ICN with Edge for 5G: Exploiting In-network Caching in ICN-based Edge Computing for 5G Networks

Rehmat Ullah[a], Muhammad Atif Ur Rehman[b], Muhammad Ali Naeem[c], Byung-Seo Kim[d,*], Spyridon Mastorakis[e]

[a]*Department of Computer Engineering, Gachon University, Seongnam 13120, Republic of Korea*
[b]*Department of Electronics & Computer Engineering, Hongik University, Sejong 30016, Republic of Korea*
[c]*InterNetworks Research Laboratory, School of Computing, University Utara Malaysia, Sintok 06010, Malaysia*
[d]*Department of Software & Communications Engineering, Hongik University, Sejong 30016, Republic of Korea*
[e]*Department of Computer Science, College of Information Science & Technology, University of Nebraska, Omaha, USA*

Abstract

In recent years, edge computing and Information Centric Networking (ICN) have been introduced as emerging technologies for distributing content closer to the end users. The former promises to increase the performance of several applications by using data locality and relieve the core network by addressing the increasing bandwidth demands caused by increased data volume. The latter makes the content directly addressable and routable in the network. Enabling ICN with edge computing in Radio Access Network (RAN) can improve the efficiency of content distribution and communication performance by reducing the distance between users and services. In line with this assertion, in this paper, we propose an ICN-capable RAN architecture for 5G edge computing environments that offers device to device communication and ICN application layer support at base stations. Moreover, ICN and edge caching is only useful for static content–requests for dynamic content have to traverse the core network increasing the latency in 5G networks. To address this issue, we provide a content prefetching strategy based on ICN naming. Experimental results show that our proposed scheme results in high cache hit ratios and lower delays compared to conventional edge systems and prior work on ICN for 5G.

Keywords: 5G, radio access network, mobile edge caching, information-centric networking, device to device communication, content popularity, backhaul traffic reduction, future networks.

---

1. Introduction

The Internet was originally designed to enable communication and resource sharing between end-to-end hosts. Over time, the Internet's scope has changed with technological advances such as broadband and mobile devices. Every day, massive volumes of content are now searched and uploaded through platforms like Twitter, YouTube, Facebook, Flicker, and Google. This content is increasing exponentially, and multimedia content is expected to account for most Internet traffic in the near future. According to the Global Mobile Data Traffic Forecast Update 2016-2021, global mobile data traffic grew by 63 percent in 2016 and is expected to reach 49 exabytes by 2021 [1]. According to the Ericsson Mobility report [2], millions of cars are already connected via fourth generation (4G) cellular access, and cellular connectivity with the Internet of Things (IoT) is expected to grow significantly by 2024 [3], [4]. Therefore, various technologies have been deployed for content dissemination, such as Content Delivery Networks (CDNs) and Peer-to-Peer (P2P) networks. Edge Computing (EC) has also been proposed as an emerging paradigm for the deployment of resources at the network edge, which has attracted great interest; in particular, EC moves computation, bandwidth, and storage resources closer to the end user in order to reduce backbone network traffic and response latency. Along with EC, the research community has recently advanced the emerging paradigm of Information-Centric Networking (ICN). Various ICN architectures have been proposed that share common ideas and principles [5], [6] related to content naming and pull-based communication. In general, a name is assigned to the content, and that content is retrieved without knowledge of the location where it resides (unlike traditional host-oriented systems). Additionally, rather than securing the communication channel, ICN emphasizes content security, consumer mobility, and multihoming.

ICN can be an enabler of the EC vision, since it gives the communication power to end-users [7], [8], [9], [10]. Before EC, all user content requests had to pass through the core network fetching the content from the cloud/provider. This resulted in high latency due to the core network and Wide Area Network (WAN) delay [11], [12]. To mitigate that, the concept of CDNs was proposed as a means of bringing content closer to the user by deploying multiple edge servers around the world. The edge can be anywhere in the end user's vicinity (e.g., Base

---

[*] Corresponding author.

Email addresses: `rehmatullah@gachon.ac.kr` (Rehmat Ullah), `atif_r@outllook.com` (Muhammad Atif Ur Rehman), `malinaeem7@gmail.com` (Muhammad Ali Naeem), `jsnbs@hongik.ac.kr` (Byung-Seo Kim), `smastorakis@unomaha.edu` (Spyridon Mastorakis)

DOI: 10.1016/j.future.2020.04.033

Station (BS), coffee shop, shopping mall). In addition to that, ICN inherently supports client mobility, multihoming, and content security, being suited for fifth generation (5G) networks due to name-based forwarding and routing, which can be leveraged to realize content awareness directly at the network layer. Existing IP-based caching at the Radio Access Network (RAN) level is not context-aware, resulting in limited scalability in 5G networks [13]. ICN offers in-network caching that can reduce the overall content retrieval delay. To this end, researchers have recently proposed the use of ICN with EC for 5G networks [14], [15]. However, ICN is enabled between the BS and the core network only, in part to reduce consumer handoff delay when mobile users move from one BS to another. We argue that ICN should also be enabled at the device level and at BSs [16], [17], so that devices first request content in their vicinity (i.e., from nearby devices), and, subsequently requests are sent towards the BS, if the content is not available within the device vicinity.

Moreover, ICN and edge caching face challenges with dynamic content, and caching content at edge servers is not a good fit for dynamic content, since dynamic content is always delivered from the origin server via the core network. This works well for static content (content that does not change over time) such as files or images [18], [19], but it is not useful for search results or real-time content retrieval (e.g., sports scores). To make effective use of emerging technologies like ICN and EC, it is important to consider the nature of the content.

The main contributions of our paper are as follows:

1. To exploit the caching capability of ICN at the device level for device-to-device (D2D) communication, we propose an ICN-edge enabled 5G network architecture.
2. To address the issue of dynamic content, we propose a named-based content prefetching strategy for fetching in advance the latest dynamic content from the origin server.
3. We implement the ICN functionality and core data structures such as Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB) in the .NET framework.
4. We propose the .NET web Application Programming Interface (API) for RESTful web services to enable mobile devices to communicate with the BSs at the edge.
5. Through extensive simulations (using the ndnSIM simulator [20] and an ICN-edge testbed that we developed), we present evaluation results and establish a correlation between the cache hit ratio and the average content retrieval delay.

The rest of the paper is organized as follows. Section 2 describes the research background and related work on EC, ICN and the benefits of ICN with EC for 5G, as well as our problem scenario. The proposed scheme is provided in Section 3. In Section 4, we discuss the performance evaluations in the testbed and simulation environment. In Section 5 we provide open issues, challenges and future research directions of ICN with EC for 5G network. Finally Section 6 concludes the paper.

2. Background and related work

Our proposed scheme comprises of ICN with EC for 5G network. Therefore, before our proposed architecture is discussed, it is important to elaborate upon conventional content delivery with the EC and then the role of ICN in EC for 5G is presented.

2.1. Edge Computing

Cloud computing was proposed to centralize computing, storage and network management. Such clouds were referred as cloud data centers, backbone IP networks and cellular core networks [21]. However, due to technological evolution such as smartphones, laptops and tablets, the demands for services and applications were increased. Even though today's mobile devices are powerful in resources, these may not be able to accommodate the requirements of some applications such as augmented reality (AR) and virtual reality (VR) [22]. To prolong the lifetime of battery and to accommodate high processing applications, the concept of Mobile Cloud Computing (MCC) was introduced [23]. In MCC, users are allowed to offload their tasks to cloud data centers via the core network of a mobile operator and Internet.

Due to emergence of IoT, the large numbers of interconnected IoT devices generate massive amounts of data. Moreover, applications may require real time processing such as drone flight control applications, AR/VR, and online gaming, with requirements of latencies below a few tens of milliseconds. To overcome these challenges, the idea of EC was proposed where the cloud services are moved closer to end users. The term "cloudlet" was coined in 2009 as the first realization of the EC paradigm [24], bringing computation and storage resources in vicinity of end users. In 2012, CISCO, for the first-time, proposed Fog computing [25] by enabling services, applications, and content storage in close vicinity to mobile end users. In the Fog computing paradigm, data processing happens locally rather than being sent to the Cloud servers. Fog computing supports offloading, caching, location awareness, and mobility information. It has many advantages for applications that are delay sensitive.

From the perspective of mobile users, all the EC proposals have the drawback of Quality of Service (QoS) and Quality of Experience (QoE). One of the main reasons is that computing was not enabled for mobile networks. To this end, the European Telecommunications Standards Institute (ETSI) [26] introduced the EC in the mobile network architecture and was named as *"Mobile Edge Computing (MEC)"*. The main purpose of MEC is to enable an efficient and seamless integration of the cloud computing functionalities into the mobile network. Initially it was only intended for mobile networks and now both fix and mobile networks are using MEC. Therefore, the MEC acronym no longer refers to *"Mobile Edge Computing (MEC)"* and instead stands for *"Multiple-access Edge Computing"* [27]. In the literature there are many confusions about the terms such as Edge, Fog, MEC, Cloudlet and MCC. These are all concepts under the umbrella of EC. However, the implementation speciation differs at different aspects for each proposal. For better understanding on these terms and differences between all the edge proposal the interested readers are referred to [28], [29].

2.2. Multi-access Edge computing

In this section, we explain the conventional data delivery method of MEC (Fig. 1). The architecture comprises of mobile users, BSs with edge servers installed (MEC servers), the core network, and the cloud/provider. The MEC servers are typically mini clouds that are deployed by providers (e.g., telecom operators). These servers are connected to data centers via the Internet whereas the mobile devices and MEC servers are connected via wireless links [30]. Whenever a user needs some data, it sends a request to the MEC server. The first time, if the data are not found in the MEC server, the request is forwarded to the cloud through the core network. The provider replies with the content and sends the data back to the MEC server. The MEC server stores the content and, in the future, whenever requests for the same content is received by the server, then the content is provided by the server instead from the provider.

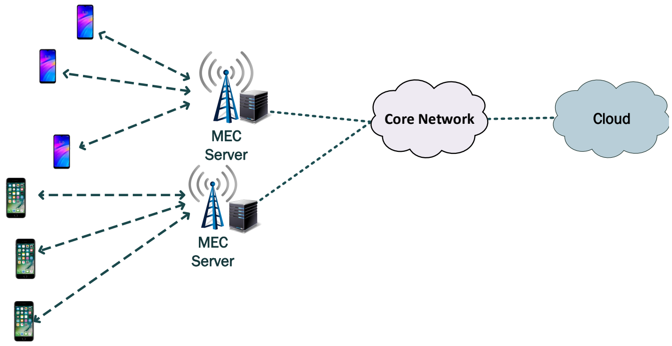


Figure 1: Example of architecture of the MEC systems

The idea of having MEC servers communicating directly with mobile users resulted in substantial latency reductions. In the case of mobility, this can result in handoff delays, since users that move from one BS to another need to establish a new session for content retrieval. To address this issue, researchers introduced ICN in MEC using the ICN features, such as in-networking caching and name-based routing to facilitate content delivery and mobility for users.

2.3. Information-Centric Networking

ICN [5], [6] is an emerging paradigm where a name is assigned to Data/content, and that Data/content is retrieved without knowing the physical location of the content provider unlike IP-based networks. In the literature various architectures have been proposed under the umbrella of ICN paradigm such as MobilityFirst, Network of Information (NetInf), Data Oriented Network Architecture (DONA), Publish Subscribe Internet Technology (PURSUIT), content mediator architecture for content-aware networks (COMET) and CONVERGENCE [31]. Name Data Networking (NDN) [32] is considered as widely used architecture among all ICN architectures due to its active development and involvement in the community. NDN follows pull-based communication model and two type of packets e.g., Interest and Data are used for communication. The Consumer(s) sends Interest packets containing the name of the requested content. The provider or any node when receives the interest message may reply with the content and all the nodes on the path may cache the content subject to the caching policies. The NDN architecture resulted as an extension of the Content Centric Networking (CCN) [33] and follows the same communicating model of interest and data exchange process. However, NDN is slightly modified in architectural prospective. Generally, in pure ICN communication model, the CS check is the very first operation when a content is received at any node. However, in the latest NDN communication model and implementation, the PIT is checked at first and then the CS is checked. It is stated in the literature that the CS is much larger than PIT, therefore, to minimize the lookup delay NDN first checks the PIT [34]. The detailed forwarding process of NDN interest and Data packet exchange is shown in Fig. 2 and are discussed as follows:

When a consumer requires some content, it expresses an Interest packet. Any node when receives an Interest packet, it first performs PIT operations. If an entry for that content is found in PIT, the Interest is discarded. Otherwise the CS is checked for data availability; if a match is found, the node sends the data back to the consumer. Otherwise, a new PIT entry is created, and the Interest is sent upstream via interface(s) stored in the FIB. Once an Interest reaches a data producer, the producer will send back a Data packet, which is forwarded back to the requesting consumer(s) following the created state (PIT entry) at each router. In this paper, we utilize the NDN communication model and use the terms ICN and NDN interchangeably.

2.4. ICN with Edge Computing for 5G

Although conventional MEC architectures bring the content closer to users via EC, it has some challenges and this approach cannot reduce the latency due to extra overhead on the core network–if the data is not found at the edge server, the edge server will forward the requests via core network to the provider/cloud. In this regards, very few efforts have been made towards 5G-ICN.

He Li et al. [14] used ICN-enabled forwarding devices between BS and the core network that cache the content. Thus, there is no need to access the core network if a cache hit occurs on the ICN-enabled forwarding devices before the core network. After receiving the request from the end user device, the edge server sends the content request with the ICN protocol to ICN-enabled forwarding devices that are deployed between the BS and the core network, since it is possible that ICN-enabled forwarding devices in the forwarding route have cached the required content. If the required content is not cached on ICN nodes in RAN, the edge server still needs to download the content from the core network.

Z. Zhang et al. [15] also used ICN-enabled forwarding devices between the BS and the core network (Fig. 3). The authors proposed an ICN caching approach, taking into account user mobility and content popularity and demonstrating that if

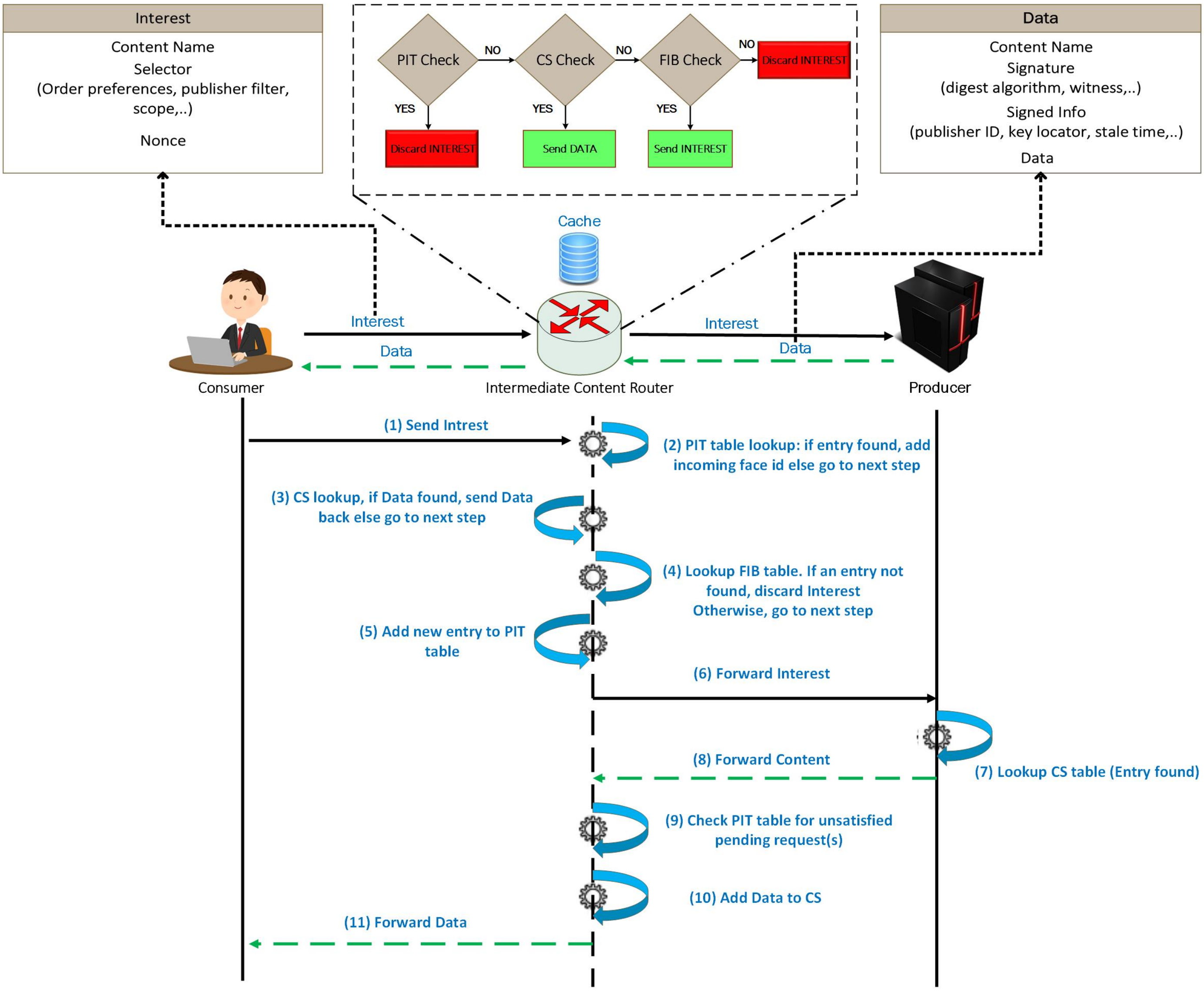


Figure 2: NDN communication model

popular videos are cached at the ICN-enabled forwarding devices, mobile users can fetch the content from these devices directly when a handoff occurs. This approach significantly improved the QoE while reducing the handoff delay.

R. Ravindran et al. [35] proposed a 5G-ICN architecture and compared their results to the 3GPP-based mobile architectures. 5G-ICN is designed to offer Mobility as a Service (MaaS). To minimize cache redundancy, cooperative ICN caching has been proposed in [36]. Caching decisions are made on the popularity of content in a probabilistic way. However, they did not propose a scheme to reduce the core network delay which is the main issue in 5G networks. In case of dynamic content, requests have to go via the core network which brings additional latency in 5G networks.

C. Liang et al. proposed an information-centric virtualization architecture for 5G [37]. Caching and resource allocation is formulated as a joint optimization problem to minimize the content access delay. However, this approach did not enable D2D communication. Specifically, in 5G, ICN-D2D scenarios can be enabled where mobile users can share their resources e.g., storage and bandwidth resource for efficient data dissemination. In this regard, *Grassi et al.* utilized ICN in VANETs and enabled D2D communication [38]. Unlike vehicles which are powerful in terms of storage, energy and other resources, mobile devices are often resource constrained and may be unable to perform complex tasks. Therefore, edge caching should also be considered for balancing the load on mobile devices.

M. Sheng et al.[39] proposed caching both at mobile devices and at the BSs and enabled multihop D2D communication. However, the cache could also be enabled by employing ICN-enabled devices between the BS and core network which could further limit the access to the core network. Moreover, the nature of the content is important and should be considered in account for caching the content at various places in 5G architecture. In our work, we did not only enable ICN at devices level for D2D communication, but also at BSs. We also provide

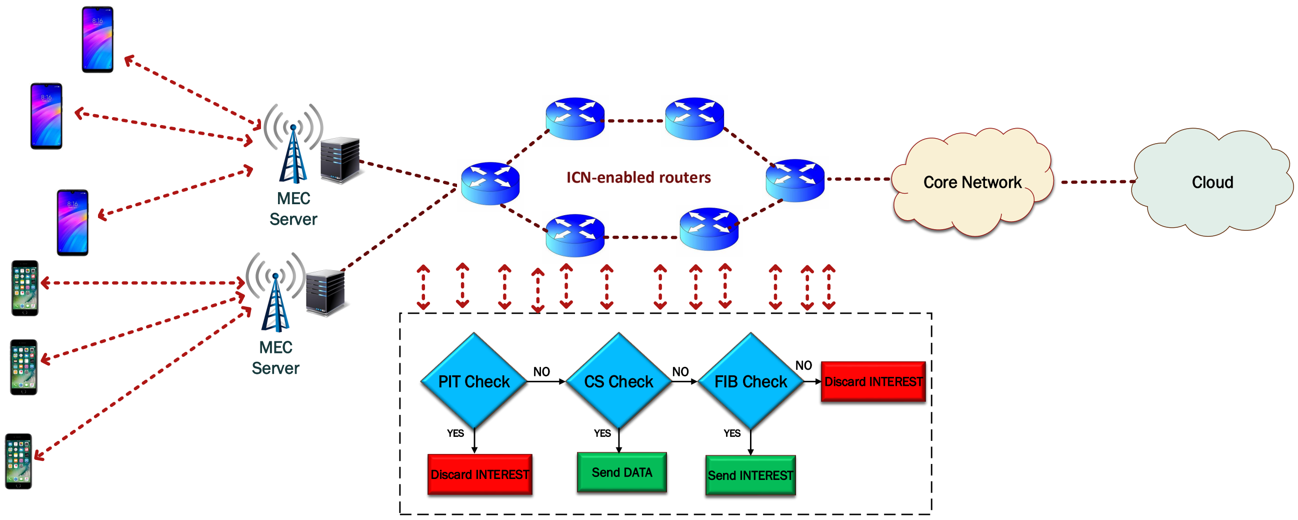


Figure 3: Example of architecture of the ICN enabled MEC system

Table 1: Comparison of existing works

Research efforts	5G	ICN-D2D	ICN-edge caching	ICN-RAN routers	Dynamic content
[14]	✓	✗	✗	✓	✗
[15]	✓	✗	✗	✓	✗
[35]	✓	✗	✗	✗	✗
[36]	✓	✗	✗	✓	✗
[37]	✓	✗	✗	✗	✗
[38]	✗	✓	✗	✗	✗
[39]	✓	✓	✗	✗	✗
Proposed Solution	✓	✓	✓	✓	✓

content prefetching scheme for addressing the dynamic content issue in 5G networks. The comparison of existing works with our work is shown in Table 1.

In the following we shed light on the use of ICN in 5G networks and the possible benefits of ICN for 5G networks.

2.5. *How ICN can help 5G?*

The ICN paradigm promises to provide the required features currently not addressed by the existing 5G wireless research. ICN has many features that are well suited for 5G networks such as consumer mobility support, name-based forwarding, multihoming, content security, and caching. In the following we highlight some of major matching features of ICN and 5G systems [40].

1. **Naming and contextual communication:** ICN provides naming of content and functions called Named Function Networking (NFN) [41] via hierarchal naming schemes. The content names are independent of the host who generates them. With the help of content naming, any ICN node can recognize the content type and services as well as via semantically meaningful names at the network layer. Moreover, consumer mobility comes natively with ICN, which is beneficial for handoffs during the movement of users from one BS to another.
2. **Multihoming support:** ICN provides native support for multihoming. That means ICN nodes may have multiple interfaces and the most convenient interface can be used based on the information embedded in packets. Multiple interfaces can also be used to increase the utilized bandwidth. Multihoming support is beneficial in 5G networks due to presence of heterogenous networks which will require multiple interface support.
3. **In-network caching:** Due to in-network caching, the nodes in ICN can cache the content closer to the user subject to caching policies. For example, content is cached at the device itself, the BS, or at the RAN forwarding devices, which will result in the reduction of network traffic on the core network and will also reduce the content retrieval delay.
4. **Content security:** ICN features content security with signatures at the network layer in the lieu of securing commu-

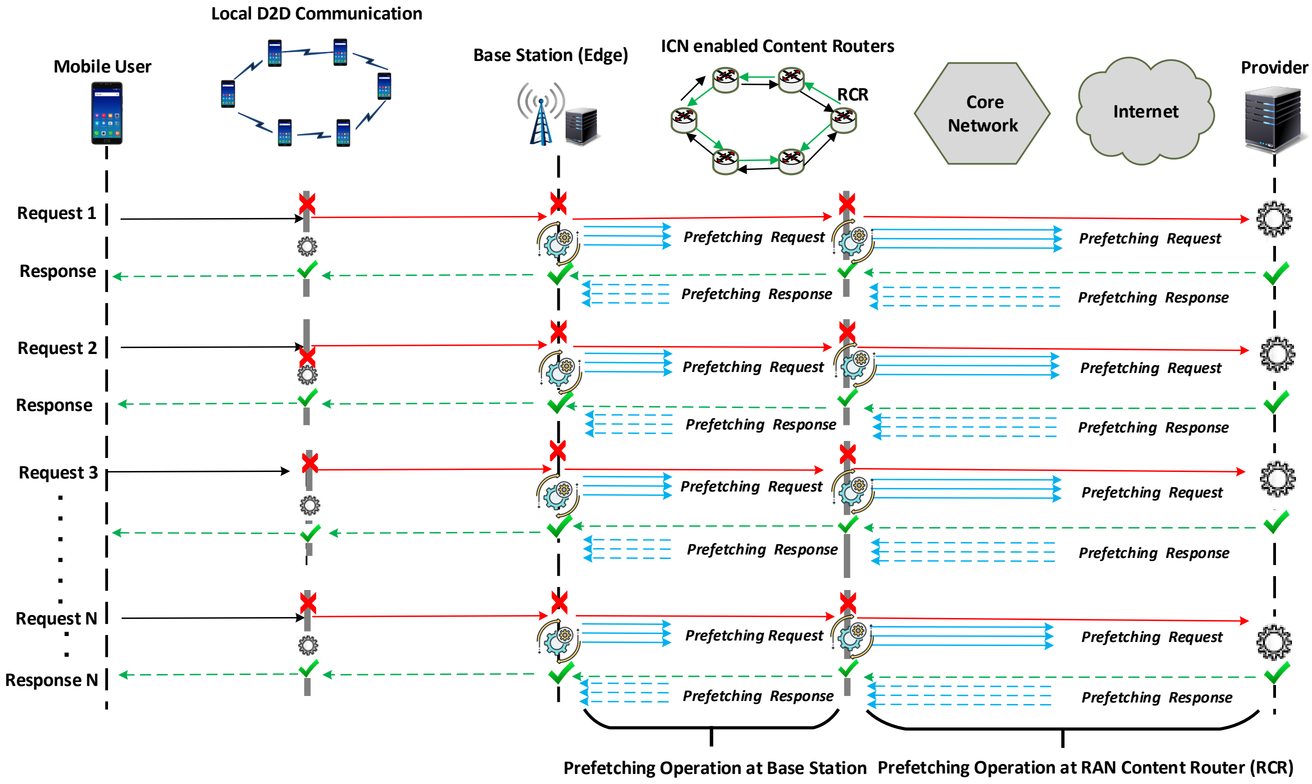


Figure 4: Example of content retrieval process in the proposed architecture

nication channels [42]. In existing Long-Term Evolution (LTE) systems, mobile users are authenticated, while the authentication phase introduces additional latency. On the other hand, the ICN built-in content security feature can be vital in 5G networks to guarantee the security of content instead of security the channel. However, there are many challenges yet to be explored in this area [43].

2.6. Problem Scenario and an illustrative use-case: ESPN

As a relevant use case scenario, we chose Entertainment and Sports Programming Network (ESPN) [44]. In live games, team scores are dynamically updated from origin servers–all requests for scores are forwarded to these servers. In the context of our work, we consider the issue of live scoring in three games: Soccer, Cricket, and Baseball. Note that similar challenges also apply to smart campus scenarios, where IoT devices are connected to the cloud to send and retrieve relevant content, or in the case of an ultra-dense stadium network [45]. In conventional MEC systems, when a user requests a score, the request passes each time via the edge server and core network to the origin server, resulting in high latency and overhead, as millions of users across the world may request the same live scores. Moreover, mobile users may request static content that can be retrieved from the MEC server; in this case, each request from each mobile user has to go to the BS. Through the device caching capability, such static content could be retrieved from a nearby device. Finally, dynamic content could be prefetched to limit core network access for each requests.

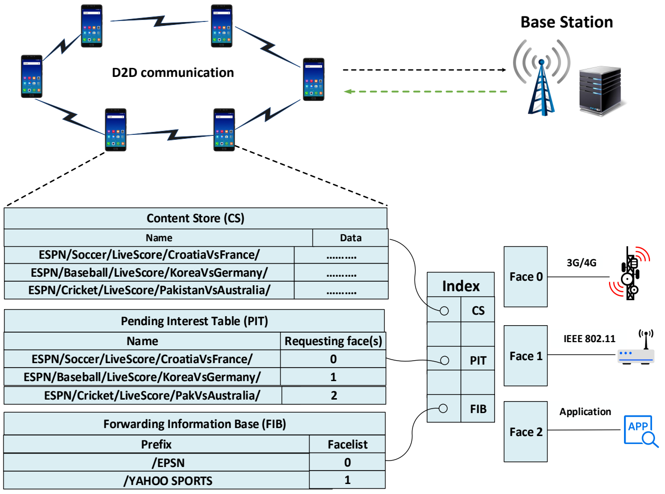
3. Proposed ICN-edge enabled 5G Architecture

For clarity, we first provide an overview of the proposed architecture before describing it in detail. Fig. 4 shows an example of the proposed architecture's messaging sequence, comprising two main elements: named-based D2D communication and a named-based prefetching mechanism for dynamic content. To enable upcoming 5G networks with ICN and EC and to reduce the need to access the core network, we first enabled ICN locally at device level, allowing devices to communicate with each other. Second, we enabled ICN at the application layer of BS, where we implemented the main ICN data structures, including PIT, CS, and FIB. We also developed a hierarchical name-based content prefetching strategy to deal with dynamic content in 5G networks.

As shown in Fig. 4, a mobile user seeking content sends an interest packet to a nearby mobile device. After receiving this packet, a device searches its memory cache for the requested content. If the content is found, it is sent back to the user. If the content is not found in the cache memory of any local mobile device, the request is forwarded to the BS. For the purposes of an ICN-based edge system, we provided application layer implementation at the BS. When the BS receives a request, it searches its CS for the content. If found, the content is sent back to the mobile user; otherwise, the BS forwards this interest packet to the ICN-enabled router(s) deployed between the BS and the core network. After the interest packet is received, the router checks its cache memory for the interest name. If the content is found, it is sent back to the BS. All neighboring ICN-enabled routers are checked for the content; if it is not found, the request is sent to the provider/cloud via the core network. This reduces the content retrieval delay and improves the QoE for mobile users by reducing the need to access the core network.

As discussed earlier, dynamic content changes over time, and requests for such content must pass through the core network to the origin server. Therefore, the proposed architecture also enables a name-based content prefetching strategy at the BS and at the ICN-enabled routers known as a RAN content routers (RCRs) connected to the core network. By keeping track of the number of requests from mobile users, the BS determines their popularity and maintains a certain popularity threshold. Once the requests for specific content exceeds this threshold, a prefetching request is sent to the cloud to download the latest content in advance (as shown after the BS in Fig. 4). As the content is simultaneously prefetched from the origin server and cached at the BS, it can be provided to mobile users without forwarding requests to the origin server.

Moreover, in 5G networks, there may be multiple BSs, each with different trends, depending on interests in that local area. As traffic from all BSs goes to the origin server via ICN-enabled routers and the core network, we also enabled a content prefetching strategy before the core network at RCR, which is directly connected to the core network. The RCR also keeps track of requests from all BSs and checks content popularity. However, the threshold value at the RCR is higher than at the BS because the RCR receives requests from many BSs. Once requests for specific content exceed the popularity threshold of the RCR, the prefetching requests are sent to the cloud to download the content in advance (as shown before the core network in Fig. 4). By doing so, the latest content is prefetched at the RCR. This mechanism reduces the need for access to the core network and may eventually result in latency reduction in 5G networks. D2D communication and the hierarchical name-based content prefetching strategy are described in detail in the next Section, and details of ICN application layer implementation are provided in Section 4. After discussing D2D communication, including limitations and assumptions, we go on to describe the name-based prefetching mechanism for dynamic content.



Content Store (CS)	
Name	Data
ESPN/Soccer/LiveScore/CroatiaVsFrance/	....
ESPN/Baseball/LiveScore/KoreaVsGermany/	....
ESPN/Cricket/LiveScore/PakistanVsAustralia/	....

Pending Interest Table (PIT)	
Name	Requesting face(s)
ESPN/Soccer/LiveScore/CroatiaVsFrance/	0
ESPN/Baseball/LiveScore/KoreaVsGermany/	1
ESPN/Cricket/LiveScore/PakVsAustralia/	2

Forwarding Information Base (FIB)	
Prefix	Facelist
/EPSN	0
/YAHOO SPORTS	1

Figure 5: D2D communication in the proposed architecture

3.1. D2D communication

Fig. 5 shows the proposed D2D communication setup in the proposed architecture. D2D communication is enabled in an ad hoc manner to allow nearby devices to communicate with each other. However, D2D communication in cellular networks poses technical challenges. The first issue is that as devices communicate with each other via BSs in the licensed cellular bandwidth, ad hoc D2D communication must take account of how the devices will be billed for local communications. In the first four generations of cellular networks, D2D communication was not considered, as communication could only occur via BSs. A further challenge relates to the security of data passing through other devices-that is, if a device serves as a relay for another device, how can the data be secured, and how can data privacy be ensured? Additionally, if no BS is involved, how can the connection setup, interference management, and resource allocation be managed. These limitations are discussed below. To realize the goal of ICN-based D2D communication in 5G networks, we have made a number of assumptions, which are outlined next.

Assumptions:

Our first assumption relates to D2D communication via ad hoc wireless connections in a local environment. As all devices are authenticated, there is no privacy issue when passing data through other devices. We also assume that the devices involved in D2D communication are not charging for battery and bandwidth usage while passing data through; the communication environment is based entirely on friendly devices that make resources freely available to other devices. When a mobile device/user requires content, the process is as follows.

Step 1. The mobile user sends content request(s) to the nearest device(s)

Step 2. On receiving a request, the nearest device(s) follow the NDN/ICN communication model to return the content. If the requested content is found, the request is satisfied, and the data are sent back to the requester(s)

Step 3. If the content is not found locally, the consumer sends the request to the MEC server, which is one hop away

Step 4. The consumer commences communication with the BS

Once the consumer node initiates communication with the BS and the request arrives, the BS then checks for content availability using the hierarchical naming scheme. However, as the edge node/BS is a real device, the ICN mechanism will not work, and it cannot check the content on the edge node by means of the ICN naming scheme. As our architecture is ICN-based, we implemented ICN data structures at the application layer of the edge node to realize ICN at the BS and to check the content on real devices by means of the ICN naming scheme. The reason for using ICN at the application layer is to realize

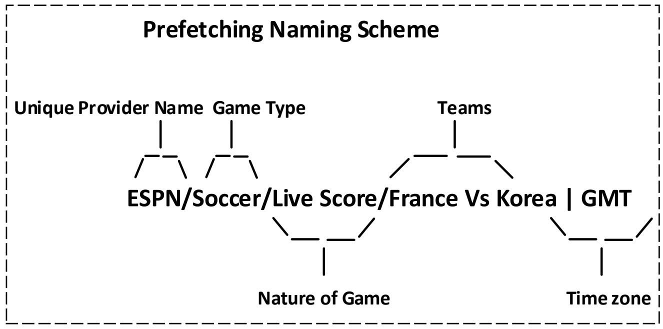


Figure 6: Content prefetching: Naming scheme

ICN at the edge so that the mechanism can be followed on the edge node. The next section describes the name-based prefetching strategy for dynamic content.

3.2. Content Prefetching

Our content prefetching strategy is based on ICN hierarchical naming, which can be used for cases beyond the case of dynamic content discussed here. For example, the dynamic content can be trending on Twitter, YouTube etc. After explaining the naming scheme, we discuss the content prefetching strategy.

3.2.1. Prefetching Naming Scheme

The proposed prefetching naming scheme comprises of a number of components. As shown in Fig. 6, the first component shows the unique provider name–in our use-case, the ESPN name prefix. The second component is the name of the content to be requested from the provider (e.g., the type of game such as Soccer, Cricket, Baseball). The third component is the nature of the data (e.g., live scores, live streaming). The fourth component shows the teams (e.g., Pakistan, Korea). The final component shows the time zone and the time at which the interest packets were sent. Our naming scheme is flexible and can change according to user requirements. For example, in smart buildings, nodes are usually deployed to monitor, sense, or control various parameters such as temperature, pressure, humidity, and air quality. The provider name might be a smart campus using the IoT, with components such as Hongik/buildingD/Room425/Temperature/GMT showing the temperature in room number 425 in building D of Hongik University. The system could also be used for performing actions such as turning lights or air conditioning on or off [46].

3.2.2. Content Prefetching Strategy

Fig. 7 shows the complete architecture of the proposed ICN-enabled 5G system, including mobile users, MEC server, ICN routers, core network, and cloud. The left side of Fig. 7 shows the D2D communication setup as discussed above. When mobile devices send requests to the MEC server, it checks for content availability. If the content is found, it is sent back to the mobile user. However, as discussed previously, requests for dynamic content must go to the cloud via the core network. Therefore, we propose content prefetching strategy to limit the access to the core network. To rank the content's popularity and simultaneously download that content from servers, we proposed a content prefetching strategy at the BS and at the RCR node.

Prefetching Strategy at the BS. First, to acquire the most popular content, we enabled our proposed content prefetching strategy at the BS. We kept track of the number of requests from mobile users to the BS through a table on edge node that measures the frequency of such requests. We also maintained a threshold based on content popularity at each BS. For instance, if requests for cricket-related content equal or exceed a certain threshold value, the content is classified as popular. In the proposed scheme, we assumed a threshold value of 1000 requests for a specific request at each BS. This threshold value could be changed according to use case requirements, and optimized values could be used.

After receiving interest packets from mobile users, the BS checks the frequency of requests. If mobile users request content whose frequency exceeds the threshold, the BS sends a request with the name of the content and Greenwich Mean Time (GMT). For example, if GMT at the edge node is 13:20:48, the request to download the latest content is sent to the origin server after that time. When the content arrives at the BS, the CS of the BS is checked for space availability. If space is available, the content is stored for future requests; if not, the least requested content is removed from the CS.

For instance, during the World Cup, the frequency of requests for soccer scores from users all over the world is very high. The edge node would begin to download all the latest relevant content in advance from the origin server by sending a request to the server for the latest content, accompanied by timing information that is then used by the origin server to forward the required content. Such content would be forwarded to mobile users and cached along the path, reducing the need to access the core network and ultimately reducing latency and traffic on the origin server with a higher cache hit rate in RAN.

However, 5G networks may include multiple BSs. As shown in Fig. 7, MEC servers are connected to the core network via ICN content routers, enabling requests from all BSs to pass through the RCR router, which is connected to the core network. As each BS keeps a record of user requests, request trends at other BSs may be different. In that case, the RCR node checks for accumulative popular content and decides which content to prefetch from the origin server.

Prefetching strategy at RAN content router (RCR). Let us consider three BSs, all of which are sending data to ICN-enabled routers deployed between the core network and the BSs; these routers then send data to the core network. For an ICN router directly connected to the core network, let us suppose that the frequency of end users differs at the BSs. For instance, at BS-1, the request frequency is 1000 for Cricket scores, 500 for Soccer, and 700 for Baseball. If we assume that the BS threshold value is 1000, requests equal to or greater than that threshold will be considered as popular content. BS-1 sends the prefetching request for popular content (in this case, Cricket) to the origin server with the corresponding GMT time. Suppose now that

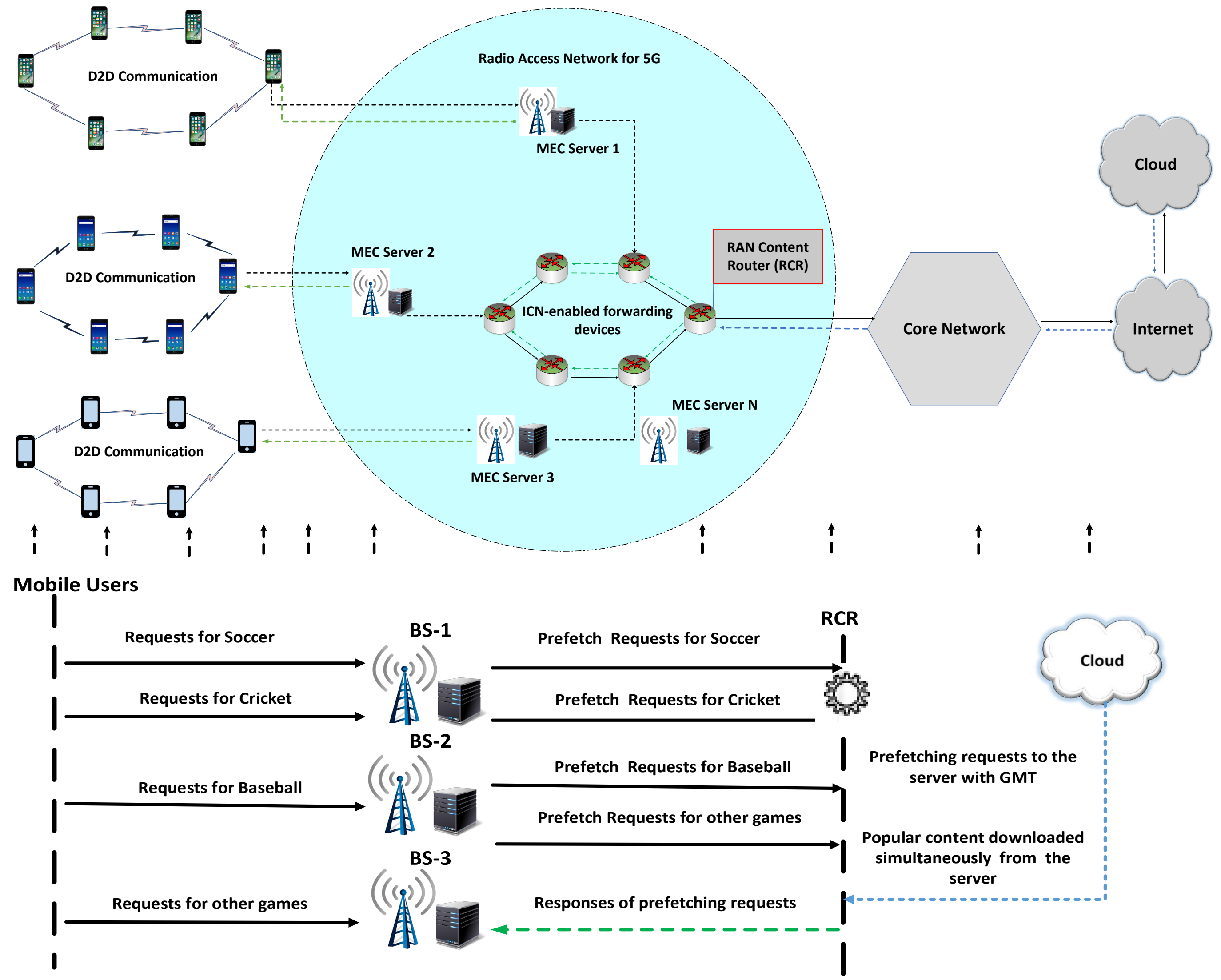


Figure 7: Architectural diagram of ICN-edge-enabled 5G system architecture

requests arriving at BS-2 have a different trend (1000 for Soccer, 800 for Cricket, and 600 for Baseball). BS-2 prefetches the latest content for Soccer requests and stores it in the CS (subject to CS resource availability). At BS-3, the trend is again different (Baseball 1000, Cricket 400, and Soccer 900); BS-3 will prefetch the latest Baseball content from the origin server. When all such requests pass through the RCR node, content popularity is checked because RCR is receiving requests from different BSs. The threshold value for the RCR node may differ from the BS threshold value because this node may be connected to multiple BS servers with different trends at each.

Fig. 7 shows an example where multiple MEC servers are deployed, all of which receive requests from mobile users. The MEC servers then forward these requests to ICN-enabled content routers deployed between the core network and the BSs. As the traffic from all MEC servers passes through the RCR node, the volume of traffic at RCR will be higher than at the BSs. The RCR node checks the popularity of requests from all BSs and prefetches the latest popular content in its CS. The Least Recently Used (LRU) replacement policy was employed in our example, but other replacement policies such as Least Frequently Used (LFU) can also be employed. Based on the scores in the above example, the RCR node receives a total of 2200 requests for Cricket, 2300 for Baseball, and 2400 for Soccer. Assuming a threshold value of 2000 at RCR, the most frequent requests (in our case Soccer) would be prefetched from the origin server. This means that access to the core network is minimized, and content is prefetched at the RCR and BSs.

4. Performance Evaluation

Evaluation of ICN-based edge computing related proposals is an unwieldy task since there is no dedicated open source testbed/simulator available. To evaluate our proposed ICN edge-enabled 5G architecture, we develop our own evaluation system. In this section, we first introduce the experimental

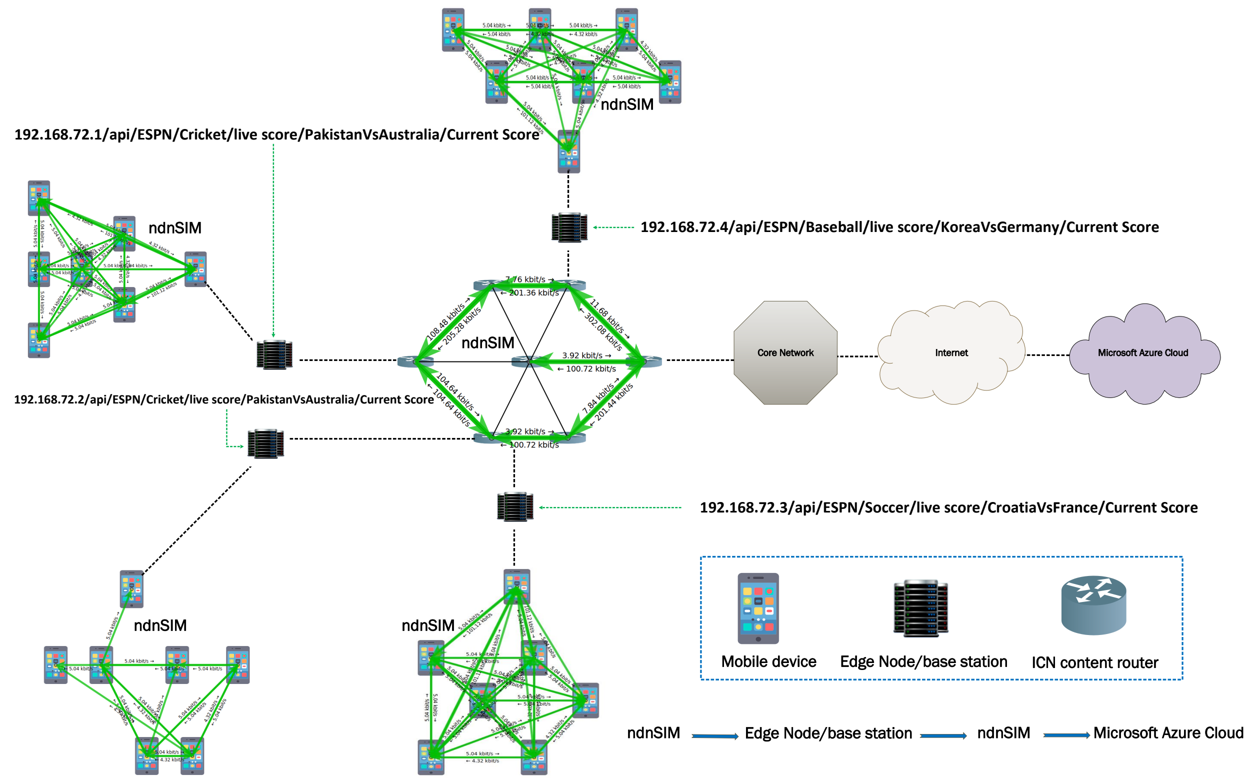


Figure 8: Network Model for Simulations

setup. After that ICN application layer implementation details for the testbed is presented.

4.1. Experimental Setup

For the evaluation of our architecture, we make amendments to the ICN-based edge computing testbed that we have developed [47]. The implementation details of the edge application is given in the next subsection. Fig. 8 shows our testbed model for the evaluation of our proposed architecture which comprises of ndnSIM simulator [20] [48], an Edge nodes, and a Cloud machine (Microsoft Azure). Hence, the real devices in our evaluations are Microsoft Azure Cloud machine and Edge PC. For D2D communication, we generate requests from ndnSIM to the Edge node (e.g., BS). ndnSIM is running on a Linux virtual machine having 8 GB RAM and 4 cores of CPU. For the determination of the performance of in-network caching in ICN, we adopted the Mandelbrot-Zipf (MZipf) [49] distribution in ndnSIM to represent content popularity with the settings: p = 5.0, and α= 0.7-1.2. These are common values used in literature where actual traces observed from Internet Service Providers (ISPs) and CDNs [50], [51]. We vary the catalogue size from 1000 to 5000 data objects and we use a certain threshold for content popularity at BS and RCR node to highlight the benefits of our scheme. The total simulation time is 120 seconds.

For the BS, we developed our own EC application. We employed various popular .NET framework technologies such as .NET framework class libraries, SQL Database for content storage, JSON for the FIB entries, Memory Cache for short time storage of PIT entries, Entity Framework 6 for the communication between our ICN-based EC application to Database, and LINQ-to-JSON for the communication between our application and the FIB JSON file. For API deployment, we are using the Microsoft Internet Information Services (IIS) which host websites, web applications, and services needed by users or developers. The ICN EC application is hosted one hop away from ndnSIM on a system equipped with 16 GBs of memory and an i7-4710HQ CPU @2.40GHz.

As shown in Fig. 8, the mobile devices use ndnSIM for generating requests for content. We deployed a total number of 28 nodes (mobile devices as shown in Fig. 8).The local ICN mobile devices are connected with the BS device located one hop away through a web API. As we consider the use case of ESPN live scoring, requests carry the name of the content and the IP address of the BS. For instance, a request name would be in the form 192.168.72.1/api/ESPN/Soccer/live-score/KoreaVsGermany/Current-Score. The first part of the name represents the IP address where the edge node (BS) is located. The second part of the name represents the content itself. Once the request arrives at the BS, the BS will check whether it has the content cached locally. If not, the BS forwards the interest packet to ICN-enabled forwarding devices deployed between the BS and the core network. As we do not have real ICN-enabled forwarding devices, we also used ndnSIM to simulate such devices. The content is checked in ICN-enabled forwarding devices. If found, it sends back to the requested user via BS. Otherwise it is forwarded to the cloud via passing through core network. It is to be noted that ndnSIM is running at two different virtual machines (e.g., one at local level for D2D and the other is between the BS and core network).

4.2. Application layer ICN implementation for BS

The proposed architecture consists of an application layer ICN implementation with multiple sub layers such as API layer, CS Implementation layer, PIT Entries layers, Forwarder layer and Data Access layer as illustrated in Fig. 9. The implementation details of all sublayers are discussed in detail as follows.

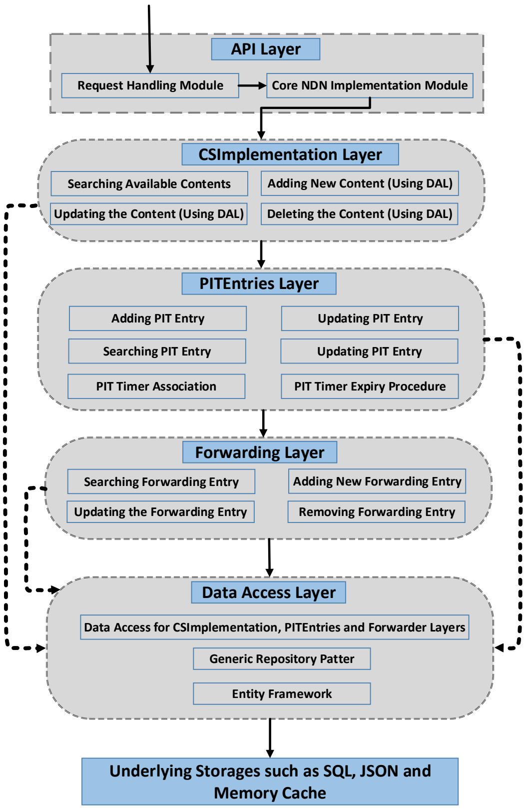


Figure 9: Application layer ICN implementation for base station

4.2.1. API layer

We provide an API layer at the BS device that has two core responsibilities: 1) handling requests from mobile users; and 2) implementing the fundamental ICN/NDN data structures. All the requests from mobile users (e.g., from ndnSIM) comes in the form of an API and hit the *RequestHandler Controller* of Web API (Microsoft ASP.NET Web API 2.0). These requests are initially transformed into the ICN Interest format inside *RequestProcessing Action* and then forwarded to the *ICNImplemention class*. *ICNImplementation* is a separate *class* inside an API layer which implements the fundamental algorithms of ICN. The *ICNImplementation* class processes the Interest based on the mechanisms of ICN and returns the Data to the *RequestHandler/RequestProcessing Action*, which then returns the result to the mobile user. *ICNImplemention class* employs various .NET class libraries (layers) in order to provide the functionalities of ICN. More details about these class libraries are presented in following subsections.

4.2.2. CS implementation layer

CS Implementation layer is responsible to provide all necessary `C#` methods such as checking CS data, listing the names of all contents that are available in CS (SQL database), adding new content in CS, deleting the content from CS. All of these methods are written in *CSImplementation class* which call the Data Access Layer (DAL) in order to communicate with SQL-based storage. The DAL is explained further in the next subsections.

4.2.3. PIT entries layer

The PIT Entries layer is responsible for providing all necessary features such adding a new PIT entry in the table, setting an event-based timer, associating the timer with the PIT entry, removing the PIT entry after timer expiry, removing the PIT entry after receiving Data packet, and searching through all PIT entries. We used Cache for the storage of PIT entries in the form of a `C#` list. Caching enables to store data in memory for rapid access to improve performance and scalability. One can cache information by using classes in the *System.Runtime.Caching namespace*. The *caching classes* in this namespace provide the following features:

- Abstract types that provide the foundation for creating custom cache implementations.
- A concrete in-memory object cache implementation.

The abstract base caching class (ObjectCache) defines the following caching tasks:

- Creating and managing cache entries.
- Specifying expiration and eviction information.
- Triggering events that are raised in response to changes in cache entries.

4.2.4. Forwarder layer

The responsibility of the forwarder layer is to provide the information of the next forwarding node for a specific name. This layer implements the *Forwarder class* which has various `C#` methods such as adding a new *Forwarder* entry in the JSON collection, finding a *Forwarder* entry by Interest name, removing an entry, and updating an existing entry. For the storage of *Forwarder* entries, we employed a JSON file as a storage system and LINQ-to-JSON `C#` library for the communication of our application with the JSON file storage. The rationale of using JSON is: 1) it is a lightweight data interchange format; 2) it is human-readable; and 3) and it can be used for cross-platform applications. Moreover, it is also convenient to port and embed the JSON file from one EC application to another, if the routes information is the same or in the case of failures.

4.2.5. Data access layer

DAL is the layer where data management occurs thought the use of a database such as SQL, MySQL, Oracle MongoDB, and JSON files. In our implementation, we used an SQL database as CS and a JSON file for storing *Forwarder* information. The software modules in DAL are triggered based on either when communication is needed with the SQL database or with the JSON file. Both of these calls triggered from their respective layer (e.g., *CSImplemention* or *Forwarder*). To communicate with the SQL database, we are using Microsoft Entity (an underlying communication framework between a database and an application). The Entity framework automates all the database related activities for our application. Automated database commands are generated for reading or writing data in the database. The Entity Framework provides relevant libraries and executes relevant queries in the database to create results for our application. For JSON file communications, we employed LINQ-to-JSON library; an API for working with JSON objects and designed with LINQ in order to enable quick querying and creation of JSON objects. Furthermore, we employed repository pattern in order to make separate Data Access classes for SQL database and JSON file. Fig. 9 illustrates the application layer ICN implementation for BS.

4.3. Evaluation Metrics

We evaluate our work with the most relevant work on ICN-capable EC over 5G named ECCN [14] as well as with the existing IP-based RAN caching of a conventional MEC system (cache contents at BSs). To evaluate our framework, we consider the following metrics:

- Average Cache Hit Ratio (CHR): CHR is a measure of how many interest packets a cache can satisfy, compared to how many interests are forwarded in the network.
- Average latency: The average latency is the time consumed by the mobile users, BSs, ICN-enabled forwarding devices, and the core network to fulfill Interest-Data exchanges.

4.4. Results and Analysis

4.4.1. Impact of Cache Hit Ratio without Prefetching

In this round of experiments, we investigate the impact of CHR in our ICN-edge enabled 5G architecture without the proposed prefetching mechanism. Fig. 10 shows the impact of CHR on the catalogue size at the devices level, at the BS, and the ICN forwarding nodes in-between the core network and the BS. The cache size of each Content Router (CR) is fixed at 10 data objects while we vary the catalogue size from 1000 to 5000 data objects. A total of 12000 interest packets are generated. Our results show that for catalogue size of 3000, 4.15% of the requests are satisfied at the D2D level, while the remaining requests are forwarded to edge devices (e.g., BS). When requests arrive at the edge devices, 25.48 % of these requests (2930 requests) are satisfied at the edge devices, and 8571 number of requests are sent to the ICN forwarding nodes. These nodes satisfied 15.24% of the received requests (1307 requests), while 7265 requests are sent to the provider (e.g., Cloud). Our results demonstrated that enabling ICN in EC systems may result in significant reduction of core network traffic and lower latency due to limiting the access of the core network.

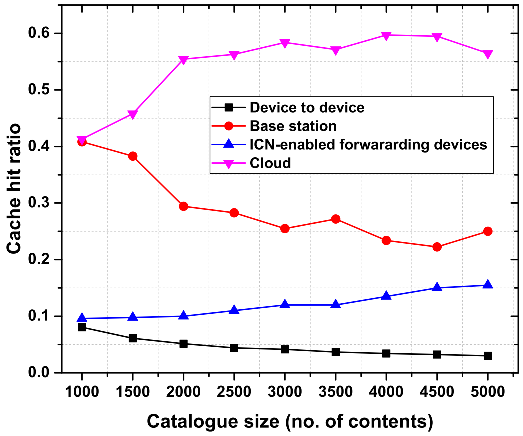


Figure 10: Cache hit ratio as a function of catalogue size (without prefetching)

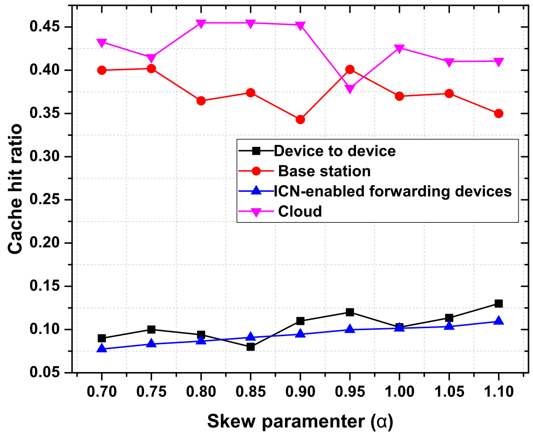


Figure 11: Cache hit ratio as a function of skew parameter (without prefetching)

Fig. 11 shows the impact of CHR with content popularity on our architecture. We consider 9 different values of the skew parameter (α= 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 1.0, 1.05, 1.10). The closer the skew parameter will be to 0, the more ineffective in-network caching will be for the same catalogue size. To illustrate the behavior of skew parameters with respect to the CHR, we chose one value (popular value: α= 1.0) of the skew parameter and described how many requests could be satisfied. A total of 12000 interest packets are generated. We investigate the request satisfaction rate at different levels of our architecture. Our results indicate that for α= 1.0, the D2D handles 13% of the requests, the BS handles of the 35% requests, the ICN forwarding nodes handle 11% of the requests, and 41% of the requests are sent to the cloud via the core network. We also observed that the CHR is inversely proportional to the latency; as more requests can be satisfied from in-network caches, the latency to fetch the requested content is reduced.

4.4.2. Impact of Cache Hit Ratio and Latency with Prefetching

In this round of experiments, we investigated the CHR and latency when we enable the proposed prefetching mechanism. We also evaluated and compared our proposed architecture with recent work on Edge-Centric Computing and Content-Centric Networking in the 5G RAN named ECCN [14] and with the existing IP-based RAN caching of conventional MEC systems denoted as "MEC" for comparison purposes. We evaluated the CHR performance as we vary the catalogue size from 1000 to 5000 data objects. We used the prefetching mechanism at BSs and at the RCR node between the core network and the BS. Fig. 12 illustrates the prefetching impact on our architecture. Our results indicate that for a catalogue size of 3000 objects and for 12000 generated requests, the D2D handles 7% of the requests, BS handles 30% of the requests, the ICN forwarding devices handles 16% of the requests, while 47% of the requests are sent to the cloud via the core network. To this end, our prefetching mechanism results in 8.3% less traffic being forwarded to the core network compared to cases that content prefetching is not enabled.

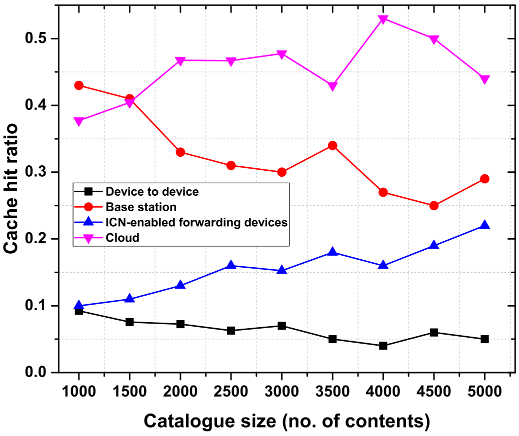


Figure 12: Cache hit ratio as a function of catalogue size (with prefetching)

We also measured the CHR performance with content popularity using our prefetching scheme. Fig. 13 shows CHR as a function of the varying skew parameter (α= 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 1.0, 1.05, 1.10). In Fig. 10, we confirmed that when prefetching is not enabled 55.13% requests are sent to the cloud via the core network. However, when we apply our proposed prefetching mechanism, we observe a different trend at various locations of our architecture. For instance, when 12000 interests are generated using skew parameter (α= 1.10), the D2D handles 15% of the requests, BS handles 40% of the requests, the ICN forwarding devices handles 16% of the requests, and 29% of the requests are sent to the cloud. That being said, 71% of the traffic is handled in the 5G RAN and only 29% of the traffic is sent to the cloud. This shows that with prefetching, 26.13% less traffic is forwarded through the core network, limiting the access to the core network and ultimately reducing the content retrieval latency.

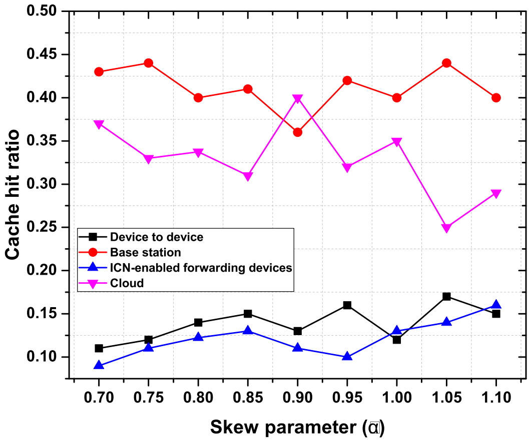


Figure 13: Cache hit ratio as a function of skew parameter (with prefetching)

To study the effect of prefetching on latency, we investigated the content retrieval latency with and without prefetching. Fig. 14 depicts the average latency as a function of skew parameter. Our results show that when prefetching is not enabled, the latency is higher in comparison with cases where prefetching is enabled. The proposed prefetching scheme brings the content closer to users in the RAN and limits the access to the core network, reducing the overall latency. For (α= 1.10) without prefetching, the overall latency in the network is 61ms. However, when the proposed prefetching scheme is enabled, the latency reduces to 39ms. The main reason is that prefetching downloads the content in advance form the server and brings in the 5G RAN. As a result, the access to the core network is limited resulting in considerable latency reduction in the network.

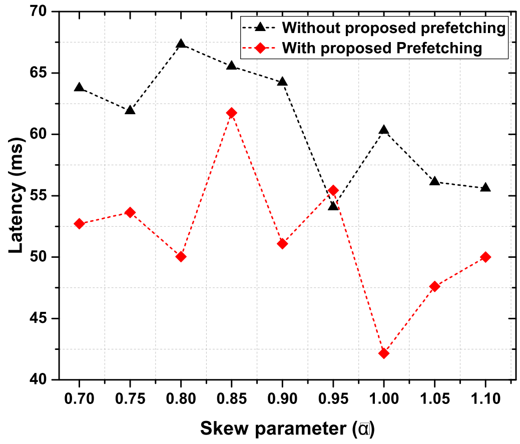


Figure 14: Average latency as a function of skew parameter (with and without prefetching)

Fig. 15 shows CHR as a function of the catalogue size. The cache size is fixed at 10 data objects while we vary the catalogue size from 1000 to 5000 data objects. In this case, we

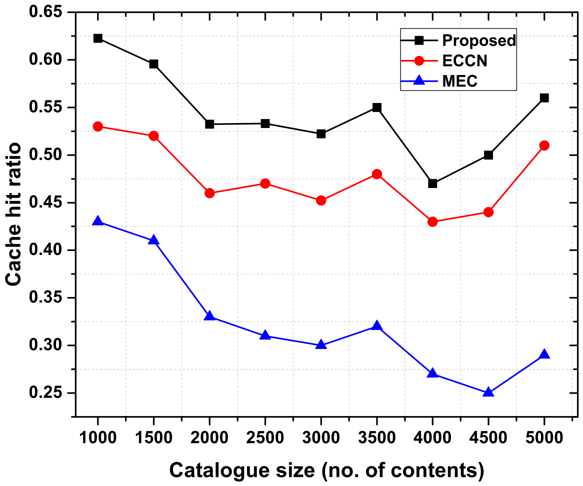


Figure 15: Cache hit ratio as a function of catalogue size

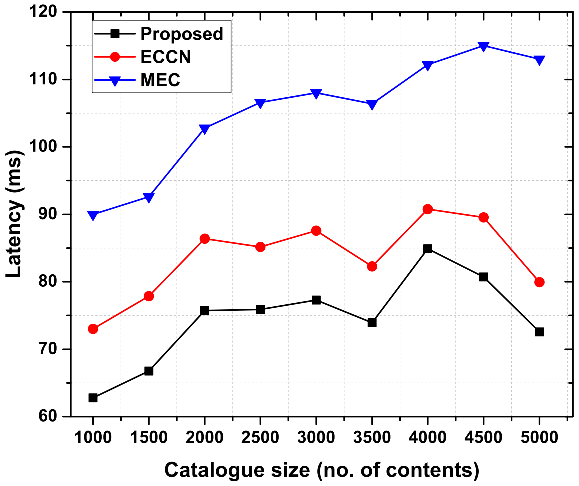


Figure 16: Average latency as a function of catalogue size

use a fixed value for the skew parameter (α= 1.10) that demonstrates reasonable content popularity. Our results show that our proposed method performs better than the conventional MEC and ECC. Since we enable D2D communication in architecture, before sending requests to the BS, devices try to retrieve the content from nearby devices. About 10%–12% of cache hits occur due to the local D2D communication, while content prefetching also contributes to bringing the content closer to mobile users in the RAN. Our results show that for a catalogue size of 2000 objects, the CHR of our method in RAN is 54%, while it is 47% for ECCN and 33% for MEC. In other words, 46% of the traffic is sent to the cloud in our proposed architecture, while 53% of the traffic is sent to cloud in ECCN and 67% in MEC. This is due to the fact that in ECCN, ICN forwarding devices are used between the core network and the BS, therefore, the content may be cached on the forwarding path to the core network; however, in the case of MEC, if the content is not found at the edge server, the BS sends the request(s) directly to the cloud via the core network.

We also investigated the correlation between cache hits and latency. The results of Fig. 16 demonstrate that network latency is inversely related to CHR. The higher the cache hit is, the lower latency will be. Our results also show that for a catalogue size of 2000, the average latency of our proposed architecture is 75ms, while the latency of ECCN is 86ms and the latency of MEC is 102ms. The reason is that in our proposed architecture most of the content is cached at local ICN (D2D communication), BSs and ICN forwarding devices. Moreover, the proposed content prefetching mechanism brings more content in the RAN, limiting the access to the core network, therefore, reducing the overall latency. On the other hand, in ECC the ICN forwarding devices cache the content after the BSs, while, in MEC, only the BSs cache the content and most of the requests when not fulfilled from the BSs are sent to the cloud via the core network.

5. Challenges and Future Research Directions

Enabling ICN with EC for 5G networks is not an easy task and this area needs further heed. There are several open issues and challenges for future research yet to be explored. Enabling ICN for 5G may result in various open research challenges especially in D2D, Edge, RAN, core network, backhauling, caching, and resource management. To this end, we discuss the open issues, challenges, and future directions for researchers below.

5.1. *Pricing*

In conventional cellular systems, the operator charges users based on their usage. In such systems, the operator directly controls the devices and get charges for such control. However, in direct D2D communication, there will be no direct involvement of the operator. Therefore, the operator should answer the question of *what to charge for* in D2D communication. In our architecture, we enabled direct D2D communication without the control of the operator. Therefore, it is of upmost importance to design a business model and a policy on how the operator will charge the devices in such a D2D environment. The decision of pricing models for users in D2D communication needs an extensive analysis of usage cases and new business models. Moreover, it should be noted that the pricing model should be based on the resource utilization of users relaying devices. In other words, in D2D communication, the devices will act as relays for other users, having their battery, storage, bandwidth and other resources utilized due to performing such relaying functions for other users.

5.2. *Security*

As the data of users will be relayed via other devices in D2D communication, data security and privacy are critical factors. In our work, we assumed that the devices are trusted. However, in realistic scenarios, it is impractical to assume trusted devices. It is also possible that a small-sized network such as a university or a local federated environment can make a list of devices in order to authenticate these devices. Such devices can protect the confidentiality of the shared data through encryption. However, for a complete system deployment, one should explore the possible solutions for security issues in D2D communication [52], [53] both in software and in hardware [54].

5.3. Interest broadcast storm and producer mobility

In ICN, multiple Interests may request successive content. Therefore, the interest transmission rate and traffic flow should be controlled according to the network resources [55], [56]. Moreover, the devices in D2D communication aim to fetch the content from nearby devices. Although ICN provides built-in support for consumer mobility, producer mobility will also be frequent in our environment. To this end, mechanisms to make the producer reachable while moving need to be further explored [57], [58], [59]. If the data cannot be fetched from the producer, consumers will need to explore Interest retransmission mechanisms hoping that they will eventually reach the producer. Alternatively, producers could store the data they produced at data repositories at the edge of the network, so that consumer can retrieve it from there even when producers are not reachable [60], [61].

5.4. Content and Context Namespace Design

Enabling ICN in a 5G architecture creates naming issues with content and context in terms of how to design naming schemes that can embed several contextual characteristics of content. One solution might be to use naming conventions or to move descriptive information that is explicitly linked in a manifest to metadata objects. At the same time, ICN naming should be flexible enough to accommodate various types of content, which can be retrieved under different conditions (e.g., data transfers at various locations between end user mobile devices, edge devices, backhaul and core network). Further research investigation is needed for the design of such naming schemes for ICN-capable edge computing.

5.5. Name based network and protocol design

ICN inherently supports multicast communication, an important feature that is actively used by modern content sharing applications such as YouTube and Netflix. In recent years, it has also become a fundamental requirement for AR applications. However, existing applications try to use the IP-based network architecture for multicasting, while its application scope is rather limited, since it can be only enabled by large service providers. To make matters worse, service providers do not have direct business interest in supporting multicast therefore, they often do not put effort in addressing security and scalability issues. Fundamentally though, the IP-based architecture itself was not created with the consideration of enabling multicast communication. All these issues signify the importance of designing scalable and secure ICN-based solutions for multicast communication to address the communication needs of modern content sharing applications.

5.6. Optimization of network functions at the edge

In edge computing, computation is largely or totally performed on distributed edge computing servers as opposed to primarily taking place in a centralized cloud environment. However, the conditions are highly dynamic at the network edge, therefore, we need to investigate orchestration mechanisms for seamless resource allocation with minimal changes in the overall system [62]. To cope with such situations, it is important to develop features to test the optimization and service performance of the entire system.

5.7. Distribution of computation, communication, control and storage

An ICN-based EC architecture should allow computing, storage, and networking tasks to be dynamically relocated among the Edge, the core network, and the mobile devices. In this case, the fundamental challenges have to do with determining which tasks should be distributed for execution to the edge devices/base stations, the core network, and the cloud, as well as how edge devices, edge, and cloud are supposed to interact with each other.

Another issue might be the fact that edge devices may request a mix of both light-weight and non-trivial computation. As a result, light-weight parts of computational tasks can be executed locally on the devices, while compute-intensive parts need to be offloaded to edge and cloud servers. A typical example may be the case of mobile AR, where multiple mobile AR devices simultaneously start sensing the environment, producing raw videos, and capturing user gestures via their cameras and sensors. After local processing, some of the data will be offloaded to other edge devices with execution instructions. Compute-intensive parts of tasks might take a longer time for computation than light-weight ones [63]. Therefore, the design of robust mechanisms to merge compute-intensive with light-weight parts of tasks need to be investigated.

5.8. Efficient resource and service discovery methods

5G will support the requirements of applications, such as mobile AR, for ultra-low latency. To provide real-time feedback as needed by mobile AR applications, it is necessary to provide the required computing resources and services efficiently and seamlessly [64].In cases of resource (e.g., CPU and RAM capacity) and service availability discovery, a mobile device should have knowledge of which edge node/BS has the adequate available resources and can accommodate its request for a service. As a result, we avoid unnecessary delays due to forwarding a service request towards multiple different edge nodes until we are able to find one that can provide the necessary computing resources. To fulfill this demand, it is important to design an efficient resource/service discovery mechanism using ICN for edge computing [8] as well as exploring computation reuse mechanisms [65].

5.9. Physical and MAC layer amendments and concerns

Latency is a fundamental unit of 5G networks, which can be further enhanced by modifications of the physical and MAC layers. Further investigation is needed to explore promising technologies at the physical layer, such as Millimeter wave, which bring massive new spectrum for communications in the 3-300GHz band. Protocol enhancements at the link layer should also be explored to enable efficient error and loss detection as well as rate adaptation [66]. Moreover, in the absence of a centralized controller in D2D communication, the devices connection setup, interference management, and resource allocation will be challenging. In this regard, distributed mechanisms are needed to establish D2D communication. This area needs to be further explored to understand the impact of D2D communication on cellular systems.

5.10. Content Caching Issues

Recently, edge caching received significant attraction from both academia and industry [67]. Edge caching plays a vital role in latency reduction as well as spectral and energy efficiency improvements. However, it is not a silver bullet and still several research problems are open and need further heed. Specifically, researchers should explore how the latency would be impacted by increasing/decreasing the size of the caches, efficient cache replacement policies, and wireless channel parameters. Overall, researchers are encouraged to study various trade-offs such as capacity versus latency and other parameters while designing efficient cache mechanisms.

Additionally, it is worth mentioning that the popularity graph of content may change over time particularly for the case of dynamic content and use of constant threshold value may decrease the performance of the caching systems. There are multiple ways to optimize the threshold value; either by running extensive simulation experiments, or by analyzing the traffic on the production system. Moreover, various statistical modeling techniques and machine learning algorithms can be employed for the prediction of dynamic network traffic, analyze the incoming traffic on edges and make the threshold value adaptive over time.

5.11. Hybrid Artificial Intelligence-Based Edge Architectures

In addition to all previous topics, an interesting research direction would be to explore the intersection of Software-Defined Networking (SDN) and Artificial Intelligence (AI) with ICN-based edge computing architectures [68]. For example, new SDN protocols for data forwarding in ICN-based EC for 5G could be designed. SDN provides an opportunity because of its programmable and practical nature, since most SDN protocols, such as OpenFlow, are compatible with various networks and devices [69]. Moreover, Edge AI is an important factor for assessing the required computing resources, memory bandwidth, and caching capacity, which are all necessary for supporting 5G services. In this regard, researchers should investigate efficient resource scheduling and coordination schemes and models.

6. Conclusion

In the upcoming 5G networks, the latency of content distribution will be a critical issue and will introduce challenges for the existing network architecture. ICN and EC are considered as promising technologies to reduce network latency by bringing the content closer to the end users. To achieve that, in this paper, we proposed an ICN edge computing architecture for 5G, enabling D2D communication and offering ICN application layer support for BSs in a 5G RAN. Moreover, we proposed a prefetching strategy for dynamic content based on ICN hierarchical naming and content popularity. Our experiments demonstrate that if we enable ICN with EC at various locations in a 5G RAN, the access to the core network and the content access latency are minimized. Finally, we presented challenges and issues associated with ICN and EC for 5G as well as we presented future research directions.

Acknowledgement

This work was partially supported by the International Research & Development Program of the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No. NRF-2018K1A3A1A39086819), and by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (No. 2018R1A2B6002399

References

[1] 5g america, white paper, Available online: https://www.5gamericas.org/5g-at-the-edge/ (accessed on 15 September 2019) (October 2019).

[2] Ericsson mobility report, Available online: https://www.ericsson.com/49d1d9/assets/local/mobility-report/documents/2019/ericsson-mobility-reportjune-2019.pdf (accessed on 16 September 2019) (June 2019 edition).

[3] B. Nour, K. Sharif, F. Li, S. Biswas, H. Moungla, M. Guizani, Y. Wang, A survey of internet of things communication using icn: A use case perspective, Computer Communications 142-143 (2019) 95 – 123. URL: http://www.sciencedirect.com/science/article/pii/S0140366418309228. doi:`https://doi.org/10.1016/j.comcom.2019.05.010`.

[4] B. Nour, K. Sharif, F. Li, H. Moungla, Y. Liu, A unified hybrid information-centric naming scheme for iot applications, Computer Communications 150 (2020) 103 – 114. URL: http://www.sciencedirect.com/science/article/pii/S0140366419306553. doi:`https://doi.org/10.1016/j.comcom.2019.11.020`.

[5] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, IEEE Communications Magazine 50 (2012) 26–36. doi:`10.1109/MCOM.2012.6231276`.

[6] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, G. C. Polyzos, A survey of information-centric networking research, IEEE Communications Surveys Tutorials 16 (2014) 1024–1049. doi:`10.1109/SURV.2013.070813.00063`.

[7] A. Mtibaa, R. Tourani, S. Misra, J. Burke, L. Zhang, Towards edge computing over named data networking, in: 2018 IEEE International Conference on Edge Computing (EDGE), 2018, pp. 117–120. doi:`10.1109/EDGE.2018.00023`.

[8] S. Mastorakis, A. Mtibaa, Towards service discovery and invocation in data-centric edge networks, in: 2019 IEEE 27th International Conference on Network Protocols (ICNP), IEEE, 2019, pp. 1–6.

[9] S. Mastorakis, A. Mtibaa, J. Lee, S. Misra, ICedge: When Edge Computing Meets Information-Centric Networking, IEEE Internet of Things Journal (2020).

[10] M. Król, S. Mastorakis, D. Oran, D. Kutscher, Compute first networking: Distributed computing meets icn, in: Proceedings of the 6th ACM Conference on Information-Centric Networking, 2019, pp. 67–77.

[11] J. Pan, J. McElhannon, Future edge cloud and edge computing for internet of things applications, IEEE Internet of Things Journal 5 (2018) 439–449. doi:`10.1109/JIOT.2017.2767608`.

[12] M. Chiang, T. Zhang, Fog and iot: An overview of research opportunities, IEEE Internet of Things Journal 3 (2016) 854–864. doi:`10.1109/JIOT.2016.2584538`.

[13] X. Wang, M. Chen, T. Taleb, A. Ksentini, V. C. M. Leung, Cache in the air: exploiting content caching and delivery techniques for 5g systems, IEEE Communications Magazine 52 (2014) 131–139. doi:`10.1109/MCOM.2014.6736753`.

[14] H. Li, K. Ota, M. Dong, Eccn: Orchestration of edge-centric computing and content-centric networking in the 5g radio access network, IEEE Wireless Communications 25 (2018) 88–93. doi:`10.1109/MWC.2018.1700315`.

[15] Z. Zhang, C.-H. Lung, I. Lambadaris, M. St-Hilaire, When 5g meets icn: An icn-based caching approach for mobile video in 5g networks, Computer Communications 118 (2018) 81 – 92. URL: http://www.sciencedirect.com/science/article/pii/S0140366417305066. doi:`https://doi.org/10.1016/j.comcom.2017.10.002`.

[16] R. Ravindran, P. Suthar, D. Trossen, C. Wang, G. White, Enabling icn in 3gpp's 5g nextgen core architecture, Technical Report, ICN Research Group, Internet-Draft (2019.).

[17] P. Suthar, M. Stolic, A.Jangam, D. Trossen, R. Ravindran, Native deployment of icn in lte, 4g mobile networks, Technical Report, ICN Research Group, Internet-Draft (2019.).

[18] S. Mastorakis, A. Afanasyev, Y. Yu, L. Zhang, nTorrent: Peer-to-peer file sharing in named data networking, in: 2017 26th International Conference on Computer Communication and Networks (ICCCN), IEEE, 2017, pp. 1–10.

[19] S. Mastorakis, Peer-to-peer data sharing in named data networking, Ph.D. thesis, UCLA, 2019.

[20] S. Mastorakis, A. Afanasyev, L. Zhang, On the evolution of ndnsim: An open-source simulator for ndn experimentation, SIGCOMM Comput. Commun. Rev. 47 (2017) 19–33. URL: http://doi.acm.org/10.1145/3138808.3138812. doi:`10.1145/3138808.3138812`.

[21] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, D. Sabella, On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration, IEEE Communications Surveys Tutorials 19 (2017) 1657–1681. doi:`10.1109/COMST.2017.2705720`.

[22] S. Shannigrahi, S. Mastorakis, F. R. Ortega, Next-generation networking and edge computing for mixed reality real-time interactive systems, in: 2020 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2020.

[23] H. T. Dinh, C. Lee, D. Niyato, P. Wang, A survey of mobile cloud computing: architecture, applications, and approaches, Wireless Communications and Mobile Computing 13 (2013) 1587–1611. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/wcm.1203. doi:`10.1002/wcm.1203`.

[24] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for vm-based cloudlets in mobile computing, IEEE Pervasive Computing 8 (2009) 14–23. doi:`10.1109/MPRV.2009.82`.

[25] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, MCC '12, ACM, New York, NY, USA, 2012, pp. 13–16. URL: http://doi.acm.org/10.1145/2342509.2342513. doi:`10.1145/2342509.2342513`.

[26] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, V. Young, Mobile edge computing-a key technology towards 5g, ETSI, Sophia Antipolis, France, White Paper 11 (2015).

[27] M. Paoloni, Power at the edge: Processing and storage move from central core to the network edge,, Greater Seattle Area, Senza Fili Consulting, Tech. Rep., Available online: http://www.senzafiliconsulting.com 11 (Feb. 2017.).

[28] R. Ullah, S. H. Ahmed, B. Kim, Information-centric networking with edge computing for iot: Research challenges and future directions, IEEE Access 6 (2018) 73465–73488. doi:`10.1109/ACCESS.2018.2884536`.

[29] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, P. A. Polakos, A comprehensive survey on fog computing: State-of-the-art and research challenges, IEEE Communications Surveys & Tutorials 20 (2017) 416–464.

[30] Y. Mao, C. You, J. Zhang, K. Huang, K. B. Letaief, A survey on mobile edge computing: The communication perspective, IEEE Communications Surveys Tutorials 19 (2017) 2322–2358. doi:`10.1109/COMST.2017.2745201`.

[31] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, IEEE Communications Magazine 50 (2012) 26–36. doi:`10.1109/MCOM.2012.6231276`.

[32] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang, Named data networking, ACM SIGCOMM Computer Communication Review 44 (2014) 66–73.

[33] Ccnx-content centric networking, Available online: https://www.ccnx.org/ (accessed on 20 July, 2019).

[34] A. Afanasyev, J. Shi, et al., NFD Developer's Guide, Technical Report NDN-0021, NDN, 2015.

[35] R. Ravindran, A. Chakraborti, S. O. Amin, A. Azgin, G. Wang, 5g-icn: Delivering icn services over 5g using network slicing, IEEE Communications Magazine 55 (2017) 101–107.

[36] Y. Cao, A cooperation-driven icn-based caching scheme for mobile content chunk delivery at ran (2017).

[37] C. Liang, F. R. Yu, X. Zhang, Information-centric network function virtualization over 5g mobile wireless networks, IEEE Network 29 (2015) 68–74. doi:`10.1109/MNET.2015.7113228`.

[38] G. Grassi, D. Pesavento, G. Pau, R. Vuyyuru, R. Wakikawa, L. Zhang, Vanet via named data networking, in: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2014, pp. 410–415. doi:`10.1109/INFCOMW.2014.6849267`.

[39] M. Sheng, C. Xu, J. Liu, J. Song, X. Ma, J. Li, Enhancement for content delivery with proximity communications in caching enabled wireless networks: architecture and challenges, IEEE Communications Magazine 54 (2016) 70–76. doi:`10.1109/MCOM.2016.7537179`.

[40] M. Amadeo, C. Campolo, A. Molinaro, Empowering 5g network softwarization through information centric networking, Internet Technology Letters 1 (2018) e30. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/itl2.30. doi:`10.1002/itl2.30`.

[41] M. Sifalakis, B. Kohler, C. Scherb, C. Tschudin, An information centric network for computing the distribution of computations, in: Proceedings of the 1st ACM Conference on Information-Centric Networking, ACM-ICN '14, ACM, New York, NY, USA, 2014, pp. 137–146. URL: http://doi.acm.org/10.1145/2660129.2660150. doi:`10.1145/2660129.2660150`.

[42] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev, L. Zhang, An overview of security support in named data networking, IEEE Communications Magazine 56 (2018) 62–68. doi:`10.1109/MCOM.2018.1701147`.

[43] B. Nour, K. Sharif, F. Li, Y. Wang, Security and privacy challenges in information-centric wireless internet of things networks, IEEE Security Privacy (2019) 1–1. doi:`10.1109/MSEC.2019.2925337`.

[44] Entertainment and sports programming network, https://www.espn.com/, accessed on 10 September, 2019.

[45] M. A. U. Rehman, D. Kim, K. Choi, R. Ullah, B. S. Kim, A statistical performance analysis of named data ultra dense networks, Applied Sciences 9 (2019). URL: https://www.mdpi.com/2076-3417/9/18/3714. doi:`10.3390/app9183714`.

[46] M. A. U. Rehman, R. Ullah, B. S. Kim, Ninq: Name-integrated query framework for named-data networking of things, Sensors 19 (2019). URL: https://www.mdpi.com/1424-8220/19/13/2906. doi:`10.3390/s19132906`.

[47] R. Ullah, M. A. U. Rehman, B. Kim, Design and implementation of an open source framework and prototype for named data networking-based edge cloud computing system, IEEE Access 7 (2019) 57741–57759. doi:`10.1109/ACCESS.2019.2914067`.

[48] S. Mastorakis, A. Afanasyev, I. Moiseenko, L. Zhang, ndnSIM 2.0: A new version of the NDN simulator for NS-3, NDN, Technical Report NDN-0028 (2015).

[49] M. Hefeeda, O. Saleh, Traffic modeling and proportional partial caching for peer-to-peer systems, IEEE/ACM Transactions on Networking 16 (2008) 1447–1460. doi:`10.1109/TNET.2008.918081`.

[50] S. K. Fayazbakhsh, Y. Lin, A. Tootoonchian, A. Ghodsi, T. Koponen, B. Maggs, K. Ng, V. Sekar, S. Shenker, Less pain, most of the gain:

Incrementally deployable icn, SIGCOMM Comput. Commun. Rev. 43 (2013) 147–158. URL: http://doi.acm.org/10.1145/2534169.2486023. doi:`10.1145/2534169.2486023`.

[51] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, S. Moon, I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system, in: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, IMC '07, ACM, New York, NY, USA, 2007, pp. 1–14. URL: http://doi.acm.org/10.1145/1298306.1298309. doi:`10.1145/1298306.1298309`.

[52] F. Ahmad, C. A. Kerrache, F. Kurugollu, R. Hussain, Realization of blockchain in named data networking-based internet-of-vehicles, IT Professional 21 (2019) 41–47. doi:`10.1109/MITP.2019.2912142`.

[53] E. Barka, C. A. Kerrache, R. Hussain, N. Lagraa, A. Lakas, S. H. Bouk, A trusted lightweight communication strategy for flying named data networking, Sensors 18 (2018). URL: https://www.mdpi.com/1424-8220/18/8/2683. doi:`10.3390/s18082683`.

[54] S. Mastorakis, T. Ahmed, J. Pisharath, Isa-based trusted network functions and server applications in the untrusted cloud, arXiv preprint arXiv:1802.06970 (2018).

[55] A. Benmoussa, A. e. K. Tahari, N. Lagaa, A. Lakas, F. Ahmad, R. Hussain, C. A. Kerrache, F. Kurugollu, A novel congestion-aware interest flooding attacks detection mechanism in named data networking, in: 2019 28th International Conference on Computer Communication and Networks (ICCCN), 2019, pp. 1–6. doi:`10.1109/ICCCN.2019.8847146`.

[56] R. Ullah, M. A. U. Rehman, B. S. Kim, Hierarchical name-based mechanism for push-data broadcast control in information-centric multihop wireless networks, Sensors 19 (2019). URL: https://www.mdpi.com/1424-8220/19/14/3034. doi:`10.3390/s19143034`.

[57] B. Nour, H. Ibn Khedher, H. Moungla, H. Afifi, F. Li, K. Sharif, H. Khelifi, M. Guizani, Internet of things mobility over information-centric/named-data networking, IEEE Internet Computing (2019) 1–1. doi:`10.1109/MIC.2019.2963187`.

[58] Y. Zhang, Z. Xia, S. Mastorakis, L. Zhang, Kite: producer mobility support in named data networking, in: Proceedings of the 5th ACM Conference on Information-Centric Networking, ACM, 2018, pp. 125–136.

[59] C. Xu, P. Zhang, S. Jia, M. Wang, G. Muntean, Video streaming in content-centric mobile networks: Challenges and solutions, IEEE Wireless Communications 24 (2017) 157–165.

[60] R. Ullah, M. K. Afzal, B. S. Kim, A novel parallel processing mechanism for data transmission in wireless content-centric networking, Journal of Intelligent and Fuzzy Systems 35 (2018) 5815–5825.

[61] A.-C. Nicolaescu, O. Ascigil, I. Psaras, Edge data repositories-the design of a store-process-send system at the edge, in: Proceedings of the 1st ACM CoNEXT Workshop on Emerging in-Network Computing Paradigms, 2019, pp. 41–47.

[62] C. Argyropoulos, S. Mastorakis, K. Giotis, G. Androulidakis, D. Kalogeras, V. Maglaris, Control-plane slicing methods in multi-tenant software defined networks, in: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE, 2015, pp. 612–618.

[63] R. Ullah, M. A. Ur Rehman, B. Kim, B. Sonkoly, J. Tapolcai, On pending interest table in named data networking based edge computing: The case of mobile augmented reality, in: 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), 2019, pp. 263–265. doi:`10.1109/ICUFN.2019.8805923`.

[64] R. Ullah, M. A. U. Rehman, B. S. Kim, Poster: A testbed implementation of ndn-based edge computing for mobile augmented reality, in: Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications, HotMobile '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 181. URL: https://doi.org/10.1145/3301293.3309565. doi:`10.1145/3301293.3309565`.

[65] J. Lee, A. Mtibaa, S. Mastorakis, A case for compute reuse in future edge systems: An empirical study, in: 2019 IEEE Globecom Workshops (GC Wkshps), IEEE, 2019, pp. 1–6.

[66] S. Vusirikala, S. Mastorakis, A. Afanasyev, L. Zhang, Hop-by-hop best effort link layer reliability in named data networking, NDN, Technical Report NDN-0041 (2016).

[67] X. Chen, C. Xu, M. Wang, T. Cao, L. Zhong, G. Muntean, Optimal coded caching in 5g information-centric device-to-device communications, in: 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1–7.

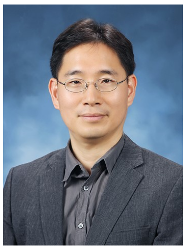
[68] K. Chan, B. Ko, S. Mastorakis, A. Afanasyev, L. Zhang, Fuzzy interest forwarding, in: Proceedings of the Asian Internet Engineering Conference, 2017, pp. 31–37.

[69] H. Khelifi, S. Luo, B. Nour, A. Sellami, H. Moungla, S. H. Ahmed, M. Guizani, Bringing deep learning at the edge of information-centric internet of things, IEEE Communications Letters 23 (2019) 52–55. doi:`10.1109/LCOMM.2018.2875978`.

Rehmat Ullah, Ph.D., is currently working as an Assistant Professor with the Department of Computer Engineering at Gachon University, Seongnam, South Korea. He received the B.S. and M.S. degrees in computer science (major in wireless communications and networks) from COMSATS University Islamabad, Pakistan, in 2013 and 2016, respectively and the Ph.D. degree in electronics and computer engineering from Hongik University, South Korea, in February 2020. Prior to starting his position at Gachon University, he was a Postdoctoral Researcher with the Broadband Convergence Networks Laboratory, Hongik University, South Korea. His research focuses on the broader area of future Internet and network systems, particularly the development of architectures, algorithms, and protocols for emerging paradigms, such as information centric networking/named data networking (ICN/NDN), Internet of Things (IoT), cloud/edge/fog computing for IoT, and 5G and beyond. This includes the design, measurement studies, prototyping, testbed development, and performance evaluations. Furthermore, he serves as a regular reviewer for the IEEE Journal on Selected Areas in Communications, the IEEE Wireless Communications Magazine, Transactions on Emerging Telecommunications Technologies, Elsevier Future Generation Computer Systems, the IEEE Communications Magazine, IEEE Access, the IEEE Network and the ACM Computing Surveys. He also served as a reviewer and/or TPC for various international conferences and workshops including ACM MobiHoc (Los Angeles, USA), IEEE CCNC (Las Vegas, USA), VTC2018-Spring (Porto, Portugal), SIGCSE 2019 (Special Sessions: Minneapolis, Minnesota, USA) and ICWMC 2020 (Athens, Greece).

Muhammad Atif Ur Rehman received the B.S. degree in Electronics & Communication from The University of Lahore, Lahore, Pakistan, in 2013, and the M.S. degree in Computer Science from the COMSATS Institute of Information Technology, Islamabad, Pakistan, in 2016. He is currently pursuing the Ph.D. degree with the Broadband Convergence Networks Laboratory with the Department of Electronics and Computer Engineering, Hongik University, South Korea. His major interests are in the field of Information Centric Wireless Networks, Named Data Networking, Wireless Sensor Networking, Edge Computing, Internet of Things and 5th Generation communication. He serves as a regular reviewer for the various IEEE & Elsevier Journals. Moreover, he also served as a reviewer and/or TPC for the international conferences and workshops. From 2013 to 2018, he was working as software engineer & architect in leading IT companies in Pakistan. His responsibilities were to write well design, testable, and efficient code, collaborate with other team members to determine functional and non-functional requirements for new software or application, provide technical guidelines to other team members and ensure software meets all requirements of quality. He worked closely with Quality Assurance team to deliver high quality and reliable products.

Muhammad Ali Naeem is a Ph.D Scholar at InterNetWorks Research Lab, School of Computing, College of Arts and Sciences, Universiti Utara Malaysia. He received his M.Sc. degree with 76% in Computer Science at COMSATS Institute of Information Technology Pakistan. He is currently pursuing PhD in networking (Information Centric Networking).



Byung-Seo Kim, Ph.D., (M'02-SM'17) received his B.S. degree in Electrical Engineering from In-Ha University, In-Chon, Korea in 1998 and his M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Florida in 2001 and 2004, respectively. His Ph.D. study was supervised by Dr. Yuguang Fang. Between 1997 and 1999, he worked for Motorola Korea Ltd., PaJu, Korea as a Computer Integrated Manufacturing (CIM) Engineer in Advanced Technology Research and Development (ATR&D). From January 2005 to August 2007, he worked for Motorola Inc., Schaumburg Illinois, as a Senior Software Engineer in Networks and Enterprises. His research focuses in Motorola Inc. were designing protocol and network architecture of wireless broadband mission critical communications. From 2012 to 2014, he was the Chairman with the Department of Software and Communications Engineering, Hongik University, South Korea, where he is currently a Professor. He served as the General Chair for General Chair of 3rd IWWCN 2017, and the TPC member for the IEEE VTC 2014-Spring and the EAI FUTURE2016, and ICGHIC 2016 2019 conferences. He served as Guest Editors of special issues of International Journal of Distributed Sensor Networks (SAGE), IEEE Access, and Journal of the Institute of Electrics and Information Engineers. He was also served as the Member of Sejong-city Construction Review Committee and Ansan-city Design Advisory Board. His work has appeared in around 167 publications and 22 patents. He is IEEE Senior Member and Associative Editor of IEEE Access. His research interests include the design and development of efficient wireless/wired networks including link-adaptable/cross-layer-based protocols, multi-protocol structures, wireless CCNs/NDNs, Mobile Edge Computing, physical layer design for broadband PLC, and resource allocation algorithms for wireless networks.

Spyridon Mastorakis, Ph.D., is an Assistant Professor in Computer Science at the University of Nebraska, Omaha. He received his Ph.D. in Computer Science from the University of California, Los Angeles (UCLA) in 2019. He also received an MS in Computer Science from UCLA in 2017 and a 5-year diploma (equivalent to M.Eng.) in Electrical and Computer Engineering from the National Technical University of Athens (NTUA) in 2014. His research interests include network systems, Internet architectures and protocols (such as Information-Centric Networking and Named-Data Networking), and edge computing.