Needs and Challenges of Edge Computing in Software Engineering for the Internet of Things (IoT)

Elizabeth High, Sayeed Sajal
Computer Science
Minot State University
500 University Ave W, Minot, ND, 58707
Sayeed.Sajal@minotstateu.edu

Abstract

With billions of Internet of Things (IoT) connected to the network and an increase in data usage and connectivity, the conventional cloud computing architecture is insufficient to handle the future demands while maintaining low latency and the conservation of bandwidth. The edge computing paradigm will help to provide a more seamless network closer to the user. While this is a promising solution, there are still challenges to overcome during the development, deployment, and transition to this new system. In this work, we will discuss the reasons for this inevitable need for an improved network architecture and also challenges for developing and implementing these paradigms.

1. Introduction

The development of cloud computing changed the world by providing faster than ever internet speeds; while still an incredibly important component of the internet network architecture, emerging technologies, and IoT devices demand more than the cloud can deliver alone. A solution to this problem has been in development with the introduction of edge computing, a revolutionary idea. Edge computing will increase the conservation of bandwidth and provide a real-time response, spectral efficiency, and ultra-low latency. With all of these improved features of the network, problems in other important aspects of society will be increased. Time-sensitive applications will only be sent long distances to the central cloud if the edge cloud nodes are unable to process the information. In addition, not all information collected at the edge of the network will be sent to the cloud either; for instance, it is a waste of resources and not necessary to send all video surveillance to the cloud. Another incredible feature of the edge is the ability for it to adapt quickly to local issues. A cloud center far away does not have the ability to adapt in real time to frequent changes in a local network.

Different edge computing infrastructures have been proposed and are in development to bring the internet closer to the user. The idea of mobile edge computing (MEC) was proposed initially by IBM and Nokia Siemens, and fog computing was presented by Cisco in 2012 [2]. These infrastructures will both provide ultra-low latency, decreased energy consumption, and conservation of bandwidth which will result in increased network availability by bringing the network closer to the users and IoT devices. These ideas will also be context-aware and implement intelligence algorithms to reduce unnecessary resource consumption and analyze data to determine the most efficient level to process the data. Despite all of these common characteristics, both provide services in different ways and satisfy different needs.

While edge computing is still in its beginning stages [4], it is rapidly developing and many are researching different aspects of this subject. In this work, a general overview of edge computing and its different proposed infrastructures, identify problems and discuss possible solutions. In Section II mobile edge computing infrastructure and software features are defined and discussed. The paradigm of fog computing will be covered in Section III. Section IV will cover challenges of development and deployment. Finally, Section V concludes this work.

2. Mobile Edge Computing

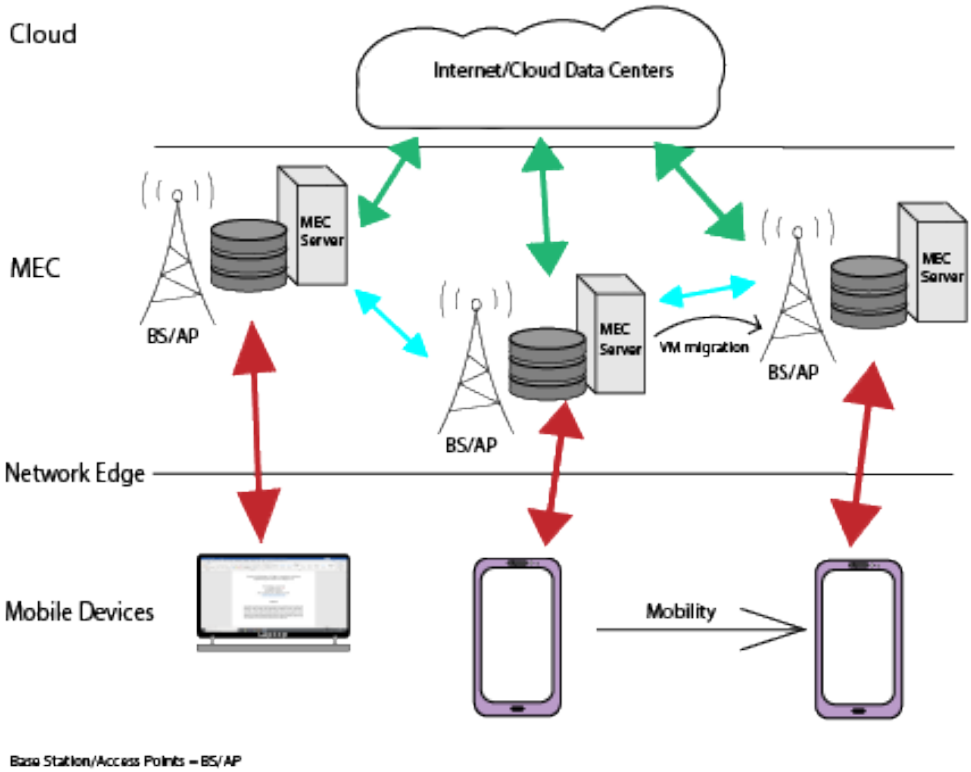

Figure 1: MEC Network Design

2.1. MEC Architecture

MEC will bring the edge closer to end users (EU), shown in figure 1, through the deployment of distributed, localized, privately owned microdata centers that will reduce traffic to the cloud. These centers will be located and integrated within radio network controllers, access points, base stations, and sites that handle aggregation [1]. They perform many cloud functions on a smaller scale, due to the limited space and cost, and process as much data as possible before sending it to the central cloud or another micro data center. Seemingly insignificant data processed at one of these centers can have a significant impact on the speed of the internet by reducing internet traffic over long distances and reducing the workload of the cloud data center; this allows for faster processing of computation intensive processes. Since MEC centers are local, the area's needs can be better met by allowing customization of the center, and third parties are able to be approved to deploy applications. This localization can allow for quick adaptive change when there is a shift or influx of data in the network.

Open and generic architecture is a necessity. Considering the main components of MEC, the cloud, MEC centers, and IoT devices, it is understandable that the middle level must be receptive to an unknown variety of devices. Virtual machines (VM) run MEC applications within the infrastructure. Not only do VMs provide support for various devices, services, and applications, but it is incredibly useful for portability of end devices.

2.2. MEC Applications

MEC has many critical characteristics that will greatly improve jobs and lives; these characteristics are made possible by artificial intelligence (AI). Without AI, MEC will not be an improvement to the current cloud network. Context awareness, latency improvement, mobility, and computation offloading are some tasks that will need to incorporate intelligent functions.

Context awareness is a critical part of the success of MEC. These microdata centers must be able to analyze data and make real-time decisions based on the results. Consider a traffic scenario which utilizes sensors and cameras. These devices would collect data from the local area, send it to a smart traffic controller at a local MEC node where it would determine the number of pedestrians and vehicles, then change the light accordingly.

Latency would be improved tremendously by processing small data right at the node instead of unnecessarily routing it all the way to the cloud and back. A MEC node will have a response time of several milliseconds [9]. The node will process as much data as possible, with respect to the state of network traffic. If the node is unable to process all of the data due to a full network, it will send it to a center which is able to process it. If the computation load is too intensive, it will be sent to the cloud. This will impact basic EUs but, more importantly, emergency response devices and services, such as body sensors, cameras that scan vehicles can send real-time updates on suspect vehicles, and jet engine failures.

As stated above, mobility is carried out by VMs. A user can send data to the network that is received by a MEC server. The EU moves farther from the initial receiving server and into the scope of another different server. The servers communicate this device movement and initiate a VM migration. The optimal server should always have the user's VM.

Computation offloading has the potential to decrease energy consumption of end devices considerably. IoT devices do not have the resources to perform intensive computations, so these computations must be sent to the network to allow MEC nodes or the cloud to compute and return the result. While end devices do not have the capabilities that MEC servers or the cloud do, they can process small data. While offloading, the end devices must decide whether to fully offload or partially offload.

3. Fog Computing

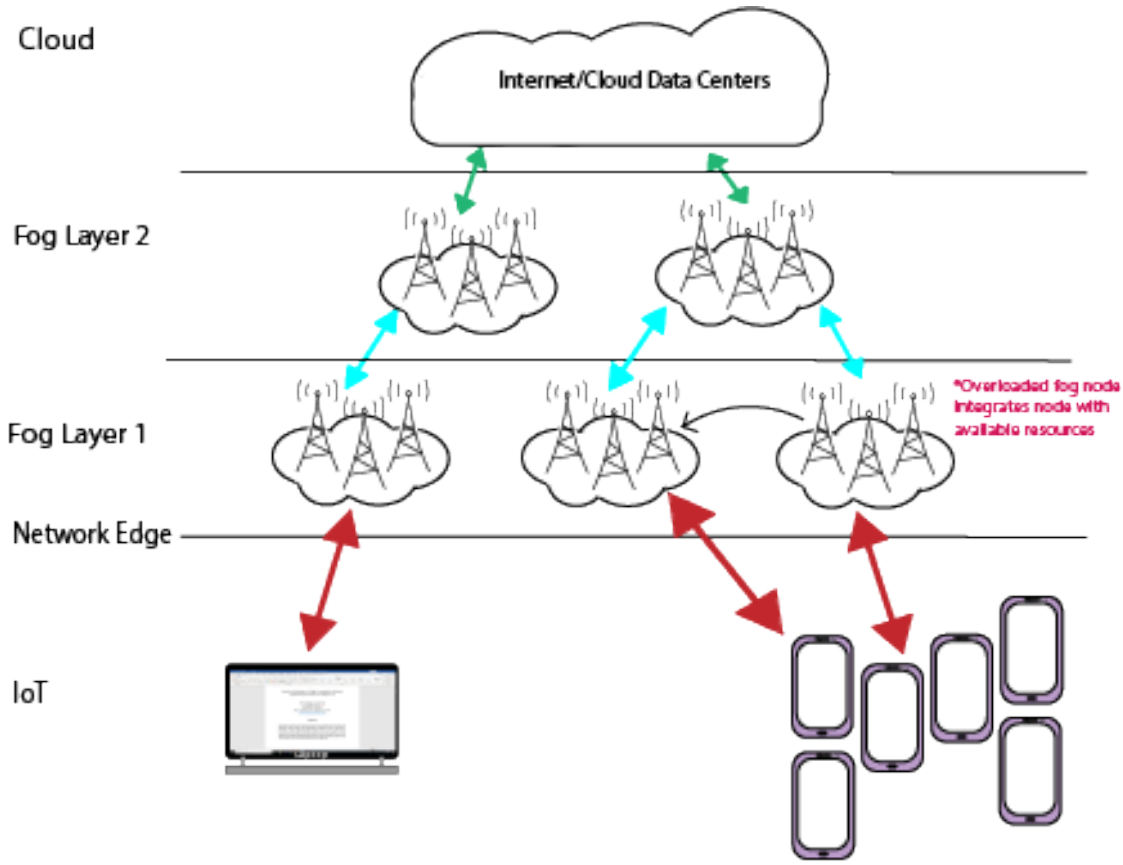

Figure 2: Fog computing architecture

3.1. Fog Computing Architecture

Fog computing is a complementary extension of the cloud and not a replacement [7]. This architecture, shown in figure 3, can carry out distribution, orchestration, management, and provide security across the network. When deployed, it has a hierarchical structure consisting of three tiers: the cloud, fog nodes, and IoT devices. Fog nodes will interact with local access points to the network, i.e. routers or gateways. Unlike MEC, these nodes will not be privately owned but maintained by enterprises.

Fog nodes are the key aspect of this architecture. They will handle less intensive computations, communicate with the rest of the network, and store data while intermittently sending updates to the cloud. Fog nodes will implement and utilize the invaluable properties of a horizontal architecture to help handle incoming traffic and improving Quality of Service (QoS) and Quality of Experience (QoE) of all EU. The horizontal architecture will allow fog nodes to send data to other fog nodes to utilize parallel processing. This will significantly impact several aspects of the network. For example, the amount of traffic to the cloud will decrease, bandwidth availability will be enhanced, responses will be real time, and the offloading to idle resources will relieve stress on nodes dealing with heavy network traffic.

3.2. The software of Fog Computing

Fog computing needs to provide flexibility to this promising network architecture through the use of intelligence algorithms. This will encapsulate mobility, scalability, and high availability to billions of IoT devices. The software must be able to effectively and efficiently analyze data in order to provide these essential functions.

Intelligence is one of the most crucial parts of the entire software architecture. The implementation of fog computing will need machine learning (ML) algorithms to process and analyze local data to fit the current traffic flow in such a way that the fog nodes are not overwhelmed. In addition, decisions will need to be made at the edge whether to offload processing to another node or send it to the cloud.

Fog nodes will be more local than the cloud. A need for mobility is one of the most important functions of the fog. As a user moves, their connection should seamlessly move over to another node or traditional coverage if fog nodes are unavailable. This implements the use of containers. Containers are software that collects and packages any and all code needed to run an executable. Unlike VMs, they do not create virtualized computer components but are abstractions at the application layer which results in faster booting, packaging, and unwrapping of these executables [12].

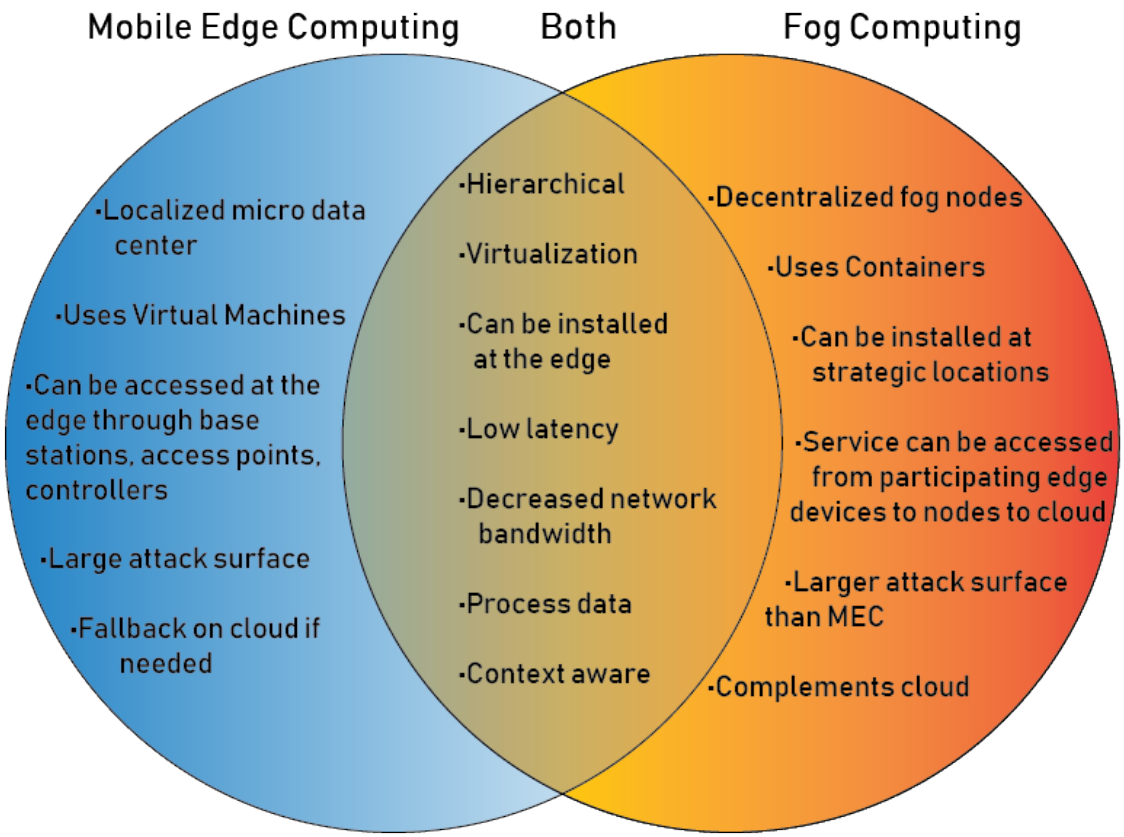

Figure 3: Venn Diagram of MEC and Fog

4. Challenges

While edge computing has incredible potential and promise, there are challenges to overcome before the deployment of this future network. Edge computing architectures bring the network closer to the EU's in the form of mini data centers or nodes, see figure 3 for a brief overview. As a result, fog nodes and MEC servers have limited resources due to space and cost constraints which introduce multiple security issues and offloading challenges.

As stated previously, smaller nodes and microdata centers will have less computational power and limited resources. This presents a red flag on security. In the MEC architecture, security must be provided on the microdata centers; for fog, the security functions must be handled by any nodes that participate in this infrastructure. The traditional cloud is vulnerable to many attacks such as the man in the middle and denial of service; edge processing units are more susceptible to these attacks without added security features. A current possible solution to DoS attacks is the plugin Defence4All for the OpenDayLight software-defined network (SDN) controller [1]. This specific application diverts suspicious data to a different facility for further analysis to confirm the threat level. This is a potential solution but still poses the challenges of increased bandwidth usage during redirection.

Most proposed approaches are not dynamic which does not meet the requirements of this new extremely flexible paradigm.

Another problem that must be improved on before adopting edge computing is offloading issues. The current tradeoff for offloading is decreased energy consumption for possible latency issues. This is a bigger challenge for the MEC system than fog because of the different implementations of virtualization, i.e. VM vs containers. Also, flexibility is crucial for this function because different applications need different levels of offloading. When offloaded, the nodes and centers need to determine how much data can be processed at the edge and what needs more computational power.

5. Conclusion

Despite the challenges of these network architectures, there is a tremendous amount of promise. With the explosion of technology, solutions will be proposed and continually improved to provide increased QoE and QoS. The incredible flexibility that edge computing will bring to the network is invaluable and a necessity for the future of technology. Innovation will be positively influenced and current time-sensitive applications will no longer face the burden of latency.

References

[1] R. Rapuzzi and M. Repetto, "Building situational awareness for network threats in fog/edge computing: Emerging paradigms beyond the security perimeter model," Future Generation Computer Systems, vol. 85, pp. 235–249, Aug. 2018.

[2] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges," Future Generation Computer Systems, vol. 78, pp. 680–698, Jan. 2018.

[3] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, "Secure integration of IoT and Cloud Computing," Future Generation Computer Systems, vol. 78, pp. 964–975, Jan. 2018.

[4] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. D. Foy, and Y. Zhang, "Mobile Edge Cloud System: Architectures, Challenges, and Approaches," IEEE Systems Journal, vol. 12, no. 3, pp. 2495–2508, Sep. 2017.

[5] M. Chen, Y. Qian, Y. Hao, Y. Li, and J. Song, "Data-Driven Computing and Caching in 5G Networks: Architecture and Delay Analysis," IEEE Wireless Communications, vol. 25, no. 1, pp. 70–75, Feb. 2018.

[6] V. Gruhn, R. Striemer, S. Nastic, and S. Dustdar, "Towards Deviceless Edge Computing: Challenges, Design Aspects, and Models for Serverless Paradigm at the Edge," in The Essence of Software Engineering, Springer International Publishing, 2018, pp. 121–136.

[7] Q. D. La, M. V. Ngo, T. Q. Dinh, T. Q. Quek, and H. Shin, "Enabling intelligence in fog computing to achieve energy and latency reduction," *Digital Communications and Networks*, vol. 5, no. 1, pp. 3–9, 2019.

[8] L. Huang, X. Feng, C. Zhang, L. Qian, and Y. Wu, "Deep reinforcement learning-based joint task offloading and bandwidth allocation for multi-user mobile edge computing," *Digital Communications and Networks*, vol. 5, no. 1, pp. 10–17, 2019.

[9] S. Taherizadeh, A. C. Jones, I. Taylor, Z. Zhao, and V. Stankovski, "Monitoring self-adaptive applications within edge computing frameworks: A state-of-the-art review," *Journal of Systems and Software*, vol. 136, pp. 19–38, Feb. 2018.

[10] N. Srinidhi, S. D. Kumar, and K. Venugopal, "Network optimizations in the Internet of Things: A review," *Engineering Science and Technology, an International Journal*, vol. 22, no. 1, pp. 1–21, Feb. 2019.

[11] Y. Ai, M. Peng, K. Zhang, "Edge computing technologies for the Internet of Things: a primer," *Digital Communications and Networks*, vol. 4, no. 2, pp. 77-86, Apr. 2018.

[12] "What is a Container?," *Docker*. [Online]. Available: https://www.docker.com/resources/what-container. [Accessed: 27-Mar-2019].