Control-Plane Load Balancing and Auto-Scaling in 5G and Beyond Networks

Wriddhiraaj Dev
Dept. of CST, IIEST, Shibpur, India
Email: wriddhiraajdev@gmail.com

Samaresh Bera, *Senior Member, IEEE*
Dept. of CSE, IIT Jammu, India
Email: s.bera.1989@ieee.org

Anutosh Tiwari
Dept. of CSE, IIT Jammu, India
Email: anutosh05@gmail.com

***Abstract*—In this paper, we study an auto-scaling and load balancing approach, called `ScaleLB`, in a 5G core control plane with an aim to minimize control plane latency while maximizing the resource utilization of active network functions. Specifically, we focus on the latency involved in gNB association and end-user authentication through the access and mobility management function (AMF) in the core network. `ScaleLB` considers two approaches for auto-scaling and load balancing – `proactive` and `reactive` – compared to the static approach. To minimize control plane latency, the `proactive` approach scales the number of AMFs based on a predefined threshold irrespective of future service requests. In contrast, on receiving new requests, the `reactive` approach scales the number of AMFs when the existing AMFs are fully utilized. We develop a prototype of the system using open-source software tools. Experiment results present the trade-off between latency and the number of active AMFs with proactive and reactive schemes while comparing them with the static approach. Furthermore, the proactive approach yields competitive performance with the static approach when the threshold value for scaling up is carefully considered.**

***Index Terms*—Load balancing, Auto-scaling, 5G core network, Control plane**

I. Introduction

The control plane overhead and signaling is one of the biggest concerns over the years in mobile core networks [1], [2]. Furthermore, with the introduction of emerging IoT applications and millions of devices to be supported by 5G (and beyond) network, issues with control plane signaling become more challenging to address with traditional hardware-based networking. Thanks to the service-based architecture of 5G network (5G-SA), which enables virtualized network function placement, specifically, at the 5G core networks [3]. The 5G core network is divided into control and user planes, where the former handles the control plane signals and the latter takes care of data forwarding. With 5G-SA, it is possible to place multiple virtualized network functions at the core network in the form of virtual machines or containers instead of hardware-based functions.

In 5G-SA, the access and mobility management function (AMF) handles the control signals for association of a base-station (gNB) and end-user authentication to the core network. Therefore, AMF is the crucial entity in the 5G core network. Recent studies focused on control-plane load balancing by placing multiple AMFs [2], [4]. These studies broadly focused on different algorithms, such as round-robin and constant hashing, for load balancing to the AMFs. However, they assumed that all the AMFs are always available, which leads to inefficient resource utilization and energy consumption in the absence of large number of gNBs and end-users. Furthermore, the core network should be abstracted from the radio access network (RAN) comprising of gNBs and end-users to improve the security.

In this paper, we propose a load balancing with auto-scaling approach, called `ScaleLB`, to scale the number of active AMFs while abstracting the 5G core network components from the RAN. We consider two approaches for scaling up the AMFs – proactive and reactive. In proactive approach, `ScaleLB` scales up the number of AMFs based on a predefined load threshold per AMF. In contrast, on receiving new requests, the reactive approach scales up the number of AMFs when the existing AMFs are fully utilized. We consider the round-robin algorithm for load balancing among AMFs. We implement `ScaleLB` inside Open5GS (https://open5gs.org/) 5G core network platform. Experiment results indicate that `ScaleLB` outperforms static approach in terms of number of active AMFs and achieves competitive performance in terms of control plane latency. In brief, `ScaleLB` provides three fold advantages: a) abstraction between the 5G core and RAN; b) load balancing of control plane traffic among AMFs; and c) auto-scaling of number of AMFs based on the network status.

The rest of the paper is organized as follows. Section II presents an overview of the existing works. Section III presents the proposed architecture with objectives. Section IV discusses the network setup and implementation of `ScaleLB`. The experiments results are presented in Section V. Finally, Section VI concludes the paper with future research directions.

II. Related Work

In this section, we review the existing works on load balancing and auto-scaling [2], [4]–[13] of network functions in mobile networks while identifying their limitations. Nguyen et al. [2], [8] proposed dynamic control plane load balancing approaches based on three parameters – load on the AMF, service time, and pending number of requests at AMF. The authors showed that the proposed dynamic load balancing approaches yield improved performance than the static load balancing algorithms, such as round-robin and consistent hashing. Similarly, Buyakar et al. [4] developed a prototype for load balancing of control plane traffic among multiple AMFs with an aim to reduce the control plane latency.

Harutyunyan et al. [11] studied a joint optimization problem of user association, placement of service function chains (SFCs), and VNF scaling with an aim to minimize the service provisioning cost. Bello et al. [6] proposed a predictive auto-scaling approach for the evolved packet core (EPC) in 4G network based on the CPU utilization of the associated network functions. The authors used container-based implementation of EPC to ensure efficient scaling of the network functions instead of using virtual machine (VM) based approaches. Similarly, Nguyen et al. [7] proposed an auto-scaling and load balancing of user plane gateways in 5G network. The authors combined the auto-scaling and load balancing for efficient user plane services based on the load in the network.

While there have been recent studies on the auto-scaling and load balancing on network functions, latency involved in control plane signaling is always a concern to meet stringent latency requirements of emerging 5G applications. Consequently, we revisit the load balancing and auto-scaling problem at the 5G core control plane.

III. Auto Scaling and Load Balancing

We consider the standalone architecture (SA) of 5G network [3]. Figure 1 shows the schematic view of the network architecture with different network functions at the 5G RAN and core networks. As depicted in Figure 1, the auto-scaling and load balancing module is placed between the RAN and AMF. The module abstracts the core network components from the RAN. Therefore, the control messages between the RAN and control planes are exchanged through the load balancing and auto-scaling module, called `ScaleLB` module.

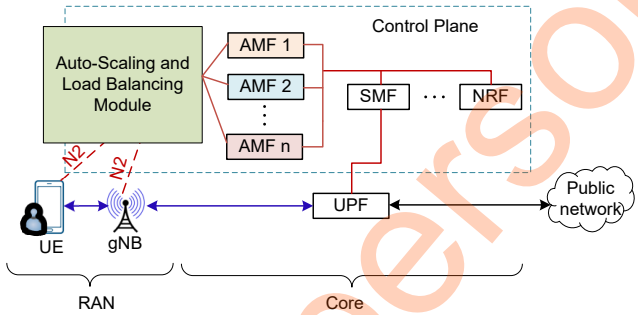


Fig. 1: Schematic view of the control-plane architecture of the 5G network with auto-scaling and load balancing module

Objective: The objectives of the proposed approach, called `ScaleLB`, are as follows:

- Automatically scale-up the number of AMFs based on the number of active gNBs to maximize the resource utilization.
- Balance the load from gNBs to AMFs to reduce the control-plane latency.

To achieve the above mentioned objectives, we propose two simple yet effective approaches, `proactive` and `reactive` auto-scaling, as presented in Algorithm 1. In `proactive` approach, `ScaleLB` continuously monitors the network states, i.e., number of active gNBs and AMFs. Whenever the utilization of AMFs reaches a (predefined) threshold value, γ, `ScaleLB` triggers the network controller to scale up the number of AMFs. Whereas in `reactive` approach, on receiving a new gNB association request, `ScaleLB` checks whether the existing AMFs are fully utilized and acts accordingly. Therefore, there exists a trade-off between the `proactive` and `reactive` approaches. While the former helps in reducing control-plane latency, it may lead to AMF under-utilization in the absence of new requests. On the other hand, the latter increases control-plane latency on receiving new requests, but improves the AMF utilization.

Algorithm 1 Auto-Scaling Approach

Inputs: Number of gNBs: B, Threshold on the number of gNBs per AMF: α; Predefined utilization threshold: γ
Number of active AMFs: A; Approach: `Reactive` or `Proactive`;

Output: Auto-scale the number of AMFs and gNB-AMF association

1: **if** Approach == `Proactive` **then**
2: **while** (1) **do**
3: `Get` the current network status (B, A);
4: **if** $\frac{B}{A \times \alpha} \geq \gamma$ **then**
5: `ScaleUP`(B, A) and `Update` A;
6: **if** Approach == `Reactive` **then**
7: Receives a new gNB association request;
8: `Update`: $B \leftarrow B + 1$;
9: **if** $\frac{B}{A \times \alpha} > 1$ **then**
10: `ScaleUP`(B, A) and `Update` A;

IV. Network Setup and Implementation

A. Overview of the Network Setup and Implementation

The proposed approach is implemented in a Kubernetes platform, where each network function (NF) runs in a containerized form. Figure 2 presents the connections between NFs with their IP addresses and interfaces. The components of the RAN, i.e., UEs and gNBs, exchange control messages with the AMFs through the `ScaleLB` module. On receiving requests from RAN, the proposed approach, `ScaleLB`, maps the messages with one of the appropriate AMFs based on the underlying load-balancing technique. Therefore, UEs and gNBs are always abstracted from the core network and enjoy seamless connectivity with the AMF. We use `Open5GS` (https://open5gs.org/) and `my5G-RANTester` (https://github.com/my5G/my5G-RANTester) to deploy the 5G core and RAN, respectively. We develop `ScaleLB` based on LoxiLB (https://www.loxilb.io/) in which we add the functionalities related to the auto-scaling.

B. Abstraction of AMFs in control messages between RAN and AMF

Figure 3(a) presents the gNB association and UE authentication messages that are exchanged between RAN, `ScaleLB` module, and AMF with time. The messages presented in

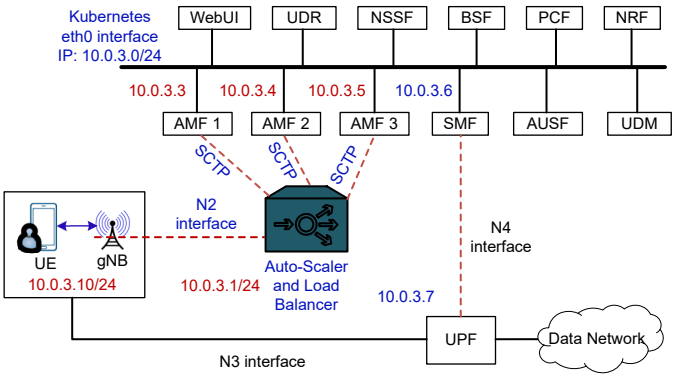


Fig. 2: Network details with auto-scaler and load balancer

Figure 3(a) are further validated through the packet capture at the ScaleLB module, as shown in Figure 3(b). This ensures the abstraction between the RAN and control plane in the core network which improves the core network security.

V. Results and Discussion

We compare the performance of ScaleLB with Static approach. As discussed in Section III, ScaleLB has two variants, ScaleLB-Pro and ScaleLB-React, to represent the proactive and reactive scenarios, respectively. ScaleLB-Pro continuously monitors the number of active gNBs and AMFs, and proactively scales the number of AMFs. Whereas in ScaleLB-React, AMFs are re-actively scaled when there is no AMF to serve new requests. In contrast, in Static approach, all the AMFs are always active irrespective of the number of active gNBs in the network. We consider a total of 90 UE-gNB pairs. Each pair of UE-gNB arrive uniform randomly in the network and sends the association request to AMF. Furthermore, we set the utilization threshold γ to different values, such as 0.7, 0.8, 0.9, and 1. We note that we consider a maximum of 30 UE-gNB pairs that can be served by an AMF in this experiment. Consequently, the total number of AMFs is set to three to serve all UE-gNB pairs in the network.

A. Auto-Scaling of AMFs

Figure 4 presents the number of AMFs with different threshold values for the ScaleLB-Pro approach with a comparison to ScaleLB-React and Static. The experiment results show the effectiveness of ScaleLB-Pro compared to ScaleLB-React and Static. In Static, all three AMFs are always active irrespective of the number of gNBs in the network. On the other hand, on receiving new requests, ScaleLB-React scales up the number of AMFs when the existing AMFs are fully utilized. In contrast, ScaleLB-Pro automatically scales the AMFs considering the number of gNBs in the network and the utilization threshold. The four different instances reflect the dynamic behavior of ScaleLB-Pro, as shown in Figures 4(a), 4(b), 4(c), and 4(d). Intuitively, we say that the proposed approach ScaleLB-Pro is energy-efficient compared to ScaleLB-React, which is further energy-efficient compared to Static, due to the dynamic scaling of AMFs. We note that ScaleLB-Pro and Scale-React yield similar performance in terms of number of AMFs when the threshold value γ is set to 1, as presented in Figure 4(d). Furthermore, by intuition, we say that ScaleLB-Pro and Scale-React achieve higher AMF utilization compared to Static.

B. Latency in gNB Association

We measure the latency for the gNB association with AMF for all approaches – ScaleLB-Pro with different thresholds, ScaleLB-React, and Static, which is presented in Figure 5. It is evident that ScaleLB-Pro yields competitive performance with Static without requiring to switch on all AMFs for all time when the threshold value is well-within the maximum capacity of the AMF. Furthermore, ScaleLB-Pro yields similar latency performance with ScaleLB-React when the threshold value is 1 similar to Figure 4(d). The spikes in latency for both ScaleLB-Pro and ScaleLB-React are due to the unavailability of resources of existing AMFs to serve new requests and an AMF takes roughly 40 seconds to be fully functional from its creation. Consequently, the association process between gNB and AMF is queued until a new AMF is available.

In summary, ScaleLB-Pro and ScaleLB-React dynamically scales the AMF based on the number of UE-gNB pairs in the network, while achieving competitive latency in gNB association when compared to Static. Furthermore, there exist a trade-off between the latency and number of active AMFs. With a low threshold value, latency is minimized but number of active AMFs is high, and vice-versa. Therefore, the threshold value can be properly tuned considering the arrival rate of UE-gNB pairs to minimize the control plane latency and maximize the resource utilization of AMFs.

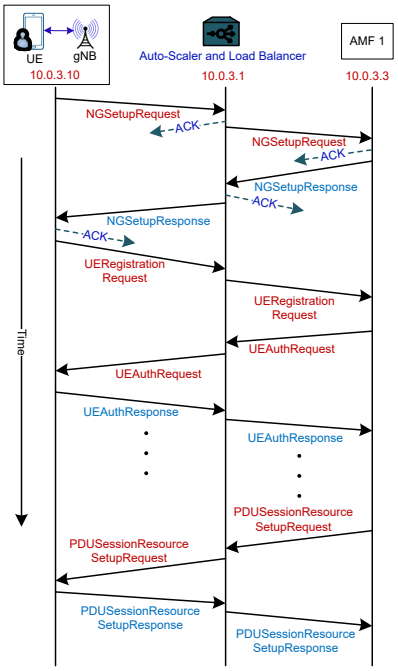
VI. Conclusion

In this paper, we studied the AMF auto-scaling and load balancing in 5G core control plane. We propose simple yet effective approaches – proactive and reactive – for the auto-scaling and load balancing of AMFs. We developed a prototype of the system using open-source software tools. The experiment results showed that the proposed proactive and reactive approaches are useful for providing improved control plane latency and resource utilization of active AMFs, respectively, compared to the static approaches.

The proactive approach scales the number of AMFs based on a predefined threshold on the maximum capacity of AMF without considering the future requirements. This may lead to inefficient resource utilization in the network. Consequently, a machine learning based prediction approach can be integrated with the proposed proactive scenario to improve the resource utilization. We consider this as a future research direction of this work.

Acknowledgment

The work is partially supported by the Indian Institute of Technology Jammu (IIT Jammu), India, and ISEA project

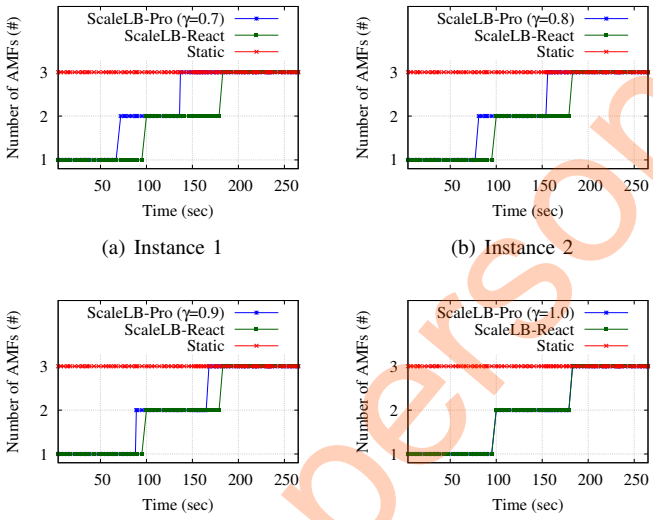


(a) Messages exchanged

```
11 10.0.3.1   10.0.3.3   NGAP          NGSetupRequest
12 10.0.3.3   10.0.3.1   SCTP          SACK (Ack=0, Arwnd=106433)
13 10.0.3.3   10.0.3.1   NGAP          NGSetupResponse
14 10.0.3.1   10.0.3.3   SCTP          SACK (Ack=0, Arwnd=106439)
15 10.0.3.1   10.0.3.10  NGAP          NGSetupResponse
16 10.0.3.10  10.0.3.1   SCTP          SACK (Ack=0, Arwnd=106439)
17 10.0.3.10  10.0.3.1   NGAP/NAS-5GS  InitialUEMessage, Registration request
18 10.0.3.1   10.0.3.3   NGAP/NAS-5GS  InitialUEMessage, Registration request
19 10.0.3.3   10.0.3.1   NGAP/NAS-5GS  SACK (Ack=1, Arwnd=106496) , DownlinkNASTransport, Authentication request
20 10.0.3.1   10.0.3.10  NGAP/NAS-5GS  SACK (Ack=1, Arwnd=106496) , DownlinkNASTransport, Authentication request
21 10.0.3.10  10.0.3.1   NGAP/NAS-5GS  SACK (Ack=1, Arwnd=106496) , UplinkNASTransport, Authentication response
22 10.0.3.1   10.0.3.3   NGAP/NAS-5GS  SACK (Ack=1, Arwnd=106496) , UplinkNASTransport, Authentication response
23 10.0.3.3   10.0.3.1   NGAP/NAS-5GS  SACK (Ack=2, Arwnd=106496) , DownlinkNASTransport, Security mode command
24 10.0.3.1   10.0.3.10  NGAP/NAS-5GS  SACK (Ack=2, Arwnd=106496) , DownlinkNASTransport, Security mode command
25 10.0.3.10  10.0.3.1   NGAP/NAS-5GS  SACK (Ack=2, Arwnd=106496) , UplinkNASTransport
26 10.0.3.1   10.0.3.3   NGAP/NAS-5GS  SACK (Ack=2, Arwnd=106496) , UplinkNASTransport
27 10.0.3.3   10.0.3.1   NGAP/NAS-5GS  SACK (Ack=3, Arwnd=106496) , InitialContextSetupRequest
28 10.0.3.1   10.0.3.10  NGAP/NAS-5GS  SACK (Ack=3, Arwnd=106496) , InitialContextSetupRequest
29 10.0.3.10  10.0.3.1   NGAP          SACK (Ack=3, Arwnd=106496) , InitialContextSetupResponse
30 10.0.3.1   10.0.3.3   NGAP          SACK (Ack=3, Arwnd=106496) , InitialContextSetupResponse
31 10.0.3.1   10.0.3.10  SCTP          SACK (Ack=4, Arwnd=106496)
32 10.0.3.3   10.0.3.1   SCTP          SACK (Ack=4, Arwnd=106496)
33 10.0.3.10  10.0.3.1   NGAP/NAS-5GS  UplinkNASTransport, UplinkNASTransport
34 10.0.3.1   10.0.3.3   NGAP/NAS-5GS  UplinkNASTransport
35 10.0.3.3   10.0.3.1   NGAP/NAS-5GS  SACK (Ack=5, Arwnd=106496) , DownlinkNASTransport
36 10.0.3.1   10.0.3.3   NGAP/NAS-5GS  SACK (Ack=4, Arwnd=106432) , UplinkNASTransport
37 10.0.3.1   10.0.3.10  NGAP/NAS-5GS  SACK (Ack=6, Arwnd=106496) , DownlinkNASTransport
38 10.0.3.3   10.0.3.1   NGAP/NAS-5GS  SACK (Ack=6, Arwnd=106496) , PDUSessionResourceSetupRequest
39 10.0.3.10  10.0.3.1   SCTP          SACK (Ack=4, Arwnd=106496)
40 10.0.3.1   10.0.3.10  NGAP/NAS-5GS  PDUSessionResourceSetupRequest
41 10.0.3.10  10.0.3.1   NGAP          SACK (Ack=5, Arwnd=106496) , PDUSessionResourceSetupResponse
42 10.0.3.1   10.0.3.3   NGAP          SACK (Ack=5, Arwnd=106496) , PDUSessionResourceSetupResponse
```

(b) Packet traces

Fig. 3: Messages exchanged between RAN, load balancer, and AMF with packet traces



(a) Instance 1

(b) Instance 2

(c) Instance 3

(d) Instance 4

Fig. 4: Auto-scaling of AMFs in different instances

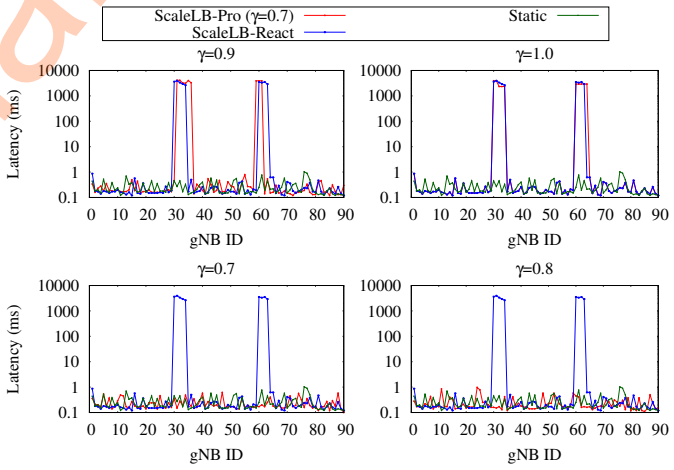


Fig. 5: gNB-AMF association latency (instantaneous)

Phase III (Sanction Letter Number: L-14017/1/2022-HRD), Govt. of India, at IIT Jammu. The work is also partially supported by the RISE-UP internship program at IIT Jammu.

REFERENCES

[1] "You need a robust signaling solution in 5G too!" https://www.ericsson.com/en/blog/2019/10/you-need-a-robust-signaling-solution-in-5g-too.

[2] V.-G. Nguyen, K.-J. Grinnemo, J. Taheri, and A. Brunstrom, "Adaptive and Latency-aware Load Balancing for Control Plane Traffic in the 4G/5G Core," in *Joint European Conference on Networks and Communications & 6G Summit*, Jun. 2021, pp. 365–370.

[3] 3GPP, "5G System Overview," https://www.3gpp.org/technologies/5g-system-overview.

[4] T. V. K. Buyakar, A. K. Rangisetti, A. A. Franklin, and B. R. Tamma, "Auto scaling of data plane VNFs in 5G networks," in *CNSM*, Nov. 2017, pp. 1–4.

[5] Y. Ren, T. Phung-Duc, J.-C. Chen, and Z.-W. Yu, "Dynamic Auto Scaling Algorithm (DASA) for 5G Mobile Networks," in *IEEE GLOBECOM*, Dec. 2016, pp. 1–6.

[6] Y. Bello, A. A. Abdellatif, M. S. Allahham, A. R. Hussein, A. Erbad,

A. Mohamed, and M. Guizani, "B5G: Predictive Container Auto-Scaling for Cellular Evolved Packet Core," *IEEE Access*, vol. 9, pp. 158 204–158 214, 2021.
[7] V.-G. Nguyen, K.-J. Grinnemo, J. Taheri, J. Forsman, T. Le Duc, and A. Brunstrom, "On Auto-scaling and Load Balancing for User-plane Gateways in a Softwarized 5G Network," in *CNSM*, Oct. 2021, pp. 132–138.
[8] V.-G. Nguyen, K.-J. Grinnemo, J. Taheri, and A. Brunstrom, "On Load Balancing for a Virtual and Distributed MME in the 5G Core," in *PIMRC*, Sep. 2018, pp. 1–7.
[9] T. V. Kiran Buyakar, H. Agarwal, B. R. Tamma, and A. A. Franklin, "Prototyping and Load Balancing the Service Based Architecture of 5G Core Using NFV," in *IEEE NetSoft*, Jun. 2019, pp. 228–232.
[10] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A survey on 5G usage scenarios and traffic models," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 905–929, 2020.
[11] D. Harutyunyan, R. Behravesh, and N. Slamnik-Kriještorac, "Cost-efficient placement and scaling of 5G core network and MEC-enabled application VNFs," in *IFIP/IEEE IM*, 2021, pp. 241–249.
[12] H. Farhady, H. Lee, and A. Nakao, "Software-defined networking: A survey," *Computer Networks*, vol. 81, pp. 79–95, 2015.
[13] J. d. J. Gil Herrera and J. F. Botero Vega, "Network functions virtualization: A survey," *IEEE Latin America Transactions*, vol. 14, no. 2, pp. 983–997, 2016.