news, sport, etc.). The edge node is responsible for extracting the content's name attributes and calculating the matching degree for all received Interests. Then, it makes the caching decision based on the extracted information as well as content popularity. However, and from the naming perspective, some of the content may not fall into a common theme that has not been taken into consideration in this work. Moreover, regardless of the increase of content over time, such classification may raise computation overhead at the edge nodes. Kamiyama *et al.* [124] apply a hash function to the content name in order to enhance the content diversity in the network. The authors use different naming features to avoid content redundancy in the network's content-store. The core idea is to build a matching between the hash value of the content name, and the candidate node name that wants to cache the content. The mechanism is mainly based on the hashing of the name, in which each node enables caching assigned with binary ID. Therefore, the content name is hashed using a binary function. However, the authors did not explain the hashing process, such as hashing collision and the effect on routing/forwarding.

Naming and Security and Privacy ICN adopts a content-based security concept by applying security mechanisms to the content itself, and using self certification, such a technique can improve the security in the network, however, using plain-text and knowing the semantics of the name may pose different issues. Jung *et al.* [127, 132] propose a naming scheme for NDN architecture in order to improve the security of content naming in sensitive applications and use cases such as military applications. The main idea is that the requester uses a symmetric key to first encrypt a part of the content name in the Interest packet. This allows a secure information exchange (the Interest and Data packets) only between trusted nodes that are in the path while hiding the real content name. A three-way handshake mechanism is used to transfer the key between the intermediate nodes in order to decrypt the real content name. The comparison with related schemes records a higher interest satisfaction since the proposed approach supports the in-network caching feature. However, such security reinforcement may affect latency and mobility.

The performance of any network is mainly based on the performance of naming/addressing lookup. For instance, a DNS service lookup is the first step used to resolve an URL to an IP address. However, DNS is only used with address resolution and does not support additional functionalities (e.g., content placement and movement). In contrast, ICN expanses the naming functionalities in order to promote different additional functions and semantics (e.g., mobility support, security, caching, etc.). Indeed, a few efforts have been made in this area as different aspects were not considered in case of designing a new naming application structure. Meanwhile, some field has not yet explored, such as computation based naming.

2.6.2 In-network Data Caching

In 5G-based ICN networks, as illustrated in Figure 2.13, the network is aware of forwarding the Interest towards any content-store that has the requested content [133]. If the content is not cached at the core network, the Interest will be forwarded upstream to the original provider or cloud. During the content delivery phase, each node has the ability to cache the content, and serve it for future demands in a fully transparent way aiming at improving the quality of service [133].

In the case of the cache-store runs out of memory, the node evaluates the worthless content in order to keep room for the newly arrived content, this process called the content replacement policy. However, in the literature, these topics have less attention from the research community. The most used content replacement policies are First In First Out (FIFO): that evacuates the first entered content to keeping the newcomer one, Least Recently Used (LRU): that attempts to remove the less recently used content, and Least Frequency Used (LFU): that removes the content that has a smaller frequency rate.

The content cache placement is an active research topic where various research efforts attempt to provide seamless and efficient schemes. In this context, we differentiate two main concepts as illustrated in Figure 2.14: (i) *Vertical Management* tends to select the optimal content cache placement to satisfy a well-defined application scenario (e.g., video streaming), and (ii) *Horizontal Management* tends to push the popular content towards the edge network to satisfy as many users as possible. In this section, we provide a systematic review of the existing cache placement schemes in the context of the 5G based

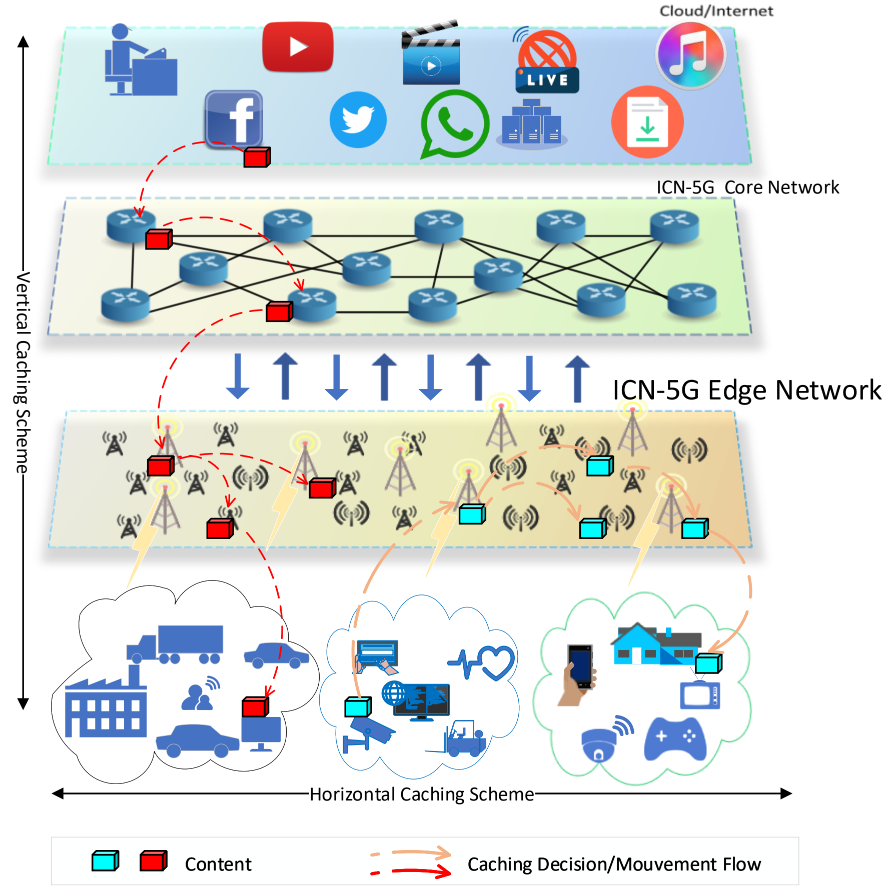


Figure 2.13: Content caching in 5G-based ICN networks.

ICN networks. Table 2.3 provides a summary of main ICN cache schemes that will be used as a reference in most of the reviewed solutions later.

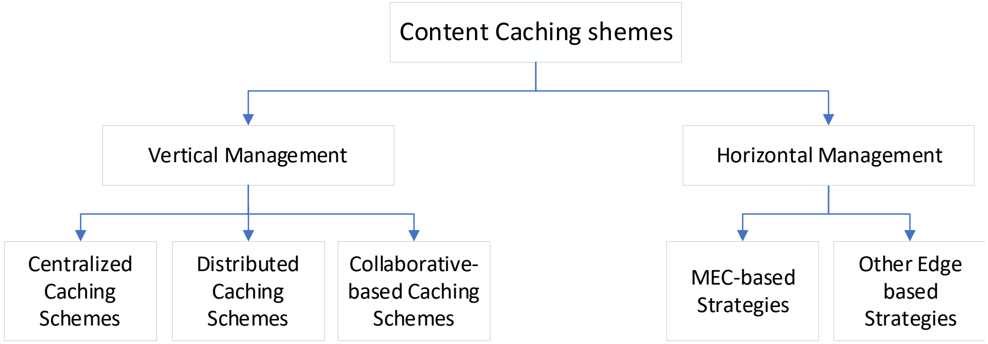


Figure 2.14: Taxonomy of existing content caching solutions.

Vertical Management In the following, we comprehensively review the existing content caching techniques that fit under the vertical management concept for both edge and core networks. Based on the controlled nature that influences the cache decisions, we classify the existing strategies into three main categories: centralized, distributed, and collaborative.Table 2.3 provides a summary of these techniques.

A. Centralized Caching Schemes: In centralized caching schemes, a controller entity or a server is used as an imperative network element to decide the cache placement. This entity is responsible for selecting a set of nodes in order to cache a replica version of the content. Centralized schemes provide absolute control over the number of copies of the content, which reduces the cache redundancy and enhances the network [93, 144]. For instance, Li*et al.* [145] propose a content cache strategy in order to avoid the miss-utilization of the storage capacity in ICN nodes by decreasing the cache redundancy in the network. The authors design a probabilistic approach to determine the matching between the content popularity and the node level. The node level is obtained by using different metrics such as the node's betweenness centrality, node distance, and the node cache capacity. Then, they apply a Grey algorithm [146] to ensure the correlation between the different node's parameters. The proposed scheme is designed to place the content in the appropriate place according to the content popularity and the node level value. Popular content has more probability of being cached at the edge nodes; meanwhile, less-popular content is cached in the core nodes. Although the proposed scheme achieves better efficiency compared to other strategies, the main drawback is the length of the path. In other words, with a larger path with multiple hops, the process will be more complex and require more computation overhead. Moreover, the replacement policy at the node is not considered at all in this work.

Focusing on content distribution, Meng *et al.* [147] propose a cache strategy that aims at enhancing content distribution in the network while reducing data redundancy. In this strategy, nodes are extended with: (a) a statistical table to collect the frequency demands for each content, and (b) a caching mechanism that works as follows: it places all content at a centralized node along the delivery path. The centralized node caches the popular content while storing

Table 2.3: Main ICN content caching schemes.

Ref.	**Scheme**	**Working Principle**	**Feature**	**Limitations**	**Control**
[134]	Leave Copy Everywhere (LCE)	All routers cache all passed content	• Reduce server load • Improve retrieval delay • Simple implementation	• Rise cache utilization • High replacement operation • Low diversity ratio	Distributed
[135]	Leave Copy Down (LCD)	Moves the content one hop from the producer	• Reduce server load • Maintain cache utilization • Consider content popularity	• Rise the Interest trip • Increases the retrieval time	Distributed
[136]	Edge Caching(EC)	Only edge nodes cache content	• Decrease latency • Maintain cache utilization • Improve retrieval delay	• Ignore the core enable nodes • Introduce content redundancy	Distributed
[137]	Betweenness	Cache content with maximum betweenness centrality node	• Maintain cache resources • Enhance content delivery	• More overhead on the centralized nodes • Ignore content popularity	Distributed
[138]	PropCache	Decide the cache based on probability value (cache size and path traffic)	• Reduce caching redundancy • Efficient utilization of cache resources	• Ignore content popularity	Centralized
[139]	Prob(α)	Caches the content based on given probability model	• Reduce cache utilization • Reduce replacement operation • Efficient utilization of cache resources	• Ignore content popularity • May keep unpopular content	Centralized
[140]	WAVE	Collaboratively adjusts the content chunk then progressively push them toward the edge	• Consider the chunk level • Increases hit ratio • Consider content popularity	• Introduce inter-traffic overhead • Complex implementation • Rising replacement operation	Collaborative
[141]	Cache Capacity Aware Cache (CCAC)	Estimate the available cache capacity to cache new coming content	• Efficient utilization of cache resources • High cache hit ratio • Cache store awareness	• Ignore content popularity • Low diversity ratio	Distributed
[142]	HighEst Cost Item Caching (HECTIC)	Work at the chunk level, group them based on the chunk-utility then make the caching decision	• Reduce cache utilization • Increases hit ratio	• Algorithm complexity • Low diversity ratio	Centralized
[143]	Random	Cache the content in one or more node in the downstream	• Improve hit ratio • Enhance content delivery	• Unguaranteed efficiency • Ignore content popularity	Centralized

Table 2.4: Main ICN content caching schemes.

Ref.	Scheme	Working Principle	Feature	Limitations	Control
[134]	Leave Copy Everywhere (LCE)	All routers cache all passed content	• Reduce server load • Improve retrieval delay • Simple implementation	• Rise cache utilization • High replacement operation • Low diversity ratio	Distributed
[135]	Leave Copy Down (LCD)	Moves the content one hop from the producer	• Reduce server load • Maintain cache utilization • Consider content popularity	• Rise the Interest trip • Increases the retrieval time	Distributed
[136]	Edge Caching(EC)	Only edge nodes cache content	• Decrease latency • Maintain cache utilization • Improve retrieval delay	• Ignore the core enable nodes • Introduce content redundancy	Distributed
[137]	Betweenness	Cache content with maximum betweenness centrality node	• Maintain cache resources • Enhance content delivery	• More overhead on the centralized nodes • Ignore content popularity	Distributed
[138]	PropCache	Decide the cache based on probability value (cache size and path traffic)	• Reduce caching redundancy • Efficient utilization of cache resources	• Ignore content popularity	Centralized
[139]	Prob(α)	Caches the content based on given probability model	• Reduce cache utilization • Reduce replacement operation • Efficient utilization of cache resources	• Ignore content popularity • May keep unpopular content	Centralized
[140]	WAVE	Collaboratively adjusts the content chunk then progressively push them toward the edge	• Consider the chunk level • Increases hit ratio • Consider content popularity	• Introduce inter-traffic overhead • Complex implementation • Rising replacement operation	Collaborative
[141]	Cache Capacity Aware Cache (CCAC)	Estimate the available cache capacity to cache new coming content	• Efficient utilization of cache resources • High cache hit ratio • Cache store awareness	• Ignore content popularity • Low diversity ratio	Distributed
[142]	HighEst Cost Item Caching (HECTIC)	Work at the chunk level, group them based on the chunk-utility then make the caching decision	• Reduce cache utilization • Increases hit ratio	• Algorithm complexity • Low diversity ratio	Centralized
[143]	Random	Cache the content in one or more node in the downstream	• Improve hit ratio • Enhance content delivery	• Unguaranteed efficiency • Ignore content popularity	Centralized

a copy at the edge node. The popular content is known based on a threshold value (frequency demands). At the same time, less popular content is also kept at the edge node in order to improve the content's availability. The simulation results show the efficiency of the proposed scheme in terms of content diversity, cache hit ratio, and height hops reduction. However, this strategy requires additional information to be stored for each content, which may affect spacial complexity and management. Moreover, the replacement policy is not considered in this strategy.

Zheng *et al.* [148] propose a cache placement strategy, namely Betweenness and Edge Popularity (BEP) strategy, which is considered as an improvement of the traditional betweenness strategy [149]. The latter selects the node with a higher degree as the appropriate node to cache the content. This decision is justified as most requests pass through the node (e.g., common node in the path). On the other hand, BEP aims at reducing the load on the server by caching the content according to the node's importance and content popularity. The core idea is that the upstream edge nodes compute the content popularity and the node degree. The edge node computes the content popularity with statistic technique taking the old and the current demands of content into account. In the downstream, the BEP algorithm puts the contents according to the node metric and the content popularity. Thus, the popular content will be cached at the highest node degree. The simulation results show that BEP outperforms other strategies in terms of hit ratio and server load reduction. However, by caching the highest requested content at a high degree, more cache hit occurs, and extra overhead is generated, which may diminish the QoS. Moreover, the content replacement policy is not mentioned in this work, which is vital to reduce the overhead on the node. Nguyen *et al.* [150] propose a Progressive Popularity-Aware Caching Scheme (PPCS) in order to avoid the cache redundancy of the data cached at the network. PPCS works on the chunk level, which is based on the received demands of content in a period of time. It then decides whether the content is popular or unpopular compared to their access rate with a defined threshold value. For popular content, PPCS caches the first chunk at the edge node then progressively continues to put the rest of chunks at the upstream according to the node level. When the access rate of a chunk at the edge increases over time, PPCS progressively regroups all the chunks of the requested content at the same edge. Besides, the related replacement policy in PPCS tries to remove the least served content obtained from the size of the content over the received demands. The simulation results show that PPCS reduces the load on the server, and outperforms the other strategies in term of cache utilization and hops ratio. However, in the simulation testbed, the popularity of the content follows Zipf distribution [151]. Zipf represents the access demands for the web content; it is mentioned that the chunk level can follow the Zipf distribution except that the chunks delivered in the sequential order [152], which is not the case for all content in real-world networks. Moreover, the exchanged messages between nodes may create extra overhead and congestion.

B. Distributed Caching Schemes: In a distributed caching scheme, there is no centralized entity to control the caching decision. Nodes use local factors and metrics to decide the content caching, which improves network efficiency by reducing the exchanged information between nodes and/or centralized entities.

Ren *et al.* [153] propose a distributed caching strategy, namely Distributed MAx-Gain (MAGIC). MAGIC aims at reducing bandwidth consumption and avoiding unnecessary content replacement. In this scheme, both Interest and Data packets are extended with gain attributes. During interest forwarding, each node computes the local gain value and then updates the value in the Interest packets only if it is bigger than the existing value in the Interest itself. The local gain is obtained by involving existing local information, such as the demands rate, hop reduction, replacement penalty, etc. In the data delivery process, when a node receives a data packet, it caches the content only if the local gain equals to the maximum gain in the Data packet. The simulation results show that MAGIC has more improvements compared to related schemes in terms of hops reduction ratio, cache utilization, and caching operation. However, the design of this scheme is limited to one replica node selection along the delivery path, which is not recommended in large-scale networks with high content's diversity, besides some content must be replicated to meet the massive demands from users. Banerjee *et al.* [154] design a greedy cache placement algorithm in order to decrease content delivery latency. The designed strategy runs in a distributed way where each node caches the content based on its popularity. The greedy algorithm constructs a directed acyclic graph by leveraging information from the nodes' routing table. Then, it considers the missed demands on the upstream for each router and extracts the content popularity by computing the missed demands (along the communication path towards the producer). The authors compare their algorithm with other strategies in diverse network topologies and multiple scenarios, including multi producers. The obtained results show the efficiency of the proposed strategy in terms of cache utilization, cache hit ratio, and hop reduction. However, the latency of the lookup of content in the nodes is not considered. Moreover, the generated graph may take more computational power, especially in the case where the content is far away from the consumers, which may impact the speed of cache decision making.

Similarly, Wu *et al.* [155] design a distributed probabilistic caching strategy in ICN that aims at enhancing the efficiency of in-network content caching. The authors integrate local information to decide the content caching. Each node in the network takes individual decision to either cache the content or not using a probabilistic formula. The probabilistic formula combines the content popularity and the benefit metric, and generates a probabilistic value. The benefit metric refers to the hop reduction between the requester and the cache store's node or the original server. The authors also propose a content popularity prediction technique that estimates future content's popularity based on the stored historical demands locally at the node. The authors validate their solution in a testbed using diverse network topologies. Results show the proposed solution is better than other strategies by caching popular content. However, the popularity prediction technique is based only on local popularity without considering the network perspective. This may influence the replication decision in other nodes for similar popular content. Moreover, storing historical information (for popularity) for each content in the node excessively affects memory usage and overhead the node.

Kamiyama *et al.* [156, 124] propose a distributed caching strategy, namely Spatially Dispersed Caching (SDC), where each node makes an autonomous cache decision. SDC aims at avoiding content duplication among nodes in the same area and caching only the highest popular content. SDC works in four main steps: (1) each router is assigned a single binary ID of - bit, while nearby routers have different values in the higher bit; (2) setting a hash value for each content; (3) if the hash value of the content matches with the router ID, this content is acceptable to the next phase; (4) based on the content popularity the router classifies the accepted content into multiple classes and then caches the popular one. By applying all these steps, SDC aims at guaranteeing the content's spread over the network. Although the idea is similar to [157], the key addition is in the caching decision. Besides, for each acceptable content, the SDC node classifies them into - , 2 groups based on the content demands rate, hence, the router differentiates the number of nodes that probability can cache the content based on group popularity then changing the number of bits checked if the node can host the content. The simulation results show that SDC marks a considerable improvement in terms of cache utilization and hop reduction compared with other strategies. However, in the case table of group popularity inputs raising over time, it dramatically affects the node memory as well as the time of the caching decision. The authors do not consider this limit that can degrade the effectiveness of the node retroactively in the real environment with massive content demands. Moreover, the question remains open on its efficiency in the case of multi-provider scenarios.

C. Collaborative-based Caching Schemes: Nodes in the communication path may build cooperation between each other and exchange information in order to explore more capabilities and better use the network resource. In the following, we review the existing caching schemes use this mechanism.

Micket *et al.* [158] propose a collaborative multi-hop caching strategy based on a bloom filter technique that aims at enhancing content diversity and latency while reducing intercommunication overhead between collaborative nodes. In this strategy, each node assembles the cache information between = hops reached nodes (neighborhood); then, the caching information is collected and stored in a set of attenuated bloomfilters. Whenever a request reaches a node, the node first checks the stored information regarding the neighborhood. If the content does not exist, it forwards the Interest towards the producer. In the meantime, to achieve content diversity and reduce content redundancy, the eviction policy uses the forwarding information base in order to avoid the caching content in near neighbors. The simulation results show the efficiency of the proposed scheme in terms of the cache hit ratio and latency. However, in large-scale networks, the probability of getting a false positive of the bloom filter technique is high, which may be considered the major drawback of such an approach. Moreover, in the simulation setup, the cache utilization is ignored. Huang *et al.* [159] design a collaborative cache strategy where nodes in the delivery path collaborate to maximize the use of the network-embedded caching. The authors focus on reducing the cost of traffic by considering both the received demands and the content size. They propose a payoff solution function Least Caching Utility (LCU), in order to solve the traffic minimization problem. LCU is used as a content replacement policy along the delivery path, which tends to evict the content with the least utility value taking the demand rates and distance between the requester and the original content provider into account. The node exchanges information about the content (Time Since Birth – TSB) by injecting it into the Data packet. This mechanism improves the collaboration mechanism between the node while reducing the extra overhead of the exchanged messages. Simulation results show benefits of the proposed solution in terms of low communication overhead (i.e., exchange messages between nodes). However, such a policy may introduce a high redundancy of content caching, especially in large-scale networks. Moreover, with the expectation of increasing the content in the network, checking the TSB for each passed data packet affects the router's processing resource and adds extra delay.

Zhi *et al.* [160] propose Gain-Aware 2-Round Cooperative (GAC) caching scheme to enhance the content diversity while reducing the data redundancy in the network. In GAC, the node decides the content caching based on the decisions made by other nodes along with the path delivery. GAC's core idea is that the upstream nodes compute the potential gain value and include it in the Interest packet with their node ID. The potential gain value takes the number of received demands and the number of hops reduction if the node caches the content into consideration. In the data delivery path, the node computes the final gain and takes the cache decision accordingly. If the gain value is higher than a pre-defined threshold, the node updates the hop-value to 0 in the Data packet. The hop-value describes the number of hops needed to get the content from the replica node or the original producer. The simulation results show the efficiency of the proposed scheme in terms of the cache hit ratio, hop reduction ratio, and cache utilization. However, since the cache decision must be line speed in ICN routers [139], the packet processing for each content may introduce a considerable effect on network delay, especially in the case of multi-producer and massive contents requesting. Moreover, adding the node's ID to each packet is not feasible and may need changes in the nature of ICN packet. Furthermore, defining an ID for each node is a challenging task that needs more investigation to ensure a unique global ID, it also takes the paradigm back to hostcentric networks.

Liu *et al.* [161] propose a collaborative Cache On Demand (COD) strategy. COD tends to reduce the number of nodes that cache the content in the network. COD operates based on the number of received demands and the temporal patterns of content popularity. According to the study in [162], the authors rely on the change of content popularity by applying two stages: (a) the early stage, when requesting content in the initiator until up to the peak in popularity; and (b) the late stage, the change and degradation of the popularity of the content. Depending on a probabilistic formula, the node chooses to cache the content or push it to the adjacent nodes with more cache size. COD aims to keep the content in the early state close to the producer, and over time, depending on the received demands, change the content to the late stage. The authors provide several measures to show the efficiency of their solution in terms of cache consumption, energy consumption, and content diversity. However, this work is based on the popularity of the content that needs to be known beforehand. The scheme is not applied to dynamic content or content generated based on demands. Besides, each node is obligated to compute the content state periodically, which produces computation overhead. Moreover, the user's mobility may create a major drawback of such a strategy since the demands are not stable. Noh *et al.* [161] propose a caching strategy for video streaming. The main objective is to provide a seamless video streaming service. Firstly, the authors design an algorithm to create the meta-file for each content that adapts with a multi-caching node environment in ICN. The meta-file regroups information about content chunks as the priority and the bandwidth requirement. In this strategy, each node selects its range of chunks to cache using the associated content meta-file, therefore avoid the redundant caching of the same chunk in the adjacent nodes. Nodes exchange information about cached content using modified Interest/Data packets. At the reception of this information, the proposed algorithm progressively adjusts the range according to chunks' priority hop number and demands rate. The obtained results show that the proposed solution provides seamless video streaming by reducing network stress. Yet, this efficiency is at the expense of the cache utilization, which not considers in this strategy. Moreover, creating meta-files and exchanging information may produce more overhead on the network, especially in profoundly changing content's popularity.

Horizontal Content Caching Management In this part, we explore the management techniques that take advantage of its location (closer to the consumer) to exploit the benefits of the powerful interaction between the Edge-enabled caching node and the consumers. Table 2.5 provides a summary of these techniques.

A. MEC-based Strategies: Several edge caching strategies have been designed that use MEC servers and exploit their robust computation to provide better content distribution at the edge network and enhance QoS for the end-users. In this part, we review the recent existing works based on MEC founded in the literature. For instance, Zhang *et al.* [152] propose an edge caching architecture for 5G networks that combines the caching and computing resources of MEC and ICN-based edge nodes, including a base station, small base station, and user nodes. The proposed architecture aims at enhancing the cache performances within the global edge network. The main idea is to achieve a tradeoff between computing cost and caching gains. In doing so, the authors use MEC computing resources in order to compress the content. This will improve the caching capability at the edge and, by consequence, improve the content availability. The simulation results show a significant improvement in both the cache hit and the round trip latency. However, the major drawback is the high complexity of scheduling of all resources simultaneously, especially when takes the nodes behaviors as mobility into consideration. Moreover, some nodes are resource-constraints (e.g., IoT devices), hence using them as content-stores may affect their performances while ignoring them may lead to overhead computation.

Huang *et al.* [170] propose an Online Learning Framework (OLCB) that aims to reduce the backhaul bandwidth consumption by enhancing the edge nodes' cache efficiency. The proposed strategy initially works to construct the context by capturing the user group access behavior, including connecting period and mobility trace. The Exploitation-Exploitation machine learning

Table 2.5: Summary of horizontal content caching strategies.

Ref.	Feature	Comparison	Evaluation	Limitation	Year
MEC Based Strategies					
[52]	• Improve caching capacity • Enhance content availability	• Cache without MEC	• Latency improvement • Cache hit rate enhancement	• Compression overhead • Computation expenses	2018
[163]	• Context integration • User-behavior aware	• PoPCache, Greedy Algorithm, MPC	• Cache hit improvement • Considerable running time	• Users' privacy • High complexity	2019
[164]	• Recommended system integration	• Random, LCE, SVD Based	• High retrieval delay • Cache hit Ratio enhancement	• Focus only on video contents • High computation overhead	2019
[165]	• User location aware • Consider users' mobility • Content prefetching	• FiFo, LRU, LFU	• Cache hit ratio improvement • Reduce access time	• Users' privacy	2019
[166]	• Enhance MEC resources • Improve user energy	/	/	• Ignore users' behaviors • Lack for the practical part	2019
[167]	• Improve video streaming service	• Non-MEC based	• Data retrieval improvement	• Ignore wireless packets loss	2019
Other Edge Based Strategies					
[123]	• User preferences aware • Leveragr content naming	• EC, Prob(α)	• Efficient cache hit ratio • Latency improvement	• Edge node overhead • Naming schemes not evaluated	2017
[168]	• Include social network knowledge	• Random caching	• Considerable caching enhancement	• Ignore users behavior • Extra computation overhead	2019
[169]	• Employ deep learning	• EC with different replacement strategies	• Cache hit ratio enhancement	• Learning process	2019

model is adopted to create a relationship between the requested content and their specific context under a distinct time period and estimate the reward of caching the content. The problem is modeled as a Knapsack problem, where each requested content is associated with a specific context that has a particular reward. The objective is to maximize the reward value under specific node storage memory. In another work, Zeng *et al.* [163] propose an algorithm to solve the knapsack problem leveraging branch and bound approach. A smart caching a heuristic algorithm is proposed that applies the mining on the central effect of consumer behavior in order to extract the consumer content preference in each time slot. Based on the user's content preference, the proposed scheme makes realtime learning-based caching decisions by predicting the variety of content popularity that can share content preferences under specific contexts. The authors simulate the scheme using a real dataset of user records to show the efficiency of the cache hit rate. However, the differences in demands for diverse content is questionable. The solution needs more investigation to deal with context management which needs more computation. Moreover, collecting data for learning purposes is challenging due to the users' privacy issues. Similarly, Tang *et al.* [165] design a cache strategy scheme for mobile multimedia devices, namely Edge IoT equipment-Assisted Caching Multimedia (EACM). EACM tends to improve the quality of experience. The overall idea is to cache the content at the edge node (e.g., SBS), taking the importance of the requested content of the users' group and the users' mobility into consideration. The proposed strategy benefits from the computation capability of the edge by using Recurrent Neural Network (RNN) to predict the next localization through learning the users' trace. At the same time, the FB-growth algorithm [171] is adopted to extract the common users' group interest. This coordination ensures to pre-fetching the common content from the remote server and guarantees their distribution along with the edge nodes, according to the users' needs in a specific area. Furthermore, a hybrid LRU and LFU policy are designed to combine and manage respective features, both of them in the double queueing and handle cache-store decisions. The simulation results show that EACM diminishes the load on the server-side, while the replacement policy outperforms other policies, reduces the access time, and optimizes the cache hit ratio. However, such an approach may create redundancy for the popular content in the edge nodes, which is not taken into account. Moreover, re-fetching the most requested content for each area may create overhead demands on the remote server for the same content, which increases the bandwidth consumption. Finally, any changes in the network setting can affect directly on RNN model's precision [164].

Tang *et al.* [166] tend to enhance the memory of the MEC using the ICN paradigm. In doing so, the authors assume that each edge node has specific caching resources. The objective is selecting the nearest BS caching node to the MEC. The chosen node is based on consumer proximity to the base station, while the global objective is diminishing the user side's energy consumption. However, the users' behavior (e.g., mobility) has not been considered. Moreover, the proposed solution needs to trade-off between the content popularity and the memory resource selected by the MEC, which needs more investigation towards real implementation since the authors present only the theory part. Han *et al.* [167] aim at improving video streaming delivery in the MEC based ICN paradigm by enhancing the Forward Error Correction (FEC) mechanism of ICN architecture. The authors propose a framework running on top of MEC platform, in which FEC generates the packet loos at the network edge instead of the remote server. A deep learning algorithm is then used to learn the received information from FEC and use the FEC rate to monitor network states (by deciding whether to correct the packet error or use the re-transmission). Compared with conventional ICN (non-MEC based), the simulation results prove the efficiency of the proposed. However, this framework treats only the error transmission between MEC and the original producer, where some packets may get lost due to the interference.

B. Non MEC Based Strategies: In this section, we present an edge caching solution that not consider the MEC server as a single caching management entity. These works assume that each edge node (BS, SBS, User Equipment, etc.) can use the computation and decision.

Shan *et al.* [123] propose a cache strategy based on the users' group interest preference. The proposed strategy aims at reducing the network delay by caching the most preferred contents in the edge node. The authors use a classification model for the contents under the assumption that the content name contains two attributes theme (e.g., sport, news) and a type (e.g., text, video). The main idea is to ensure the matching between the preferred interest and response data. The edge node extracts the user's group interest from users' requests. When data reaches the edge, it decides whether to cache the content or not according to defined threshold judgment. The simulation results show the efficiency of the proposed strategy in terms of cache utilization and hop reduction. However, this strategy uses only the edge node as the main node to compute, store, and deliver content, which may lead, as a consequence, to overhead and degrade the QoE, especially in the case of highly mobile users. Moreover, due to the less storage memory at the edge node, the computation and communication may congest the backhaul links. Liu *et al.* [168] propose an analytical framework based on the ICN edge caching scheme. The proposed framework aims at putting the commonly requested content at a nearby location of the users. In the first stage, the framework records the user check-in to get the connected edge node. The authors then leverage a location-based social network technique that considers: the distance between users, the user interest, and the contact rate. Since the problem is an NP-hard, a metaheuristic simulated annealing algorithm [172] has been adopted to get near optimality solution. Compared with other solutions, the proposed framework shows high performance in terms of content propagation ratio. However, this framework needs extra information about users. Moreover, this work considers only the users' common interests and ignore users' behaviors. Wang *et al.* [169] propose a caching scheme that tends to be adapted to the network environment. In doing so, a deep reinforcement learning algorithm has been presented based on an asynchronous actor-critic agent mechanism. A learning agent is evolved (using the reward or punishment) during interactions with the network environment. Hence, the framework decides whether to keep the requested content at the edge node or not. It also uses the substitution strategy in case the cache is full. The decision is linked to learned network status. The simulation results show the high performance of the proposed strategy, notably in terms of improving the cache hit ratio. However, the results are strongly related to the capacity of the user data in the learning process. Therefore, the relearning of the model at each network may appear as a challenge.

Horizontal-based caching strategies at the edge are more resilient and flexible as the cache hit is near to users. The content's usefulness is strongly associated with how close it is to the consumer. Therefore, bring content closer to the users comes with many profits, such as reducing retrieval time, decreasing the backhaul traffic, etc. Exploiting the necessary methods to achieve this purpose is the aim of most horizontal-based approaches. Their decisions have a strong dependency on the correlation between the consumer/interests and its behavior. Using advanced techniques to achieve a better user/behavior adaptation may be paved the road to the intelligent edge paradigm. The machine learning approaches are taken place near to the consumer at the network edge and on the top of a powerful computation platform as MEC [173]. However, several issues raised, including the choice of the appropriate model, the learning process, and content privacy.

2.7 ICN Mobility Support

The fifth-generation (5G) network is expected to support significantly a huge number of wireless connections and a large amount of mobile data traffic. 5G solutions should support mobility on demand for many scenarios, ranging from very high mobility to low mobility or stationary devices. With the manned increase in the number of mobile devices, many efforts are looking to provide seamless mobility management within the IP paradigm. All of these efforts fall into high access rates due to there complexity. Which prompted the search for a radical change of the communication manner. ICN provides a set of features such as abstraction of content from the location through names [62, 19], consumer mobility support [20], in-network content caching, and content-based security [21]. The content in ICN is decoupled from its original provider by using a content name. The name is the basic communication element used by the consumer. From the mobility perspective, two main questions occur:

- *Consumer Mobility:* the consumer sends an Interest packet to fetch content and then moves to another network or access point. The content is sent back to the previous location of the consumer but not successfully delivered.
- *Producer Mobility:* the content provider moves from a network to another one. The issue is the network cannot locate the new location of the provider and hence cannot deliver the request.

For ICN architectures that use flat names, the node's mobility performance is directly related to the used resolution service mechanism. Thus, it worth noting that ICN, in such architectures, can handle more than 10^{12} billion content in the near future [174]. Indeed, it can be seen as a drawback since the nodes' mobility management in such a design is directly related to how swift to handle the object at the resolution services [99, 175]. On the other side, consumer mobility may not impede ICN design that uses hierarchical naming. For instance, NDN and CCN support consumer mobility by design but cannot handle producer mobility issues. Wang *et al.* [176] show that CCN can handle up to 97% of the requests during the consumer's high mobility. Unfortunately, leveraging a seamless consumers' mobility feature increases the producer mobility drawbacks. For this reason, researchers are trying to find cost-less methods to make additional support for producer mobility. The desirable method should neither be complicated nor require fundamental changes in the naming prefix or the working principle.Adhatarao *et al.* [177] propose a Network Mobility in ICN (NeMoL) approach. The proposed focuses to reduce the signaling overhead at each update mobility (for both producers and consumers) especially in the case of the network on the move. The main idea is assigning a single namespace to the target moving network in which each prefix is aggregated in that namespace. A distributed name resolution agents exchange the routing information about the current attached point of each namespace in order to updates the routing path towards the target network at each node in the rest networks' nodes. Sivaraman *et al.* [178] propose a hop count based forwarding strategy to support producer's mobility in NDN. The authors suggest injecting the number of hops traversed by the interest in the interest packet. The core idea is whenever the interest reaches a router, the router decides either to forward it toward the matched interface if it exists or broadcast the interest to all available interfaces. The results show the efficiency of the proposed solution in terms of network throughput and interests succeeded. However, the proposed solution requires extra resources. The broadcast forwarding can occur an extra overhead, especially in the case of producer handover between different networks due to the heterogeneity. Azgin *et al.* [125] propose a mobility service targeting hierarchical naming in ICN. The solution aims at reducing the time of content retrieval from a mobile producer. The overall idea is to provide a service that allows any entries in the hierarchical naming to be mobile. The proposed system consists of many principal blocks that interact with each other in order to reduce the Interest Flooding and redirect the Interest packet towards the current mobile producer attached point. In addition to the high complexity of their working principle, such an approach requires additional material resources (servers), which increases the deployment cost.

Zhang *et al.* [126] introduce trace-based mobility support for NDN, namely KITE. KITE tends to maintain the structure's ease of the NDN paradigm, where the mobile producer acts as a consumer for a fixed rendezvous server. At each location/network changing, the mobile producer sends an interest to the server in order to keep the trace of its new location. The server is responsible for updating the forwarding rules in the FIB table of intermediate nodes in the network. In doing so, the authors suggest injecting a trace attribute in the routing prefix's name to the content generated by the target mobile producer. For the consumer's side, if it finds the trace of the mobile producer during Interest forwarding, it follows the trace to get the content. The simulation results prove the efficiency of KITE in ensuring seamless producer mobility. However, the architecture suffers from the overhead signaling by the producer, which may create a network disaffection. Moreover, time-sensitive content may not fit well with the proposed solution. Li *et al.* [179] apply a neural network machine learning to solve producer mobility issues. In order to predict the next access point, the learning model holds the record of the historical movement of the mobile entity. The intermediate router modifies the Interest packet in the FIB table to route it towards the predicted producer location. However, due to the privacy and security of the mobile node information, this approach may not be applied in real-world networks. Besides, the main drawback of the neural network is that it is strongly related to the primary parameters, any change in the parameter means a change in the results of the performance of the solution. Korla *et al.* [179] propose to exploit the caching capability in network nodes in order to improve producer mobility. The overall idea is to keep the popular content at the edge node before the producer handover occurs purpose to reduce the incoming Interest loss. However, such an approach may not be suitable for real-time content delivery since it ignores unpopular content. Moreover, the efficiency of the proposed solution may directly be affected by a decrease in cache size in some nodes due to the redundancy of content. Auge *et al.* [180] propose map-me mobility support for ICN that tends to skip the delay of FIBs update in ICN due to the producer's movement. Map-me consists of a pack of protocol purposes at a dynamic update of the FIBs. When the mobile producer changes the attachment from one point to a new one, Map-me tries to update the FIBs along the path between the previous and the new attachment point. Although the simulation results prove the efficiency of the proposed, it brings additional protocols on the top of ICN architecture that may reduce its effectiveness. Hence, can draw us back to the actual IP complexity.

User mobility is an essential aspect of the next 5G networks, especially for time-sensitive applications. Although ICN has native support for user mobility, which may provide a seamless user movement, the frequency change of the user attachment point between multiple access technologies may impact the QoE as well as QoS, which leads to service degradation. Furthermore, the main drawback appears if the same user act as a producer (that produces a certain content). In this case, the routing based naming process uses a flooding Interest for path recovery. Thus, it increases the network traffic, which leads to the network conjunction. In this regard, various works have been done concerning producer mobility issue. However, none of them uses a flexible naming scheme that supports multiple features, e.g., mobility support, security enhancement, scalability, etc.

2.8 Conclusion

In this chapter, we have defined the building concepts used in this thesis, mainly, 5G network and ICN architecture. We provided a comprehensive survey of different enabling technologies that come into this domain. We have also presented different pieces related to the 5G networks. Then, we argued the necessity and feasibility of the ICN paradigm within the 5G network. We also provided an in-depth overview of the basic component of the ICN architecture, including the naming efforts and the caching strategies. We reviewed recent and essential solutions that have been classified into different categories based on their working principle. Finally, we have also detailed recent solutions and architecture related to ICN mobility support and reviewed deeply their working mechanism. We believe that ICN will provide an open view of the 5G implementation by helping to realize the 5G missions and visions. In the following parts, we will present our contributions within in network caching management, naming, and mobility handling.

Chapter 3

A Cache and Split Content Caching Scheme

3.1 Introduction

Most of the current Internet use cases are now shifting toward the content centric paradigm, where the data content is the main element in the infrastructure. The 5G-based ICN network will bring new opportunities and vision based on a world built on fast communication. In this chapter, we design a cache and split content caching scheme for the future 5G based ICN network. The proposed strategy target distributed content placement problems for the next 5G network and compared with the existing works it provides a set of builtin features such as fair content distribution, efficient resource management, and reduce delivery delay.

This chapter is organized as follows; we first highlight the requirements and challenges for content cache placement in ICN. Then, we present the system model and formulate the placement problem. Then, we detail the proposed scheme. Finally, we present the implementation and evaluation part.

3.2 Caching Requirements and Challenges

In 5G-based ICN networks, each cache store node contains a limited storage capacity. The concept of content caching placement consists of choosing the optimal place to cache the content. A set of nodes is selected to cache content in order to bring it closer to consumers and hence improve the overall network performance. In order to design an efficient cache placement plane in 5G based ICN network, various issues and challenges need to be addressed. We classify these challenges into three main aspects:

- **Optimality Challenge:** 5G-ICN provides a transparent in-network cache feature leading to enhance the overall network performance. Hence, data can be placed anywhere in the network, which raises a variety of options, for instance, (1) keeping the data close to the provider to reduce strain on the server; (2) or placing the content close to the consumers in order to improve the latency, and provide low delivery delay; (3) or even putting the data in the core network will reduce the backhaul consumption. In this regard, An efficient cache placement scheme should select optimal content placement in the network.
- **Scalability and Complexity Challenges:** in a widely distributed 5G-ICN network, that contains various connected devices, cache placement scalability is a challenging aspect. A suitable cache strategy should be address scalability and be distributed in nature. However, to fit the scalability issues, a suitable solution might be a self-caching decision for each node. Yet, due to the nodes' resource constraints, the caching decision process must be accomplished with low complexity overhead.
- **Content Replacement-related Challenges:** removing content for the content store to keep room for newly arrived content may lead to removing the popular one, in case of the cache-store is full, the replacement policy may lead to removing the popular content, and hence, all future demands should be fulfilled by the original producer. An efficient cache replacement policy must remove only the nonpopular content.

3.3 Cache and Split Scheme

In the following, we define the system model, formulate the cache placement problem, and describe the proposed scheme.

Table 3.1: CnS List of used notations.

Notation	*Description*
N	Set of nodes
A	Set of links
R	$R \subset N$ Set of cache-enabled routers
P	$P \subset N$ Set of content publishers
Q	$Q \subset N$ Set of consumers nodes
D	Set of contents
S_r	Cache memory size of node $r \in R$
S_d	Size of content $d \in D$
φ_d^r	Total number of demand for content d received by r
ρ_d	Popularity of the content d
$C_{q,r}$	Cost of content delivery between consumer q and router r
$H_{p,q}$	Number of hops between producer p and consumer q
$H_{r,q}$	Number of hops between router r and consumer q
x_d^r	0 - 1 Response to data demand : $x_d^r = 1$ if the content $d \in D$ is located in the cache store of router $r \in R$ and $x_d^r = 0$ otherwise

3.3.1 System Model and Problem Formulation

We represent a 5G-based ICN network as a graph $G = (N, A)$, where N is a set of nodes and A set of links. A node can be a consumer $q \in Q$, an ICN router $r \in R$ with caching capabilities (including edge routers and SBS), or a content publisher $p \in P$. Each cache-enabled node $r \in R$ has a limited cache memory S_r. We assume that all content chunks have the same size of S_d. Content request issued by consumer $d \in D$ may be satisfied by a cache router $x_d^r = 1$. Let φ_d^r denote the total number of demands for content d received by the router r for a period of time, and ρ_d denote the content popularity. Table 3.1 summaries the most used notations in the paper.

3.3.2 Problem Formulation

In this work, our content caching problem is defined as a problem of dual-objective optimization. These objectives are detailed as follows.

OBJECTIVE 1. Maximize the Cache Utilization: In order to maximize the cache utilization, we need to maximize the content diversity in the network. This can be done by caching as much as possible of different contents, in other words, minimizing the number of duplicated contents in the overall network. Eq. 3.1 expresses this objectives by minimizing the number of involved routers to cache the same content.

$$\min \sum_{r \in R} x_d^r, \quad \forall d \in D \tag{3.1}$$

OBJECTIVE 2. Minimize the End-to-End Delay: In order to improve content retrieval, we need to place the content at a common point to server maximum number of demands and minimize the end-to-end delay between the consumer(s) and the router that stored the requested content. Eq. 3.2 expresses this objective by selecting the common routers with the minimum end-to-end network delay.

$$\min \sum_{d \in D} x_d^r \, . \, C_{q,r}, \quad \forall r \in R, q \in Q \tag{3.2}$$

Where $C_{q,r}$ denote the cost of content delivery (e.g., end-to-end network delay) between q and r.

GLOBAL OBJECTIVE FUNCTION: The global objective function can be written as shown in Eq. 3.3.

$$\min \left(\sum_{r \in R} x_d^r + \sum_{d \in D} x_d^r \, . \, C_{q,r} \right) \tag{3.3}$$

subject to:

$$\sum_{d \in D} x_d^r \, S_d \leq S_r, \quad \forall r \in R \tag{3.4}$$

$$x_d^r \in \{1,0\}, \quad \forall d \in D, r \in R \tag{3.5}$$

Constraint (3.4) imposes that the overall size of cached contents should not exceed the maximum cache size of the router. While constraint (3.5) describes the possible values of the variable decision x_d^r.

3.3.3 Proposed Scheme

The overall idea of **C**ache a**n**d **S**plit (CnS) scheme consists of two main steps. The first step aims at calculating the content popularity and determining if its worth to be cached (cache-able content) or not. The second step is executed only if the content is cache-able, hereby we design three cache place actions during the whole process:(a) cache the content at local cache-store, (b) push the content down, or (c) split the content and push it down to selected neighbors. All the three options contribute to reducing the content redundancy in the network, moving the content as closely as possible to consumers, and improving the network performance. Figure 3.1 illustrates the big picture of CnS concept in 5G-based ICN networks.

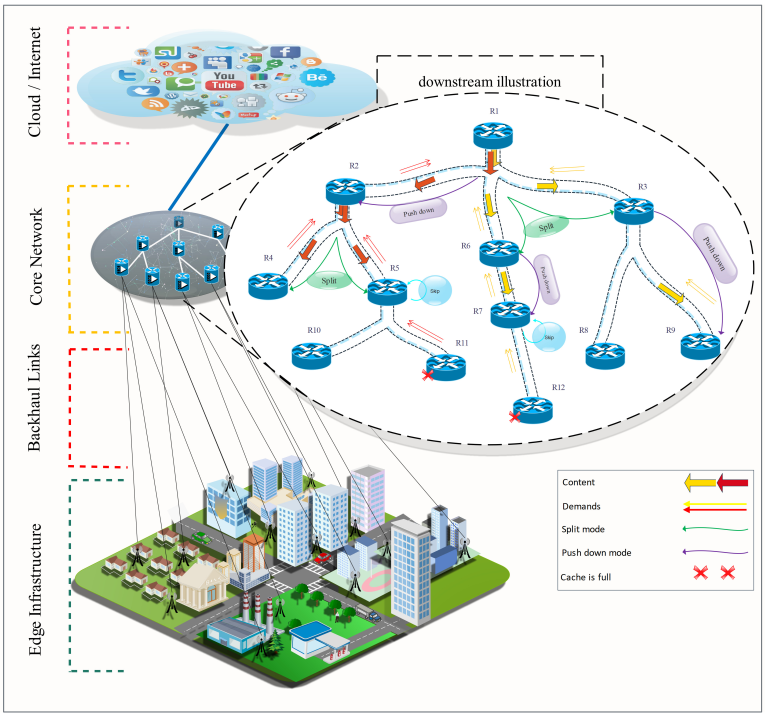


Figure 3.1: Virtual Network Topology based on CnS Strategy.

It is worth nothing to highlight that the term *split*, in this context, does not mean splitting the content into chunks. It rather means splitting the cache responsibility to more than one cache-store.

3.3.4 Content Popularity

In the first step, the caching plane is required to calculate the content popularity factor (ρ_d). In this work, we use three main parameters to calculate the content popularity:

- Content Freshness (f): Since the content popularity changes over time and during the content lifetime, we combine the content published time t_p and current time of requesting/consuming content t_n (both in hours) to express the content freshness in the interval $[0-1]$, as shown in Eq. 3.6.

$$f = \frac{1}{|t_n - t_p|} \tag{3.6}$$

- Distance Factor (η): The diversity of contents in real-world scenarios is one of the major challenges for content caching, especially for equal popularity contents. CnS distributes the content based on the number of demands. Hence, the position of the content in the network is an important factor (e.g., popular content should be cached at the edge, less-popular at the core network). We measure the distance based on the producer, cache-store, and requesters' position, as shown in Eq. 3.7.

  $$\eta = \frac{H_{p,q} - H_{r,q} - 1}{H_{p,q} - 1} \tag{3.7}$$

  where $H_{p,q}$ denotes the number of hops from the original producer p to the farthest consumer q, and $H_{r,q}$ denotes the number of hops from the cache-store r to the farthest consumer q.

- Access Frequency Rate (χ): Satisfying demands from the cache-store means that the content is placed at a desirable place (even if not the optimal one). The access frequency rate aims at measuring how many times demand has been fulfilled from the cache-store, and thus deciding to push it down towards consumers to reach an optimal place. Eq 3.8 expresses how χ is calculated.

  $$\chi = \frac{\varphi_d^r}{\varphi^r} \tag{3.8}$$

  where φ_d^r denotes the total number of demands received for the content d in the cache-store r, and φ_r denotes all demands received on node r.

The popularity factor is defined by $\rho_d = \alpha f + \beta\eta + \gamma\chi$, where α, β, and γ are weight parameters used to tweak the model based on the network needs, in which their sum equals to 1.

3.3.5 Cache Decision Process

It is worth noting that content placement is an NP-Hard problem. Hence, we tend to design a cache placement scheme that achieves better results with less cost. In this process, we note two points of view: Theoretical and Practical. Each point of view is illustrated respectively in Figure 3.1 and Figure 3.2.

Theoretical Point of View Figure 3.1 depicts the big picture of CnS strategy. ICN demands are spread upstream searching for the desired content from the requester side to the core network and up to the cloud/Internet level. When the content if found in a cache-store or the original content producer, a data packet is forwarded downstream towards the requester. During the data delivery path, the first intermediate node (i.e., R1) who receives the content calculates the content popularity (ρ_d) and checks if the content is cache-able (depending on the popularity). Based on ρ_d, the intermediate node performs one of the following operations:

- Push the content down towards the downstream forwarder only if the current router has one forwarding interface where the demand has been received (e.g., R1 → R2).
- Split the cache decision to all forwarding interfaces (downstream) in case the demand has been received from multiple interfaces. The current router is involved to decide the most desirable interface(s) to push the content down based on the number of received demands (e.g., R1 → R6 and R1 → R3).
- Cache the content in the local cache-store in case all downstream neighbors are full (e.g., R5 and R7).

This process is executed in all intermediate routers upon the reception of each content downstream. Doing so, the popular contents will be pushed down towards the edge routers, while less-popular content will remain in the core network.

Practical Point of View Figure 3.2 depicts the overall CnS cache plane process. Here, we have three main modules: (i) *Cache Placement*: to decide the content caching, (ii) *Push & Split*: to decide if the content should be pushed to one neighbor downstream or split the cache decision to a list of neighbors, and (iii) *Cache Replacement*: to decide which content should be evicted from the cache-store. The overall CnS process is summarized in Algorithm 1 and detailed as follows.

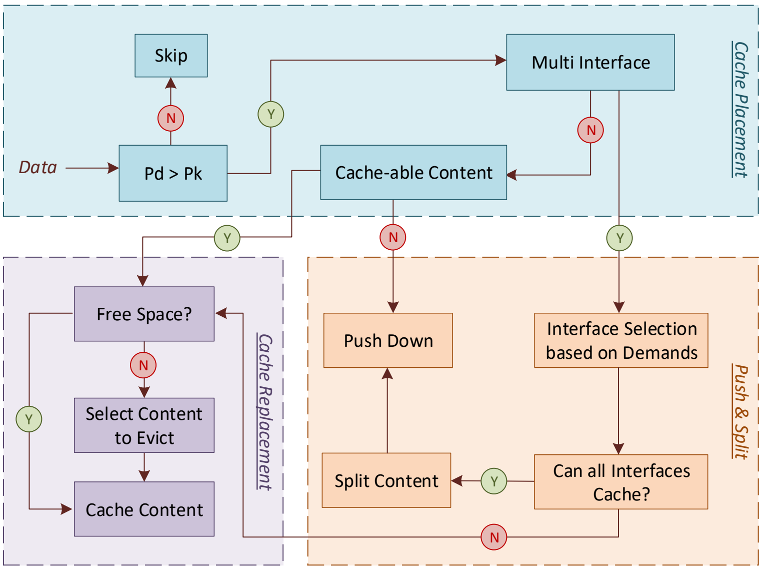


Figure 3.2: CnS Cache Plane.

Algorithm 1: Cache and Split Algorithm

```
Input: $d$: data packet.
Output: Output
calculate $\rho_d$;
if ($\rho_d > \rho_k$) then
    if (neighbor is consumer) then
        local cache policy();
    end
    if (demands come from single interface) then
        if (neighbor can cache) then
            push data down();
        else
            local cache policy();
        end
    else
        select interfaces with highest demands();
        if (selected interface can cache) then
            split & push down();
        else
            local cache policy();
        end
    end
end
```

1. Cache Placement Decision: In the first step, we tend to decide the content caching based on its popularity, then decide if the content should be cached locally or pushed down towards the edge network.

2. Split & Push Decision: The second step tries to split the cache decision towards the downstream routers and determine the suitable neighbor nodes depending on the number of demands. In such a case, we differentiate two cases:

- Single Interface Case: in this case, the node is connected to one neighbor. This means placing the content at that neighbor cache-store is better than the current node. Hence, we push the content down and cache it,
- Multiple Interfaces Case: if there are many interfaces received the demand, we must select a set of neighbor cache-stores based on the number of received demands and split the cache decision over them.

3. Cache Replacement Decision: The step consists of selecting which content should be evicted from the cache-store to retain room for the arrived content. The new arrived content d_n should have a higher popularity value (merit to be cached) and also its popularity must be higher than the evicted content.

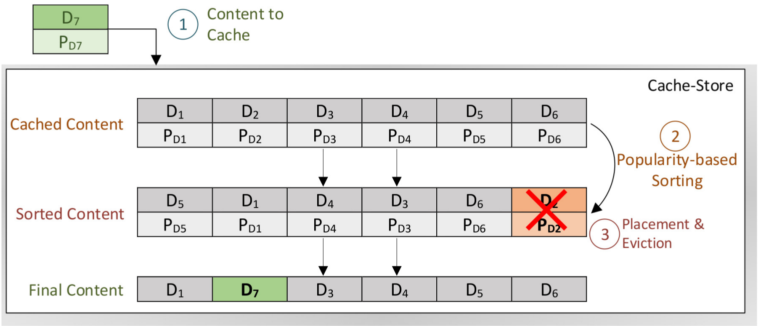


Figure 3.3: Popularity-based Content Replacement Policy.

The overall replacement policy is illustrated in Figure 3.3. All content in the cache-store are sorted based on their popularity values in a descending order. We place the content d_n in the appropriate place in the sorted list and then remove the content at the tail. Although this process is similar to LRU, we incorporate content popularity as the pillar decision factor.

3.4 Implementation and Evaluation

In this section, we describe the performance evaluation of our proposed caching scheme against different existing strategies in a large-scale network topology.

3.4.1 Simulation Setup

We have implemented CnS using Python 3 programming language. We have used *Cogent*[1], a large-scale network topology that connects most of the European countries with the USA. The network consists of 190 routers placed over Europe and USA cities. We have also used Zipf distribution to model the demands from end-users. Table 6.1 provides a summary of the simulation setup.

Table 3.2: CnS Simulation Parameters.

Parameter	Value
Number of routers	100 / 195
Size of content chunk	64MB
Number of content providers	[1 - 10]
Average size of cache store	[20 - 50]
Demand's distribution	Zipf

Furthermore, we have proposed two different scenarios:

- Single Content Provider: In this scenario, we used only one content provider and all demands are initially satisfied by this provider.
- Multiple Content Providers: In this scenario, we used a random number of content providers, range from 1 to 10, that provide different contents. This scenario is more realistic and model today's Internet behavior.

3.4.2 Evaluation Metrics

To evaluate the performance of CnS, we have measured, for each scenario, different metrics including: average network delay, cache utilization, and hop reduction ratio.

---

[1] The Internet Topology Zoo: www.topology-zoo.org

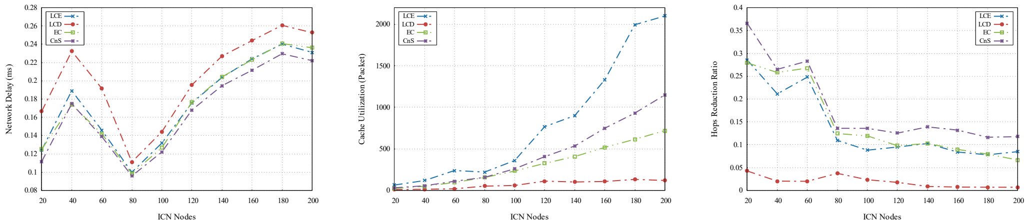
- Network delay: is the required time duration to satisfy demand issued by the consumer. In the simulation, we calculate the average network time over all demands.
- Cache utilization: is the total number of packets cached in all network's cache-stores. In the simulation, we calculate the average cache utilization over all demands and all routers.
- Hop reduction ratio: is the required number of hops that a demand traversed to fetch content from the cache-store over the provider. In the simulation, the hop reduction ratio is calculated as the average number over all demands.

Moreover, we have benchmarked CnS against different caching strategies such as LCE, LCD, and EC. We assume all contents have the same size.

3.4.3 Results and Discussion

In this following, we describe and discuss the results obtained by performing extensive simulation and analysis.

Single Content Provider Figure 3.4 presents the evaluation performance for single content provider scenario.



(a) Average Network Delay. **(b)** Cache Utilization. **(c)** Hop Reduction Ratio.

Figure 3.4: Performance Evaluation using single content provider.

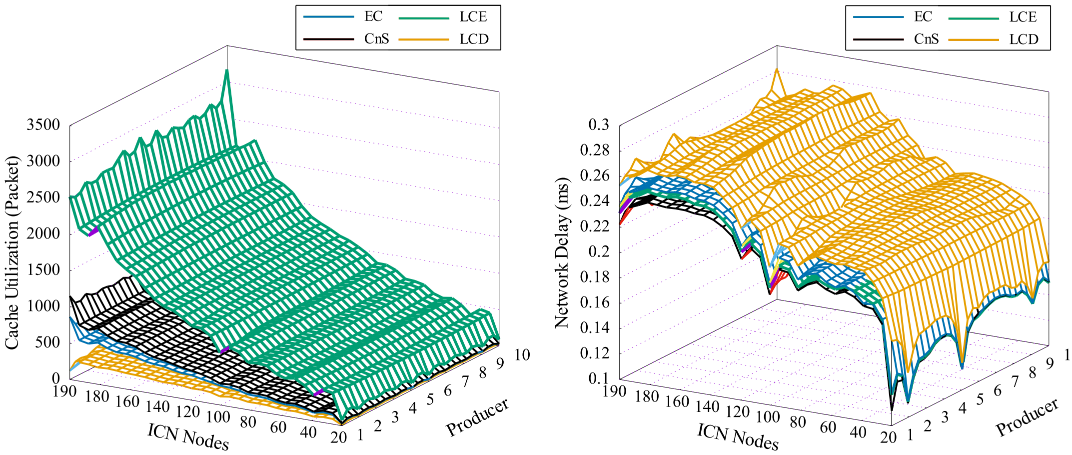
Figure 3.4(a) shows the average network delay. As long as LCD depends on caching a copy of the content one hop from the producer, the content remains far away from the consumer which reflects on the considerable network delay. EC caches the content near to consumers by storing a copy on the edge, while LCE replicates the content in each node in the delivery path including the edge router. Hence, we observe that the network delay of LCE and EC is almost overlapping. On the other hand, CnS works in pushing the popular content down. It produces a large delay at the beginning of the process, but it converges towards a small delay when the content is placed at the optimal place.

Figure 3.4(b) depicts the cache utilization based on the number of ICN nodes. LCD has a small cache utilization it only one cache-store is involved in the process. Similarly, EC caches the content at the edge node, hence we can notice that the cache utilization is semi-stable. On the other hand, LCE exploits all cache-stores in the communication path, hence its cache utilization is exponentially increased when the number of nodes increases. In CnS, the node decides to retain only the most popular content, then decides to cache or split the decision to one of more neighbors until reaching the edge router. Hence, it has a higher cache utilization compared to EC and LCD. Besides, it exploits the core routers by leaving the less-popular contents to ensure more cache hits.

Figure 3.4(c) shows the hops reduction ratio. We observe that LCD achieves a low hop reduction since the content is moved only one-hop from the producer. EC satisfies the requests of one-hop far from the consumers, where LCE satisfies any request from any node in the network depending on where the content is located (usually at the edge). Hence, both strategies achieve a remarkable level of hop reduction. CnS brings the popular content to the edge, holds over the less-popular in the core network, thus it gains better hop reduction after a set of pushing operations.

Multiple Content Providers Scenario Figure 3.5 presents the evaluation performance for multiple content providers scenario. Here, we increased the number of providers to 10, in which they provide different content.

Figure 3.5(a) shows the cache utilization versus the number of ICN nodes and the producers. For LCD, whenever the number of producers increases, the cache utilization increases. Since only one cache-store is used in LCD whenever the number of ICN nodes increases, LCD consumes low cache memory. Similarly, EC uses only the edge router as a cache-store, which explains the reason that their cache utilization is semi-static. LCE, on the other hand, consumes more cache utilization as it exploits all cache-stores. Its memory is drastically



(a) Cache Utilization performance.

(b) Network Delay performance.

Figure 3.5: Performance Evaluation using multiple content providers.

increasing as the number of ICN nodes and producers increase. CnS reduces the cache redundancy in the network and increases the content diversity even if the number of producers or nodes increases due to the use of split decision and push operation.

Figure 3.5(b) depicts the network delay versus the number of ICN nodes and producers. LCD registers the highest network delay, due to the long distance between the requester and the selected cache-store. LCE satisfies requests from any node in the network, including the edge router, where EC meets the demands at the edge. Given the principle of its operation that reduces the distance, we note that both strategies achieve less delay than LCD. CnS exploits both edge and core network resources. Therefore, we observe that it achieves lowest network delay when the content converges towards the edge (during the content pushing). Hence, the increasing number of ICN nodes or producers may not affect the performance of CnS even in large-scale networks.

3.5 Conclusion

In this chapter, we designed **C**ache a**n**d **S**plit (CnS) cache placement scheme. CnS aims at moving the popular content from the provider level towards the edge nodes closer to consumers while keeping the less-popular content at the core network. CnS is based on splitting the cache responsibility with the suitable neighbor cache-stores by pushing the popular content downstream. We evaluated the proposed scheme in large-scale topology and compared it with other strategies. Obtained results prove the efficiency and scalability of CnS in terms of network delay, cache utilization, and hop reduction ratio. In the next chapter, we will address the large scale content caching problem with a focus on IoT use case in 5G based ICN network.

Chapter 4

A Profit-based Cache Placement Scheme

4.1 Introduction

In-network content caching is a performance improvement mechanism that consists of replicating content in multiple places/nodes, named cache-stores. These cache-stores fulfill future demands rather than forwarding them to the original producer, and result in diminishing the congestion at the core network, decreasing the load at the original producer, and reducing the content delivery delay.

This chapter is organized as follows: we first highlight the requirements and challenges for an efficient large scale cache placement place in 5G based ICN network. Then, we detail the proposed cache placement strategy. Finally, we present the implementation and the performance evaluation part.

4.2 Large-Scale Caching Requirements

In-network caching is one of the fundamental components in 5G-ICN network. Such a mechanism operates to keeps and or delivers the data from the network layer in a fully transparent manner. Content requests may get satisfied from the cache stores of the intermediate node without being forwarded to the content publisher/provider [181]. Hence, the cache will enhance the overall network performance, avoid a single point of failure, improve the content distribution in the network, and reduce the network load as well as latency.

However, consider a large scale network with dynamic changes in the content distribution and their demands, a huge amount of the cached content in the network give rise to two main challenges: (1) deciding which content should be cached and (2) at which node. Hence, these challenges are the principal motivation behind this work. Thus, by capitalizing on this improvements, we will maintain a centralized caching management scheme that considers both data properties and network characteristics.

4.3 Profit-based Cache Placement Scheme

4.3.1 System Model and Problem Formulation

In this section, we describe our system model to build an optimal cache placement scheme for ICN networks. We elaborate the profit design as a building block to make the content caching decision.

4.3.2 System Model

In ICN network, we denote the set of original content objects by $O = \{o_i \mid i = 1, 2, \ldots, N\}$, where N denotes the number of content objects. We assume that all content objects have the same size S_o, and each content object has a popularity value ρ_o. Moreover, we denote the number of routers in the system by R, out of which $L \subseteq R = \{l_j \mid j = 1, 2, \ldots, M\}$ is the set of cache-enabled routers where M denotes the number of cache-enabled routers. Each cache-enabled router $l \in L$ has a limited cache size $\overline{S_l}$, while the available cache size is denoted S_l. Let ψ_l denote the cache capacity function in terms of cache ratio, processing, and memory performance, and λ_l denotes the content replacement rate. Table 4.1 summaries the most used notations in the work.

4.3.3 Profit Model

The profit model describes the gain that the network may get after caching content at a specific place. Indeed, this model incorporates multiple parameters such as content caching value, the cost of content delivery, and the content replacement reward. Moreover, this model can be used to decide which content might be evicted in case the cache store reaches the memory limits.

Table 4.1: PbCP List of used notations.

Notation	*Description*
O	Set of content objects
R	Set of routers
L	Set of cache-enabled routers
N	Number of content objects
M	Number of routers
S_o	Size of content
$\overline{S_l}$	Total size of cache store
S_l	Available size of cache store
ψ_l	Cache capacity
λ_l	Cache replacement rate
ρ_o	Content popularity
k_o^l	Cache placement variable
$\zeta_{o,l}$	content caching value
$\omega_{o,l}$	content delivery value
$\chi_{o,l}$	cache replacement reward

PROFIT VALUE (P): For a given content object $o \in O$, the Profit Value (**P**) is calculated as the sum of all content chunks' profit. The function is derived with respect to the *content caching value*, *content delivery value*, and *cache replacement reward*. The content caching value ($\zeta_{o,l}$) evaluates the profit of caching the content o at the router l and is calculated according to the cache capacity, content popularity, and content replacement rate. The content delivery value ($\omega_{o,l}$) evaluates the profit of moving the content o from the publisher to the cache store at the router l and is calculated based on the end-to-end network delay to deliver the requested content between the publisher (or another cache store) to the current cache store regarding the set of consumers. The content replacement reward ($\chi_{o,l}$) evaluates the reward that the network may gain by removing a content from the cache store l and replace it with the new received content o and is calculated based on the size of the cache store and content to cache. Eq. 4.1 defines the profit value function.

$$\mathbf{P} = \sum_{o=1}^{N} \sum_{l=1}^{M} k_o^l \cdot (\zeta_{o,l} + \omega_{o,l} + \chi_{o,l}) \tag{4.1}$$

where k_o^l expresses the cache placement variable, if $k_o^l = 1$: the content o is cached at l, otherwise $k_o^l = 0$; $\zeta_{o,l}$ expresses the content caching value; $\omega_{o,l}$ expresses the content delivery value; and $\chi_{o,l}$ expresses the cache replacement reward. More details about these factors are given in the next subsections.

The main benefits of using such a function is to assigning to each cache placement a profit value, this profit is directly related with wining cost of cache and delivery the content and affected by the cache replacement operation(s) when evicting another content in case of full memory. Hence, the profit value tends to select the best cache position for popular content with highest chance to serve large number of consumers, highest chance to select stable cache store with less replacement operation, and short end-to-end delay.

1. CONTENT CACHING VALUE (ζ): The main purpose of using the content caching value ($\zeta_{o,l}$) is to evaluate the profit of caching a content object on certain cache store, it is the pillar factor in the formula and plays an important rule to select the optimal cache store. This value is highly depended on both cache store characteristic (due to the diversity of router's configuration) and the content itself (due to the diversity of content's popularity). We design the content caching value as a function of cache store capacities (ψ_l), content popularity (ρ_o), and cache replacement rate (λ_l).

(a) Cache Store Capacity (ψ): The cache store capacity can be expressed based on multiple functions. Eq. 4.2 expresses a general definition of cache store capacity for n functions. $\Phi_l^{(k)}$ represents the k^{th} function which is calculated in a time period, and θ_k the corresponding weight parameter.

$$\psi_l = \sqrt[n]{\sum_{k=1}^{n} \theta_k \Phi_l^{(k)}} \tag{4.2}$$

In this work, we integrate three main functions that have a direct impact of the cache store capacity:

- Available Cache Ratio: the available cache ratio shown in the next equation is necessary to calculate the cache value. It depends on the proportion of available caching space in a continuous manner.

$$\Phi_l^{(1)} = \frac{\dot{v}_l - min(\dot{v}_l)}{max(\dot{v}_l) - min(\dot{v}_l)}$$

where $\dot{v}_l = \frac{S_l}{\overline{S_l}}$, $\overline{S_l}$ denotes the total size of cache store and S_l is the available cache size. If $S_l = 0$, then $\dot{v}_l = 0$ and $\Phi_l^{(1)}$ converges to zero.

- Processing Performance: the processing performance, as expressed in bellow, is an important factor to judge the overall performance of a router and how fast is it to execute operations. Routers with highest performance are always preferable for content caching.

$$\Phi_l^{(2)} = \frac{\bar{v}_l - min(\bar{v}_l)}{max(\bar{v}_l) - min(\bar{v}_l)}$$

  where $\bar{v}_l$ denotes the Million Instructions per Second generated by the router l in a period of time.

- Memory Performance: the memory performance, expressed in the next equation, is also an important factor to show how fast and efficient the access to content in the cache store is. It is represented in terms of average memory access time. Routers will small memory performance may affect the content delivery performance and hence degrade the quality of service.

$$\Phi_l^{(3)} = \frac{\hat{v}_l - min(\hat{v}_l)}{max(\hat{v}_l) - min(\hat{v}_l)}$$

  where $\hat{v}_j$ denotes the Average Memory Access Time in the cache store.

(b) Content Popularity: The content popularity is as important as the cache capacity. It is used to decide the cache placement, caching popular content near to consumers drastically enhances the cache utilization and cache hit ratio by satisfying most demands from the near cache store. On the other hand, the popularity of the content must be used to select which content to evict. Popular content should be kept in the cache store for a long time, less popular content can be placed some hops far away, while non popular content can be removed from the cache store. Eq. 4.3 represents the content popularity, which is depending on the request frequency of all content chunks. It is important to highlight that content popularity can be changed during its life-time, for example a new movie or news are popular for the first appearance period, and can be not popular after some period.

$$\rho_o = \frac{f_o}{\sum_{o=1}^{N} f_o} \cdot \frac{1}{\tau_{now} - \tau_{last}} \cdot \frac{1}{(\tau_{last} - \tau_{first})/f_o} \tag{4.3}$$

where f_o is the number of request; and τ_{first}, τ_{last}, and τ_{now} denote the first time, the last time the content has been request, and current time, respectively. The first terms in the equation ($\frac{f_o}{\sum_{o=1}^{N} f_o}$) denotes the request frequency of content object, ($\tau_{now} - \tau_{last}$) denotes the recent content request time interval, and ($(\tau_{last} - \tau_{first})/f_o$) denotes the average time interval of content requests.

(c) Content Replacement Rate: The cache replacement operation refers to the operation of removing content from the cache store when it is full of memory, and to cache the new arrived content. According to the cache store configuration and the distribution of content, a node may frequently replace contents. If content is removed from the cache store, all future demands will be forwarded upstream toward the publisher. Multiple cache replacement operations result in extra cost to the communication and the overall system. Therefore, caching content on a cache store with high cache replacement rate has more probability to get evicted soon. Hence, it is always preferable to cache content at a cache store will lower cache replacement rate. Eq. 4.4 expresses the content replacement rate.

$$\lambda_l = \frac{1}{\overline{S_l}} \cdot \sum_{j=1}^{k} S_j \tag{4.4}$$

where k is the number of cache replacement times and S_j is the size of the content to be replaced in the j^{th} replacement operation.

The content caching value can be expressed as shown in Eq. 4.5. Here, we recall that we prefer to cache content with highest popular content ρ at cache store with highest capacities ψ; thus, these values are directly proportional. While we try to avoid cache stores with highest replacement rate λ; hence, it is inversely proportional to cache capacity and content popularity.

$$\zeta_{o,l} = \frac{\rho_o \cdot \psi_l}{\lambda_l} \tag{4.5}$$

2. Content Delivery Value (ω): The content delivery value ($\omega_{o,l}$) evaluates the profit of moving content o from the publisher level to the cache store l in terms of the communication cost from the point of view of number of consumers. This value can be evaluated based on the communication distance (e.g., number of hops), the required delivery time (e.g., end-to-end network day), or resources (e.g., bandwidth). To serve a large number of consumers with the minimum cost, a content should be cached as near as possible to consumers (typically at the edge level). However, the distribution of demands, position of consumers, and the available cache resources may not always be in the benefits of the edge nodes. Hence, the selection of the suitable cache store must take the content delivery cost along with the aforementioned factors in consideration. Eq. 4.6 expresses the content delivery value.

$$\omega_{o,l} = \frac{avg(h(\{c\},l))}{avg(h(\{c\},p))} \tag{4.6}$$

where $avg(h(\{c\},l))$ and $avg(h(\{c\},p))$ represent the average end-to-end network delay to deliver the content chunk between a list of consumers and cache store, and consumer and the publisher, respectively. It is important to highlight that we ca use the number of hops or the required bandwidth in place of end-to-end network delay to measure the cost.

3. Cache Replacement Reward (χ): The cache replacement reward ($\chi_{o,l}$) evaluates the profit of caching a content o when evicting another content $o\prime$. Indeed, evicting a popular content from cache store l leads to not satisfy future demands for the evicted content from the local cache store, and fundamentally forward these demands upstream toward the publisher according the routing rules. On the other hand, evicting a non popular content and replace it with a new popular content helps to provide fast data delivery, and enhance the overall network performance. This eviction leads to a reward value as expressed in Eq. 4.7.

$$\chi_{o,l} = \begin{cases} 0\,, & \text{if } S_o \leq S_j \\ \frac{S_o}{BW_{(l,l')}}\,, & \text{if } \rho_o \geq \rho_{o\prime} \\ -\frac{S_{o\prime}}{BW_{(l,l')}}\,, & \text{otherwise} \end{cases} \tag{4.7}$$

where S_o is the size of content to be cached, $S_{o\prime}$ is the size of content to be evicted, S_j the available size in the cache store, and $BW_(l, l')$ the required bandwidth between the current cache store and the publisher or a near in-path cache store that has the evicted content.

Eq. 4.7 describes the reward based on the size of the cache store and the content popularity. If the cache store l has enough memory to cache the arrived content o without evicting any content then $\chi_{o,l} = 0$. If the popularity of the arrived content ρ_o is higher than the popularity of the content to be evicted $\rho_{o\prime}$ then the reward is the required time to download that content o from the publisher. Otherwise, the reward will be a negative value and is depending of the content size of the evicted content $o\prime$ and the bandwidth between the cache store l and content provider $l\prime$ (i.e., content publisher or replica node), which refers to the required time to download the content from the provider.

4.3.4 Problem Formulation

Based on the presented profit model, we formulate the content caching problem as follows. For a given ICN network, list of contents, and the distribution of demands, we find the optimal cache store that maximizes the profit value. In another words, find the best cache placement that satisfies as much as possible of demands with less number of eviction operations, and places the popular content near to consumers by serving maximum number of consumers. Eq. 4.8 describes the objective function.

$$maximize \quad \mathbf{P} \tag{4.8}$$

subject to:

$$\sum_{o=1}^{N} k_o^l \leq \overline{S_l}, \quad \forall l \in M \tag{4.9}$$

$$1 \leq |\{l \mid k_o^l = 1, \forall o \in O\}| \leq M \tag{4.10}$$

$$k_o^l \in \{0, 1\}, \quad \forall o \in N, l \in M \tag{4.11}$$

Constraint (4.9) imposes the overall cached content size should not exceed the maximum caching size of the cache store. Constraint (4.10) enforces that the maximum number of cache stores that may be involved should not exceed the overall number of cache-enabled routers. Constraint (4.11) indicates the possible values of caching decision variable. The complexity of the presented optimization problem is exponential and depending on the number of content, number chunks that construct that content, number of cache-enabled routers, and distribution of content popularity and demands. It can not be solved with traditional optimization solutions.

Theorem 1. The problem is NP-hard.

Proof. The goal of cache placement strategy is to maximize the profit value of content caching. The problem formulated in (4.8) can be represented as a Knapsack problem which is an NP-complete problem. Therefore, the presented problem is NP-hard. ☐

4.3.5 Proposed Scheme

In this section, we propose a **P**rofit-**b**ased **C**ache **P**lacement (PbCP) scheme using an inspiration algorithm known as Tabu Search. The search space size of the presented optimization problem is exponential in terms of the number of content objects, the popularity of the content, and the number of cache-enabled routers. The required time to solve such kind of optimization problems is quite long even for small-scale networks. Therefore, a heuristic algorithm is needed to provide an efficient cache placement in large-scale networks. Tabu Search meta-heuristic is promising to find a near optimal solution. Hence, we exploit it in ICN cache placement problem.

Tabu Search (TS) [182] is an iterative search algorithm. It starts by exploring the search space for an initial local minima solution aiming at converging with the global optimal solution. Hence, it performs a set of transactions (i.e., moves) in the space of the neighborhood's current solution. In another word, it selects a better solution (from the perspective of the current solution) from the neighborhood based on an aspiration criterion (a criteria to decide if a move leads to promising solution). The process is repeated until the termination criterion (if the process reaches a total number of iterations or the best solution was found) is satisfied. Tabu Search maintains a short-term memory to avoid retracing/cycling the previous solutions. The idea consists of recording recent transactions in a list in order to prevent previous transactions from being repeated. In the following, we explain the major parts of the proposed PbCP scheme as outlined in Algorithm 2.

Algorithm 2: Profit-based Cache Placement Scheme.

```
Input: $O$: Set of content objects; $L$: Set of cache-enabled routers.
Output: $< o,\ l >$ optimal cache placement
for ($o \in O$) do
    (1) Construct Initial Solution
    $\varkappa \leftarrow 0$
    for ($l \in \mathcal{N}(Publisher)$) do
        Calculate $\zeta_{o,l}$;  ▷ According to Eq. 4.5
        Calculate $\omega_{o,l}$;  ▷ According to Eq. 4.6
        Calculate $\chi_{o,l}$;  ▷ According to Eq. 4.7
        $\mathbf{P} \leftarrow \zeta_{o,l} + \omega_{o,l} + \chi_{o,l}$;  ▷ Profit Value
        if ($\varkappa \leq \mathbf{P}$) then
            $\varkappa \leftarrow \mathbf{P}$;
            $\mathcal{I}_0 \leftarrow < o,\ l >$;
        end
    end
    (2) Short-term Memory
    TL ← list();
    i ← 0;
    $\mathcal{I} \leftarrow \mathcal{I}_0$;
    $\mathcal{I}^* \leftarrow \mathcal{I}_0$;
    (5) Termination Criterion
    while ($i < N_{Iter.}$) do
        (3) Neighborhood Solution
        Select $\mathcal{I}' \in \mathcal{N}(\mathcal{I})$;
        $\mathcal{I} \leftarrow \mathcal{I}'$;
        (4) Aspiration Criterion
        if ($\mathbf{P}(\mathcal{I}) > \mathbf{P}(\mathcal{I}^*)$) then
            $\mathcal{I}^* \leftarrow \mathcal{I}$;
        end
        TL.append($\mathcal{I}$);
        if (length(TL) > MAX_LEN_TABU_LIST) then
            TL.pop(0);
        end
    end
end
```

(1) *Construct Initial Solution*: TS process starts with an initial solution ($\mathcal{I}_0$).

The starting point consists of finding the list of neighbor cache-enabled routers that are directly connected with the content publisher. Then, and according to the number of received demands, *PbCP* calculates the *Content Caching Value* (ζ), *Content Delivery Value* (ω), and *Content Replacement Rate* (χ) for each content $o \in O$ and each neighbor cache-enabled router $l \in \mathcal{N}(\text{Publisher})$, where $\mathcal{N}$ is function that returns the cache-enabled neighbor routers. Then, it calculates the corresponding profit value (**P**) and selects the cache placement with the highest value.

(3) *Neighborhood Solution*: PbCP performs a transaction to a new solution ($\mathcal{I}'$) by exploring the neighborhood of the current solution $\mathcal{N}(\mathcal{I})$. This process refers to a local search mechanism and aims at finding a local improvement of the solution. A transaction can only be performed if the profit value of caching content at the neighbor cache-enabled router ($\mathcal{I}'$) is better than keep caching it at the current node ($\mathcal{I}$).

(4) *Aspiration Criterion*: Aspiration Criterion defines how PbCP evaluates and recognizes the best solution. Indeed, it aims to decide how the transaction from the current solution to a new solution can be profitable. Thus, we use the Profit Value defined in Eq. 4.1 as the main criteria to evaluate the content caching value, content delivery value, and the cache replacement reward. Hence, recognize the best solution in the search space.

(5) *Termination Criterion*: The algorithm is terminated when the best solution ($\mathcal{I}^*$) does not improve for a certain number of iterations ($N_{Iter.}$). The number of iterations is fine-tuned to find a trade-off between the quality of the cache placement and algorithm execution time.

4.3.6 Complexity

The required time to construct the initial solution is $O(N \times M)$ and the required time for Tabu search to achieve the optimal cache placement is $O(N_{Iter.} \times N^2)$. Therefore, the overall complexity is not exponential rather it is polynomial and the execution overhead is relatively negligible $O(N_{Iter.} \times N^2)$, where N is the number of contents, M the number of cache-enabled routers, and $N_{Iter.}$ the number of iterations required by the core Tabu Search algorithm.

4.3.7 Optimality

PbCS converges from local optimal solution to a global optimal solution. To prove this optimality convergence, we conduct simulation using a tree with the high of 5. The root is the content publisher and the leaves are requesters. We have used two different scenarios: in the first scenario, all demands are coming from the same network (i.e., same tree branch were all requesters are connected to the same edge router), in the second scenario, demands are coming out from different networks (i.e., different branches), two of these networks have a common router at level 2, while all of them are connected via router of level 3.

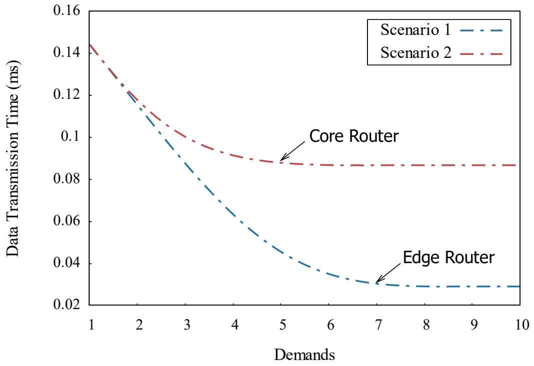


Figure 4.1: Optimal solution convergence.

Figure 4.1 illustrates the obtained results. In the first scenario, after 7 demands, the content is placed at the edge router which is the optimal cache placement. Hence, all future demands have the same data transmission time. However, content is cached at the first common router (at level 3) after 5 demands. This core router is the optimal place that handles all demands from all connected networks with the optimal data transmission time.

4.4 Implementation and Evaluation

In this section, we evaluate the performance of PbCP scheme under various experimental settings. We implemented PbCP using Python programming language and Gurobi optimizer to solve the cache placement problem. The rest of this section is organized as follows. First, we present the simulation setup for benchmarking PbCP against other caching strategies. Second, we describe the evaluation metrics. Then, we evaluate Tabu Search against Gurobi solver. Last, we discuss the obtained results for PbCS evaluation.

4.4.1 Simulation Setup

The experimental environment, shown in Figure 4.2, consists of content publisher(s), a set of forwarder routers (without caching feature), and a set of cache-enabled routers. All content demands are coming from requesters connecting via the leaf routers (omitted in the figure for simplicity purpose).

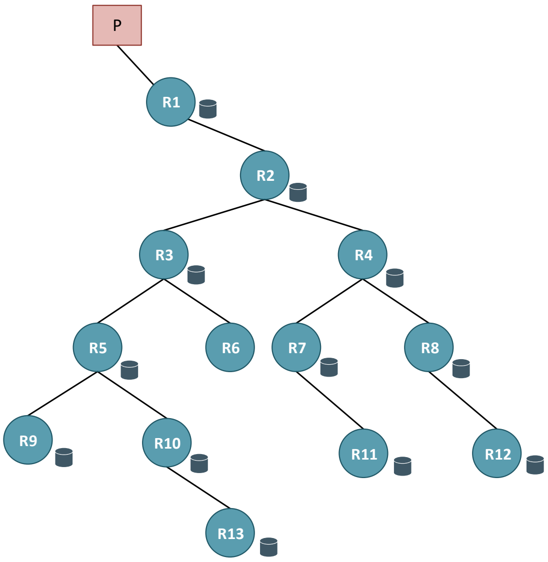


Figure 4.2: Simple network topology.

Table 4.2: PbCP Simulation Parameters.

Parameter	Value
Size of content chunk	64MB
Number of unique contents	[100, 1000]
Zipf parametres (α)	[1.6, 2.5]
Number of caching enable node	[20, 190]
Node memory capacity	[100, 1000] Giga

4.4.2 Evaluation Metrics

We evaluated the performance of PbCP against different cache placement strategies including Leave Copy Everywhere (LCE), Edge Caching (EC), and Betweenness centrality. Each experimental has been executed 10 times, and the average value has been plotted. This will help to validate the effectiveness of the cache placement decision and improve the reliability of results. The metrics that we have evaluated are:

- Average Cache Size: expresses the frequent changes within the cache memory inputs at the cache-enabled nodes. In the simulation, we focus to highlight the effects of the cache size on the proposed by progressively increase the value of the cache memory from 100 Giga up to 1 Tera.
- Average Cache Capacity: indicates the total of the overall size of all cache nodes. In the simulation, we study the impact of the cache capacity rising a number of caching nodes.
- Average Number of Content Chunks: represents the available content located in the producer, can be requested by consumers, and can also be cached by the intermediate nodes. Thus, changing the number of these chunks can reveal how far the manageability of the cache strategy. In the simulation, we progressively increase the number of available content.
- Average Content Popularity: in real-world networks, the changing of the content popularity varies over time, this phenomena can be approximated with a Zipf law distribution [151]. By using a Zipf law, we can distribute popularity values for a set of available content. At the same time, demands propagation amount the edge nodes follow Poisson distribution law [183].

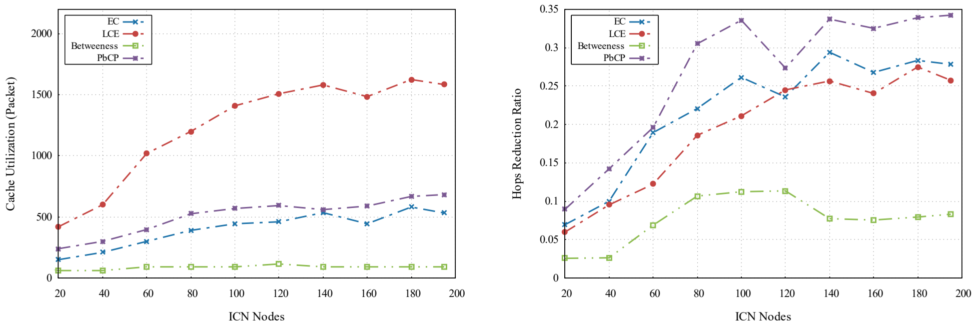
Each metric has been evaluated in terms of:

- Hop reduction ratio: is the ratio of the reduced hops that a request might be crossed to get content. In other words, it describes how many hops that the content is pushed toward the consumer.
- Number of cache replacement: is the total number of operations where the content store evicts content in order to keep a room for the new arrival content.

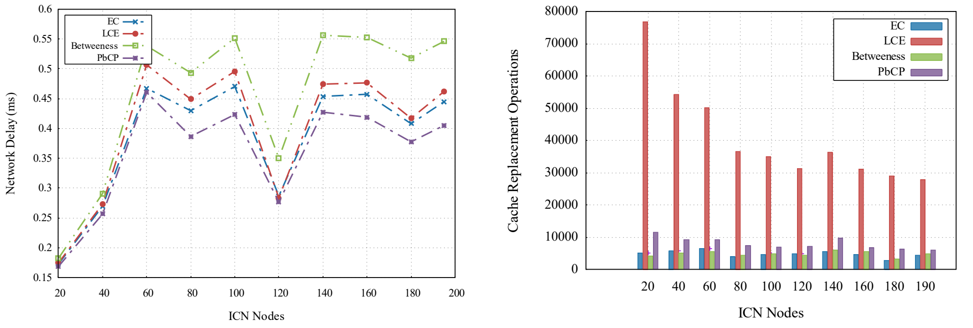
- Cache utilization: measures the number of copies of each content at the network overall. Ensuring a good content distribution means a fair of content replication and good memory usage.
- Data transmission time: indicates the needed time for the request to be sent and satisfied either from the original producer or content store.

4.4.3 PbCS Evaluation

We study the impact of cache size, cache capacity, number of data chunks, and content popularity on cache hit ratio, number of cache replacement operations, cache utilization, and data transmission time.



(a) Cache Utilization **(b)** Hop Reduction Ratio

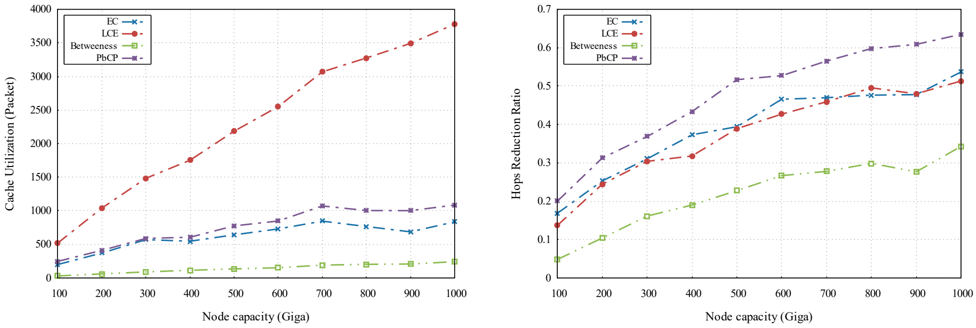


(c) Data Transmission Time **(d)** Number of Cache Replacement

Figure 4.3: Impact of cache size.

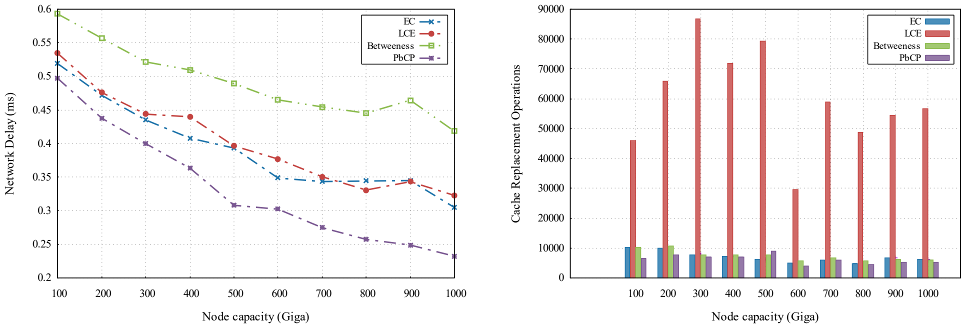
Impact of cache size In the following, we study the impact of cache size on PbCS strategy and other caching strategies. Figure 4.3 depicts the obtained results.

Figure 4.3(a) shows the cache utilization per number of nodes. We observe that the rate of cache utilization in all strategies is growing proportionally with



(a) Cache Utilization

(b) Hop Reduction Ratio

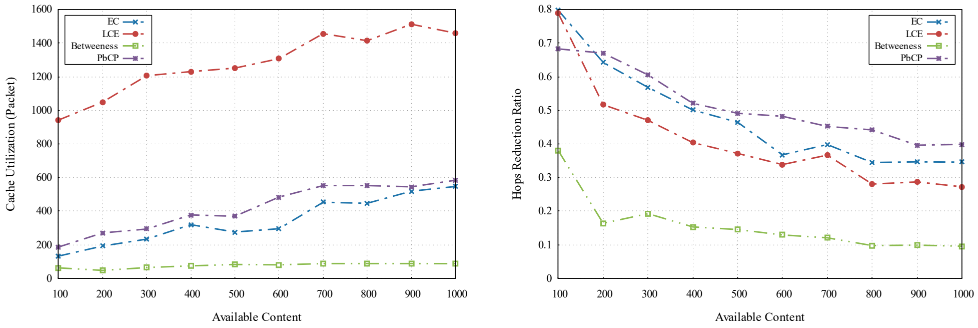


(c) Data Transmission Time

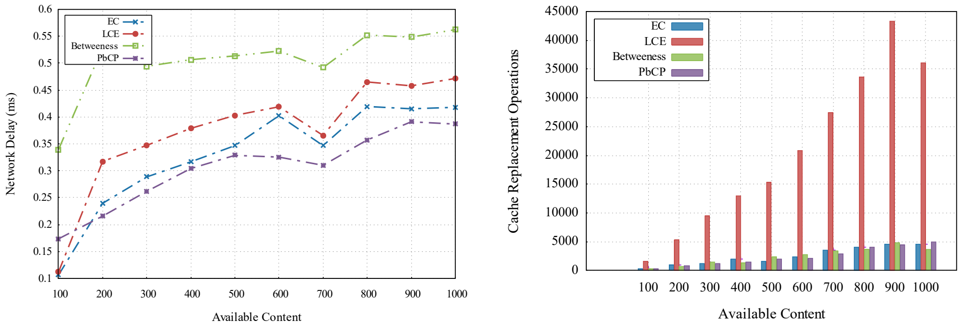
(d) Number of Cache Replacement

Figure 4.4: Impact of cache capacity.

the number of nodes. In fact, LCE records the highest cache utilization since it caches all processed content. Oppositely, Betweenness strategy shows the lowest level of cache utilization as it chooses only the high node degree to cache the content. EC, on the other side, has less cache utilization since it involves only the edge nodes as content stores. PbCP records a slight increase in cache utilization than EC strategy. Yet, it maintains the cache utilization by keeping only the content that has a high profit and pushing them near to the consumer, only one node is used as a content store with a lowest probability that multiple nodes cache the same content. This is proven in Figures 4.3(b) and 4.3(c), where PbCP has an excellent hop reduction ratio compared to the other strategies and an improved data transmission time, respectively. Finally, Figure 4.3(d) depicts the number of replacement operations versus the cache capacity. We observe that LCE has an excessive increase in the cache operation due to the huge number of cache decisions to provide room for the newly arrived content. Betweenness and EC record almost the same cache operation ratio and this is due to the limited set of the used content-store. However, PbCP reduces the cache replacement operation since it selects only popular content to cache.



(a) Cache Utilization **(b)** Hop Reduction Ratio

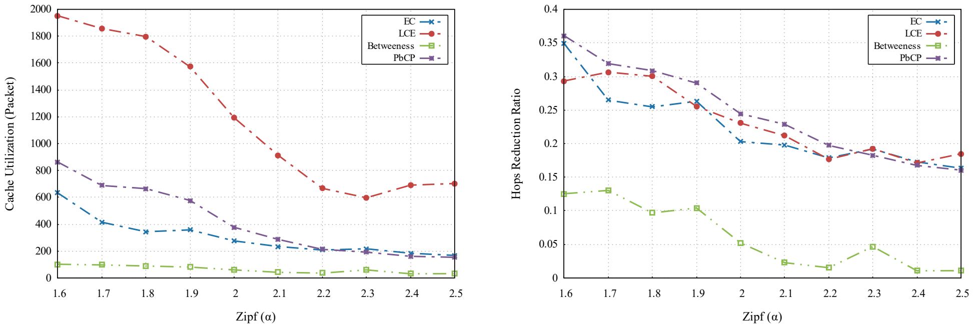


(c) Data Transmission Time **(d)** Number of Cache Replacement

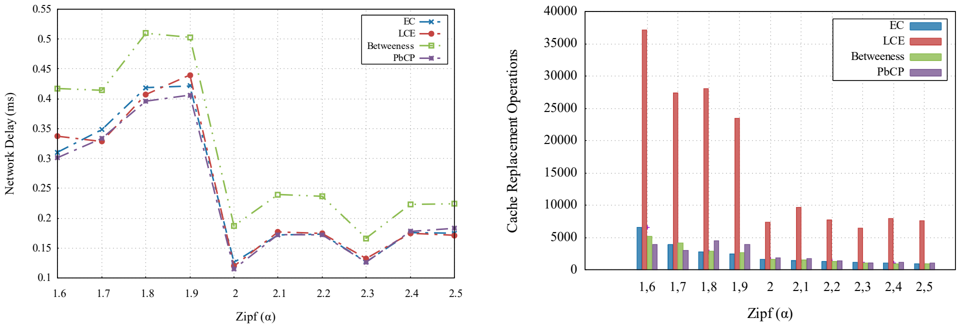
Figure 4.5: Impact of number of data chunks.

Impact of cache capacity In the following, we study the impact of the node's memory size of the cache decision and performance. In the simulation, we fixed the number of nodes to 160 and then we progressively increased the cache capacity of each node at the network. The obtained results are plotted in Figure 4.4.

Specifically, Figure 4.4(a) shows the variants of the cache utilization while increasing the nodes' cache capacity, as it is shown the LCE strategy records a considerable memory utilization while the betweenness strategy records the lowest utilization. At the beginning EC and PbCP seem almost overlapped, thereafter, the PbCP starts steadying. As long as the PbCP caches only the highest content profit, it improves the hop reduction ratio which is depicted in Figure 4.4(b). Besides, despite the excessive use of memory utilization in LCE, it shows a non-optimal hop reduction ratio which is almost similar to the EC strategy. While the Betweenness shows a weak hop reduction. Figure 4.4(c)



(a) Cache Utilization **(b)** Hop Reduction Ratio



(c) Data Transmission Time **(d)** Number of Cache Replacement

Figure 4.6: Impact of content popularity.

shows the data transmission time versus the cache capacity. the PbCP outperforms other strategies by reducing the delivery delay, resulting in a better cache resource utilization even with high available capacity. In Figure 4.4(d), LCE disgraces the cache replacement operation at every increasing step of the cache capacity yet still records the highest rate. On the other hand, the other strategies cover and show a significant replacement process.

Impact of number of data chunks In the following, we highlight the impact of the number of chunks by increasing the number of the requested content, the results are shown in Figure 4.5.

Indeed, the high availability of the embedded content in the network lets the cache strategy to rise an additional process of both the replacement process and the cache utilization. Figures 4.5(a) and 4.5(b) depict the cache utilization and the replacement process ratio versus the number of available content, respectively. LCE shows the highest cache utilization since it uses a greedy cache decision fashion accompanies with a high replacement rate. On the other hand, the Betweenness strategy uses only the node that has a top neighbors degree to keep a copy of the content, hence reflects a low usage of the cache capacity and minimum cache replacement operation. PbCP defeats the EC strategy in the cache usage as well as the replacement rate. We observe that PbCP outperforms the other strategies and shows a considerable improvement in the hop reduction, even with the rising of the number of content, since PbCP selects only the highest profitable content and progressively pushes them towards consumers (Figure 4.5(c)) which reflects to decrease the data transmission time (Figure 4.5(d)).

Impact of content popularity The real-world content popularity is unstable and varying over time, this phenomenon affects the caching plan as well as resource usage. We simulate the changes in the content popularity by varying the Zipf α parameter. The obtained results are shown in Figure 4.6.

With the increase in content popularity distribution values, all content start getting a low and close popularity level. Figure 4.6(a) shows the cache utilization against the gradual reduction of content popularity. We observe that all strategies start declining the cache usage in concur with Zipf α rising. Notably, PbCP reduces the cache usage significantly than the EC since PbCP keeps only the popular and the more profitable content. Besides, if we observe Figures 4.6(b) and Figure 4.6(d), we can see that PbCP still outperforms other strategies and records the best hop reduction rate while seems almost overlap with EC and LCE in data transmission time even for the unpopular content. Finally, as shown in Figure 4.6(d), the content popularity approximation reduces the replacement process for all of the strategies. In fact, PbCP shows a high sensibility for content popularity. Also, it improves content diversity while enhances content availability at the network while guaranteeing the high level of network resource usage.

4.5 Conclusion

In this chapter, we focused on in-network data distribution and caching issues in ICN networks. First of all, we designed PbCP, a profit-based cache placement strategy, that integrates multiple influencing factors in order to determine the optimal cache placement for each content. Hence, our PbCP strategy runs on a high level of abstraction using the content profit value as the main-decision factor. We adopted a Tabu local search algorithm that tends to guarantee the optimal content distribution. In PbCP, the high profitable content is always prioritized to be closer to consumers which reduces the delivery delay, improves the quality of service, enhances the overall network performances. Using a real-world topology as well as an exhaustive simulation considering different related factors that impact the caching scheme, our cache strategy maintained the memory resources and improved the data transmission time while enhanced content diversity. From the obtained results it is clear that a multi-metric strategy will help in placing the content at the right place which will not only improve the network performance but also increase network utilization, quality of service, and satisfy large number of demands with less network overhead. In the next chapter, we will focus on the energy efficiency in 5G network and design an energy aware cache mechanism.

Chapter 5

Energy-aware Cache Placement Scheme for IoT-based 5G-ICN

5.1 Introduction

Over the past decade, IoT devices have gained extensive attention from both the industries and academic sectors. We have witnessed a tremendous amount of research effort dedicated to improving the quality of communication in such an environment. Most of IoT devices are considered with resource-constrained and limitations in processing, memory, and energy. Despite the existing efforts, the explosive increase of IoT devices in the future 5G based ICN network will open for IoT applications in terms of energy efficiency communication where the later is a growing concern. Morover, The forecast for future ten years' traffic demand shows a massive increase in connectivity with more than 100 billion connections from IoT devices [184], which imposes a big challenge for 5G based ICN communication technology in the future [54]. The communication industry is struggling in the challenges of huge capacity demand but a low cost for future mobile networks. 5G-ICN paradigm is targeted to shed light on these contradictory demands.

This chapter is organized as follows; we first highlight the requirements and challenges for content cache placement in connected IoT with future 5G-ICN environment. Then, we present the system model and formulate the energy placement problem. We detail the proposed scheme. Finally, we present the implementation and evaluation part.

5.2 Challenges and Requirements

The Internet of Things (IoT) is overrunning different domains and future emerging applications [185]. These heterogeneous devices generate a tremendous amount of content. 5G-ICN networks aim to meet future users and application requirements. The in-network caching is a fundamental feature supported by design in 5G-ICN, which improves the network performance by providing ubiquitous caching in the network layer. Since most of IoT devices are resource-constrained with limitations in communication, processing, energy, and memory; the energy-efficiency is a prime concern in IoT deployment. Different factors may affect energy efficiency in ICN-based wireless IoT networks such as transport (communication), caching, and energy limitation. In this regard, the main challenges of an efficient cache strategy are to maximize energy-saving, improve energy efficiency and address both content transmission energy, and content caching energy.

5.3 Energy-aware Cache Placement Scheme

In the following, we define the system model, formulate the cache placement problem,and describe the proposed scheme

5.3.1 System Model

In order to design an efficient IoT network energy model, we split the model into two essential areas: the network model and the energy model.

Network Model Fig. 5.1 illustrates a scenario of caching system model for wireless IoT. An IoT network is represented as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ represents a set of nodes, and $\mathcal{A}$ set of links (wired and wireless). The network is modeled as a multi-hop model, where nodes can be classified into: Access Things (AT), Edge Things (ET), and intermediate routers ($\mathcal{R}$) with the caching capabilities. ATs represent IoT users, while ETs are the edge devices. We assume that $\mathcal{R}$s are randomly placed in the network. A set of ATs connected to the same ET is named IoT-Net.

Table 5.1: EaCP List of used Notations.

Parameters of the Model	
$\mathcal{N}$	Set of nodes
AT	Set of Access Things
ET	Set of Edge Things
$\mathcal{R}$	Set of routers with caching capabilities
$\mathcal{L}$	$\mathcal{L} = \{\mathcal{R} \cup ET\}$ Set of replica-nodes
$\mathcal{F}$	Content catalogue
$\mathcal{T}_l$	Number of content stored in the cache-store l
$\mathcal{S}_f$	Content size
$\mathcal{E}_{i,l,f}$	Overall energy required to fetch f.
$\mathcal{E}^c_{i,l,f}$	Energy required to cache f in l
ω_c	Required energy by the hardware.
$\mathcal{C}_{max,l}$	The maximum cache capacity of the node l.
$\mathcal{Q}_{max}$	QoS assignments or link transfer capacity.
$\mathcal{D}_{i,l,f}$	Data rate to fetch content from cache store l.
$\mathcal{D}_{i,l,f}$	Data rate to fetch content from original producer p.
k_l^f	{0-1} Cache storage reachability:
	$k_l^f = 1$ if replica node $l \in L$ can serve file $f \in F$
x_l^f	{0-1} Cache decision variable :
	$x_l^f = 1$ if node $l \in L$ decides to keep the content $f \in F$

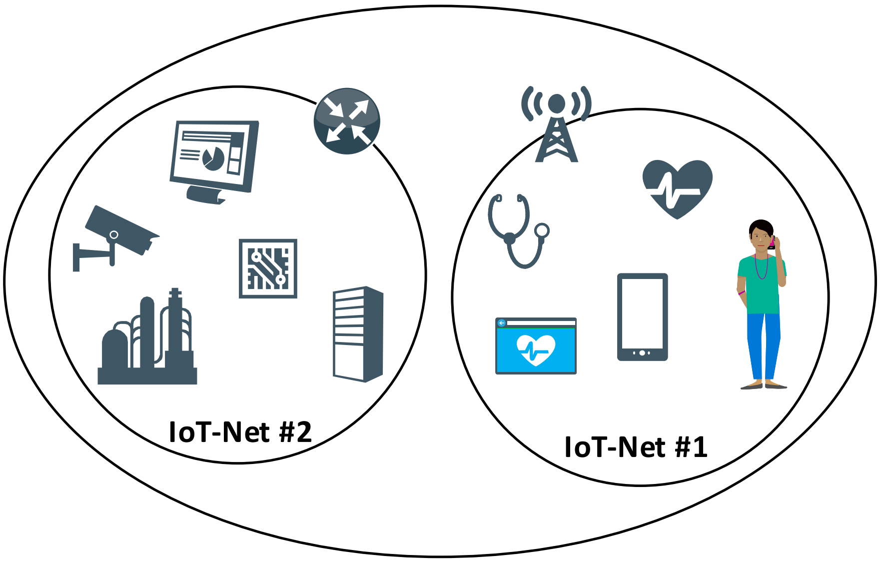


Figure 5.1: Caching System Model for Wireless IoT.

We define $\mathcal{L} = \{\mathcal{R} \cup ET\}$ as a set of nodes who can cache and offer contents (replica-nodes), and $\mathcal{F}$ content catalogue. We assume that content chunks have the same size. Each $l \in \mathcal{L}$ has a finite cache-size $\mathcal{C}_{max}$, in which the occupied size is defined by $\mathcal{T}_{l\in\mathcal{L}} \times \mathcal{S}_{f\in\mathcal{F}}$, where $\mathcal{T}_l$ denotes the number of contents stored in the cache-store, and $\mathcal{S}_f$ denotes the content size. Moreover, each link $a_l \in A$ is restricted with max capacity $\mathcal{Q}_{max}$ according to the link transfer capacity or available QoS assignment.

Energy Consumption Model In IoT network, each user $i \in \mathcal{AT}$ might be assigned at least to one $\mathcal{ET}$. User request *Req* may be satisfied by any replica node l in the IoT-Net or the original producer p.

Two primary factors influence the energy consumption in the IoT network: the content transmission energy (when performing the process of request or delivering the content); and the content caching process(when caching the content at the local node).

Content Transmission Energy: The content transmission energy varies depending on the content availability place: (a) the content is satisfied in the network by any cache-store, (b) the request reaches the original producer where the content is delivered back to the requester. In doing so, we denote k_l^d as the content storage reachability.

$$k_l^f = \begin{cases} 1 & \text{if file } f \text{ is served by cache-store } l. \\ 0 & \text{else.} \end{cases} \tag{5.1}$$

When the content is satisfied by a cache-store $l \in \mathcal{L}$, we define the data rate by:

$$\mathcal{D}_{i,l,f} = \log_2(1 + SINR(i,l,f)) \tag{5.2}$$

where $SINR(i,l,f)$ describes the signal to interference noise ration.

Otherwise, the request must be forwarded to fetch the content from the original producer p. The energy consumption in such a case is defined as:

$$\mathcal{D}_{i,p,f} = \log_2(1 + SINR(i,p,f)) \tag{5.3}$$

where $SINR(i,p,f)$ describes the signal to interference noise ratio in the system. Furthermore, as long as content delivered is imporrant, we define $\mathcal{T}^{\mathcal{T}}$ as the total throughput to fetch content $f \in F$ requested by the user $i \in AT$ over the network by:

$$\mathcal{T}^{\mathcal{T}} = (k_l^f \times \mathcal{D}_{i,l,f}) + [(1 - k_l^f) \times \mathcal{D}_{i,p,f}] \tag{5.4}$$

Content Caching Energy: Let $\mathcal{E}^c_{i,l,f}$ denote the required energy to cache content f in node l. The cache energy consumption $\mathcal{E}^c_{i,l,f}$ can also be defined based on the occupied cache size, as below:

$$\mathcal{E}^c_{i,l,f} = \mathcal{S}_f \times \omega_c \tag{5.5}$$

where ω_c denotes the required energy by the hardware in *watt/bits*, while $\mathcal{S}_f$ defines the occupied cache size. We also denote x_l^f as the cache decision variable:

$$x_l^f = \begin{cases} 1 & \text{if node } l \text{ decides to cache the content } f. \\ 0 & \text{else.} \end{cases} \tag{5.6}$$

Hence, let $\mathcal{T}^{\mathcal{E}}$ represent the caching energy required as follow:

$$\mathcal{T}^{\mathcal{E}} = x_i^f \; \mathcal{E}^c_{i,l,f} \tag{5.7}$$

Global Energy Consumption: Finally, the total IoT-Net energy consumption can be calculated as follow:

$$\mathcal{E}\mathcal{E} = \sum_{l \in \mathcal{L}} \sum_{f \in \mathcal{F}} \mathcal{T}^{\mathcal{T}} + \mathcal{T}^{\mathcal{E}} \tag{5.8}$$

5.3.2 Problem Formulation

From the ICN perspective, the total consumed energy is defined by the sum of both transport energy and cache energy. The former includes the energy consumption to forward the content from the original producer to the cache-store(s) as well as from the cache-store(s) to the consumer(s). The latter relies on the wasted energy to cache the content, and it depends on hardware technology (e.g., SSD, TCAM, DRAM, etc).

Our concern is then to extend energy efficiency by minimizing the total consumed energy, which is described as follow:

$$\min \ \mathcal{EE} \tag{5.9}$$

Subject to:

$$\sum_{i \in AT} k_l^f \, \mathcal{E}_{i,l,f} \leq \mathcal{Q}_{max}, \qquad \forall l \in \mathcal{L} \tag{5.10}$$

$$\sum_{l \in \mathcal{L}} k_l^f \, \mathcal{D}_{i,l,f} \geq \mathcal{D}_{th}, \qquad \forall i \in AT \tag{5.11}$$

$$\sum_{f \in \mathcal{F}} x_l^f \, \mathcal{T}_l \, \mathcal{S}_f \leq \mathcal{C}_{max}, \qquad \forall l \in \mathcal{L} \tag{5.12}$$

$$\sum_{e=1}^{|ET|} k_l^f \leq M, \qquad \forall f \in \mathcal{F} \tag{5.13}$$

$$x_l^f \in \{0,1\}, \qquad \forall l \in \mathcal{L} \tag{5.14}$$

$$k_l^f \in \{0,1\}, \qquad \forall l \in \mathcal{L}, \forall f \in \mathcal{F} \tag{5.15}$$

Eq. (5.9) represents the global objective function that aims to minimize the overall energy consumption of the network. Constraint (5.10) indicates that the energy to fetch the content f should be bounded by the maximum defined transmission energy $\mathcal{Q}_{max}$ (either from wired or wireless transmission capacity or defined QoS border). While constraint (5.11) represents that the data rate to cache a file f should be greater than the threshold $\mathcal{D}_{th}$. Constraint (5.12) indicates that the cache storage must not go beyond the maximum storage capacity. Similarly, constraint (5.13) indicates that a file f can be cached at maximum on M cache stores. Finally, constraints (5.14) and (5.15) imposes the non negativity on node caching decision and cache storage reachability, respectively.

5.4 EaCP: Energy-aware Cache Placement Scheme

The high diversity of real-world content embedded in the IoT networks requires considering different factors that influence and impact energy saving while ensuring a high energy efficiency level. In this section, we describe the proposed caching strategy, namely EaCP.

Basic Metrics In this part, we highlight the basic metrics used by EaCP.

- *Content Popularity Ratio* (CPR): In IoT networks, content popularity varies over time. Thus, let ζ_i^f denote the popularity value of content f at node i. The CPR is obtained by dividing the received requests for content f, by all received requests at the node i, in a defined time period, which described as follow:

$$\zeta_i^f = \frac{\text{Number of received requests for} f}{\text{Total of all received requests}} \tag{5.16}$$

- *User Proximity Ratio* (UPR): This value indicates the node's position in the network. UPR outlines the relationship linking the content size, node degree (number of neighbors requesting the content), and the sum of the arrival demands, which is described in the following equation:

$$\nu_i^f = \frac{i^\partial}{\sqrt{\varsigma_r^i \cdot \varsigma_r^p}} \tag{5.17}$$

- *Content Caching Reward* (CCR): This value is used to extract the reward value relative to client satisfaction to keep the content cached. CCR aims at making a balance among the content cache size, user satisfaction, and node importance, as described in the following equation:

$$\varpi_i^f = \sqrt{\frac{i^\partial * \zeta_i^f}{\mathcal{S}_f \, . \, \zeta_i^f}} \tag{5.18}$$

ϖ denotes the content caching reward, and i^∂ denotes the node's neighborhood degree.

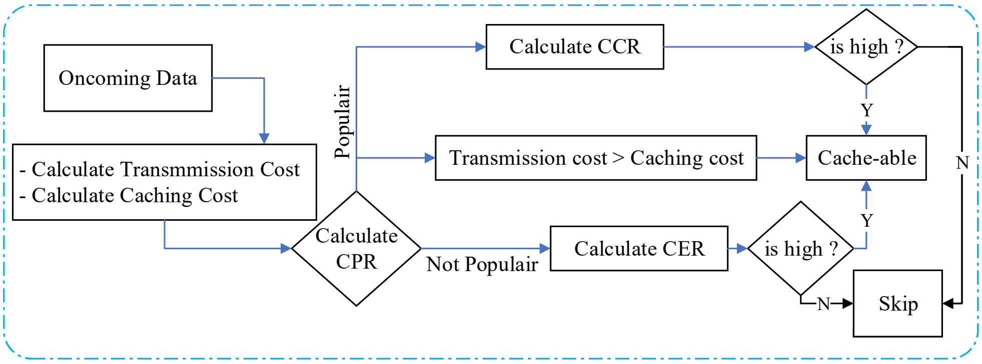


Figure 5.2: EaCP working mechanism.

- *Content Energy Reward* (CER): This metric aims to generate the reward energy value of content in the case of a caching decision. As shown in Eq. 5.19, CER tends to balance the content fetching energy, the caching energy, and user demand satisfaction by prioritizing the most requested content.

$$\psi_i^f = 1 - e^{\frac{\mathcal{T}^{\mathcal{T}} * \zeta_i^f}{\mathcal{T}^{\mathcal{E}}}} \tag{5.19}$$

Working Mechanism In this part, we discuss the fundamental pieces of EaCP scheme. For simplicity purposes, we divide its working mechanism into multiple steps, as depicted in Fig. 5.2 and discussed below.

- *Step 1.* The process starts by collecting the node-local information, including the content-cache capacity cost, received demands for the content, and node importance. Then, it obtains the neighbor-nodes-local information (neighbors for each received request for the candidate content).

- *Step 2.* It aims to compute both of the required transmission energy to fetch the content from the cache-store K or the original producer P, and calculate the required energy to cache content as represented in Eq. (5.4), and Eq. (5.7), respectively.

- *Step 3.* The process starts by calculating the popularity ratio (this mechanism aims to specify if the content is popular or not by comparing the CPR value to the defined threshold at the concerned node), then matches the content in multiple possible outputs in order to handle them on the next phase.

- *Step 4.* It tends to treat the different cases related to the caching decision. The process starts by marking if the content is cache-able or not based on different factors.
  - Case 1: If the fetching energy is higher than the caching energy and the content is popular, then mark the content as cache-able. The node with the maximum UPR is cached.
  - Case 2: If the content is not popular and the transmission energy is low than the caching energy, then calculate the content caching reward. The node with the maximum CCR is cached.
  - Case 3: If the content is not popular and the transmission energy is high than the caching energy, we use the energy reward to decide the content caching where the node with maximum CER is selected.
  - Case 4: If the content is not popular and the caching energy is high than the fetching energy, we do not cache the content.

Due to the massive amount of content generated by IoT devices, the EaCP scheme fits on multi-stage filtering rules, where the simplicity behind the EaCP mechanism aims to make fluent and optimal cache decisions with low complexity and overhead.

5.5 Performance Evaluation

In this section, we describe the performance evaluation of the EaCP caching scheme against the most used caching strategies.

5.5.1 Simulation Setup

We have implemented the EaCP scheme using the Python programming language. We have used the Barabasi network as a network core that connects with multiple groups of IoTs networks. The distribution of IoT nodes among these networks follows Normal Distribution low. We have also used Zipf distribution, which is suitable for this scenario, to model content requests' density. Table 5.2 provides a summary of the simulation setup as well as the used parameters.

Table 5.2: EaCP Simulation Parameters.

Parameter	Value
Size of content	[512 Byte, 9 Kilobyte]
Number of unique contents	[100, 1000]
Zipf parameter (α)	[1.1, 1.5]
Number of node	[20, 600]
Number of IoT-Net	[4, 40]
Memory power efficiency	6.25×10^{-12} (watt/bit)
Transport equipment (Core Nodes)	1.7×10^{-8} (Joule/bit)
Transport equipment (Gateway nodes)	1.38×10^{-7} (Joule/bit)
Transport equipment (Links)	5×10^{-9} (Joule/bit)
Transport equipment (Edge Nodes)	2.63×10^{-8} (Joule/bit)

5.5.2 Evaluation Metrics

- Cache Utilization: measures how many times the same content has been duplicated in the network's cache-store.
- Cache Hit Ratio: measures how many times a received request has been satisfied by the local node's cache-store than the overall received requests.
- Energy Saving: measures the saved energy by calculating the difference between the energy consumption if the content is served from the original producer or a replica cache-store.
- Energy Efficiency: shows the percentage of the saved energy compared to the consumed energy during the cache replacement process.

5.5.3 Results and Discussion

To show the proposed strategy's efficiency, we have bench-marked EaCP against popular caching schemes in ICN. We have also conducted simulations considering different aspects such as memory usage and energy efficiency. In doing so, we first started with a fixed content size and then extended it to a large-scale scenario.

Identical Content Size Experiment. Fig. 5.3 shows the cache utilization performance when the number of IoT nodes increases. We can observe the number of data copies constantly increasing in each of the strategies. Since

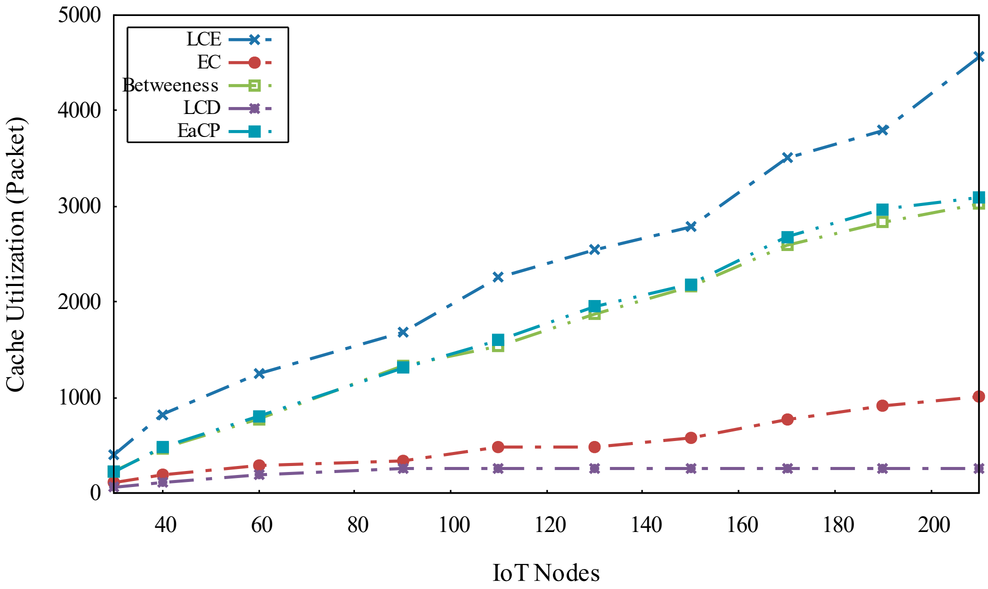
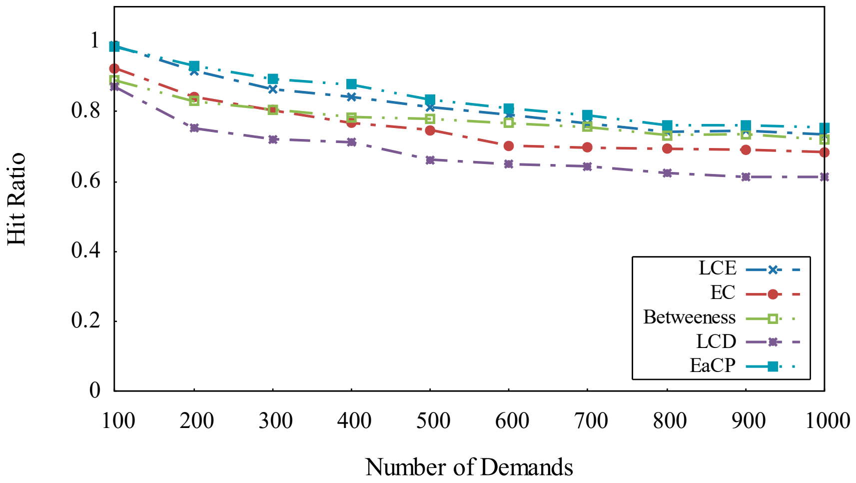


Figure 5.3: Cache Utilization.

LCD caches the content only on one node after the producer, we can notice that the cache utilization increases slightly. Similarly, EC has more extensive performance but imperceptibly different since we might have multiple edge nodes (based on the network topology). Additionally, LCE has the worse cache utilization since it caches content (high duplicated copies) along the communication path. Meanwhile, EaCP and Betweeness have similar output since they select only nodes with the optimal placement. This is because of the identical content size used in this experiment where it is an essential metric in deciding the caching energy reward in EaCP.

Fig. 5.4 depicts the average of the cache hit ratio when the number of requests increases. Both LCD and EC show a lower hit ratio performance due to the small number of involved nodes to cache content. Betweenness and LCE present considerable improvement. Besides that, EaCP shows better hit ratio improvements.

Large-scale Network with Content Size Variation. In order to study the performance of the proposed scheme in a large-scale network, we have increased the number of nodes and used a content size variation instead of identical size. Here, the prime objective is to show the caching scheme's real performance in terms of energy usage and energy efficiency.



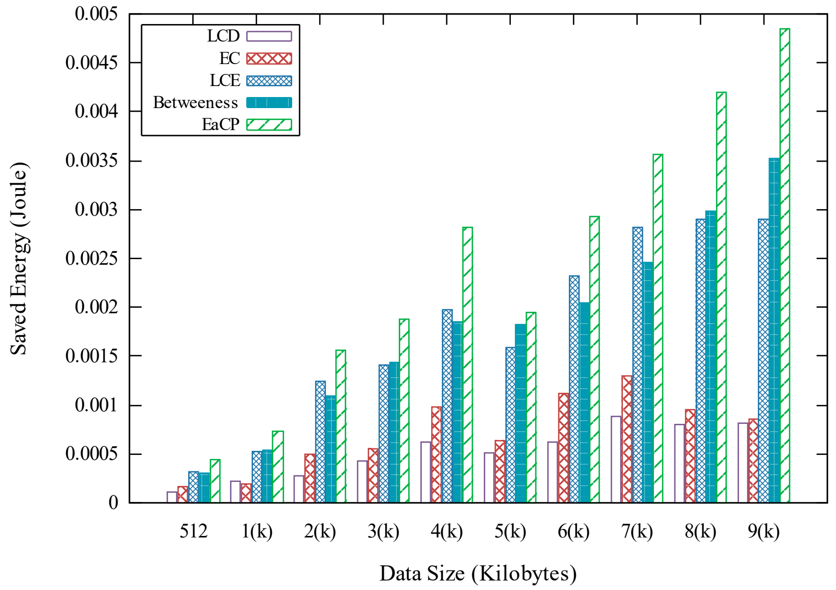


Figure 5.5: Energy Saving.

Fig. 5.5 shows the energy-saving ratio against the rising of content size. We observe that the energy-saving varies for each data size value. EC and LCD record the lowest energy saving among the rest of the strategies. Along with it, we observe that for the first content size LCE, Betwennes and EaCP show almost a little saving energy value and this is due to the small content size that improves the content available at the network. However, when the content size starts increasing, EaCP gives a high rise in energy saving.

Fig. 5.6 shows the obtained results for energy efficiency. We observe that LCE

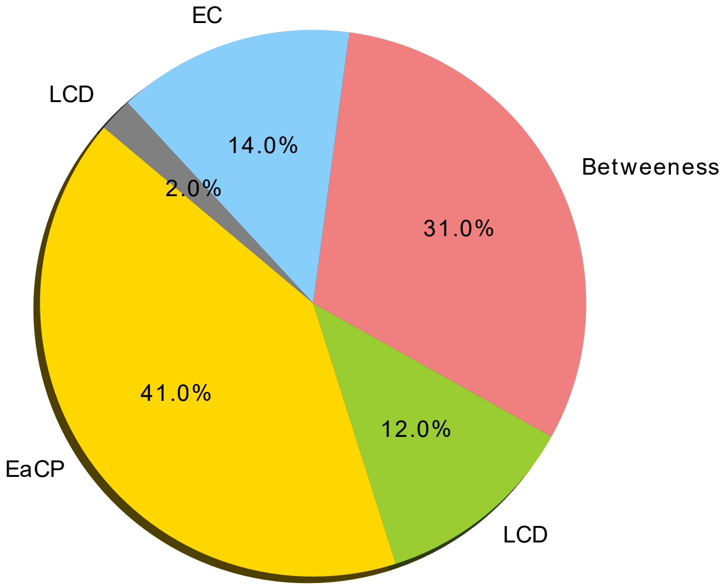


Figure 5.6: Energy Efficiency.

shows almost non-existent energy efficiency. Indeed, it performs a large replacement execution, which diminishes its efficiency. EC and LCD use a limited number of nodes that reflect to limited energy consumption. Furthermore, Betwennes shows considerable efficiency, while EaCP is more efficient and outperforms the other strategies. This proves that EaCP has a notable achievement as it is able to save energy with low overhead.

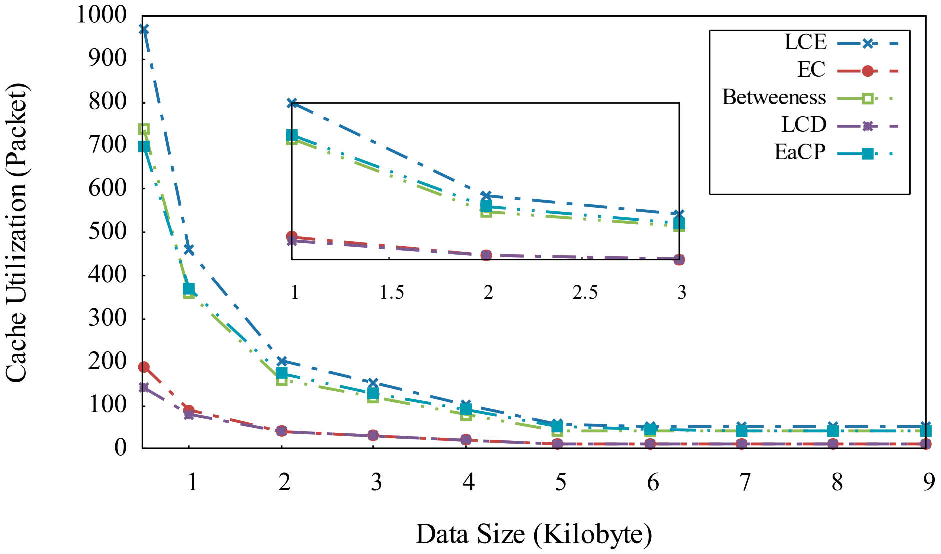


Figure 5.7: Cache utilization per data size.

Fig. 5.7 depicts the cache utilization against the rising of data size in a fixed number of network nodes along with the increase of the content size. Due to the cache-store's limited memory size, we observe a continuous decline in the data replication for all of the strategies. More specifically, we notice that LCD and EC start with low data replication and diminishing until stabilizing. In this scenario, EaCP shows a low data replication compared to LCE and Betweenness and reviles a wise memory usage.

5.6 Conclusion

In this chapter, we designed EaCP strategy in order to improve the energy efficiency of IoT-based ICN 5G networks. EaCP tends to balance the caching cost and the transmission cost in order to achieve a better content placement position. To doing so, we considered content transmission and caching energy, the node position. and content popularity. EaCP algorithm aims to achieve the optimal placement setting in a distributed fashion. In addition, we conducted an in-depth simulation considering various metrics. The obtained results proved the effectiveness of our schemes in terms of energy-saving, cache utilization, and considerable improvement in the cache hit ratio. We believe that the EaCP scheme is able to extend further the battery lives for IoT devices while strongly reduce the communication overhead for 5G based ICN networks. In the next chapter, we will focus on the mobility issues within the 5G-ICN network.

Chapter 6

A Label-based Producer Mobility Support Architecture

6.1 Introduction

With the manner increase in the number of wireless devices, mobility management became one of the fundamental requirements for future networks. In particular, the content in 5G-ICN is first-class citizens instead of the host. ICN names the content rather than the owner, and the demands are driven by the receiver. Although these changes contribute to the ease of consumers' mobility, producer mobility is considered a challenging issue in ICN.

This chapter is organized as follows; we first exhibit the existing issues and challenges related to mobility in ICN. Then, we describe in detail the proposed label-based producer mobility support architecture. Finally, we present the implementation and experimental results as well as analysis part.

6.2 Mobility Requirements and Challenges

Regardless of the existing solutions (see in chapter 2 section 2.7) aimed to develop proper ICN mobility support, mobility in ICN still needs more effort. In the following, we will summarize existing issues and challenges.

- Consumer Mobility Issues: Consumer mobility support is usually supported by design in ICN. The consumer is able to re-issue any unsatisfied interest during the mobility. However, considering 5G networks and due to the continuous handover of the consumer between different access points, the time-sensitive applications may be affected. In which, a fixed and stable session communication may not guarantee due to the mobility of the consumers at network heterogeneity.

- Producer Mobility Issues: The producer mobility remains an open issue in ICN. The producer must dynamically update all forwarding rules in all intermediate nodes FIB tables or the name resolution database in order to keep its content reachable after the movement.
- Cache-Store Mobility Issues: Cache-stores are non-control-able entities. Any node can cache the content and move from a network to another. The mobility of cache-stores is as important as the mobility of producer, otherwise, the benefit of in-network caching will be ignored.
- Name Related Issues: The use of location-based names to identify content may not be suitable for mobility. When a node moves from a network to another one and get attached with another routers, the content name might be updated according to the recent path. This mechanism is not always preferable as the producer may be active in mobility or can move back to his home network. The use of non-location based names is more recommended.
- Architecture Related Issues: Some features provided by the architecture cannot help mobility support. For instance, interest aggregation can help to reduce the load on the network by aggregating interest upstream, yet it may lead to loss of content delivery if the producer moves or the requests arrived in a disjointed manner.

In light of these issues and challenges, we present in the following section, our proposed LbPM architecture that aims to tackle the producer mobility issue.

6.3 Label-based Producer Mobility Support

In Interest-based ICN architectures, a consumer requests content in form of an Interest packet using only the name of the content. The network is aware of routing the request to the appropriate cache-store or original producer, and delivering the Data packet to the consumer without knowing its location. The use of location-based naming remains an open challenge in terms of mobility when the producer changes its location. In the following, we present Label-based Producer Mobility Support Architecture that tends to overcome this issue.

6.3.1 Problem Description

Let's consider the following scenario, depicted in Figure 6.1. The content D is generated only by the producer P, which is in a moving state. According to the popularity of the generated content, different consumers may request the same content. Let C1 and C2 two consumers connected via two different networks (e.g., heterogeneous networks). It is important to highlight that the Interest request is aggregated at the first common router (ICN design primitive). If the request is somehow lost or cannot successfully reach the producer due to its mobility state, both consumers will miss the content. Therefore, we find two cases during the handover of producer:

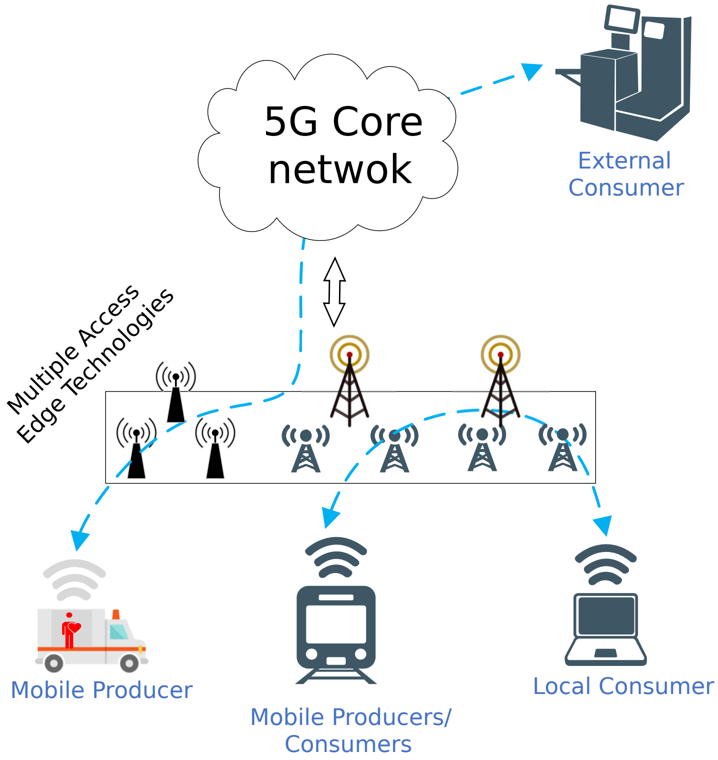


Figure 6.1: Producer Mobility Issues.

- Interest Missing during Handover: The Interest packet is forwarded hop-by-hop using intermediate nodes' FIB rules to reach the producer and kept tracing the path in the PIT table. However, and due to producer mobility phenomena, the Interest packet will be lost in the network and won't reach the producer. This issue appears during the handover of the producer. In this case, we have invalid FIB entries, useless PIT entries, and fail in request/content delivery.
- Interest Succeed Delivering Before Handover: The Interest packet successfully reaches the producer according to FIB rules before its mobility. However, and just before delivering the content back to the consumer, the producer moves to another place. In this case, we have valid FIB entries, useless PIT entries, successful interest forwarding, but fail in content delivery.

In either case, the name of the content has a direct impact on request forwarding and content delivery. It is worth noting that using location-based names requires the system to update all FIB's and PIT's entries in the whole network, which is not a feasible solution due to the number of contented devices in 5G networks. Therefore, we recommend using label-based names that do not require updates in the names but only in the forwarding interfaces.

6.3.2 Label-based Naming Scheme

In this work, we are not designing a new naming scheme but using label-based names to solve the mobility issue. Instead of applying a new naming scheme to the whole network, we extend the FIB table with a new field, called *label*. This label is only a new representation of the location-based name. Here, we apply a hashing function (e.g., CRC, MD5, etc.) on the original name to generate the label. Then, a new routing update is triggered to update the FIB tables with the new label and reaching interface. Figure 6.2 depicts the whole process.

FIB Updates The FIB consists of a set of fields including the name prefix and the outgoing interface (next hop) to reach the content. By adding the label field, the routing updates will contain the original name prefix, the label, reaching interface, and a timer to indicate the period of time to keep the interface active

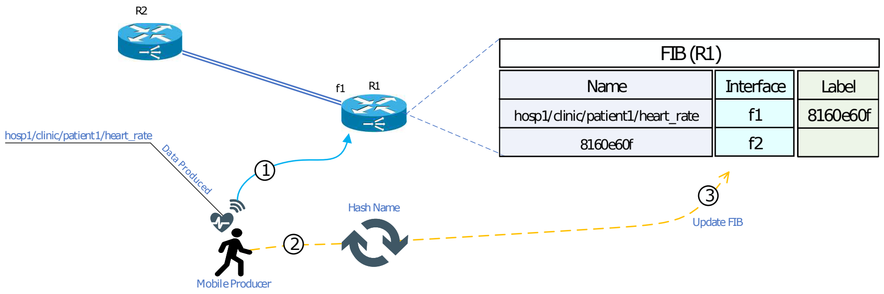


Figure 6.2: Label generation and FIB updating process.

in the FIB. The proposed LbPM can work on top of any named routing protocol (e.g., Named OSPF or Named LSP) [186].

6.3.3 Interest Forwarding Scheme

After the handover and propagation of the new label and the associated interface, the Interest packet is forwarded according to the new label and not the original name. Algorithm 1 summaries the forwarding process, and explained as follows:

Algorithm 3: LbPM forwarding scheme.

```
if (content in CS) then
    Forward Data;
else if (name in PIT) then
    Interest Aggregation;
else if (name is FIB) then
    if (label) then
        Lookup for label in FIB;
        Label-based forwarding;
    else
        name-prefix-based forwarding;
    end
else
    Drop Interest;
end
```

The consumer node has no information about the generated label, it always uses the original content name. When an intermediate node receives a new Interest packet, it performs a regular CS and PIT table as in native ICN architecture. The only change is in the FIB table, in which it checks if the name in the entry has a label or not. The forwarding process always prefers the label than the original name. After knowing the associated label, it performs a lookup in the FIB to find an outgoing interface to reach the label, which means reach the mobile producer.

6.4 Implementation and Evaluation

In this section, we describe the performance evaluation of our proposed architecture against NDN in a large-scale network topology.

6.4.1 Simulation Setup

We have implemented LbPM using Python 3 programming language. We have used *Cogent*[1], a large-scale network topology that connects most of the European countries with the USA. The network consists of 190 routers placed over Europe and USA cities. Table 6.1 provides a summary of the simulation setup.

Table 6.1: LbPM Simulation Parameters.

Parameter	Value
Number of Routers	190
Number of Access Point	50
Number of Interest	[100 - 1000]
Mobility Model	3D Random Mobility
Number of Simulation	20

In order to adapt our proposal solution to the almost real-world scenario, especially for 5G networks where the producer mobility can be either between different access points within the same network (between SBSs) or from a network to another one (in/between RANs). We use the random mobility model to reflect the global behavior of producers' mobility. Other mobility model can also be used in our architecture. We also use two types of mobility (i.e., low and high mobility), in which the connectivity with an access point can be short of period or long accordingly.

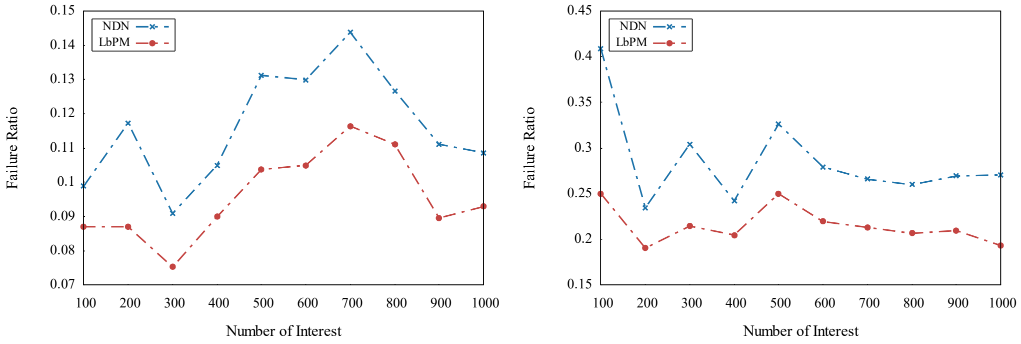
---

[1] The Internet Topology Zoo: www.topology-zoo.org

6.4.2 Evaluated Metrics

To evaluate the performance of LbPM, we have measured different metrics including: failure ratio and end-to-end network latency.

- Failure Ratio: refers to the number of missed interest over the sum of all issued interest sent to get the content. In the simulation, we use the average of all consumers over all producers.
- End-to-End Network Latency: indicates the latency from sending the first interest till receiving the content. In the simulation, we calculate the average latency for all demands.



(a) Low Producer Mobility.

(b) High Producer Mobility.

Figure 6.3: Failure Ratio Performance.

6.4.3 Results & Discussion

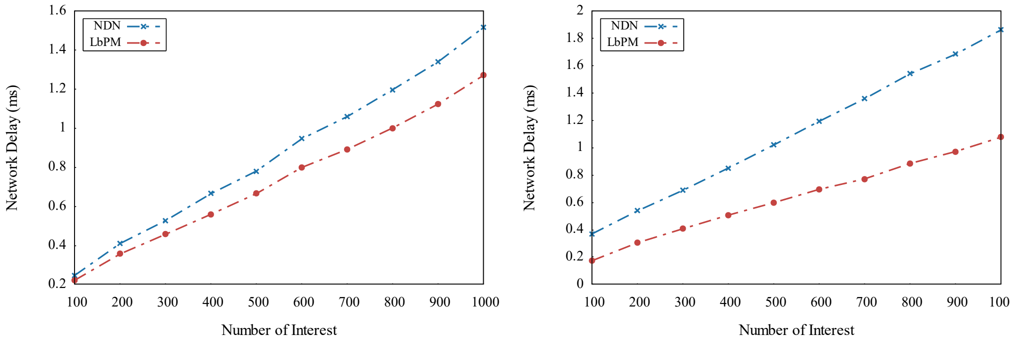
We evaluate LbPM using two different scenarios. The handover and switching speed of mobile producer vary from w low to a high speed.

Low Speed Mobility Scenario In this scenario, the mobile producer changes its access point with a low probability value. Which means, it may connect with an access point for a long period of time. Figures 6.3a and 6.4a show the evaluation results of LbPM with a low mobility scenario in terms of Interest failure Ratio and Network Delay respectively.

Figure 6.3(a) shows the number of Interest sent by consumers towards a mobile producer. The producer mobility is initiated with low movement. Hence, by the increase in the number of requests, NDN records a considerable failure ratio as most of the Interest packets have not been successfully satisfied, where LbPM shows a low failure ratio due to the fact that Interest packets are forwarded accordingly. Similarly, Figure 6.4(a) shows the network delay for interest forwarding until receiving the content. We observe the network delay rises with increasing the number of Interest. NDN uses the default forwarding/flooding scheme in order to resend the unsatisfied Interests which reflects on the high network delay to retrieve and get the content. Besides, LbPM records a low delay ratio compared with default NDN in the low mobility scenario.

High Speed Mobility Scenario In this scenario, the mobile producer moves between attachment points with a high speed, frequently and randomly. Figures 6.3(b) and 6.4(b) show the performance of failure ratio and network delay with a high speed respectively.

Figure 6.3(b) displays the number of Interest sent by consumers towards a mobile producer but with high mobility, which means the probability of being connected with an access point is lower due to its speed. Thus, by the increase in the number of requests, NDN records a significant failure ratio as requests are lost during the mobility and cannot find the producer, while LbPM exhibits a low failure ratio. Some failure in LbPM is due to the high speed of movement, yet it is extremely less compared to NDN. In the same way, Figure 6.4(b) shows the network delay against the number of Interest sent by the consumer towards the producer node that moves with high speed. We observe that NDN records a considerable delay when the number of Interest increases, meanwhile LbPM exhibits a low network delay.



(a) Low Producer Mobility.

(b) High Producer Mobility.

Figure 6.4: Network Delay Performance.

6.5 Conclusion

In this chapter, we studied the issue of producer mobility in 5G-based ICN networks. We first reviewed several solutions, identified different issues and challenges in this area, and then designed a producer mobility architecture based on the Label naming scheme, namely LbPM. We used this naming scheme in order to provide location-free producer mobility without changing the location-based hierarchical names. Without exploring extra resources or designing a complex protocol, LbPM exploited the naming feature that is mainly the fundamental building block of ICN. We evaluated our architecture using different scenarios, the obtained results confirmed the effectiveness of our solution in terms of network delay and successful Interest/Data delivery during producer mobility.

Chapter 7

Conclusions and Perspectives

The current host-centric model still faces various difficulties, such as scalability, mobility, security, etc., primarily due: the sharp increase in the number of devices and variety of the new emerging use cases, the nature of applications, and the focus for content by the users instead of the data location. In contrast, ICN expanses add extra network functionalities and features in order to promote different additional functions and semantics (e.g., mobility support, security, caching, etc.). The 5G-based ICN network will bring new Internet usage and vision based on a world built on fast communication. 5G can be realized based on multiple new technologies, while ICN helps in realizing a clean and efficient for this next-generation network. The content-centric paradigm gives a fresh perspective to communication, by eliminating the need to find a specific destination host in the network.

Summary of Our Contributions:

In this thesis, we presented and discussed the use of ICN as a communication enabler for the 5G network. We gave an overview of the 5G ecosystem and the ICN paradigm and discussed the applicability of ICN in 5G from different perspectives. Subsequently, we presented state-of the-art solutions in the area.

In the first contribution, we focused on distributed caching management, and we proposed Cache and Splite Scheme for 5G based ICN network. The scheme integrates different mechanisms and functionalities, including self node decision making; it also considers the content characteristics such as content popularity, the value of content freshness, distance factor, and demand rate. The proposed scheme dynamically decides the content cache or split the decision to downstream cache-stores. Obtained results prove the efficiency and scalability of our scheme in terms of network delay, cache utilization, and hop reduction ratio.

Our second contribution addressed massive growth connected devices and their huge generated data, we designed large-scale content caching schemes. We presented the problem formulation as well as algorithm adaptation. The simulation results showed a considerable improvement, especially in terms of resource utilization, data transmission time, content diversity ratio, and the number of cache replacement operations.

In the third contribution, designed and implemented the producer mobility support for the 5G-ICN network. We highlighted the mobility related challenges, and then proposed a label based producer mobility support. This solution proves its effectiveness by reducing network congestion and enhances end-to-end content delivery.

Perspectives and Future Directions:

Despite the existing 5G-ICN solutions proposed in the literature and our proposed solutions in this work, there are still limitations and challenges in the area of smooth integration between 5G ecosystem and ICN paradigm. In the following, we list some perspectives that can give potential enhancements to our study and future research directions for 5G based ICN networks.

Focus on content naming: Content naming provides content identification and a routing mechanism, which leads to improving content retrieval and enhancing the security level. However, ICN naming may have several issues related to the retrieval of real-time applications, including multimedia that requested by millions of users every day. Potential solutions such as enhancing the naming with attribute-value based names can help to solve this issue.

Focus on data caching: To perform adequate caching management, matching user preferences, behaviors, and context is indispensable to ensure a reliable caching decision. Thus, system complexity could be very high. Deep Reinforcement Learning is an advanced machine learning technique that is considered a promising technology to handle a large amount of input data [187]. Such a solution may help to provide distributed and scalable caching schemes. Besides, caching the frequently requested content (i.e., popular content) without considering additional parameters (e.g., content size) leads to retention and accumulation of small-sized content. Consequently, it impacts the global network caching performance, reduces caching capacity, overloads the lookup process, and adds extra weights to the content store. Deep learning solutions can help to predict content popularity by combing multiple attributes.

Focus on mobility: User mobility is an essential aspect of the next 5G networks, especially for time-sensitive applications. Although ICN has native support for user mobility, which may provide a seamless user movement, the frequency change of the user attachment point between multiple access technologies may impact the QoE as well as QoS, which leads to service degradation. Furthermore, the main drawback appears if the same user act as a producer (that produces a certain content). In such a case, the routing based naming process occurs a flooding Interest for path recovery. Thus, it increases the network traffic, which leads to the network conjunction. In this regard, various works have been done concerning producer mobility issue. However, none of them uses a flexible naming scheme that supports multiple features, e.g., mobility support, security enhancement, scalability, etc.

Focus on emerging use cases and application: 5G-based ICN networks tend to support multiple applications use cases. From this perspective, a few efforts consider improving the naming schemes targeting specifics 5G applications and use cases. For instance, some IoT actuators and sensors probably request content with a larger name length, which affects the routing tables and lookup performance. Therefore, a trade-off between naming schemes and the data size should be considered. Applying encoding schemes can be a potential solution without losing name semantic or structure.

Focus on In network Computation: The ICN-assist MEC approach can help to enhance content caching and computation at the same level. A new research question arises regarding the feasibility of cache computation of previously executed services and not only content. This will help to avoid multiple computations and, in result, enhance end-to-end delay. The computation reuse paradigm can be applied at the MEC server without violating ICN primitives.

Bibliography

[1] P. Curwen and J. Whalley. "5g literature review". In: *Understanding 5G Mobile Networks*. 2021. DOI: `10.1108/978-1-80071-036-820210003`.

[2] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman. "A survey of information-centric networking". In: *IEEE Communications Magazine* 50.7 (2012). DOI: `10.1109/MCOM.2012.6231276`.

[3] C. Gündoğan, P. Kietzmann, T. C. Schmidt, and M. Wählisch. "Information-Centric Networking for the Industrial Internet of Things". In: *Wireless Networks and Industrial IoT*. Springer, 2021, pp. 171–189. DOI: `10.1007/978-3-030-51473-0_9`.

[4] O. Waltari and J. Kangasharju. "Content-centric networking in the internet of things ". In: *Consumer Communications & Networking Conference (CCNC)*. IEEE. 2016, pp. 73–78. DOI: `10.1109/CCNC.2016.7444734`.

[5] I. Turcanu, T. Engel, and C. Sommer. "Adaptive content seeding for information-centric networking under high topology dynamics". In: *IEEE Vehicular Technology Magazine* 16.2 (2021), pp. 68–75. DOI: `10.1109/MVT.2021.3050728`.

[6] 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: *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*. 2007, pp. 181–192. DOI: `10.1145/1282427.1282402`.

[7] D. Lagutin, K. Visala, and S. Tarkoma. "Publish/subscribe for internet: Psirp perspective". In: *Towards the Future Internet*. IoS Press, 2010, pp. 75–84. DOI: `10.3233/978-1-60750-539-6-75`.

[8] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren, and H. Karl. "Network of information (netinf)–an information-centric networking architecture". In: *Computer Communications* 36.7 (2013), pp. 721–735. DOI: `10.1016/j.comcom.2013.01.009`.

[9] D. Perino and M. Varvello. “A reality check for content centric networking”. In: *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*. 2011, pp. 44–49. DOI: 10.1145/2018584.2018596.

[10] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang. “Named data networking”. In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), pp. 66–73. DOI: 10.5958/2319-6890.2017.00040.X.

[11] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, and C. Papadopoulos. “Named Data Networking (NDN) Project”. In: *Technical Report NDN-0001, Xerox Palo Alto Research Center-PARC* (2010).

[12] B. Nour, H. Khelifi, R. Hussain, S. Mastorakis, and H. Moungla. “Access Control Mechanisms in Named Data Networks: A Comprehensive Survey”. In: *ACM Computing Surveys* 54.3 (Apr. 2021). IF: 8.959, Q1. DOI: 10.1145/3442150.

[13] K. Venkateswararao, P. Swain, C. Christophorou, and A. Pitsillides. “Using UE-VBS for dynamic virtual small cells deployment and backhauling in 5G Ultra-Dense networks”. In: *Computer Networks* 189 (2021), p. 107926. DOI: 10.1016/j.comnet.2021.107926.

[14] O. Serhane, K. Yahyaoui, B. Nour, and H. Moungla. “A survey of ICN content naming and in-network caching in 5G and beyond networks”. In: *IEEE Internet of Things Journal* (2020). DOI: 10.1109/JIOT.2020.3022243.

[15] O. Serhane, K. Yahyaoui, B. Nour, and H. Moungla. “CnS: A Cache and Split Scheme for 5G-enabled ICN Networks”. In: *IEEE ICC Conference*. IEEE, June 2020. DOI: 10.1109/ICC40277.2020.9149332.

[16] O. Serhane, K. Yahyaoui, B. Nour, and H. Moungla. “Energy-aware Cache Placement Scheme for IoT-based ICN Networks”. In: *IEEE International Conference on Communications (ICC)*. 2021. DOI: 10.1109/ICC42927.2021.9500341.

[17] O. Serhane, K. Yahyaoui, B. Nour, and H. Moungla. “A Label-based Producer Mobility Support in 5G-enabled ICN Networks”. In: *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE. 2020, pp. 2094–2099. DOI: 10.1109/IWCMC48107.2020.9148578.

[18] N. Alliance. “5G white paper”. In: *Next generation mobile networks, white paper* 1 (2015).

[19] F. Messaoudi, P. Bertin, and A. Ksentini. “Towards the quest for 5G Network Slicing”. In: *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE. 2020, pp. 1–7. DOI: `10.1109/CCNC46108.2020.9045327`.

[20] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin, et al. “MEC in 5G networks”. In: *ETSI white paper* 28 (2018), pp. 1–28. DOI: `10.1109/ICUFN.2018.8436806`.

[21] D. S. Rana, S. A. Dhondiyal, and S. K. Chamoli. “Software defined networking (SDN) challenges, issues and solution”. In: *International journal of computer sciences and engineering* 7.1 (2019), pp. 884–889. DOI: `10.26438/ijcse/v7i1.884889`.

[22] I. Farris, T. Taleb, Y. Khettab, and J. Song. “A survey on emerging SDN and NFV security mechanisms for IoT systems”. In: *IEEE Communications Surveys & Tutorials* 21.1 (2018), pp. 812–837. DOI: `10.1109/COMST.2018.2862350`.

[23] B. Nour, S. Mastorakis, and A. Mtibaa. “Compute-Less Networking: Perspectives, Challenges, and Opportunities”. In: *IEEE Network* 34.6 (2020). IF: 8.808, Q1, pp. 259–265. DOI: `10.1109/MNET.011.2000180`.

[24] 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”. In: *IEEE Communications Surveys & Tutorials* 16.2 (2013), pp. 1024–1049. DOI: `10.1109/SURV.2013.070813.00063`.

[25] Q. C. Li, H. Niu, A. T. Papathanassiou, and G. Wu. “5G network capacity: Key elements and technologies”. In: *IEEE Vehicular Technology Magazine* 9.1 (2014), pp. 71–78. DOI: `10.1109/MVT.2013.2295070`.

[26] B. Nour, K. Sharif, F. Li, S. Yang, H. Moungla, and Y. Wang. “ICN publisher-subscriber models: Challenges and group-based communication”. In: *IEEE Network* 33.6 (2019), pp. 156–163. DOI: `10.1109/MNET.2019.1800551`.

[27] H. Khelifi, S. Luo, B. Nour, H. Moungla, Y. Faheem, R. Hussain, and A. Ksentini. “Named data networking in vehicular ad hoc networks: State-of-the-art and challenges”. In: *IEEE Communications Surveys & Tutorials* 22.1 (2019), pp. 320–351. DOI: `10.1109/COMST.2019.2894816`.

[28] R. Ravindran, P. Suthar, A. Chakraborti, S. O. Amin, A. Azgin, and G. Wang. “Deploying ICN in 3GPP’s 5G NextGen Core Architecture”. In: *IEEE 5G World Forum*. IEEE. 2018, pp. 26–32. DOI: 10.1109/5GWF.2018.8517046.

[29] C. Liang, F. R. Yu, and X. Zhang. “Information-centric network function virtualization over 5G mobile wireless networks”. In: *IEEE Network* 29.3 (2015), pp. 68–74. DOI: 10.1109/MNET.2015.7113228.

[30] M. Amadeo, O. Briante, C. Campolo, A. Molinaro, and G. Ruggeri. “Information-centric networking for M2M communications: Design and deployment”. In: *Computer Communications* 89 (2016), pp. 105–116. DOI: 10.1016/j.comcom.2016.03.009.

[31] A. Martin, J. Egaña, J. Flórez, J. Montalban, I. G. Olaizola, M. Quartulli, R. Viola, and M. Zorrilla. “Network resource allocation system for QoE-aware delivery of media services in 5G networks”. In: *IEEE Transactions on Broadcasting* 64.2 (2018), pp. 561–574. DOI: 10.1109/TBC.2018.2828608.

[32] M. Series. “IMT Vision–Framework and overall objectives of the future development of IMT for 2020 and beyond”. In: *Recommendation ITU* 2083 (2015).

[33] Z. Li, M. A. Uusitalo, H. Shariatmadari, and B. Singh. “5G URLLC: Design challenges and system concepts”. In: *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE. 2018, pp. 1–6. DOI: 10.1109/ISWCS.2018.8491078.

[34] S. Mastorakis, A. Skiadopoulos, S. Shannigrahi, A. Likens, B. Nour, and N. Stergiou. “Networking and Computing in Biomechanical Research: Challenges and Directions”. In: *IEEE Communications Magazine* 59.6 (2021). IF: 11.052, Q1, pp. 103–109. DOI: 10.1109/MCOM.001.2000639.

[35] M. Patzold. “5G is coming around the corner [mobile radio]”. In: *IEEE Vehicular Technology Magazine* 14.1 (2019), pp. 4–10. DOI: 10.1109/MVT.2018.2884042.

[36] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang. “What will 5G be?” In: *IEEE Journal on selected areas in communications* 32.6 (2014), pp. 1065–1082. DOI: 10.1109/JSAC.2014.2328098.

[37] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications". In: *IEEE internet of things journal* 4.5 (2017), pp. 1125–1142. DOI: 10.1109/JIOT.2017.2683200.

[38] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. "A survey on mobile edge computing: The communication perspective". In: *IEEE Communications Surveys & Tutorials* 19.4 (2017), pp. 2322–2358. DOI: 10.1109/COMST.2017.2745201.

[39] Z. Piao, M. Peng, Y. Liu, and M. Daneshmand. "Recent Advances of Edge Cache in Radio Access Networks for Internet of Things: Techniques, Performances, and Challenges". In: *IEEE Internet Things Journal* (2018). DOI: 10.1109/JIOT.2018.2866709.

[40] B. P. Rimal, D. P. Van, and M. Maier. "Mobile edge computing empowered fiber-wireless access networks in the 5G era". In: *IEEE Communications Magazine* 55.2 (2017), pp. 192–200. DOI: 10.1109/MCOM.2017.1600156CM.

[41] S. Yi, Z. Hao, Z. Qin, and Q. Li. "Fog computing: Platform and applications". In: *2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb)*. IEEE. 2015, pp. 73–78. DOI: 10.1109/HotWeb.2015.22.

[42] M. Laroui, B. Nour, H. Moungla, M. A. Cherif, H. Afifi, and M. Guizani. "Edge and Fog Computing for IoT: A Survey on Current Research Activities & Future Directions". In: *Computer Communications* 180 (2021). IF: 3.167, Q1, pp. 210–231. DOI: 10.1016/j.comcom.2021.09.003.

[43] B. Omoniwa, R. Hussain, M. A. Javed, S. H. Bouk, and S. A. Malik. "Fog/edge computing-based IoT (FECIoT): Architecture, applications, and research issues". In: *IEEE Internet of Things Journal* 6.3 (2018), pp. 4118–4149. DOI: 10.1109/JIOT.2018.2875544.

[44] J. Pan and J. McElhannon. "Future edge cloud and edge computing for internet of things applications". In: *IEEE Internet of Things Journal* 5.1 (2017), pp. 439–449.

[45] B. Varghese and R. Buyya. "Next generation cloud computing: New trends and research directions". In: *Future Generation Computer Systems* 79 (2018), pp. 849–861. DOI: 10.1016/j.future.2017.09.020.

[46] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang. "Predictive offloading in cloud-driven vehicles: Using mobile-edge computing for a promising network paradigm". In: *IEEE Vehicular Technology Magazine* 12.2 (2017). DOI: 10.1109/MVT.2017.2668838.

[47] A. Filali, B. Nour, S. Cherkaoui, and A. Kobbane. "Communication and Computation O-RAN Resource Slicing for URLLC Services Using Deep Reinforcement Learning". In: *arXiv 2202.06439* (2022).

[48] B. Nour, S. Mastorakis, and A. Mtibaa. "Compute-less networking: Perspectives, challenges, and opportunities". In: *IEEE Network* 34.6 (2020), pp. 259–265. DOI: 10.1109/MNET.011.2000180.

[49] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. "The case for vm-based cloudlets in mobile computing". In: *IEEE pervasive Computing* 8.4 (2009), pp. 14–23. DOI: 10.1109/MPRV.2009.82.

[50] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young. "Mobile edge computing—A key technology towards 5G". In: *ETSI white paper* 11.11 (2015), pp. 1–16. DOI: 10.13140/RG.2.2.21415.06566.

[51] J. Dolezal, Z. Becvar, and T. Zeman. "Performance evaluation of computation offloading from mobile device to the edge of mobile network". In: *IEEE CSCN Conference*. IEEE. 2016, pp. 1–7. DOI: 10.1109/CSCN.2016.7785153.

[52] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang. "Cooperative content caching in 5G networks with mobile edge computing". In: *IEEE Wireless Communications* 25.3 (2018), pp. 80–87. DOI: 10.1109/MWC.2018.1700303.

[53] Y. Li and M. Chen. "Software-defined network function virtualization: A survey". In: *IEEE Access* 3 (2015), pp. 2542–2553. DOI: 10.1109/ACCESS.2015.2499271.

[54] I. Alam, K. Sharif, F. Li, Z. Latif, M. Karim, S. Biswas, B. Nour, and Y. Wang. "A Survey of Network Virtualization Techniques for Internet of Things using SDN and NFV". In: *ACM Computing Surveys* 53.2 (Apr. 2020). IF: 8.959, Q1. DOI: 10.1145/3379444.

[55] *Open Networking Foundation: Software-Defined Networking: The New Norm for Networks, ONF white paper (2012).*

[56] F. Hu, Q. Hao, and K. Bao. "A survey on software-defined network and openflow: From concept to implementation". In: *IEEE Communications*

Surveys & Tutorials 16.4 (2014), pp. 2181–2206. DOI: 10.1109/COMST.2014.2326417.

[57] H. Kim and N. Feamster. "Improving network management with software defined networking". In: *IEEE Communications Magazine* 51.2 (2013), pp. 114–119. DOI: 10.1109/MCOM.2013.6461195.

[58] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, et al. "Onix: A distributed control platform for large-scale production networks." In: *OSDI*. Vol. 10. 2010, pp. 1–6. URL: http://www.usenix.org/event/osdi10/tech/full%5C_papers/osdi10%5C_proceedings.pdf.

[59] K. Phemius, M. Bouet, and J. Leguay. "Disco: Distributed multi-domain sdn controllers". In: *2014 IEEE Network Operations and Management Symposium (NOMS)*. IEEE. 2014, pp. 1–4. DOI: 10.1109/NOMS.2014.6838330.

[60] A. Dixit, F. Hao, S. Mukherjee, T. Lakshman, and R. R. Kompella. "ElastiCon; an elastic distributed SDN controller". In: *ACM/IEEE SANCS*. IEEE. 2014, pp. 17–27. DOI: 10.1145/2534169.2491193.

[61] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, et al. "ONOS: towards an open, distributed SDN OS". In: *Workshop on Hot topics in software defined networking*. ACM. 2014, pp. 1–6. DOI: 10.1145/2620728.2620744.

[62] M. Agiwal, A. Roy, and N. Saxena. "Next generation 5G wireless networks: A comprehensive survey". In: *IEEE Communications Surveys & Tutorials* 18.3 (2016), pp. 1617–1655. DOI: 10.1109/COMST.2016.2532458.

[63] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar. "Making middleboxes someone else's problem: Network processing as a cloud service". In: *ACM SIGCOMM Computer Communication Review* 42.4 (2012), pp. 13–24. DOI: 10.1145/2342356.2342359.

[64] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines. "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges". In: *Computer Networks* 167 (2020), p. 106984.

[65] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee. "Network function virtualization: Challenges and opportunities for innovations". In: *IEEE Communications Magazine* 53.2 (2015), pp. 90–97. DOI: 10.35940/ijitee.D8483.0310521.

[66] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati. “Network function virtualization in 5G”. In: *IEEE Communications Magazine* 54.4 (2016), pp. 84–91. DOI: `10.1109/MCOM.2016.7452271`.

[67] L. Laaziz, N. Kara, R. Rabipour, C. Edstrom, and Y. Lemieux. “FASTSCALE: A fast and scalable evolutionary algorithm for the joint placement and chaining of virtualized services”. In: *Journal of Network and Computer Applications* 148 (2019), p. 102429. DOI: `10.1016/j.jnca.2019.102429`.

[68] S. OpenFlow-Enabled and N. F. Virtualization. *ONF Solution Brief*. 2014.

[69] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider. “NFV and SDN—Key technology enablers for 5G networks”. In: *IEEE Journal on Selected Areas in Communications* 35.11 (2017), pp. 2468–2478. DOI: `10.1109/JSAC.2017.2760418`.

[70] F. Zhang, G. Liu, X. Fu, and R. Yahyapour. “A survey on virtual machine migration: Challenges, techniques, and open issues”. In: *IEEE Communications Surveys & Tutorials* 20.2 (2018), pp. 1206–1243. DOI: `10.1109/COMST.2018.2794881`.

[71] B. Nour, K. Sharif, F. Li, S. Biswas, H. Moungla, M. Guizani, and Y. Wang. “A Survey of Internet of Things Communication using ICN: A Use Case Perspective”. In: *Computer Communications* 142-143 (June 2019), pp. 95–123. DOI: `10.1016/j.comcom.2019.05.010`.

[72] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and B. Mathieu. “A survey of naming and routing in information-centric networks”. In: *IEEE Communications Magazine* 50.12 (2012). DOI: `10.1109/MCOM.2012.6384450`.

[73] J. Pan, S. Paul, and R. Jain. “A survey of the research on future internet architectures”. In: *IEEE Communications Magazine* 49.7 (2011), pp. 26–36. DOI: `10.1109/MCOM.2011.5936152`.

[74] G. Garcıa, A. Beben, F. J. Ramón, A. Maeso, I. Psaras, G. Pavlou, N. Wang, J. Śliwiński, S. Spirou, S. Soursos, et al. “COMET: Content mediator architecture for content-aware networks”. In: *2011 Future Network & Mobile Summit*. IEEE. 2011, pp. 1–8.

[75] N. B. Melazzi, S. Salsano, A. Detti, G. Tropea, L. Chiariglione, A. Difino, A.-C. G. Anadiotis, A. S. Mousas, I. S. Venieris, and C. Z. Patrikakis. “Publish/subscribe over information centric networks: A Standardized

approach in CONVERGENCE”. In: *2012 Future Network & Mobile Summit (FutureNetw)*. IEEE. 2012, pp. 1–8.

[76] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri. “Mobilityfirst future internet architecture project”. In: *Proceedings of the 7th Asian Internet Engineering Conference*. 2011, pp. 1–3.

[77] M. Brunner. “Scalable & adaptive Internet solutions (SAIL)”. In: *Future Internet Assembly* (2010).

[78] N. Fotiou, P. Nikander, D. Trossen, and G. C. Polyzos. “Developing information networking further: From PSIRP to PURSUIT”. In: *International Conference on Broadband Communications, Networks and Systems*. Springer. 2010, pp. 1–13.

[79] T. Anderson, K. Birman, R. Broberg, M. Caesar, D. Comer, C. Cotton, M. J. Freedman, A. Haeberlen, Z. G. Ives, and A. Krishnamurthy. “The nebula future internet architecture”. In: *The Future Internet Assembly*. Springer. 2013, pp. 16–26. DOI: `10.1007/978-3-642-38082-2_2`.

[80] A. Anand, F. Dogar, D. Han, B. Li, H. Lim, M. Machado, W. Wu, A. Akella, D. G. Andersen, and J. W. Byers. “XIA: An architecture for an evolvable and trustworthy Internet”. In: *ACM Workshop on Hot Topics in Networks*. 2011, p. 2. DOI: `10.1145/2070562.2070564`.

[81] T. Wolf, J. Griffioen, K. L. Calvert, R. Dutta, G. N. Rouskas, I. Baldin, and A. Nagurney. “ChoiceNet: toward an economy plane for the Internet”. In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), pp. 58–65. DOI: `10.1145/2656877.2656886`.

[82] V. Jacobson, M. Mosko, D. Smetters, and J. Garcia-Luna-Aceves. “Content-centric networking”. In: *Whitepaper, Palo Alto Research Center* (2007), pp. 2–4.

[83] A. Tagami and M. Arumaithurai. “GreenICN Project: Architecture and applications of green information centric networking”. In: *IEICE Transactions on Communications* 99.12 (2016), pp. 2470–2476.

[84] Y. Ye, S. Dixit, and M. Ali. “On joint protection/restoration in IP-centric DWDM based optical transport networks”. In: *IEEE Communications Magazine* 38.6 (2000), pp. 174–183. DOI: `10.1109/35.846091`.

[85] A. Ioannou and S. Weber. "A survey of caching policies and forwarding mechanisms in information-centric networking". In: *IEEE Communications Surveys & Tutorials* 18.4 (2016), pp. 2847–2886. DOI: 10.1109/COMST.2016.2565541.

[86] C. Kerrche, F. Ahmad, M. Elhoseny, A. Adnane, Z. Ahmad, and B. Nour. "Internet of Vehicles Over Named Data Networking: Current Status and Future Challenges". In: *Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks*. Springer, 2020, pp. 83–99. DOI: 10.1007/978-3-030-22773-9_7.

[87] R. Tourani, S. Misra, T. Mick, and G. Panwar. "Security, privacy, and access control in information-centric networking: A survey". In: *IEEE communications surveys & tutorials* 20.1 (2017), pp. 566–600. DOI: 10.1109/COMST.2017.2749508.

[88] H. Khelifi, S. Luo, B. Nour, and H. Moungla. "LQCC: A Link Quality-based Congestion Control Scheme in Named Data Networks". In: *IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, June 2019. DOI: 10.1109/WCNC.2019.8885909.

[89] B. Nour, H. Ibn-Khedher, H. Moungla, H. Afifi, F. Li, K. Sharif, H. Khelifi, and M. Guizani. "Internet of things mobility over information-centric/named-data networking". In: *IEEE Internet Computing* 24.1 (2019), pp. 14–24. DOI: 10.1109/MIC.2019.2963187.

[90] 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 Global Communications Conference (GLOBECOM)*. Abu Dhabi, United Arab Emirates: IEEE, Dec. 2018, pp. 1–6. DOI: 10.1109/GLOCOM.2018.8647629.

[91] H. Khelifi, S. Luo, B. Nour, and H. Moungla. "A QoS-aware Cache Replacement Policy for Vehicular Named Data Networks". In: *IEEE GLOBECOM Conference*. IEEE, Dec. 2019, pp. 1–6. DOI: 10.1109/GLOBECOM38437.2019.9013461.

[92] B. Nour, K. Sharif, F. Li, and Y. Wang. "Security and privacy challenges in information-centric wireless internet of things networks". In: *IEEE Security & Privacy* 18.2 (2019), pp. 35–45. DOI: 10.1109/MSEC.2019.2925337.

[93] H. Khelifi, S. Luo, B. Nour, H. Moungla, S. Ahmed, and M. Guizani. "A Blockchain-based Architecture for Secure Vehicular Named Data Networks". In: *Computers & Electrical Engineering* 86 (2020), p. 106715. ISSN: 0045-7906. DOI: `10.1016/j.compeleceng.2020.106715`.

[94] H. Khelifi, S. Luo, B. Nour, and C. Shah. "Security and Privacy Issues in Vehicular Named Data Networks: An Overview". In: *Mobile Information Systems* 2018 (Sept. 2018), pp. 1–11. DOI: `10.1155/2018/5672154`.

[95] J. Rexford and C. Dovrolis. "Future internet architecture: clean-slate versus evolutionary research". In: *Communications of the ACM* 53.9 (2010), pp. 36–40. DOI: `10.1145/1810891.1810906`.

[96] B. Nour, F. Li, H. Khelifi, H. Moungla, and A. Ksentini. "Coexistence of ICN and IP Networks: An NFV as a Service Approach". In: *IEEE Global Communications Conference (GLOBECOM)*. Waikoloa, HI, USA: IEEE, Dec. 2019, pp. 1–6. DOI: `10.1109/GLOBECOM38437.2019.9013881`.

[97] X. Jiang, J. Bi, G. Nan, and Z. Li. "A survey on information-centric networking: rationales, designs and debates". In: *China Communications* 12.7 (2015), pp. 1–12. DOI: `10.1109/CC.2015.7188520`.

[98] Y. Rekhter, T. Li, S. Hares, et al. *A border gateway protocol 4 (BGP-4)*. 1994.

[99] G. Tyson, N. Sastry, I. Rimac, R. Cuevas, and A. Mauthe. "A survey of mobility in information-centric networks: Challenges and research directions". In: *ACM Workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*. ACM. 2012, pp. 1–6. DOI: `10.1145/2248361.2248363`.

[100] M. Arumaithurai, J. Chen, E. Monticelli, X. Fu, and K. K. Ramakrishnan. "Exploiting ICN for flexible management of software-defined networks". In: *ACM ICN Conference*. 2014, pp. 107–116. DOI: `10.1145/2660129.2660147`.

[101] S. Salsano, N. Blefari-Melazzi, A. Detti, G. Morabito, and L. Veltri. "Information centric networking over SDN and OpenFlow: Architectural aspects and experiments on the OFELIA testbed". In: *Computer Networks* 57.16 (2013), pp. 3207–3221. DOI: `10.1016/j.comnet.2013.07.031`.

[102] Y. Zhang, D. Raychadhuri, L. A. Grieco, E. Baccelli, J. Burke, R. Ravindran, G. Wang, A. Lindgren, B. Ahlgren, and O. Schelén. "Requirements

and Challenges for IoT over ICN: IETF Internet-Draft draft-zhang-icnrg-icniot-requirements-01”. In: (2016).

[103] Y. Tian, X. Wang, N. L. da Fonseca, and Z. Zhu. “When ICN Meets 5G: Architecting 5G@ ICN over Programmable Data Plane”. In: (). DOI: `10.13140/RG.2.2.29440.79362`.

[104] H. Li, K. Ota, and M. Dong. “ECCN: Orchestration of edge-centric computing and content-centric networking in the 5G radio access network”. In: *IEEE Wireless Communications* 25.3 (2018), pp. 88–93. DOI: `10.1109/MWC.2018.1700315`.

[105] A. Lertsinsrubtavee, M. Selimi, A. Sathiaseelan, L. Cerdà-Alabern, L. Navarro, and J. Crowcroft. “Information-centric multi-access edge computing platform for community mesh networks”. In: *ACM SIGCAS Conference on Computing and Sustainable Societies*. 2018, pp. 1–12. DOI: `10.1145/3209811.3209867`.

[106] I. Benkacem, M. Bagaa, T. Taleb, Q. Nguyen, T. Toshitaka, and T. Sato. “Integrated ICN and CDN Slice as a Service”. In: *IEEE GLOBECOM Conference*. IEEE. 2018, pp. 1–7. DOI: `10.1109/GLOCOM.2018.8648051`.

[107] R. Tourani, S. Misra, and T. Mick. “IC-MCN: An architecture for an information-centric mobile converged network”. In: *IEEE Communications Magazine* 54.9 (2016), pp. 43–49. DOI: `10.1109/MCOM.2016.7565186`.

[108] X. Zhang and Q. Zhu. “Information-Centric Virtualization for Software-Defined Statistical QoS Provisioning Over 5G Multimedia Big Data Wireless Networks”. In: *IEEE Journal on Selected Areas in Communications* 37.8 (2019), pp. 1721–1738. DOI: `10.1109/JSAC.2019.2927088`.

[109] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo. “Communications, Caching, and Computing for Next Generation HetNets”. In: *IEEE Wireless Communications* 25.4 (2018), pp. 104–111. DOI: `10.1109/MWC.2017.1700167`.

[110] A. Carzaniga, M. J. Rutherford, and A. L. Wolf. “A routing scheme for content-based networking”. In: *IEEE INFOCOM Conference*. Vol. 2. IEEE. 2004, pp. 918–928. DOI: `10.1109/INFCOM.2004.1356979`.

[111] Y. Zhang, Z. Xia, A. Afanasyev, and L. Zhang. “A Note on Routing Scalability in Named Data Networking”. In: *IEEE ICC Conference Workshops*. IEEE. 2019, pp. 1–6. DOI: `10.1109/ICCW.2019.8756677`.

[112] S. S. Adhatarao, J. Chen, M. Arumaithurai, X. Fu, and K. Ramakrishnan. "Comparison of naming schema in ICN". In: *IEEE LANMAN*. IEEE. 2016, pp. 1–6. DOI: 10.1109/LANMAN.2016.7548856.

[113] B. Nour, H. Khelifi, R. Hussain, H. Moungla, and S. H. Bouk. "A collaborative multi-metric interface ranking scheme for named data networks". In: *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE. 2020, pp. 2088–2093. DOI: 10.1109/IWCMC48107.2020.9148196.

[114] B. Nour, K. Sharif, F. Li, H. Moungla, and Y. Liu. "A Unified Hybrid Information-Centric Naming Scheme for IoT Applications". In: *Computer Communications* 150 (2019), pp. 103–114. DOI: 10.1016/j.comcom.2019.11.020.

[115] S. Arshad, M. A. Azam, S. H. Ahmed, and J. Loo. "Towards Information-Centric Networking (ICN) Naming for Internet of Things (IoT) The Case of Smart Campus". In: *ICFNDS Conference*. 2017, pp. 1–6. DOI: 10.1145/3102304.3102345.

[116] S. Arshad, B. Shahzaad, M. A. Azam, J. Loo, S. H. Ahmed, and S. Aslam. "Hierarchical and flat-based hybrid naming scheme in content-centric networks of things". In: *IEEE Internet Things Journal* 5.2 (2018), pp. 1070–1080. DOI: 10.1109/JIOT.2018.2792016.

[117] M. A. U. Rehman, R. Ullah, and B. S. Kim. "NINQ: Name-Integrated Query Framework for Named-Data Networking of Things". In: *Sensors* 19.13 (2019), p. 2906. DOI: 10.3390/s19132906.

[118] S. S. Adhatarao, M. Arumaithurai, D. Kutscher, and X. Fu. "ISI: Integrate sensor networks to Internet with ICN". In: *IEEE Internet Things Journal* 5.2 (2017), pp. 491–499. DOI: 10.1109/JIOT.2017.2741923.

[119] H. Khelifi, S. Luo, B. Nour, and H. Moungla. "A Name-to-Hash Encoding Scheme for Vehicular Named Data Networks". In: (June 2019), pp. 603–608. DOI: 10.1109/IWCMC.2019.8766564.

[120] S. Feng, M. Zhang, R. Zheng, and Q. Wu. "A fast name lookup method in NDN based on hash coding". In: *ICMII Conference*. Atlantis Press. 2015. DOI: 10.2991/icmii-15.2015.98.

[121] D. Saxena, V. Raychoudhury, N. Suri, C. Becker, and J. Cao. "Named data networking: a survey". In: *Computer Science Review* 19 (2016), pp. 15–55. DOI: 10.1109/ECS.2015.7124841.

[122] E. Jung. "A design of alias naming scheme for namespace in named data networking". In: *ICACT Conference*. IEEE, pp. 537–540. DOI: `10.1109/ICACT.2016.7423461`.

[123] S. Shan, C. Feng, T. Zhang, and Y. Liu. "A user interest preferences based on-path caching strategy in named data networking". In: *IEEE/CIC ICCC Conference*. IEEE. 2017, pp. 1–6. DOI: `10.1109/ICCChina.2017.8330355`.

[124] N. Kamiyama and M. Murata. "Dispersing Content Over Networks in Information-Centric Networking". In: *IEEE TNSM* (2019). DOI: `10.1109/TNSM.2019.2898229`.

[125] A. Azgin, R. Ravindran, and G. Wang. "A scalable mobility-centric architecture for named data networking". In: *arXiv preprint arXiv:1406.7049* (2014). DOI: `https://arxiv.org/abs/1406.7049`.

[126] Y. Zhang, Z. Xia, S. Mastorakis, and L. Zhang. "Kite: Producer mobility support in named data networking". In: *ACM ICN Conference*. 2018, pp. 125–136. DOI: `10.1145/3267955.3267959`.

[127] N. Leshov, M. A. Yaqub, M. T. R. Khan, S. Lee, and D. Kim. "Content Name Privacy in Tactical Named Data Networking". In: *ICUFN Conference*. IEEE. 2019, pp. 570–572. DOI: `10.1109/ICUFN.2019.8805919`.

[128] I. O. Nunes, G. Tsudik, and C. A. Wood. "Namespace tunnels in content-centric networks". In: *IEEE LCN Conference*. IEEE. 2017, pp. 35–42. DOI: `10.1109/LCN.2017.105`.

[129] 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. DOI: `10.1007/978-3-319-60033-8_26`.

[130] H. Khelifi, S. Luo, B. Nour, and H. Moungla. "A Name-to-Hash Encoding Scheme for Vehicular Named Data Networks". In: *International Wireless Communications & Mobile Computing Conference (IWCMC)*. Tangier, Morocco: IEEE, June 2019. DOI: `10.1109/IWCMC.2019.8766564`.

[131] Y. Lu, Y. Liu, C. Zhang, and J. Tan. "A Data-Deduplication-Based Matching Mechanism for URL Filtering". In: *IEEE ICC Conference*. IEEE. 2018, pp. 1–6. DOI: `10.1109/ICC.2018.8422284`.

[132] B. Nour, S. Mastorakis, R. Ullah, and N. Stergiou. "Information-Centric Networking in Wireless Environments: Security Risks and Challenges".

In: *IEEE Wireless Communications* 28.2 (2021). IF: 11.979, Q1, pp. 121–127. DOI: 10.1109/MWC.001.2000245.

[133] H. Khelifi, S. Luo, B. Nour, and H. Moungla. "In-Network Caching in ICN-based Vehicular Networks: Effectiveness & Performance Evaluation". In: *IEEE ICC Conference*. IEEE, June 2020. DOI: 10.1109/ICC40277.2020.9148950.

[134] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard. "Networking named content". In: *CoNEXT Conference*. 2009, pp. 1–12. DOI: 10.1145/1658939.1658941.

[135] N. Laoutaris, H. Che, and I. Stavrakakis. "The LCD interconnection of LRU caches and its analysis". In: *Performance Evaluation* 63.7 (2006), pp. 609–634. DOI: 10.1016/j.peva.2005.05.003.

[136] A. Dabirmoghaddam, M. M. Barijough, and J. Garcia-Luna-Aceves. "Understanding optimal caching and opportunistic caching at" the edge" of information-centric networks". In: *ACM ICN Conference*. 2014, pp. 47–56. DOI: 10.1145/2660129.2660143.

[137] W. K. Chai, D. He, I. Psaras, and G. Pavlou. "Cache "less for more" in information-centric networks (extended version)". In: *Computer Communications* 36.7 (2013), pp. 758–770. DOI: 10.1016/j.comcom.2013.01.007.

[138] I. Psaras, W. K. Chai, and G. Pavlou. "Probabilistic in-network caching for information-centric networks". In: *ACM Workshop on ICN*. 2012, pp. 55–60. DOI: users.piuha.net/blackhawk/contug/cache.pdf.

[139] S. Arianfar, P. Nikander, and J. Ott. "Packet-level caching for information-centric networking". In: *ACM SIGCOMM, ReArch Workshop*. 2010, pp. 1–16.

[140] K. Cho, M. Lee, K. Park, T. T. Kwon, Y. Choi, and S. Pack. "WAVE: Popularity-based and collaborative in-network caching for content-oriented networks". In: *IEEE INFOCOM Workshops*. IEEE. 2012, pp. 316–321. DOI: 10.1109/INFCOMW.2012.6193512.

[141] Q. Chen, R. Xie, F. R. Yu, J. Liu, T. Huang, and Y. Liu. "Transport control strategies in named data networking: A survey". In: *IEEE Communications Surveys & Tutorials* 18.3 (2016), pp. 2052–2083. DOI: 10.1109/COMST.2016.2528164.

[142] L. Wang, S. Bayhan, and J. Kangasharju. "Optimal chunking and partial caching in information-centric networks". In: *Computer Communications* 61 (2015), pp. 48–57. DOI: 10.1016/j.comcom.2014.12.009.

[143] K. Suksomboon, S. Tarnoi, Y. Ji, M. Koibuchi, K. Fukuda, S. Abe, N. Motonori, M. Aoki, S. Urushidani, and S. Yamada. "PopCache: Cache more or less based on content popularity for information-centric networking". In: *IEEE LCN Conference*. IEEE. 2013, pp. 236–243. DOI: 10.1109/LCN.2013.6761239.

[144] H. Khelifi, S. Luo, B. Nour, A. Sellami, H. Moungla, and F. Naït-Abdesselam. "An Optimized Proactive Caching Scheme based on Mobility Prediction for Vehicular Networks". In: *IEEE Global Communications Conference (GLOBECOM)*. Abu Dhabi, United Arab Emirates: IEEE, Dec. 2018, pp. 1–6. DOI: 10.1109/GLOCOM.2018.8647898.

[145] Y. Li, T. Zhang, X. Xu, Z. Zeng, and Y. Liu. "Content popularity and node level matched based probability caching for content centric networks". In: *IEEE/CIC ICCC Conference*. IEEE. 2016, pp. 1–6. DOI: 10.1109/ICCChina.2016.7636877.

[146] J.-L. Deng et al. "Control problems of grey systems." In: *Systems & Control Letters* 1.5 (1982), pp. 288–294. DOI: 10.1109/GSIS.2017.8077658.

[147] Y. Meng, M. A. Naeem, R. Ali, and B.-S. Kim. "EHCP: An Efficient Hybrid Content Placement Strategy in Named Data Network Caching". In: *IEEE Access* 7 (2019), pp. 155601–155611. DOI: 10.1109/ACCESS.2019.2946184.

[148] Q. Zheng, Y. Kan, J. Chen, and S. Wang. "A Cache Replication Strategy Based on Betweenness and Edge Popularity in Named Data Networking". In: *IEEE ICC Conference*. IEEE. 2019, pp. 1–7. DOI: 10.1109/ICC.2019.8761900.

[149] W. K. Chai, D. He, I. Psaras, and G. Pavlou. "Cache "less for more" in information-centric networks". In: *International Conference on Research in Networking*. Springer. 2012, pp. 27–40. DOI: 10.1007/978-3-642-30045-5_3.

[150] Q. N. Nguyen, J. Liu, Z. Pan, I. Benkacem, T. Tsuda, T. Taleb, S. Shimamoto, and T. Sato. "PPCS: A Progressive Popularity-Aware Caching Scheme for Edge-Based Cache Redundancy Avoidance in Information-Centric Networks". In: *Sensors* 19.3 (2019), p. 694. DOI: 10.3390/s19030694.

[151] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, et al. "Web caching and Zipf-like distributions: Evidence and implications". In: *IEEE INFOCOM Conference*. Vol. 1. 1. IEEE. 1999, pp. 126–134. DOI: 10.1109/INFCOM.1999.749260.

[152] G. Zhang, Y. Li, and T. Lin. "Caching in information centric networking: A survey". In: *Computer Networks* 57.16 (2013), pp. 3128–3141. DOI: 10.1016/j.comnet.2013.07.007.

[153] J. Ren, W. Qi, C. Westphal, J. Wang, K. Lu, S. Liu, and S. Wang. "Magic: A distributed max-gain in-network caching strategy in information-centric networks". In: *IEEE INFOCOM Workshops*. IEEE. 2014, pp. 470–475. DOI: 10.1109/INFCOMW.2014.6849277.

[154] B. Banerjee, A. Seetharam, and C. Tellambura. "Greedy caching: A latency-aware caching strategy for information-centric networks". In: *IFIP Conference*. IEEE. 2017, pp. 1–9. DOI: 10.23919/IFIPNetworking.2017.8264859.

[155] H. Wu, J. Li, J. Zhi, Y. Ren, and L. Li. "Design and evaluation of probabilistic caching in information-centric networking". In: *IEEE Access* 6 (2018), pp. 32754–32768. DOI: 10.1109/ACCESS.2018.2841417.

[156] N. Kamiyama and M. Murata. "Spatially-Dispersed Caching in Information-Centric Networking". In: *IEEE ICC Conference*. IEEE. 2018, pp. 1–7. DOI: 10.1109/ICC.2018.8422428.

[157] H. Balakrishnan, M. F. Kaashoek, D. Karger, D. Karger, R. Morris, and I. Stoica. "Looking up data in P2P systems". In: *Communications of the ACM* 46.2 (2003), pp. 43–48. DOI: 10.1145/606272.606299.

[158] T. Mick, R. Tourani, and S. Misra. "Muncc: Multi-hop neighborhood collaborative caching in information centric networks". In: *ACM ICN Conference*. ACM. 2016, pp. 93–101. DOI: 10.1145/2984356.2984375.

[159] L. Huang, Y. Guan, X. Zhang, and Z. Guo. "On-path collaborative in-network caching for information-centric networks". In: *IEEE INFOCOM Workshops*. IEEE. 2017, pp. 175–180. DOI: 10.1109/INFCOMW.2017.8116372.

[160] J. Zhi, J. Li, H. Wu, and Y. Ren. "GAC: Gain-Aware 2-Round Cooperative Caching Approach in Information-Centric Networking". In: *IEEE IPCCC Conference*. IEEE. 2018, pp. 1–8. DOI: 10.1109/PCCC.2018.8711125.

[161] W.-X. Liu, J. Li, J. Cai, Y. Wang, X.-C. Liu, and S.-Z. Yu. "COD: caching on demand in information-centric networking". In: *Telecommunication Systems* 69.3 (2018), pp. 303–319. DOI: `10.1007/s11235-018-0433-5`.

[162] J. Yang and J. Leskovec. "Patterns of temporal variation in online media". In: *ACM International conference on Web search and data mining*. ACM. 2011, pp. 177–186. DOI: `10.1145/1935826.1935863`.

[163] Y. Zeng, J. Xie, H. Jiang, G. Huang, S. Yi, N. Xiong, and J. Li. "Smart caching based on user behavior for mobile edge computing". In: *Information Sciences* 503 (2019), pp. 444–468. DOI: `10.1016/j.ins.2019.06.056`.

[164] Z. Zhang, C.-H. Lung, M. St-Hilaire, and I. Lambadaris. "Smart Caching: Empower the Video Delivery for 5G-ICN Networks". In: *IEEE ICC Conference*. 2019, pp. 1–7. DOI: `10.1109/ICC.2019.8762067`.

[165] 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". In: *Future Generation Computer Systems* 91 (2019), pp. 590–600. DOI: `10.1016/j.future.2018.08.019`.

[166] Y. Tang. "Minimizing Energy for Caching Resource Allocation in Information-Centric Networking with Mobile Edge Computing". In: *IEEE Conference on DASC/PiCom/CBDCom/CyberSciTech*. IEEE. 2019, pp. 301–304. DOI: `10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00062`.

[167] L. Han, T. Maksymyuk, X. Bao, J. Zhao, and Y. Liu. "Deep Learning based Loss Recovery Mechanism for Video Streaming over Mobile Information-Centric Network." In: *KSII Transactions on Internet & Information Systems* 13.9 (2019). DOI: `10.3837/tiis.2019.09.014`.

[168] Y. Liu, A. Liu, N. N. Xiong, T. Wang, and W. Gui. "Content propagation for content-centric networking systems from location-based social networks". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49.10 (2019), pp. 1946–1960. DOI: `10.1109/TSMC.2019.2898982`.

[169] C. Wang, K. Gai, J. Guo, L. Zhu, and Z. Zhang. "Content-centric Caching Using Deep Reinforcement Learning in Mobile Computing". In: *IEEE HPBD&IS Conference*. 2019, pp. 1–6. DOI: `10.1109/HPBDIS.2019.8735483`.

[170] G. Huang, H. Jiang, J. Xie, Y. Zeng, and S. Yi. "Online Learning Framework based on user-centric access behavior". In: *IEEE HPCC/SmartCity/DSS*

Conference. IEEE. 2018, pp. 290–297. DOI: 10.1109/INFCOMW.2017.8116372.

[171] J. Han, J. Pei, Y. Yin, and R. Mao. "Mining frequent patterns without candidate generation: A frequent-pattern tree approach". In: *Data mining and knowledge discovery* 8.1 (2004), pp. 53–87. DOI: 10.1023/B:DAMI.0000005258.31418.83.

[172] P. Czyzżak and A. Jaszkiewicz. "Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization". In: *Journal of Multi-Criteria Decision Analysis* 7.1 (1998), pp. 34–47. DOI: 10.1002/(SICI)1099-1360(199801)7:13.0.CO;2-6.

[173] H. Khelifi, S. Luo, B. Nour, A. Sellami, H. Moungla, S. H. Ahmed, and M. Guizani. "Bringing Deep Learning at The Edge of Information-Centric Internet of Things". In: *IEEE Communications Letters* 23.1 (Jan. 2019), pp. 52–55. DOI: 10.1109/LCOMM.2018.2875978.

[174] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox. "Information-centric networking: seeing the forest for the trees". In: *ACM Workshop on Hot Topics in Networks*. ACM. 2011, p. 1. DOI: 10.1145/2070562.2070563.

[175] G. Tyson, N. Sastry, R. Cuevas, I. Rimac, and A. Mauthe. "Where is in a name? A Survey of Mobility in Information-Centric Networks". In: *Communications of the ACM* (2013). DOI: 10.1145/2500501.

[176] J. Wang, R. Wakikawa, and L. Zhang. "DMND: Collecting data from mobiles using named data". In: *IEEE VNC Conference*. IEEE. 2010, pp. 49–56. DOI: 10.1109/VNC.2010.5698270.

[177] S. Adhatarao, M. Arumaithurai, D. Kutscher, and X. Fu. "NeMoI: Network mobility in ICN". In: *COMSNETS Conference*. 2018, pp. 220–244. DOI: 10.1109/COMSNETS.2018.8328205.

[178] V. Sivaraman and B. Sikdar. "Hop-count based forwarding for seamless producer mobility in NDN". In: *IEEE GLOBECOM Conference*. 2017, pp. 1–6. DOI: 10.1109/GLOCOM.2017.8254709.

[179] Y. Li, X. Piao, H. Zhang, and K. Lei. "NDN producer mobility management based on echo state network: a lightweight machine learning approach". In: *IEEE ICPADS Conference*. 2018, pp. 275–282. DOI: 10.1109/PADSW.2018.8644896.

[180] J. Augé, G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, and X. Zeng. “MAP-Me: Managing anchor-less producer mobility in content-centric networks”. In: *IEEE TNSM* (2018). DOI: `10.1109/TNSM.2018.2796720`.

[181] M. A. U. Rehman, R. Ullah, B.-S. Kim, B. Nour, and S. Mastorakis. “CCIC-WSN: An Architecture for Single-Channel Cluster-Based Information-Centric Wireless Sensor Networks”. In: *IEEE Internet of Things Journal* 8.9 (2020), pp. 7661–7675. DOI: `10.1109/JIOT.2020.3041096`.

[182] M. Gendreau, J.-Y. Potvin, et al. *Handbook of metaheuristics*. Vol. 2. Springer, 2010.

[183] F. Satterthwaite. “Generalized Poisson distribution”. In: *The Annals of Mathematical Statistics* 13.4 (1942), pp. 410–417. DOI: `10.1002/bimj.200410102`.

[184] G. Liu and D. Jiang. “5G: Vision and requirements for mobile communication system towards year 2020”. In: *Chinese Journal of Engineering* 2016.2016 (2016), p. 8. DOI: `10.1155/2016/5974586`.

[185] C. Boucetta, B. Nour, A. Cusin, and H. Moungla. “QoS in IoT Networks based on Link Quality Prediction”. In: *IEEE International Conference on Communications (ICC)*. Montreal, Canada: IEEE, June 2021, pp. 1–6. DOI: `10.1109/ICC42927.2021.9500396`.

[186] A. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang, and L. Wang. “NLSR: named-data link state routing protocol”. In: *ACM SIGCOMM workshop on ICN*. 2013, pp. 15–20. DOI: `10.1145/2491224.2491231`.

[187] Z. Bellal, B. Nour, and S. Mastorakis. “CoxNet: A Computation Reuse Architecture at the Edge”. In: *IEEE Transactions on Green Communications and Networking* 5.2 (2021), pp. 765–777. DOI: `10.1109/TGCN.2021.3071497`.