Edge and Fog Computing

Ashair Imran
Faculty of Engineering
University of Ottawa
mimra070@uottawa.ca
300071365

Alex[redacted]
Faculty of Engineering
University of Ottawa
[redacted]

***Abstract*— Edge and fog architectures involve computations to reduce latency when interacting with a cloud server. Tasks are offloaded to the edge and fog networks, where high-speed computations are performed to process large volumes of data. Response times and processing speeds are faster in the edge and fog layers when compared to the cloud layer. However, edge and fog architectures run into problems such as resource management, and scalability. To further progress edge and fog architectures it is important to solve these obstacles. This paper explores the architectures of edge and fog computing, as well as issues that these network architectures face and ways to mitigate these challenges.**

I. Introduction

As smart technology advances, the number of devices that collectively produce data increases. In 2017, 20.35 billion devices were connected to the cloud. This number reached 30.73 billion in 2020 [1]. As the number of smart devices increases, the volume of data that needs to be processed increases at an astronomical rate. The sheer velocity and volume of data that needs to be moved to the cloud cause inefficiency and infeasibility. Another issue is that time-sensitive and location-sensitive devices will simply be unable to use the cloud in such low-latency environments [2]. A new network paradigm is necessary to expand the current cloud architecture to address these aforementioned concerns. Edge and Fog technology is used to ease the burden of the cloud and reduce latency by bringing processing power closer to the source of data. This 'task offloading' is vital to improve the current network architectures and support the ever-expanding Internet of Things, or IoT.

A. Cloud

Before delving into the intricacies of Edge and Fog technology, it is important to define the common terms used throughout this report. Edge and Fog technology is used to address problems on the Cloud platform. Cloud computing provides data storage and computation power distributed over many large data centers. These data centers have massive data storage capabilities and computational power. The widespread distribution of a cloud network allows individuals to use the computation power of a cloud network whenever and wherever it is required. The cloud is the biggest data unit where processing and storage is performed, and it is only growing in importance in scale, as can be seen by the increase of devices connected to the cloud [1]. The importance of widespread, on-demand databases, networks, and processing power should not be understated. The power of the cloud has infinite use cases and applications.

B. Edge

Edge and Fog computing is a term to describe the architectures involving computations occurring at the 'edge' and the 'fog' of a network. The edge of a network can be described as the location where physical devices of a local network interact with the internet. 'edge' refers to endpoints of a specific network, such as computers, smart devices, and mobile phones. Edge computing refers to when processes and computations are performed at the edge, before being sent outwards to the network or the cloud. It is also important to note that edge computing can occur in 'edge servers', local servers that perform computations for devices on the network. These edge servers have some overlap with fog servers, and many times an existing edge server can be used as a fog node. Throughout this report, edge computing will refer to computations performed on edge devices before the data is either sent further outwards to the fog or the cloud, or the data can be sent back to the original edge device.

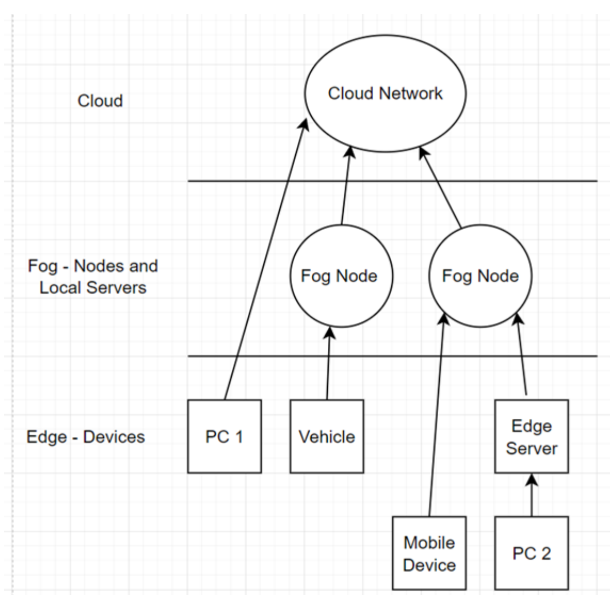

Fig. 1. Edge-Fog-Cloud Architecture

C. Fog

The ‘fog’ refers to an intentionally decentralized network infrastructure, which means that there is no centralized controller responsible for every action of the fog node. The fog is an additional layer between the edge and the cloud and consists of many fog nodes. These fog nodes can be viewed as lightweight implementations of a cloud server, distributing its resources to the local area. Like edge computing, fog computing performs computations in this fog layer before being either transferred to cloud-based servers or back to the original device. This can be seen in Fig. 2. Cisco defines fog as an “extension of cloud that goes from the center to the edge to enhance frequent services, low latency, and big data analysis” [1]. There can be many different interactions between these different layers. A PC can connect directly to the cloud, an autonomous vehicle may use a fog node, and a PC can use an edge server to perform calculations. All these various interconnections help each other.

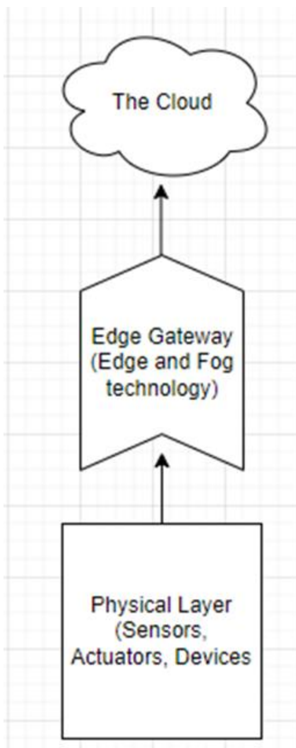

Fig. 2. Internet of Things Architecture

D. Internet of Things

The most important and widely used implementation of edge and fog technology is within the Internet of Things, or IoT. The Internet of things is the concept of physical objects embedded with networking technology. Examples include smart home devices or smart vehicles and even extend to smart factories, farming, and smart cities. The IoT provides many data-motivated functionalities, and many of these processes require not only large processing power but also low latency to maintain a high level of functionality [3]. These devices create large amounts of raw data which oftentimes need very strong computational power, and as the needs of the Cloud grow, it will become less capable to compute all the raw data produced by the Internet of Things. Large quantities of data generate greater bandwidth consumption and higher latencies. This is where the edge and the fog technology come into play to perform some of these computations and ease the burden of the Cloud. The IoT has different architecture layouts, and one such layout consists of three different layers to the IoT, as can be seen in Fig. 2. These layers include physical devices such as sensors, actuators, the devices that collect various signals. Then all this data is sent through the Edge Gateway, and then to the Cloud. Edge and Fog Computing occurs in the Edge Gateway layer, and it is the part that will be discussed the most throughout this report.

II. Description of Addressed Environment

Before analyzing the various applications and use cases of edge and fog technology, it is important to understand the wide array of environments that the technology operates. First, the intricacies and architecture of the IoT will be discussed. As discussed previously, one of the layers of the IoT is called the Edge Gateway, which is the location between the devices and the cloud network. Not only does the edge gateway connect the device sensors to the cloud platform, but it also provides local functions and a level of security between local devices and the cloud. The purpose of the edge is to bring cloud computation as close as possible to the device and we can see this applied in the IoT architecture through the edge gateway. However, it is important to note that edge and fog technology should not be seen as a replacement to cloud technology, but rather to complement cloud computing to reduce its burden [1]. The main method to distinguish whether fog computing or cloud computing should be focused on is whether the application must maintain quality of service while being latency-sensitive. Latency-sensitive applications warrant the use of fog computing [2]. A fog system will also have relatively smaller computing resources than a cloud system, however, fog systems can process data from a more diverse set of devices, and it is possible for fog systems to exist on lower-end devices [3].

A. Fog Nodes

Fog nodes and edge servers are proximal servers that provide service to local devices and connect such devices to the cloud base. There are a very large number of fog nodes that are geographically implemented to locations that allow them to cooperate and communicate with each other to allow real-time processing. These nodes offer computing, storage, networking resources that are under the infrastructure [4]. The fog nodes, however, are not typically deployed in centralized areas as is the case with cloud data centers [2]. Fog nodes are heterogeneous devices, which include servers, base stations, edge routers, each with a different allocation of resources. These fog nodes are designed according to industry standards to provide acceptable latency to any company. The primary power of fog technology is the fact that it is decentralized. This decentralization allows the fog networks to be refined on the local level in accordance with its needs [1]. The OpenFog Consortium, a consortium comprised of Cisco, Intel, Microsoft, and more, describes fog computing as “a horizontal system-level architecture that distributes computing, storage, control, and networking functions closer to the users along a cloud-to-thing continuum” [2]. It is important to note that fog technology is described as a horizontal architecture rather than a vertical one. This horizontal platform allows fog computing to be distributed along different industries and platforms.

B. Decentralized IoT

Fog computing, an inherently decentralized paradigm, can be utilized to create something called a decentralized IoT. A decentralized IoT has a different architecture than a typical IoT architecture that does not use fog nodes. A typical IoT architecture has something called a main centralized controller which is responsible for the different actions that the IoT network takes. When incorporating fog nodes, you can decentralize the architecture. What this means is that the centralized controller is no longer responsible for every little action that is taken by the various devices in the network. Decentralization also allows fog nodes to join and leave the network arbitrarily [5]. This is important because it can reduce cloud server loading and increase responsiveness. This is especially important because the IoT is used for many latency-sensitive tasks. For example, fog nodes are used to allow autonomous vehicles to communicate with one another. This task is very latency sensitive as cars move at very high speeds and it is important to be constantly updating the car's information to the fog node and relaying it to other autonomous vehicles as fast as possible.

C. WSN

Another platform that fog computing utilizes is wireless sensor networks or WSNs. These networks are used to obtain information from certain environments wirelessly. These IoT-based networks are most commonly seen in self-driving vehicles, medical care monitoring, and industrial monitoring. These WSNs suffer from many challenges, yet are an important technological advancement in the fields that they are used in. WSNs have a single network topology, limitations on energy resources, remote locations, and require a battery to power them, as they are wireless [1]. However, the fog network can enhance these WSNs to improve their functionality and reduce latency, which is vital for WSN use cases.

D. Implementations

There are many different implementations of edge and fog computing. It is sometimes helpful to identify the differences between the various architectures and implementations. Three main implementations can be identified, these are Mobile Edge Computing (MEC), Fog Computing (FC), and Cloudlet Computing (CC) [6]. MEC relies on deploying nodes on cellular networks. The base station of these cellular networks holds edge servers that allow cloud computing capabilities to the users of that cellular network. Fog computing traditionally uses machine-to-machine gateways and wireless routers to compute and store data on end devices. Cloudlets are dedicated servers, similar to data centers, but on a smaller scale, to allow task offloading similar to a data center. As an emerging technology, it is important to identify the differences and intricacies in the different edge and fog implementations

III. Applications and Use Cases

The main advantage of Edge and Fog technology is that it brings the cloud platform closer to the end user's device. This instantly resolves many issues with cloud computing and results in a variety of useful applications. Fog computing provides support to a broad range of industrial applications because of its instant response capabilities [3]. Fog nodes have their own computing, storage, and networking services, and the local operation of a fog node allows end-users to access it with a single hop. Fog computing is:

- Inexpensive
- Low Latency
- Flexibility
- Virtualized
- Portability
- Decentralized

All of these properties enhance and expand the various applications of Fog Computing in the IoT.

A. Edge Servers

Edge servers are commonly used in Content Delivery Networks (CDN) and Points of Presence (POP). A point of presence is a point in a network that allows that network to interface with other networks. A content delivery network, or CDN, is a distributed network of servers and data centers. The goal of a CDN is to provide high-quality, content containing large data streams e.g., videos, or movies, to the end-user with high performance. This is done by the use of edge servers. Edge servers are located geographically close to the end-user and contain these copies of these large files. In an ideal scenario, it is optimal to use existing Edge servers to deliver content when establishing a new CDN, which can have an advantage in cost and performance.

Another use of Edge servers is cloud gaming. Oftentimes a home desktop PC does not have the capabilities to run resource-intensive video games. As technology advances and new games are released, the demand on graphics cards and processors increases. Topically, the semiconductor shortage also has contributed to the incapability of home gaming [7]. Cloud gaming is a recent application of Edge Servers that allows people to play video games on a local edge server which is then streamed to the user's PC. This concept of task offloading is a reoccurring one in Edge and Fog computing. When a user does not have the computational power to perform a certain task, they can rely on the cloud to perform that task for them at the cost of latency. Edge and Fog nodes can reduce this latency by bringing the task closer to the end-user.

B. Fog Computing

As discussed earlier, Fog computing has its most important uses when dealing with latency-sensitive issues. Latency is the most important factor to handle in certain industries such as the medical industry and the automobile industry. In the development of self-driving vehicles, it is important for the vehicle to perform computations at an extremely fast pace. The vehicle must know the speed and location of the vehicles around it, its current speed, how hard to brake, etc. These computations must have extremely low latency. If latency is introduced into the system, the results can be life-threatening. Many computations can be performed on board to reduce latency as much as possible, but as the complication of tasks increases and the necessary resources increase, the cloud cannot be relied on. This is where fog technology can be useful, to reduce the latency of computations, increase the resources, without relying on distant cloud bases. Another use case of this technology is in the medical industry. The IoT is already widespread in health monitoring systems [8]. Once again, it is of utmost importance that the latency of these devices is reduced as low as possible as human lives are at risk. A decentralized fog system can enhance the pre-existing IoT to reduce the latency and provide more resources to the applications that require it.

IV. Challenges

In the previous sections, we have discussed the upsides and applications of edge and fog computing architectures. In order to fulfill the vision of deploying edge and fog computing on a wider scale, it is necessary to first overcome some of the challenges that these architectures face. This section will discuss the challenges and limitations that edge and fog computing already face or have the potential to face. It should also be noted that these challenges will focus mainly on the algorithms and architectures of both computing methods at the hardware and software layers. First, the challenges faced by edge computing will be addressed, then following that we shall address the challenges faced by fog computing

A. Edge Computing Challenges

1) Task offloading on edge nodes: As the scope of distributed computing environments widens, consequently, the demand for the development of new task management techniques increases. Such a technique would have to be able to ensure that tasks are executable at multiple geographic locations whilst also being efficiently partitioned. Typically, task offloading and partitioning would be handled explicitly via a programming language or management tool [9]. The challenge with offloading tasks on edge nodes is that there does not exist an automated method in which tasks can be efficiently partitioned. Therefore, until there is headway in the development of schedulers that are capable of sending partitioned tasks onto edge nodes, task offloading via edge computing will lack efficiency.

2) Lack of efficient naming mechanisms: Due to the abundance of unique and distinct applications that are being run atop edge nodes, it is important that there are effective naming schemes established in edge computing. As in many other network topologies, a good naming scheme is vital for data communication, addressing, and identification [10]. Furthermore, it improves the programmability of edge computing. The challenge that edge computing faces, as it pertains to naming mechanisms, is that there does not exist one that has been fully developed and standardized yet. As a result of the highly dynamic and mobile nature of this topology, traditional naming mechanisms such as DNS (Domain Name System) and URI (Uniform Resource Identifier) are ineffective in edge computing. Thus, edge practitioners must resort to learning numerous network and communication protocols as a means to establish communication between the heterogeneous components of the system [10]. Consequently, this method can be quite time-consuming and laborious.

3) Scalability: Regardless of the geographical location of the edge, it is imperative that all edge nodes undergo the same protocols and procedures. Due to the wide variety of demands and services that IoT devices are currently capable of, there must be a proportionate increase in resources and monitorization at edge servers. Ergo, there will be challenges with assessing and managing issues in areas with fewer financial and technical resources. Not only will there need to be more servers built at the edge, but there will also need to be an increase in scale across all IT disciplines, which can be costly. As a result, it is difficult to visualize the scale in which edge computing can be utilized.

4) Edge node discoverability: A significant challenge for edge computing remains on how to discover the nearby edge servers, as well as how to access them. The edge of the network calls for the development of new discovery mechanisms which will be tasked with locating nodes that can translate into a decentralized cloud setup [9]. Furthermore, existing discovery methods, used in cloud computing, are currently impractical in the context of edge node discovery. For an edge node discovery method to be successful, it must be automated in order to handle the sheer volume of devices available at this layer, and the wide range of heterogeneous devices that can be used [9]. Such a method must not only be able to efficiently integrate and remove nodes, but also do so without increasing latency and negatively impacting the user experience. In short, developing this edge node discovery tool will be a challenging task for the construction of edge computing architectures.

5) Programmability: One of the major benefits of cloud computing is that there exists a cloud provider that is responsible for deciding where computing occurs on the cloud. Users are able to program their code and send it to the cloud without having to worry about any intermediate tasks because the cloud provider is able to take care of it for them [10]. Programmers face issues with writing applications that may be deployed and executed at the edge due to the varying runtimes of heterogeneous edge nodes [10]. In edge computing, there is no cloud provider that is responsible for taking care of where computations are conducted in the cloud. Therefore, in order to improve the quality of service in edge computing, it is imperative to mitigate this issue.

B. Fog Computing Challenges

1) Heterogeneity: Heterogeneity, in the context of fog computing, refers to the degree to which the fog system nodes differ in computing and storage capabilities [11]. This is important because it can have a significant impact on the performance of algorithms. The challenge that fog computing faces, as it pertains to heterogeneity, is the extent to which existing computing and storage technologies are able to spare resources that can serve as fog nodes [11]. We can process device requests and store content by successfully imposing heterogeneity in terms of storage and computations. Additionally, we must also figure out how to add additional computing and storage nodes, as well as learn the dimensions and placements of these additions so that we are able to optimize the system. It is important to mitigate this issue as it greatly affects the efficiency of fog computing.

2) Inefficient resource management: A key requirement of fog computing is that it is able to operate autonomously and independently in order to ensure uninterrupted services, even when connections with the cloud are unstable. Unlike cloud computing, fog computing does not contain enough computing and storage resources [12]. This calls for the development of appropriate resource management and coordination tools that are able to guarantee the acceptable performance of applications and services, while also taking advantage of cloud capabilities. One of the main challenges in developing a dependable fog computing architecture is determining how to efficiently allocate resources in the system.

3) Inefficient application offloading: Building upon the previous challenge of inefficient resource management, application offloading is also not consistent with its efficiency in regards to; delay, bandwidth, and energy consumption due to the lack of available resources [12]. Furthermore, because the factors governing offloading performance are unknown, fog offloading is inefficient because we are uninformed of the major benefits of offloading applications' expected performance. As a result fog computing will perform inefficiently unless progress is made in the creation of a mechanism that is capable of securely offloading applications in fog environments.

4) Mobility and Scalability: Mobility and Scalability pose significant challenges in fog computing. Regarding mobility, the continuation of given services must be assured despite the IoT device's travel between several fog domains. There still also remain question marks as to whether or not fog computing is feasible, and performs well, on a large scale with a wide variety of devices. The challenge here is discovering the necessary strategies to deal with the mobility of fog computing. Such a strategy would have to ensure that services are continuously delivered as a device moves across numerous fog domains to maintain efficient functioning. Because research into this topic is so limited, mobility is a problem that fog computing has yet to overcome. Furthermore, many of the existing schemes and algorithms for fog computing do not scale to the magnitude of IoT networks [2], which is why scalability is a problem that fog computing has yet to overcome.

5) SDN defined fog computing: Several research concerns occur in fog computing, based on Software Defined Networks. Fog computing is not supported natively by SDN software [2]. A local coordinator is required to handle the dynamic and time-varying service requirements, and also must be placed strategically in order to handle fog-related task handling. furthermore, we must take into account controller design, The fog-SDN controllers need to cooperate among themselves in order to efficiently handle the limited fog resources. By improving and standardizing SDN software for fog use cases, the development of fog computing software becomes less complicated.

V. PROPOSED SOLUTIONS AND IMPROVEMENTS

The preceding section explored the various and unique challenges faced by both edge and fog computing. These issues were mainly centered on scalability, resource management, as well as task and application offloading. Although these challenges pose roadblocks in the development of fully functional fog and edge computing network architectures, there are ways to overcome them. In this section, we shall discuss some proposed solutions and improvements of edge and fog computing, based on academic research. Similar to the challenges section, this section will be broken down into two parts. The first part will discuss possible solutions for edge computing and following that, the second part will discuss possible solutions for fog computing.

A. Edge Computing Solutions and Improvements

1) Introduction of computing streams: In the context of computing, a computing stream is defined as a series of functions and computations that can be applied to the data along the data propagation path [10]. This computation can take place anywhere along the path so long as the application explicitly specifies exactly where it should be conducted [10]. By developing a computing stream, we are able to ensure that data is computed as close as possible to the data source while also reducing the data transmission cost. Furthermore, by integrating a computing stream, we are effectively able to solve the challenge of programmability in edge computing. A computing stream can help solve the challenge of programmability by helping the user to determine what functions and computing should be done, and how the data is to be propagated after the computing happens at the edge.

2) Addition of new naming mechanisms: To solve the issue of the lack of efficient naming mechanisms, a new naming mechanism would have to be developed to be able to keep up with the mobility, and highly dynamic nature of edge computing whilst also being secure and scalable. As would effectively optimize programming, addressing, identification, and data communication efficiency in edge computing. Traditional naming mechanisms such as DNS and URI lack the flexibility to adapt to the dynamic edge [10], however, there are new naming mechanisms in place that could be applied such as Name Data Networks (NDN) or MobileFirst. NDN has good edge scalability in addition to being user-friendly for service management [10]. NDN is content-centric as opposed to being endpoint-centric. The data packets are identified using hierarchically structured names rather than IP addresses. MobileFirst detaches the name from the network address for the purpose of providing better mobility. This is beneficial for edge computing networks as they are highly mobile. By finding a way to incorporate either of these two mechanisms into edge computing architectures, we come closer to being able to mitigate the challenge of the scarcity of efficient naming mechanisms.

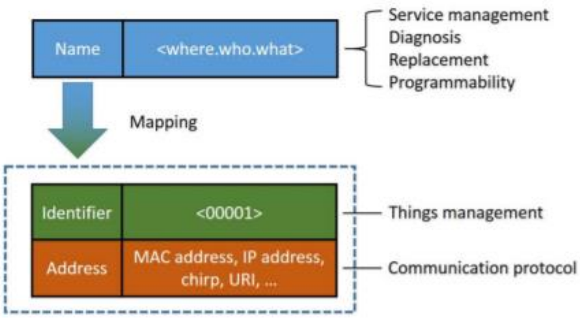

Fig. 3. Naming mechanism in edge operating system

3) Increase in the number of edge locations: By increasing the number of edge locations we are able to minimize inefficiencies in the mobility and scalability of edge computing architectures. Furthermore, expanding edge locations allows us to ensure system availability, network availability, and service availability. This grants the system the ability to ensure service anytime, and from anywhere, without degradation or interruption. By providing fully uninterrupted, and available service, edge computing can better serve real-time applications such as video streaming and vehicular edge technologies.

4) Addition of server clusters and load balancing: A possible solution to the issue of scalability in edge computing is the incorporation of server clusters and load balancing mechanisms. Server clusters refer to the act of turning multiple computer servers into a cluster of servers that act as one single system. Load balancing refers to distributing workloads across multiple computing resources and servers. Load balancing operates by a model of a min-max optimization problem. The goal is to reduce application processing time without facing any degradation or compromising the quality of service. Scalability is arguably the biggest challenge that edges computing faces, so the alleviation of this issue would be a large leap towards the mass deployment of this innovative topology.

B. Fog Computing Solutions and Improvements

1) Real-time analytics: A proposed solution to aid with application offloading is the placement of real-time analytics. By incorporating real-time analytics and being able to be privy to data as soon as it comes, we shall be able to know exactly which applications can be offloaded in order to satisfy the quality of service. Furthermore, we also become knowledgeable of what data should be sent to either the cloud or fog layer [12]. The addition of real-time analytics effectively strengthens the already existing computing infrastructure of edge computing and improves the quality of service by alleviating the burden that is placed on the cloud.

2) Virtualization and mobility patterns: By understanding the mobility pattern of IoT end devices [12], we may be able to effectively eradicate any inefficiencies regarding task assignment and resource management in fog computing. Virtualization is defined as the practice of utilizing software versions of physical computing resources. Additionally, virtualization can be introduced to perform the flexibility of a network service arrangement. Virtualization is beneficial for fog computing because it is capable of effectively eliminating the limitations regarding resource constraints on sensory-level nodes, whilst also efficiently creating custom applications for end-users. Network Function Virtualization (NFV) is a virtualization infrastructure that can be incorporated into fog computing to increase flexibility in managing applications. NFV is a method of replacing dedicated network hardware with software virtualized capabilities. By including virtualization mechanisms and technologies, such as NFV, fog computing becomes significantly more efficient in terms of mobility. Fog computing practitioners will gain the freedom of being fully mobile without having to worry about the compromise of network service.

3) Design of scalable algorithms: As previously mentioned, many contemporary fog computing techniques and algorithms are unable to handle the volume of IoT networks and the diversity of devices. Therefore, it is important for fog systems to be scalable so that they can be implemented in IoT networks. Fog computing may become more viable and widely employed if scalable methods and techniques for fog systems are developed. Some types of online job offloading, for example, could relieve an IoT device of the burden of decision-making without requiring any information from individual IoT nodes [2].

1) Enhancing and standardizing the SDN: We can improve communication between nodes and eliminate the challenges of data forwarding in smart grids in fog computing by upgrading SDN-based fog networks. The standardization of SDN-based fog environments will help solve the challenges of heterogeneity and help with managing different time requirements by different applications. OpenFlow northbound, southbound, and east-west bound are examples of SDN software that can be standardized in the fog computing paradigm [2]. OpenFlow is a programmable SDN network protocol that is used for managing and directing traffic across routers and switches from different vendors. With the numerous operators in fog systems, each with varying computing and storage capabilities, the addition and standardization of SDN architectures can enhance the quality of service in fog computing.

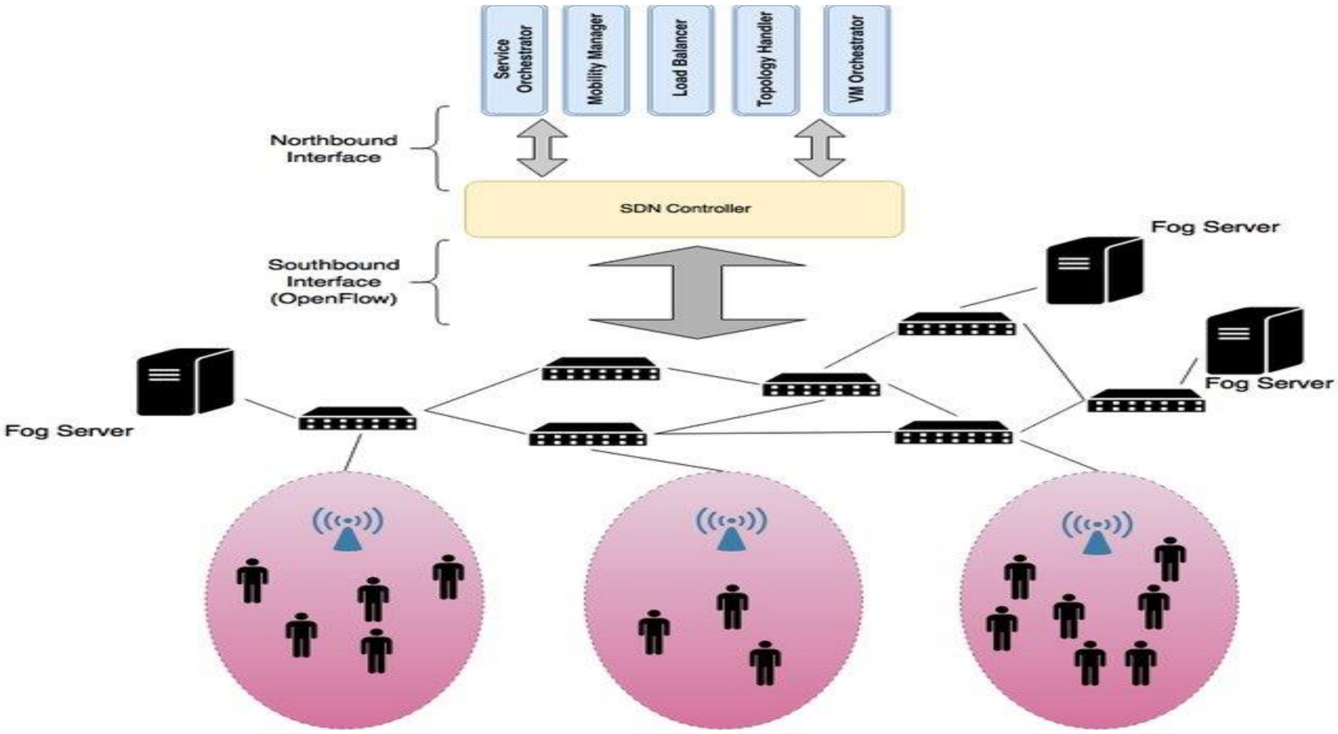

Fig. 4. SDN orchestrated fog computing

VI. Conclusion

As the Internet of Things expands, more devices import computational power from cloud-based servers. However, low-latency solutions are necessary for many applications. Edge and Fog computing act as a complement to Cloud computing by offering cheaper, low-latency resources to devices in local networks. Edge computing allows data storage and computation closer to the end-user through the implementation of edge servers. This allows for content delivery networks and other helpful low-latency alternatives to the Cloud. Fog computing is a decentralized system that allows the offloading of various tasks to increase performance and improve time-sensitive applications. The most important use cases involve instances where low latency is vital, such as in the medical or automotive industry. With the expansion of the IoT, it is important to revamp existing network infrastructure and incorporate necessary edge and fog technologies. However, although edge and fog technology provides many benefits, there are also many challenges involved when implementing the technology. These challenges are mainly centered around scalability and mobility. Edge and fog computing must consider the wide variety of IoT devices and also adapt to their movement across various network domains. By better managing resources and adopting technologies such as software-defined networks and virtualization into their infrastructures, edge, and fog computing have great potential for the future.

References

[1] G. Caiza, M. Saeteros, W. Oñate, and M. V. Garcia, “Fog computing at industrial level, architecture, latency, energy, and security: A review,” Heliyon, 08-Apr-2020.

[2] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, “All one needs to know about fog computing and related edge computing paradigms: A complete survey,” Journal of Systems Architecture, 12-Feb-2019.

[3] S. Khan, S. Parkinson, and Y. Qin, “Fog computing security: a review of current applications and security solutions,” Journal of Cloud Computing, 16-Aug-2017.

[4] A. Mebrek, L. Merghem-Boulahia and M. Esseghir, "Efficient green solution for a balanced energy consumption and delay in the IoT-Fog-Cloud computing," 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), 2017, pp. 1-4, doi: 10.1109/NCA.2017.8171359.

[5] J. Mocnej, A. Pekar, W. K. G. Seah, P. Papcun, E. Kajati, D. Cupkova, J. Koziorek, and I. Zolotova, “Quality-enabled decentralized IoT architecture with efficient resources utilization,” Robotics and Computer-Integrated Manufacturing, 08-Jun-2020.

[6] K. Dolui and S. K. Datta, “Comparison of edge computing implementations: Fog computing, cloudlet, and mobile edge computing,” *ResearchGate*, Jun-2017.

[7] J. Voas, N. Kshetri and J. F. DeFranco, "Scarcity and Global Insecurity: The Semiconductor Shortage," in IT Professional, vol. 23, no. 5, pp. 78-82, 1 Sept.-Oct. 2021, doi: 10.1109/MITP.2021.3105248.

[8] Mohanta, Bhabendu & Jena, Debasish & Panda, Soumyashree & Gountia, Debasis. (2019). Decentralized Secure Fog Computing in Cloud-Fog-IoT Infrastructure Using Blockchain.

[9] Varghese, Blesson & Wang, Nan & Barbhuiya, Sakil & Kilpatrick, Peter & Nikolopoulos, Dimitrios. (2016). *Challenges and Opportunities in Edge Computing*. 10.1109/SmartCloud.2016.18.

[10] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, "Edge Computing: Vision and Challenges," in IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637-646, Oct. 2016, doi: 10.1109/JIOT.2016.2579198.

[11] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," in IEEE Communications Surveys & Tutorials, vol. 20, no. 1, pp. 416-464, Firstquarter 2018, doi: 10.1109/COMST.2017.2771153.

[12] M. Mukherjee, L. Shu and D. Wang, "Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges," in IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 1826-1857, thirdquarter 2018, doi: 10.1109/COMST.2018.2814571.

[13] Laghari, Asif & Jumani, Awais & Laghari, Rashid. (2021). Review and State of Art of Fog Computing. Archives of Computational Methods in Engineering. 28. 10.1007/s11831-020-09517-y.

[14] Baktir, Ahmet Cihat & Sonmez, Cagatay & Ersoy, Cem & Ozgovde, Atay & Varghese, Blesson. (2019). Addressing the Challenges in Federating Edge Resources: Principles and Paradigms. 10.1002/9781119525080.ch2.