Binary Task Offloading Model For Mobile Edge Computing using NDN Architecture

Supratik Banerjee[1]

[1]NIIT University, Neemrana, Rajasthan
supratikbanerjee84@gmail.com

Abstract**: *Driven by the advantages of Mobile Edge Computing (MEC) and Information-Centric Networking (ICN) in next-generation networks we propose an architecture for MEC using Named Data Networking (NDN). NDN is one of the prominent architectures of ICN having features like unique-naming, in-network caching, inherit support for multicasting, and support for mobility. Placing MEC in NDN provides the additional facilities of edge computing like pushing of resource-hungry and time-critical applications of the mobile devices to the edge-computing server. Therefore, one of the research challenges is the decision regarding the task offloading process by the end-users to the edge-computing server. We propose a mathematical model that enables the end-user to take decisions in Yes/No regarding the binary task offloading process.***

Keywords**: Binary task offloading, Information-centric Networking, Mobile Edge Computing, Named Data Networking**

(Article history: Received: March 2022 and accepted May 2022)

I. INTRODUCTION

In the area of communications, fifth-generation wireless networks (5G) is a significant milestone. Hence, in the 5G era, communication technologies like millimeter-wave communications, device-to-device (D2D) communications, and hyper-dense networks are quite prominent [1] [2] [3]. Recently, a state-of-the-art internet design called Information-centric networking (ICN) came into prominence. ICN prioritizes content (i.e. data or information), and proposes to remove some of the drawbacks that current internet architecture is facing like efficiency, security, and network congestion. In addition to that, unique naming, multi-cast communication, and in-network caching are those primary features of ICN that minimize the response time and the load on the server [4]. In-network caching facilities optimize the content distribution process and relieve the transmission stress on the network due to the growing internet traffic.

While framing the caching policies the researchers consider the system-wide internet topology or vast internet area. Thus, these caching policies fail to give good results for small size or medium size networks having a single routing policy and finite set of network devices [5]. Furthermore, with the advent of wireless communication, there are additional issues like user mobility, changing topology, limited processing power, and battery life of mobile devices [6]. Edge computing is a promising area that can resolve some of these challenges. Edge computing proposes to place the computing facility at the network edge. Thus, the energy-constrained mobile devices can save their energy and computation power by offloading their resource-hungry tasks to the edge node [7]. Edge caching in ICN improves the QoE in wireless networks [8]. Placing Mobile Edge Computing (MEC) and cloud computing facilities at the network edge in ICN eliminates the disadvantages like back-haul bandwidth interference and larger latency in mobile cloud computing. Researchers are also using MEC for data delivery in cellular networks. This provides considerable improvement in cellular networks [9]. Recent advancement in MEC especially in the area of mobile wireless technology provides efficient and high-speed real-time communication services. In addition to that, it also provides better efficiency and better data rates. However, management of cloud servers and connected devices are some of the additional issues that need to be taken care of in MEC. With the increase in the number of devices the volume of information that the MEC server needs to process is also increasing. The amount of multimedia services in the networks is continuously growing. Thus, improvement in the 5G cellular network data rate is mandatory. To accomplish this goal, it is preferable to deploy more computational resources at the network edge [10].

The rest of the paper is organized as follows, in section-II, we discuss the State-of-the-Art research work. In section-III, we provide a brief overview of NDN. In section IV, we discuss the edge computing mechanism over NDN architecture. In section V, we discuss the system model and the binary task offloading model. Followed by conclusion and future work in section-VI.

II. LITERATURE REVIEW

Edge caching is playing a pivotal role for MEC to improve the overall performance of the wireless network. The MEC edge caching reduces the distance of transmission, reduce the transmission delay, minimizes energy consumption, and provides caching facility for mobile users. Thus, edge caching enhances the overall performance of the network [11]. In [12] the researchers presented the survey in the domain of Edge computing in ICN. The use of a naming mechanism instead of IP addresses and an efficient content delivery system for information dissemination paves the way for optimization of performance for future networks. Edge computing in ICN reduces the response time. Hence, it resolves issues relevant to big data management, user mobility, bandwidth, node identification, and privacy. Hence, edge computing is playing an important role in future networks [12]. In [13] researchers observed that network performance can be significantly improved by reducing the round trip time and transmission length. Edge computing achieves both these goals as it reduces the proximity to end-users. Apart from that edge computing also provide a better mechanism for network-wide content dissemination, improves the Quality of Service (QoS) of the network.

Strategic cache placement is very important to receive optimal benefits of the caching facility. Thus, a lot of researchers focused on the fundamental questions related to caching like, where to cache? What should be the caching policy? How to design optimal cache space assignment problems? Optimized cache allocation policies are formulated to optimize cache utilization and service quality [14] [15] [16]. This is quite important as an in-network caching facility in ICN improves the overall quality of experience (QoE) of the end-user [17]. In [18] the researchers proposed that Information-centric mobile edge computing is a promising technology for next-generation connected vehicles. In ICN content is the central point. This content is decoupled from the location using content identifiers. Thus, ICN architecture inherently supports user mobility, in-network caching, in-network processing, and multicast communication. However, there are open challenges that need to be addressed for connected vehicles using an Information-centric mobile edge computing facility. Context-based selection, naming strategies for service discovery, dynamic orchestration of automotive services, handling the high degree of mobility, and efficient data dissemination process through the core network to reduce the computation time [18]. In [19] researchers proposed architecture for edge computing using ICN in 5G networks. This architecture supported local D2D communication. The base stations and 5G Radio Access Network (RAN) are supported by ICN. In addition to that, the researchers also proposed a content pre-fetching strategy based on content popularity. Using adequate results researchers have shown that this architecture provides a considerable reduction in response latency [19].

Using ICN visualization with MEC, the computing and caching facilities are easily accessible to the end-users. Most of the researchers have worked on ICN and MEC separately. In this paper, we have presented these two approaches under one roof to optimize the caching, communication, and computing facilities.

The main contribution in this paper is as follows:

- To enable efficient content caching and computing we propose an ICN architecture having the MEC facility. This architecture has all the inherent features of ICN like caching, communication, and computing thereby improving the overall Quality of Service (QoS).
- We propose a model for binary task offloading for mobile devices to the MEC server. This model empowers the mobile nodes to take task offloading decisions.

III. OVERVIEW OF NAMED DATA NETWORKING (NDN)

The propounded model is implemented on the Named Data Networking (NDN) architecture. NDN is one of the prominent ICN architectures. In NDN the receiver initiates the communication i.e. the data consumers, by the interchange of Interest packet and data packet [20]. The data packet information is identified by a unique name. Therefore, the consumer needs to assign a name to the Interest packet. Thereafter, this packet is sent towards the data source. This unique name identifies the data source in the network. Forwarding mechanism drives the Interest packet towards the data source. Once this Interest packet reaches the data source, the Data packet containing the required information is reverted back to the consumer. This data packet consists of the content name, data and the signature of producer’s key for authentication. Three data structures namely, the Pending Interest Table (PIT), the Forwarding Information Base (FIB), and a Content Store (CS) are maintained by the NDN node. These data structures facilitate the forwarding of Interest and Data packets. Figure 1 depicts the same. Interest packets are forwarded using a forwarding strategy. All the unsatisfied Interests forwarded by a node is recorded in the PIT table. The PIT records the incoming and outgoing interface of these Interest packets.

When an Interest packet arrives an NDN node, NDN initially checks for the matching content in the content store of the node. If matching data exists, data packet is sent back to the requester. Otherwise, the node checks its PIT entry for a match, and if a match exists, it adds the record of the incoming face of the Interest packet in PIT. On the other hand, if no entry is there in the PIT, the router forwards the Interest towards the data producer using the FIB and forwarding policy of the node. The node only forwards the earliest received Interest packet towards the data producer, for the rest of the Interests an entry is added in the PIT table.

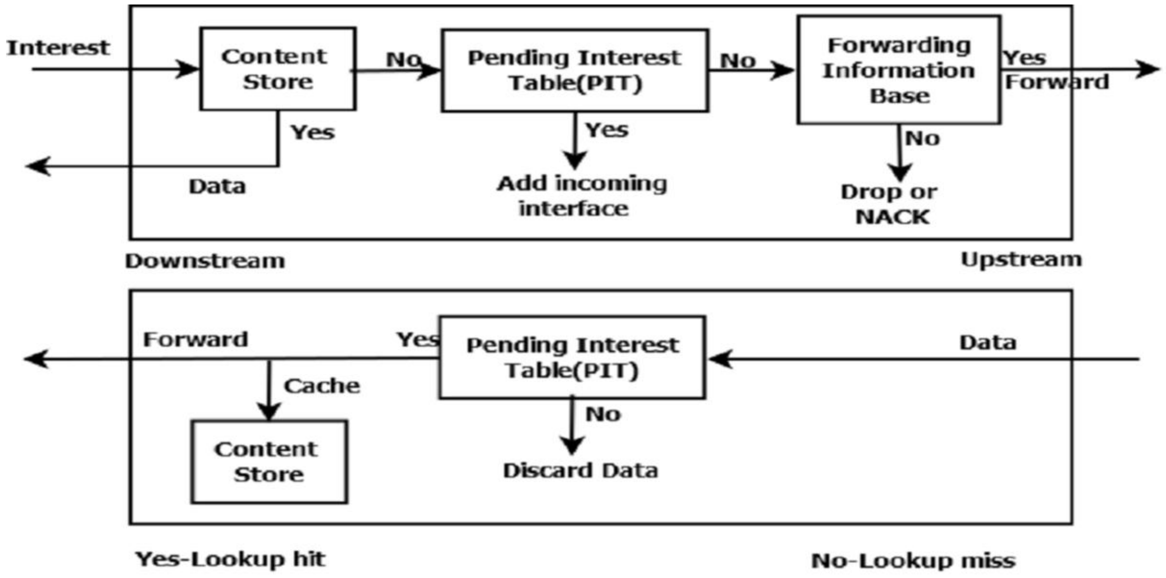

Fig. 1. The Named data networking architecture (adapted from [20])

A routing protocol fills up the FIB entries based on name prefix. However, in exceptional circumstances, the forwarding strategy drops an Interest packet if congestion exists in all upstream nodes or the Interest is suspected to be malicious. To fulfill the forwarding mechanism of the Interest packet, the longest prefix matched entry is checked in the FIB and accordingly, it takes the forwarding decision for the Interest packet. Each node has a temporary cache called the content store for storing the data packets. As the data packet arrives at the NDN router, the NDN router looks for the matching PIT entry, and thereafter the data is sent to all downstream faces recorded in that PIT entry. In the last step, the PIT entry is eliminated, and data is stored in the content store.

IV. EDGE COMPUTING OVER NDN ARCHITECTURE

Figure 2 shows an ICN network using named data networking (NDN) architecture. For implementation and validation, we use the NDN architecture. The edge computing facility is available at some of the nodes. Mobile devices can offload expensive tasks to the MEC servers for remote processing. User 1 requests the service /LM/s1. There are five MEC servers out of which the two MEC servers i.e MEC1 and MEC2 provide the service requested by the user1. The MEC servers located at the network edges need to disperse their status information. Named-data link-state routing (NLSR) is used to exchange these messages [21].

The researchers propose three approaches to address the issue of resource discovery, namely, the proactive, reactive, and the passive technique [21]. In the first approach i.e., the proactive approach edge nodes perform periodic resource advertisement. To accomplish this task the edge nodes, use a routing protocol. The time period of the advertisement must be set cautiously as a larger time period leads to sharing of stale content and a shorter time period increases the overhead on the network. In the second approach i.e., the reactive one, the edge nodes reply back their status information to the nodes requesting services from the edge node. Thus, in this approach, the edge nodes share the status information with only the nodes that send a service request. However, this approach is only suitable for tasks that are requested less frequently as this approach generates more overhead on the network. In the third approach i.e., the passive approach, the edge node waits for the task. Whenever it receives a request to perform the task, the edge node checks whether it is feasible for the edge node to accomplish the task. If the answer is yes, the edge node accepts the task, otherwise, it sends a negative acknowledgment (NACK) to the sender in the reverse direction. Thus, all the on-path routers stop sending additional requests to this node. Hence, these tasks are routed to the different edge nodes. In wireless networks having a MEC facility, mobility management is another challenging issue. In [22] researchers propose a solution for the mobility management problem.

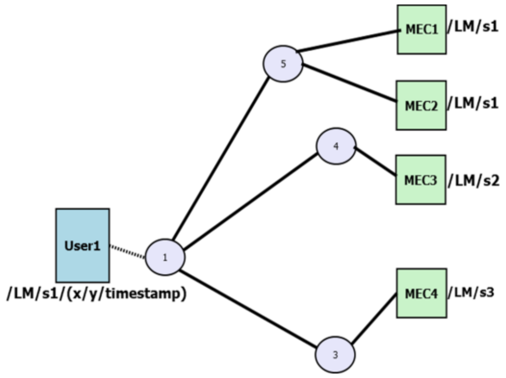

Fig. 2 Mobile edge computing over Information- Centric Networking

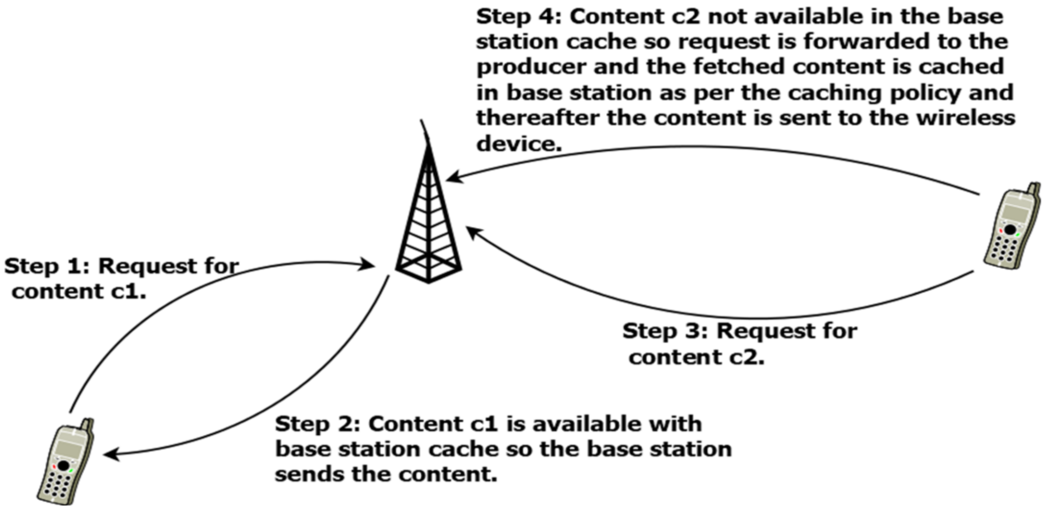

Fig. 3. The caching mechanism

In this paper, we use NDN to deploy ICN. We focus on efficient resource allocation strategies for edge caching. Thus, caching servers are in place to cache the information nearer to the end-devices. The MEC servers take care of the caching and at the network edges. Thus, the MEC servers also take care of the processing overhead of the concerned cache(s). Task resource allocation to MEC servers in a dynamic environment is a challenging problem. Hence, to address this problem, we have done a comprehensive study on the challenges faced by the MEC server.

The energy-constrained mobile device has limited computational capability. Thus, it is a wise decision to offload the computationally intensive tasks to the MEC server. This can save the power of the end-user device. Additionally, the computationally intensive tasks can be completed within a deadline by the powerful MEC server. This motivated us to carry forward our task in this direction, and come up with a mathematical model that allows the mobile device to take task offloading decisions.

V. SYSTEM MODEL

In our model, there are α base stations. We represent this as set BS = {b1, b2, b3, …, bα}. We further assume that there are β wireless devices. We represent this as set W = {w1, w2, w3, …, wβ}. As shown in figure 4, the wireless devices connect to the base stations for communication. These base stations are present at the network edge. Each base station has caching facility. A MEC is shared by one or more base stations. The wireless devices offload their task to the MEC server associated with the base station. For modeling, we consider the binary task offloading process to the MEC server. We consider that the user location status of the wireless devices is maintained by the network controller, and it also keeps track of the user movements. Whenever, a wireless device makes a content request, the caching facility at the base station caches the content as per the caching policy. However, duplicate contents are not cached. Whenever, the user device changes location, its connectivity with one base station ends and it reconnects to a new base station. In this scenario, the last base station may fail to cache the content, so the adjacent base station with which the user reconnects performs the content caching. For our system model, we consider the scenario shown in figure 4. As shown in figure 4 the wireless devices are connected to the ICN network via base stations. These base stations are connected to both the wireless devices and the core network. A MEC server and a cache is associated with each base station. This infrastructure provides low latency multimedia services.

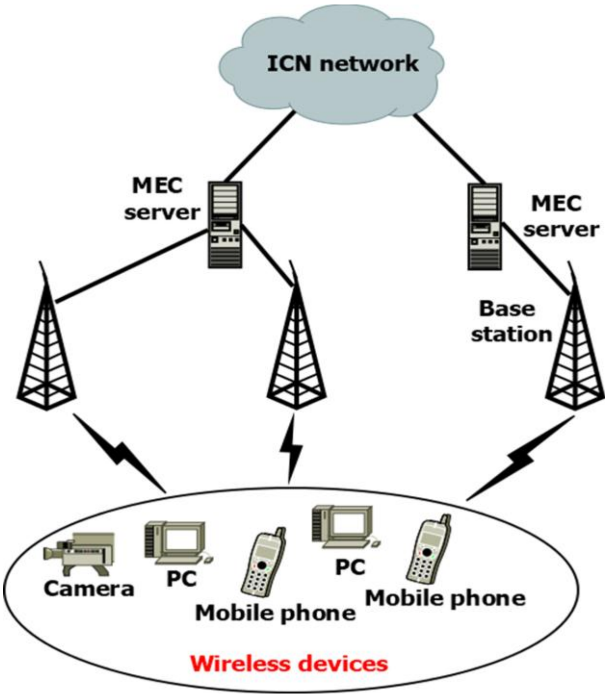

Fig. 4. The wireless devices are connected to the ICN network via base stations. These base stations are located at the edge of the network. A MEC server is associated with one or more base stations

The wireless devices offload the tasks having high computational overload to the MEC server of the nearest base station. The MEC server computes the task and responds to the wireless device. However, if the application needs support that is not available at the MEC server, the request is forwarded to the core network [10]. The resource blocks that the base station cache is shared among the other bases stations connected to the MEC server. The caching process is shown in figure 3A.

A. *The Binary offloading process:*

The system executes the integrated tasks as a single unit as their partition is not possible. Thus, offloading these tasks to the MEC server is called binary offloading. A three-field notation T(Ts, Cd, Cw) is used to represent these tasks. Where Ts represents task input-data size in bits, Cd represents the completion deadline of the task in seconds and Cw represents computation workload required for the task in bits per second. These three parameters are co-related to the type of application. Hence, they are estimated from the task profiles [23] [24]. These three parameters capture the important properties like computation and communication demands of the mobile application. Additionally, these parameters also facilitate the evaluation of execution latency and energy consumption of the task.

B. *Computation performance of mobile devices:*

In this section, we discuss the techniques to evaluate the computation performance of mobile devices. For local computation CPU available with the mobile device is the primary tool. The performance of the CPU is controlled by the CPU clock speed Ω. In some of the literature, CPU clock speed is also referred to as CPU-cycle frequency. The mobile CPU architecture allows stepping up or stepping down the CPU clock speed using the dynamic frequency and voltage scaling (DVFS) technique [25]. Stepping-up the CPU clock speed leads to more energy consumption and on the flip side-stepping down the CPU clock speed leads to less energy consumption. However, the maximum clock speed of the CPU is bounded by the threshold value Ωmax. Thus we calculate the latency to execute the task T(Ts, Cd, Cw) as,

$$£ = (Ts * Cw)/\Omega \qquad (1)$$

Equation (1) indicates that CPU clock speed directly effects the task execution latency. Hence for faster task execution higher clock speed and more energy are required. As the mobile devices have limited energy, the mobile device may offload some of these tasks to the MEC server. The energy utilization of one CPU clock cycle is given by $K\Omega^2$, where K is the constant for the hardware architecture [25]. Therefore, for the task T(Ts, Cd, Cw) with CPU clock speed Ω the energy utilization is given by equation (2),

$$Energy_{mob} = K * \Omega^2 * Ts * Cw \qquad (2)$$

From equations (1) and (2) the mobile device can decide whether it is feasible for the device to complete the task within the deadline or it should offload the task to the nearest MEC server.

VI. RESULTS AND DISCUSSIONS

A. *Task offloading decision*

We consider a computationally intensive task of X264 VBR encoding. This task requires 1300 cycles/second on ARM Cortex A8 CPU [23]. The ARM Cortex A8 CPU have a clock speed of 600 MHz-1GHz. The file size is 200 MB. And the task completion deadline is 15 seconds. So, for this task T, Ts = 200 $*10^6$ bytes, Cd = 100 seconds, Cw = 1300 cycles/byte. Considering the current clock speed of CPU as 800 $* 10^6$ Hz. From equation (1) the latency to execute the task is $(200 *10^6 * 1300)/(800 * 10^6)$ = 325 seconds. This task consumes more time than the allocated deadline, thus this task can be offloaded. The maximum achievable clock speed is 1 GHz. However, higher clock speed increases energy consumption of mobile device. In this scenario, even the maximum clock speed cannot complete the task within the deadline.

B. *Impact of zipf distribution parameter on in-network caching*

TABLE I

I	**Cache hit % a=1.3**	**Cache hit % a=1.4**	**Cache hit % a=1.5**	**Cache hit % a=1.6**	**Cache hit % a=1.7**	**Cache hit % a=1.8**
500	56.2	67.4	74.6	82	87.2	89
1000	65.49	68.9	76.8	83.39	87.9	91.7
1500	53.4	68.26	74.93	85.66	91.06	92.4
2000	59.5	70.7	78.9	84.45	89.25	91.25
2500	56.59	73.56	77.32	83.39	88.6	91.3
3000	60.13	68.72	77.16	85.92	87.96	92.47

Table showing the in-network cache hit percent with change in value of zipf distribution parameter a. I show the number of interests. The in-network cache capacity is 20 contents. And there is total 100 contents available in the network. All these contents have fixed size.

TABLE I shows the relationship between zipf distribution parameter a and the cache hit percentage. Least frequently used (LFU) is the cache eviction policy. We have written a python code for implementation. The zipf distribution parameter controls the content popularity skewness. Thus, higher values of parameter a increases the hit rate as most of the popular contents can be accommodated in the cache. Zipf distribution is often used to represent the content popularity of Internet contents. The value of parameter a should be greater than 1. We have not taken value of a more than 1.8 as it is not applicable for practical applications. More hit-ratio means less utilization of backhaul for transmission. This decreases the response latency and energy requirement for transmission as the content is available mostly at the cache available at the network edge.

C. Transmission delay for communication via base station

TABLE II

No of chunks	Transmission delay (seconds)
60	6.815
120	12.801
180	18.880
240	24.854
300	30.881
360	36.893
420	42.855

Table showing transmission delay between two end devices via base station with change in content size. Each chunk is of 1024 bytes.

TABLE II shows the transmission delay between end devices via base station. These devices are in the range of same base station. We simulated our results in ns-3 and ndnSIM 2.0.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed the architecture for NDN using MEC. This architecture provides inherent support for mobility and task offloading facility for energy-constrained mobile devices. Furthermore, we proposed a mathematical model for binary task offloading by these mobile devices. This model focus on energy consumption and CPU utilization for the task. This enables the mobile device to decide whether the task can be executed locally or the task needs to be offloaded to the MEC server for execution. In the future, combined optimization of energy, CPU utilization, and the in-network caching facility can be done. Additionally, researchers can work on the revenue generation model for this scheme.

REFERENCES

[1] J. Kim and A. Molisch, "Fast millimeter-wave beam training with receive," Journal of Communications and Networks, vol. 16, no. 5, p. 512–522, 2014.

[2] D. Kwon, S. Kim, J. Kim, and A. Mohaisen, "Interference-Aware Adaptive Beam Alignment for Hyper-Dense IEEE 802.11ax Internet-of-Things Networks," Sensors, vol. 18, no. 10, 2018.

[3] J. Kim, G. Caire and A. F. Molisch, "Quality-aware streaming and scheduling for device-to-device video delivery," IEEE/ACM Transactions, vol. 24, no. 4, p. 2319–2331, 2015.

[4] S. Banerjee, T. Naskar and S. Biswash, "The Survey, Research Challenges, and Opportunities in ICN," in Cloud Network Management - An IoT Based Framework, Chapman and Hall/CRC, 2021.

[5] L. Wu, T. Zhang, X. Xu, Z. Zeng and Y. Liu, "Grey relational analysis based cross-layer caching for content centric networking," in IEEE/CIC International Conference on Communications in China, 2015.

[6] W. Quan, C. Xu, J. Guan, H. Zhang and L. A. Grieco, "Social cooperation for information-centric multimedia streaming in highway vanets," in Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2014.

[7] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge computing: Vision and challenges," IEEE internet of things journal, vol. 3, no. 5, pp. 637-646, 2016.

[8] C. Liang, F. R. Yu and X. Zhang, "Information-centric network function virtualization over 5G mobile wireless networks," IEEE Network, vol. 29, no. 3, pp. 68-74, 2015.

[9] N. Dao, D. Vu, W. Na, J. Kim and S. Cho, "SGCO: Stabilized green crosshaul orchestration for dense IoT offloading services," IEEE Journal on Selected Areas in Communications, vol. 36, no. 11, p. 2538–2548, 2018.

[10] D. Liu, B. Chen, C. Yang and A. F. Molisch, "Caching at the wireless edge: design aspects, challenges, and future directions," IEEE Communications Magazine, vol. 54, no. 9, pp. 22-28, 2016.

[11] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," IEEE Access, vol. 5, p. 6757–6779, 2017.

[12] R. Ullah, S. H. Ahmed and B. Kim, "Information-centric networking with edge computing for IoT: Research challenges and future directions," IEEE Access, vol. 6, p. 73465–73488, 2018.

[13] O. Salman, I. Elhajj, A. Kayssi and A. Chehab, "Edge computing enabling the internet of things," in 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), 2015.

[14] S. Dustdar and W. Shi, "The promise of edge computing," Computer, vol. 49, no. 5, p. 78–81, 2016.

[15] O. Makinen, "Streaming at the edge: Local service concepts utilizing mobile edge computing," in 9th International Conference on Next Generation Mobile Applications, Services and Technologies, 2015.

[16] E. Cau, M. Corici, P. Bellavista, L. Foschini, G. Carella, A. Edmonds and T. M. Bohnert, "Efficient exploitation of mobile edge computing for virtualized 5G in EPC architectures," in 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, 2016.

[17] R. Huo, R. Xie, H. Zhang, T. Huang and Y. Liu, "What to cache: differentiated caching resource allocation and management in Information-centric Networking," China Communications, vol. 13, no. 12, pp. 261-276, 2016.

[18] D. Grewe, M. Wagner, M. Arumaithurai, I. Psaras and D. Kutscher, "Information-centric mobile edge computing for connected vehicle environments: Challenges and research directions," in Proceedings of the Workshop on Mobile Edge Communications, 2017.

[19] R. Ullah, M. A. U. Rehman, M. A. Naeem, B.-S. Kim and S. Mastorakis, "ICN with edge for 5G: Exploiting in-network caching in ICN based edge computing for 5G networks," in Future Generation Computer Systems, 2020.

[20] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang and B. Zhang, "Named data networking," SIGCOMM Comput. Commun. Rev., vol. 44, no. 3, p. 66–73, 2014.

[21] A. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang and L. Wang, "NLSR: named-data link state routing protocol," in 3rd ACM SIGCOMM workshop on Information-centric Networking, 2013.

[22] A. Mtibaa, R. Tourani, S. Misra, J. Burke and L. Zhang, "Towards Edge Computing Over Named Data Networking," 2018 IEEE International Conference on Edge Computing (EDGE), pp. 117-120, 2018.

[23] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in USENIX Conf. Hot Topics Cloud Comput. (HotCloud), Boston, MA, USA, 2010.

[24] S. McGarry and M. P. Melendez, "Computation offloading decisions for reducing completion time," in Proc. IEEE Annu. Consum. Commun. Netw. Conf. (CNCC) , Jan. 2017, Las Vegas, NV, USA, 2017.

[25] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," IEEE COMMUNICATIONS SURVEYS & TUTORIALS, vol. 19, no. 4, 2017.

AUTHOR PROFILE

Mr. Supratik Banerjee is currently serving as Assistant Professor in Computer Science & Engineering Department in NIIT University, Neemrana, Rajasthan. He received his B. Tech degree from SMIT, Sikkim in Computer Science and Engineering and M. Tech degree from ABV-IIITM, Gwalior with specialization in Advanced Networks. His research area includes Information Centric Networking, 5G and vehicular networks.