Journal of The Gujarat Research Society

ISSN: 0374-8588
Volume 22 Issue 1, January 2020

Delay-Optimized Offloading for Mobile Cloud Computing Services in Heterogeneous Networks

Mr. Anuj Kumar, Dr. Niraj Singhal

Shobhit Institute of Engineering and Technology (Deemed to be University), Meerut

Email Id- anuj.k@shobhituniversity.ac.in, niraj@shobhituniversity.ac.in

ABSTRACT: ***By executing remotely on mobile cloud computing (MCC) networks, offloading is an effective technique for extending the lifespan and speeding up the execution rate of mobile devices. Meanwhile, a heterogeneous network (HetNet), which includes a variety of radio access nodes, is generally seen as a viable method to meet the increasing traffic demand. First, we present two delay-optimized offloading control methods for LTE-Advanced heterogeneous networks in this article. Setting a threshold to limit the number of unloading users is one of them. Another method is to estimate the execution latency to see whether the users are suitable for offloading. Both take into consideration the traffic load of the serving cell and neighboring cells. Simulations in both heterogeneous and macro-only networks are used to assess the delay performance of various methods for comparative purposes.***

KEYWORDS: ***Heterogeneous Network (HetNet), Mobile Devices (MDs), Mobile cloud computing (MCC), Offloading.***

1. INTRODUCTION

Mobile cloud computing (MCC) has emerged in recent years as a means of integrating cloud computing into the mobile environment in order to overcome the constraints of mobile devices (MDs). However, since it combines two distinct disciplines, it faces many technological difficulties. Computing offloading method is one of the difficulties. Large calculations and sophisticated processing from resource-constrained MDs are suggested to be moved to distant servers with the goal of prolonging battery life and increasing operation rate without needing structural or hardware modifications. Experiments in demonstrate that offloading isn't always the most efficient method to boost performance. © Institute of Computer Sciences, Social Informatics, and Telecommunications Engineering MCC Services Delay-Optimized Offloading in 2014 Heterogeneous[1]Networks are a kind of network that is made up of different 123As a result, offloading the application to distant servers is becoming more essential. Offloading methods are divided into two categories: complete offloading and partial offloading. The term "full offloading" refers to the transfer of all mobile application computing duties from the MD to a distant cloud. Different aspects of an application's computing duties, on the other hand, may make them more or less appropriate for offloading. For further energy savings, the application may offload just the sub-parts that benefit from remote execution. As a result, partial offloading, in which the program is divided into unoffloadable tasks and offloadable portions, has gotten increasing attention. proposes a basic but fundamental approach for determining if compute offloading can save energy. It has been shown that offloading image processing to a distant server lowers energy usage by 41%.

The system is represented in a Markovian dynamic control framework [6], and the energy vs delay trade-off is investigated. [reduces power consumption by addressing two restricted optimization problems: determining the optimum clock frequency of the local processor and the data transfer rate o[2]f each time slot for cloud execution within a certain time delay. The choice to unload is then made with less energy usage in mind. proposes a dynamic offloading method based on Lyapunov optimization for making offloading decisions for all computing jobs while meeting the application execution time requirement. Although previous work has improved the basic model in various ways, there are still some unsolved issues, particularly when offloading data is transmitted over a heterogeneous network (Hetnet), in which base stations with lower transmit power) are deployed alongside macro eNodeB with high transmit power.

An application's offloading choice is based on an estimate of execution time or energy usage. However, practically no research to date has taken into account the traffic load of MD's serving cell and adjacent cells, which may influence the offloading choice. The amount of traffic in an MD's serving cell has an effect on wireless channel gains at MD scheduling instants. In the meanwhile, increasing traffic loads in adjacent cells may cause severe interference. Both of these factors may have an impact on power usage and execution time. As a result, we offer delay-optimized offloading management methods[3] in the Hetnet environment that take traffic load into account and decrease the average delay of all mobile users. The suggested scheme's efficacy in terms of average delay is shown by simulation results. This article looks at an LTE-Advanced heterogeneous network using microcells and microcells. Each sector of the microcell is equipped with a microcell eNodeB.

In a single microcell, there are M sectors. For mobile applications, a dynamic flow model with elastic traffic is considered, in which a fresh flow enters the network with a finite-length file request and[4] exits when the file is delivered. The arrival rate of user n at network cell m follows a Poisson distribution with an average arrival rate of (m) n. In cell m, 0 m M, there are Nm active users serviced by the eNodeB. Cell 0 symbolizes the microcell, while the others represent microcells. Computation off loading's primary goals are to reduce MD power usage and speed up application execution. However, depending on the details of the computing job, server load, and network connection, the aforementioned two objectives may not always be met. Offloading is more likely to help an activity with high computing and low communication needs than a job with low computation and high communication requirements. As a result, it's important to make an informed choice about whether or not to offload a computing job. A mobile application is supposed to be run either locally on the mobile device or remotely on distant cloud servers via complete offloading for the sake of simplicity. The offloading decision is described by the binary vector B(m) = b (m) n |b (m) n 0 |b (m) n 1Nm, where b (m) n = 1 indicates that the application of user n in cell m is performed locally, otherwise b (m) n = 0.

The entire execution time for an application in an offloading scenario is made up of the time spent sending the task and data to the cloud servers, waiting for the cloud to do the job, and getting the task result. However, since cloud servers have a high computational capacity, the delay induced by wireless transmission between the MD and cloud servers, particularly in the uplink, may account for the majority of the overall execution time. The data transfer rate between the mobile device and the cloud servers has a big effect on offloading choices in this instance. A two-dimensional (2-D) time-frequency grid that corresponds to a collection of

OFDM symbols and subcarriers in the time and frequency domains is the fundamental radio resource unit for OFDM transmission. The fundamental unit for data transmission in LTE-Advanced networks is a pair of resource blocks (RBs) that correspond to a 180-kHz bandwidth during a 1 ms subframe. All radio resources, i.e. K RBs, are considered to be completely reused across picocells and macrocells in this work.[5]

2. DISCUSSION

Simulated results are given in this part to assess the delay performance of the suggested offloading control methods in LTE-Advanced heterogeneous networks (HetNet). Only one microcell and three picocells are examined here. The network's performance with just macro-cells is also provided for comparison. Summarizes the detailed simulation parameters, including the channel model and system assumptions. The performance of the networks in which the applications may be offloaded according to the provided probability, i.e., threshold, is first supplied in order to establish the threshold value for offloading choice. The offloading ratio may be set anywhere between 0% and 100%. Different scheduling methods are used in these simulations for various situations. The average is shown in 1st Table In LTE-Advanced wireless networks, parameters are assumed. Parameters Values Transporter (GHz) Bandwidth 2 (MHz) 10[6].

Duration of time slot (ms) 180 Resource block spacing (kHz) The quantity of resource blocks 50VA channel model a speed of 3 km/hRate of arrival 2 sub-frames/applications File size after offloading (Kbytes) Target SNR: 10 (dB) 10eNodeB's transmit power (dBm) UE power class (Pmax): 46 in Macro/30 in Pico (dBm) 23Configuration of the antenna Tx Rx= 1 1 Tx Rx= 1 1 Tx Rx= 1 1 Tx Rx= 1 1 128.1+37.6log10(R) pathloss model in Macro, R in kmPico pathloss model 140.7+36.7log10(R), R in kilometers, Max C/I Scheduling Algorithm Controlling the power Full pathloss compensation in an open loopHeterogenous Networks with Delay-Optimized Offloading for MCC Services 129Offloading users (a) (b) the whole user base.

Users whose apps have been chosen to be offloaded will experience a delay. As can be observed, the delay increases as the offloading ratio increases. The radio resource that may be allocated to each user is reduced as the number of applications that need to be offloaded grows. As a result, consumers' attainable data rates decrease, causing transmission delays to rise. The networks using the Max C/I algorithm perform better than those with the Round Rubin (RR) algorithm because to the scheduling advantage. Furthermore, Hetnet has a significantly shorter latency than a macro-only network. It's because radio resources are shared across macro- and pico-cells, resulting in a higher number of RBs available to consumers for offloading data transmission. Under all simulated situations, it can be shown that the average latency increases with the offloading ratio and is the convex function of.

According to the study, there is an optimum offloading ratio that may be utilized as the offloading decision threshold. The average delay performance of all users whose application is offloaded is then presented where the local execution time is fixed, i.e., L = 1ms. When anticipated, as the offloading ratio increases, the average latency of all users decreases at first, then increases. The optimum offloading ratio is largely dependent on the scheduling algorithm and network conditions, it should be mentioned. In HetNet situations, for example, the optimum offloading ratio of 40% may be used as a threshold for the Max C/I scheduling method, whereas 30% can be used for the RR scheduling algorithm. As a result, we must use

simulations to determine the various threshold values for the threshold-based offloading control method. In we show the network's average delay performance with our two suggested offloading control methods, assuming various local execution times. With a higher value for local execution time, more apps are likely to offload, resulting in more radio resource competition in the network. Then, owing to the lower data rate that each user can achieve, the delay grows. Despite its simplicity, the threshold-based method does not always make the optimal choice.

In comparison to the threshold-based method,Regardless of whether scheduling method is used, the network's delay performance with the rate-prediction strategy is superior. Because the rate-prediction technique takes into consideration not just the channel status information but also the total number of users, this is the case. Users may access application software and databases via the software as a service (SaaS) paradigm. The infrastructure and platforms that operate the apps are managed by cloud providers.

SaaS is also known as "on-demand software," and it is often paid on a pay-per-use or subscription basis. Cloud providers install and run application software in the cloud, while cloud customers access the software via cloud clients under the SaaS model. The cloud infrastructure and platform over which the application operates are not managed by cloud users. This eliminates the need for the cloud user to install and operate the program on their own machines, making maintenance and support easier. Scalability is a feature of cloud computing that may be accomplished by cloning jobs onto numerous virtual machines at runtime to accommodate changing work demand. Load balancers spread the workload over a group of virtual computers.

The cloud user sees just one access point; thus, this procedure is clear to them. Cloud applications may be multitenant to support a high number of cloud users, meaning that a single server can service several cloud-user organizations. Because SaaS apps usually charge a fixed cost per user on a monthly or annual basis,[7] rates become scalable and changeable when users are added or withdrawn at any time. It's also possible that it'll be free. Proponents argue that by outsourcing hardware and software maintenance and support to the cloud provider, a company could decrease IT operating expenses. This allows the company to reallocate IT operations expenditures away from hardware/software and human costs and towards another objectives. Furthermore, since programs are hosted centrally, updates may be sent without requiring users to install new software. One disadvantage of SaaS is also that customers' data is stored on the cloud provider's server. As a consequence, there may be illegal access to the data.

Games and efficiency tools like Google Docs and Word Online are examples of SaaS applications. SaaS applications may be linked to cloud storage or file hosting services, as Google Docs is with Google Drive and Word Online with One Drive. Web app and mobile app developers are given a way to link their applications to cloud storage and cloud computing services with application programming interfaces (APIs) exposed to their applications and custom software development kits in the mobile "backend" as a service (m) model, also known as backend as a service (BaaS) (SDKs). User administration, push alerts, integration with social networking sites, and other services are available[8]. Most BaaS companies are from 2011 or later making this a relatively new cloud computing paradigm.

However, current trends show that these services are acquiring considerable mainstream momentum among corporate customers.

Serverless computing, also known as Function-as-a-Service, is a kind of computing that does not need a server (FaaSArticle in its entirety: Computing without a server- Serverless computing is a cloud computing code execution model in which the cloud provider manages the start and stop of virtual machines as needed to serve requests, and requests are billed by an abstract measure of the resources required to satisfy the request rather than per virtual machine, per hour. Despite the name, it does not include the execution of programs without the need of servers. The term "serverless computing" comes from the fact that the owner of the system does not have to buy, rent, or supply servers or virtual machines for the back-end code to operate on.Function as a service (FaaS) is a serverless computing-based remote procedure call that enables the deployment of specific functions in the cloud that execute in response to events. Some people consider FaaS to be a subset of serverless computing, while others use the words interchangeably.

Private cloud refers to cloud infrastructure that is administered exclusively for the benefit of a single company, whether it is managed internally or by a third party, and is hosted domestically or outside. A private cloud project requires substantial involvement in order to virtualized the business environment, as well as a reevaluation of current resource choices. It may boost profits, but every stage of the process presents security concerns that must be addressed to avoid severe flaws. Self-contained data centers are often expensive to operate. They have a large physical footprint, necessitating space, hardware, and environmental controls allocations. These assets must be renewed on a regular basis, which necessitates extra capital expenditures. They've been chastised since customers "still have to purchase, construct, and maintain them" and so don't profit from less hands-on management, basically "[lacking] the economic paradigm that makes cloud computing such an attractive idea." Cloud computing in the public domainSee Cloud-computing comparison for a comparison of cloud-computing software and services.

When cloud services are provided via the public Internet, they are called "public," and they may be available for a fee or for free. There are minimal architectural differences between public and private cloud services, but when services (applications, storage, and other resources) are shared by many users, security issues rise dramatically. The majority of public cloud providers provide direct-connection services, which enable clients to securely connect their traditional data centers to cloud-based applications. Several variables influence whether businesses and organizations select a public cloud or on-premises solution, including solution functionality, pricing, integrational and organizational elements, as well as safety and security.

The term "hybrid cloud" refers to a combination of a public cloud and a private environment, such as a private cloud or on-premises resources, that stay separate but are linked to provide the advantages of various deployment methods. The capacity to link collocation, managed, and/or dedicated services with cloud resources is referred to as hybrid cloud. A hybrid cloud service, according to Gartner, is a cloud computing solution that combines private, public, and community cloud services from several service providers. A hybrid cloud service spans isolation and provider borders, making it impossible to categorize it as either private, public,

or community cloud. It enables you to increase a cloud service's capacity or capabilities by aggregating, integrating, or customizing it with another cloud service.

Hybrid cloud composition has a wide range of applications. For example, a company may keep confidential client data on a private cloud application but link it to a business intelligence application offered as a software service on the public cloud. The integration of externally accessible public cloud services to the enterprise's capabilities to provide a particular business service is an example of hybrid cloud. Hybrid cloud adoption is influenced by a variety of variables, including data security and compliance needs, the degree of data management required, and the apps used by the company.

Another hybrid cloud scenario is when IT companies utilize public cloud computing resources to fulfill temporary capacity requirements that the private cloud cannot provide.

With this feature, hybrid clouds may use cloud bursting to scale across clouds.

Cloud bursting is a software deployment strategy in which an application operates in a private cloud or data center and then "bursts" to a public cloud when demand for computing power grows. One of the most significant benefits of cloud bursting and a hybrid cloud architecture is that a company only pays for more computing resources when they are required. Cloud bursting allows data centers to build an in-house IT architecture that can handle normal workloads while also using cloud resources from public or private clouds when processing needs rise. "Cross-platform Hybrid Cloud" is a customized hybrid cloud concept that is based on heterogeneous hardware. Underneath, a cross-platform hybrid cloud is often driven by various CPU architectures, such as x86-64 and ARM. Users may deploy and grow apps in the cloud without being aware of the hardware variety. The development of ARM-based system-on-chip for server-class computing has given birth to this kind of cloud Hybrid cloud architecture basically helps to overcome the constraints of private cloud networking's multi-access relay features. The benefits of virtualized interface models include more runtime flexibility and adaptive memory processing. Cloud computing in the community clouds pool infrastructure across many companies from a same community that share similar concerns (security, compliance, jurisdiction, and so on), whether managed internally or by a third party, and whether hosted internally or externally. Only a portion of the cost savings potential of cloud computing is achieved since the expenses are distributed across fewer users than a public cloud (but more than a private cloud). Cloud that is dispersA cloud computing platform may be made up of a dispersed group of computers in various places that are all linked to a single network or hub service. Two kinds of distributed clouds may be distinguished: public-resource computing and volunteer cloud. This kind of distribution is known as public-resource computing[9] [10].

3. CONCLUSION:

Due to the constraints of mobile devices, energy savings and execution speed improvements have gotten a lot of attention. We suggest two offloading control methods in this paper to satisfy the latency needs of mobile cloud applications. The suggested techniques' performance with common scheduling algorithms is investigated in a variety of situations. The threshold-based approach, which is simple but less flexible, can effectively decrease

execution time. The rate-prediction method may produce superior delay performances and can be readily adapted to different situations since it takes into consideration not only the network environment but also the channel characteristics.

REFERENCES:

[1] B. de Bruin and L. Floridi, "The Ethics of Cloud Computing," *Sci. Eng. Ethics*, 2017.

[2] J. Lee, "A view of cloud computing," *Int. J. Networked Distrib. Comput.*, 2013.

[3] V. Pushpalatha, K. B. Sudeepa, and H. N. Mahendra, "A survey on security issues in cloud computing," *Int. J. Eng. Technol.*, 2018.

[4] K. Akherfi, M. Gerndt, and H. Harroud, "Mobile cloud computing for computation offloading: Issues and challenges," *Applied Computing and Informatics*. 2018.

[5] S. Shilpashree, R. R. Patil, and C. Parvathi, "'Cloud computing an overview,'" *Int. J. Eng. Technol.*, 2018.

[6] Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, and P. Merle, "Elasticity in Cloud Computing: State of the Art and Research Challenges," *IEEE Trans. Serv. Comput.*, 2018.

[7] T. H. Noor, S. Zeadally, A. Alfazi, and Q. Z. Sheng, "Mobile cloud computing: Challenges and future research directions," *J. Netw. Comput. Appl.*, 2018.

[8] E. Mishra and A. Bhatnagar, "A survey on cloud computing," *Int. J. Innov. Technol. Explor. Eng.*, 2018.

[9] J. Angelin Jebamalar and A. Sasi Kumar, "A review on the integration of cloud computing and internet of things," *Int. J. Eng. Technol.*, 2018.

[10] S. C. Misra and A. Mondal, "Identification of a company's suitability for the adoption of cloud computing and modelling its corresponding Return on Investment," *Math. Comput. Model.*, 2011.