Improving the quality of Federated Learning processes via Software Defined Networking

Ahmad Mahmod
ahmad.nawras.mahmod@gmail.com
DIMES-University of Calabria
Rende, Italy

Giuseppe Caliciuri
g.caliciuri@dimes.unical.it
DIMES-University of Calabria
Rende, Italy

Pasquale Pace
p.pace@dimes.unical.it
DIMES-University of Calabria
CNIT - National Inter-University Consortium for Telecommunications
Rende, Italy

Antonio Iera
antonio.iera@dimes.unical.it
DIMES-University of Calabria
CNIT - National Inter-University Consortium for Telecommunications
Rende, Italy

ABSTRACT

Federated Learning (FL) is rapidly gaining popularity as an effective cooperative and distributed approach, widely used by edge devices, to train machine learning models. Several aspects shall be managed to ensure a FL process that can more precisely match the QoS requirements of the applications that use it. The heterogeneity in the dataset available to each participant in the process, the variability in computational/memory capabilities, and the different availability of communication resources to connect the clients to the server are among the most critical. In this paper we will focus on the latter issue, less investigated in the literature, with particular reference to the case where the FL is used to support time-sensitive applications. Specifically, we will focus on studying the potential of an approach that leverages the Software-Defined Networking paradigm (SDN) to maintain the distributed learning process at high levels of effectiveness and efficiency even in the presence of edge client devices that may be delayed in delivering the result of their training due to the overload conditions experienced in the communication paths to the server. It will be shown, via a proof-of-concept performance evaluation campaign, how the proposed SDN support to the FL can guarantee significant overall reductions in process time at the cost of limited signaling overhead due to traffic to and from the controller.

CCS CONCEPTS

• **Networks** → **Network architectures**; **Network performance evaluation**; • **Computing methodologies** → **Machine learning**.

KEYWORDS

Federated Learning, SDN for AI, SDN-based FL orchestration.

NetAISys '23 , June 18, 2023, Helsinki, Finland

ACM ISBN 979-8-4007-0212-9/23/06…$15.00
https://doi.org/10.1145/3597062.3597281

ACM Reference Format:
Ahmad Mahmod, Giuseppe Caliciuri, Pasquale Pace, and Antonio Iera. 2023. Improving the quality of Federated Learning processes via Software Defined Networking. In *1st International Workshop on Networked AI Systems (NetAISys '23), June 18, 2023, Helsinki, Finland*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3597062.3597281

1 INTRODUCTION

Recently a plethora of applications are emerging that are based on the use of machine learning techniques to support increasingly sophisticated tasks to be carried out in different vertical markets [1], such as Industry 4.0, e-Health, Automotive, etc. The involvement in these services of different devices distributed on the edge of emerging network platforms and the growing need to maintain strict privacy on user data have made the so-called Federated Learning (FL) approach increasingly interesting [2]. Indeed, the latter is based on training a Machine Learning (ML) algorithm on multiple decentralized edge devices that maintain data samples locally, without exchanging them across the network.

Unfortunately, a problem that goes hand in hand with the numerous advantages of Federated Learning (and distributed learning techniques, more generally) is undoubtedly the fact that very heterogeneous devices in terms of computational capacity, amount of memory and communication resources can take part in this distributed learning process, some of which could also be Internet of Things (IoT) devices and therefore constrained by their nature. This aspect, if not adequately managed, has a significant impact on the quality of the FL process and on the consequent QoS of the applications that use it.

Many works in the literature aim to enhance the process through solutions conceived starting from a typical perspective of the research community on distributed learning. Some research proposes multi-criteria approaches to the selection of FL clients taking into account CPU, memory, energy, and time [3]. Other contributions also add as a selection criterion the guarantee of fairness towards clients [4], the reputation of the clients themselves [5], or aggregate clients based on the conditions of their resources [6]. Aspects related to improvements to the FL process to make it more communication-efficient are extensively examined, for example, in [7] and [8].

However, not enough attention has yet been paid in the above body of literature to addressing FL efficiency enhancement from the perspective of the networking research community, which implies the proposal of effective methods for managing the network resources available to the various devices involved in the exchange of learning model parameters during the different phases of the process. Two aspects related to networking in fact clearly arise that may contribute to severely compromising the quality of the FL process. Firstly, the different portions of the network that connect the various clients to the server can be congested by background traffic which could have a variable nature and in some phases of the client-server data exchange could cause excessive delays due to the temporary low capacity of some links used. Furthermore, the traffic itself from the clients and the server that repeatedly traverses the network during the various cycles of the FL process adds to the background traffic and thus causes further problems.

The logical consequence of the concurrence of the two phenomena described is that the quality of learning degrades with the same duration of the process. In fact, some clients involved, despite having excellent performance in terms of computational capacity and memory, complete the training tasks but their data arrives late to the server due to the bad conditions of the underlying network segments. This causes a greater delay in completing the number of rounds that allows to obtain the target quality of service, in terms of *Accuracy* and *Loss* to be offered to applications. Therefore, sufficient performance can often only be achieved in not always reasonable time intervals.

To meet the QoS needs for the new applications that are emerging in the 5G and future 6G scenarios, studies aimed at developing methods that allow obtaining high quality in the FL process in a short time are becoming urgent.

The approach that we propose to use in this work is in the wake of the emerging *"network for AI"* paradigm, different from and complementary to the more traditional *"AI for network"*. In fact, a new generation framework based on SDN is proposed in which the network supports the FL and is not, as widely studied so far [9], the FL to be used to support the policies of the SDN controllers.

Research into the role of SDN in supporting distributed learning has yielded interesting initial work. In the one in [10] it is proposed to create an overlay network where an SDN controller is used to manage auctions with bids from various potential customers. The one in [11], on the other hand, focuses on a resource slicing scheme in SDN IoT where the proposed scheme uses the SDN controller as manager of the virtualized infrastructure to orchestrate the optimal forwarding graph for each slice appropriately. Conversely, the authors of [12] propose using SDN to ensure that IoT gateway group data is received by multiple terminals and to manage it to address some of the problems of distributed ML.

Even in scenarios characterized by mobile devices, the authors of [13] propose that SDN controllers maintain QoS levels while supporting content placement in the presence of multiple compute and caching points. Finally, the recent survey work in [14] summarizes the main challenges, technologies and solutions in applying FL over SDN, also highlighting that applying FL over SDN provides more flexible and effective mechanisms for participant collaborations.

The contribution of our research is the first, as far as the authors know, which evaluates the potential of a framework in which a SDN network explicitly supports FL clients in dealing with issues related to the load conditions of the links on which their traffic flows with a view to improving the QoS of the entire FL process.

In particular, the ways in which SDN can support ML and define the reference framework are identified, and a first proof of concept is also provided through a performance evaluation campaign. The output of the latter demonstrates how the proposed approach can provide interesting improvements in the overall quality of service of the process in terms of reducing the time required to reach a good level of estimation for the ML performance metrics. Interesting indications are also provided on the impact of the control traffic generated by the SDN controller and on the most appropriate sizing of some parameters related to the operation of the latter.

The remainder of this article is organized as follows. Section 2 presents the architecture of the designed SDN-based FL framework; Section 3 describes all the technical choices for implementing the framework and related performance metrics; the experimental results are presented in Section 4; finally, the conclusions and future work are discussed in Section 5.

2 THE REFERENCE FRAMEWORK

Figure 1 schematically describes the reference system for our study. What we refer to is a framework to implement Federated Learning in which the main players of the FL process (FL Clients and FL Server) are supported at the Application layer by an FL Orchestrator and at the Network layer by a network resource Orchestrator (SDN Controller). The role of the former player is quite "traditional", namely the training of the local models by the heterogeneous devices and the creation of the global model to be redistributed to the various devices by the remote Server, respectively. In addition, the FL Server will dialogue with the FL Orchestrator, which in turn will carry out the entire set of control functions for the management of the traffic associated with the FL process.

Obviously, in performing the client selection procedure, and client re-selection in the case of a dynamic adaptive procedure, the FL Orchestrator will also continuously exchange data with the SDN controller. The main task of the latter will always be to guarantee the optimal path between the selected clients and the FL server,

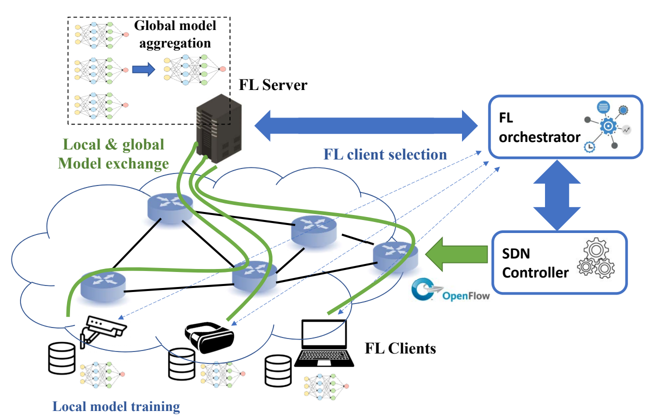

Figure 1: The proposed SDN-based framework for FL.

continuously monitoring and contrasting any dynamic overload of the network connections.

This will be implemented through SDN load balancing algorithms and dynamic path selection and will guarantee the continuous minimization of delays in the exchange of data related to the FL and therefore a higher quality of the process itself. In light of this, the main objective of our study is to understand the real potential of SDN in supporting the FL process and, mainly, to provide useful information on the delays obtainable for each path between Client and Server during each round. This information will be used to enable a continuous dynamic path selection which must lead to an increase in the overall quality of the whole learning process in terms of a reduced time to reach a certain preferred level of accuracy and loss.

This described in the present study is only a first step to understand to what extent the SDN controller policies are able to implement an effective control on the overall QoS of the FL service offered. The next step, subject to future research, may even be a wiser SDN-driven *client selection* procedure implemented by the FL Orchestrator which takes into account the communication resources available at any stage of the process in addition to the computational an memory resources of the clients.

It is worth stressing that there is ample room for future research also in the direction of using the SDN controller for the purpose of (i) intervening on the traffic load at the client side deriving from connections other than the one established with the FL server and active at the same time and (ii) implementing effective intrusion detection approaches and therefore counteracting malicious clients by intervening directly at the network level.

3 EVALUATING THE EFFECTIVENESS OF THE PROPOSED SDN-BASED FRAMEWORK

In this section, we discuss in detail the implemented testbed to validate the proposed SDN-based FL framework also presenting the main performance metrics we used to measure the goodness of the envisaged approach.

3.1 The Testbed for Performance Evaluation

The proposed SDN-based FL framework has been implemented by using the GNS3 [15] virtual environment emulator that offers an easy way to design, build, emulate, configure, test and troubleshoot virtual and real networks without the need for hardware. The network Controller is based on the OpenDaylight (ODL) architecture [16]. It is built on Model-Driven Software Engineering (MDSE) principles [17], where the network is implemented by using Beryllium 0.4.3 release. The model-driven in ODL approach provides a flexible method to manage the network using NETCONF and RESTCONF protocols and YANG modeling language. The RESTCONF protocol is a REST protocol that provides the possibility to manage, modify and retrieve data from the controller using HTTP protocol encoded with XML or JSON.

For the performance evaluation study reported in this paper, we refer to the hierarchical topology shown in Figure 2 made up of three levels of Open vSwitches (OvS) and providing multiple choices of paths between the server and the clients. The server is connected to OvS S2, so as to consider paths of different lengths connecting the server with the different clients, while the SDN controller is connected to both available first-level OvS S1 and S2. We consider three client categories according to the hardware characteristics in terms of memory and CPU (cat1 - low performance, cat 2 - medium performance and cat3 - high performance), as shown in Table 1. For the illustrated experiments we refer to a set of 8 clients whose category is randomly chosen, each connected to one switch of the third level of the topology.

Table 1: Client categories

Client Performance	HD Characteristics	
	RAM (GB)	♯ of core
Cat 1 - Low	1	1
Cat 2 - Medium	4	2
Cat 3 - High	8	4

To dynamically and continuously monitor the load on each client-server path, the controller periodically implements a quick and easy load balancing strategy, based on the Dijkstra Algorithm, that allows selecting the shortest paths from each client to the server with a cost represented by the percentage of utilization of the path's links. Obviously, in the future, smarter and more performing policies may be envisaged, but the optimization of this aspect is beyond the scope of this paper. Here, we intend to give an initial proof of concept of the effectiveness of the envisaged FL system supported by SDN.

To run the implemented simulation architecture, we used the powerful hardware platform HP Enterprise Proliant DL560 Gen10 equipped with 2 Intel Xeon-Gold 6225N processors (2.3GHz and 24 core) and 256GB of RAM.

To implement the federated learning strategy, the framework *Flower* [18] has been installed and properly configured on both client and server-side running the *FedAvg* algorithm [19]. In particular, the training process on the clients has been executed by using

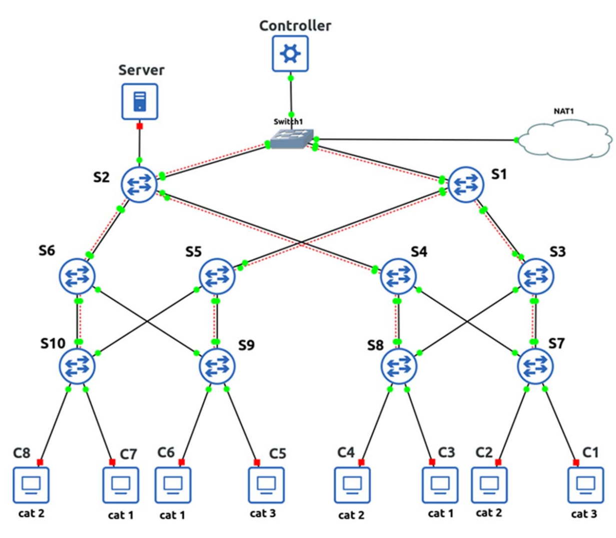

Figure 2: Network topology implemented in GNS3.

the well-known CIFAR-10 [20] dataset on two different neural networks. The first is the MobileNetV2 [21] neural network, usually used for image classification and having a size of 14MB with 3.5 million hyper-parameters. The second is DenseNet121 [22] neural network, having a size of 33MB with 8.1 million hyper-parameters. The idea is to test within the introduced framework the behavior of both a “lightweight” neural network and a “heavy” one in terms of load offered to the network during the different phases of data exchange provided by the FL mechanism implemented.

The clients participating in the federated learning process will be uniformly distributed in terms of their hardware characteristics [23]; moreover, the core network will be loaded by additional background data traffic in order to make the clients experiencing different and dynamically variable data traffic delays. 20 rounds of evolution for the federated learning process, each one consisting of 2 epochs, have been executed.

3.2 Proof-of Concept Performance Metrics

The adopted performance metrics are Accuracy and Loss, which are well-known and widely adopted metrics in machine learning. In particular, *Accuracy*, typically expressed as a percentage, is the count of predictions where the predicted value equals the true value and can provide a measure of the performance of a classification model. In contrast, *Loss* is a cost function that provides a more nuanced view of the model performance by considering the probabilities or uncertainty of a forecast based on how much the forecast varies from the true value. Unlike Accuracy, Loss is not a percentage, rather it is a sum of errors made for each sample in the training or validation sets. Both performance metrics are implemented in the Flower platform and calculated server-side at the end of each FL round.

4 RESULTS

In this section, we show the performance obtainable from the FL system implemented with and without SDN support, with a view to assessing the strengths of the proposed solution. In the proof-of-concept analysis shown, without losing generality, we use 6 of the 8 clients, 2 for each of the specified categories. The objective is to study the behavior of the system in a scenario of heterogeneous traffic in which the overall performance of the system is closely related both to the capabilities of the hardware device and to the network conditions to adequately manage the data traffic generated.

Figure 3 shows the accuracy of the federated learning process when using the MobileNetV2 neural network model to train clients with and without SDN support. The benefit of using SDN support is very clear and impressive in terms of time saved to reach an acceptable level of accuracy; in fact, when the SDN Controller uses a path update time of 5 seconds to reconfigure the paths to the server and change the flow tables accordingly, it is possible, for example, to achieve an accuracy level of about 65%, thus saving 43% of the time. The same evident increase in the quality of the FL process is also demonstrated by the trend of the Loss metric shown in figure 4, where it can be observed that SDN allows to reach the same low values in almost half the time.

The same experiments were repeated using a more complex and heavy neural network model for training (DenseNet121). This aims to discover any changes that occur in system behavior and also to verify the effectiveness of SDN support when the amount of data, generated by clients and delivered to the server to implement FL, increases significantly. Consequently, in this scenario, the links belonging to the core network will become more loaded with a consequent greater negative impact on the delay in the execution and completion of the FL process.

The figures 5 and 6 further confirm the intuition on the usefulness of using SDN to support the FL process and confirm the good performance deriving from this integration; in particular, it is possible to observe that the presence of SDN allows the system to work even better under more intense traffic load conditions, reaching the accuracy value of 75% and the loss value of 0.70 saving 55% of time.

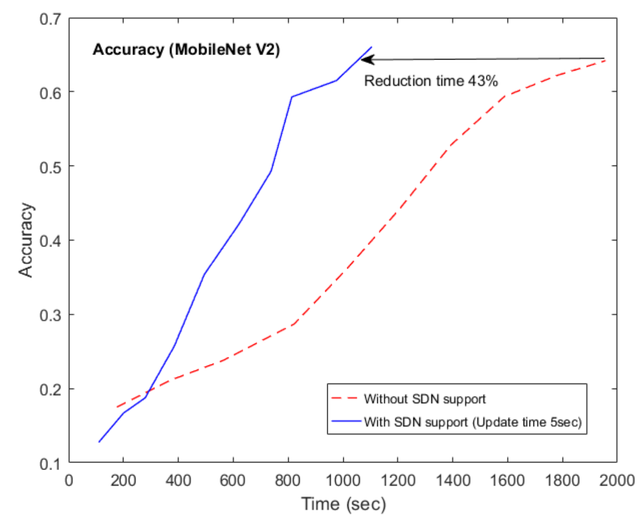

Figure 3: Accuracy of the FL process using MobileNetV2.

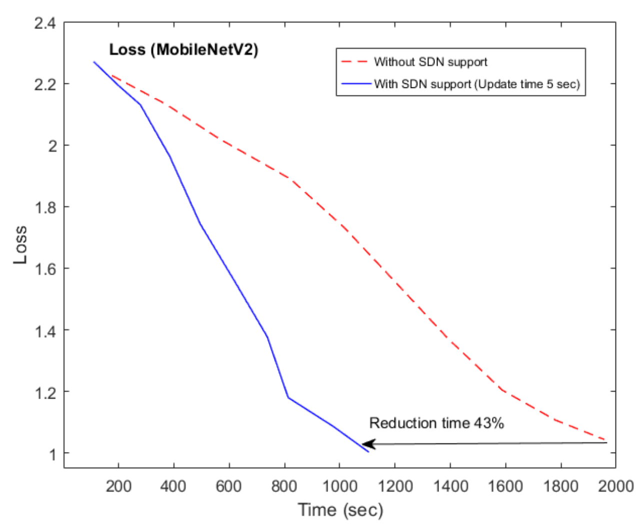

Figure 4: Loss of the FL process using MobileNetV2.

Finally, we evaluate the *"price to pay"* in terms of the amount of additional control data packets traveling to/from the SDN controller. This metric is highly relevant because it can help understand the actual impact of network resource orchestration on the proposed SDN-based FL framework.

Figure 7 shows the average amount of control data packets exchanged by varying the time interval (from 1 second to 5 seconds) for the controller to reconfigure the client-server data paths. As expected, the overhead, in terms of control data packets, increases from 425KB/s when the update time is 5 seconds, up to 916KB/s when updating at a higher rate (*that is,* every second).

Differently, as a consequence of the variation of the update time interval, the duration of the distributed training process of the FL has a minimum. Specifically, the best refresh time value that minimizes the training time of our 20-round sample process is 3

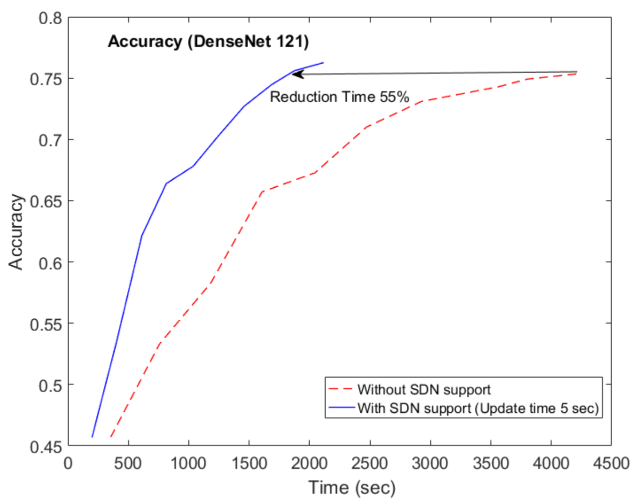

Figure 5: Accuracy of the FL process using DenseNet121.

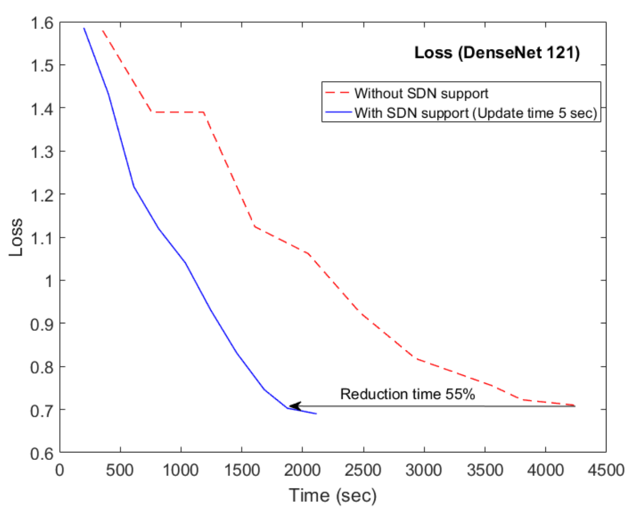

Figure 6: Loss of the FL process using DenseNet121.

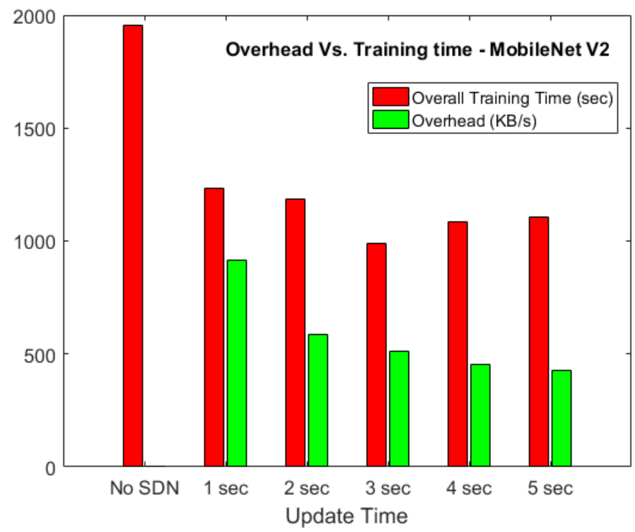

Figure 7: Overhead due to the SDN support Vs. Time to make the training process using MobileNetV2.

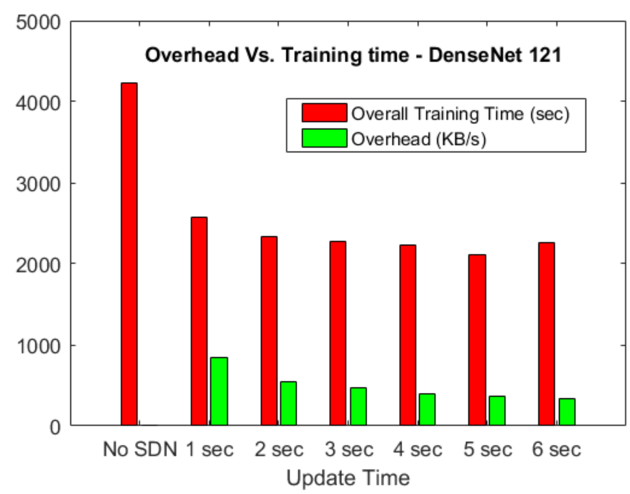

Figure 8: Overhead due to the SDN support Vs. Time to make the training process using DenseNet121.

seconds, although the corresponding additional signaling overhead of 510 KB/s isn't the minimum. This too is a widely awaited *result* since it must be considered that, if on the one hand, the increase in the update frequency of the core network path corresponds to an increase in the signaling load, on the other hand, a higher rate of updates brings with it better management of network resources and therefore a reduction in the delay in completing the learning process.

As a result, there is an inherent trade-off that leads to determining an optimal refresh rate value that is always observable. At this optimal refresh rate value (3 seconds), the performance gain of an FL process enhanced by SDN compared to one without SDN goes from 43%, already shown in the figure 3 for the value of 5 seconds, to a value equal to 49%.

Figure 8 further validates the previous analysis by referring to a different use case scenario with the heavier DenseNet121 neural network model, where the best compromise for a longer update time is reached (*i.e.*, 5 seconds) while the overhead due to the control data to provide SDN support is comparable to that obtained using the lightest training model, thus proving that the control load generated by SDN is almost independent of the FL training model used.

5 CONCLUSION

This paper addressed a problem of extreme interest in the current landscape of next-generation network infrastructures, in which a plethora of intelligent services that exploit distributed learning paradigms will presumably be delivered. Starting from the emerging "network for AI" paradigm, a communication architecture was proposed designed to support the quality of the federated learning process with particular attention to reducing training times at the same performance level. A first set of tests for performance evaluations was carried out to provide a proof of concept for the proposal. The results obtained testify to the effectiveness in guaranteeing very high-quality levels of the federated learning process. An activity planned for the future is the study and development of sophisticated client selection techniques, in which SDN can play a key role, which considers at the same time the storage and computing capabilities of the client devices and the network performance.

6 ACKNOWLEDGMENTS

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE00000001 - program “RESTART”).

REFERENCES

[1] S. K. Sood, K. S. Rawat, and D. Kumar. 2022. A visual review of artificial intelligence and Industry 4.0 in healthcare. Computers and Electrical Engineering. 2022 Jul;101:107948. doi: 10.1016/j.compeleceng.2022.107948.
[2] M. Aledhari, R. Razzak, R. M. Parizi and F. Saeed. 2020. Federated Learning: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Access, vol. 8, pp. 140699-140725, 2020, doi: 10.1109/ACCESS.2020.3013541.
[3] S. Abdulrahman, H. Tout, A. Mourad, and C. Talhi. 2021. FedMCCS: Multicriteria Client Selection Model for Optimal IoT Federated Learning. IEEE Internet of Things Journal, 2021, vol. 8, no. 6, pp. 4723-4735.
[4] T. Huang, W. Lin, W. Wu, L. He, K. Li, and A. Y.Zomaya. 2020. An efficiency-boosting client selection scheme for federated learning with fairness guarantee. IEEE Transactions on Parallel and Distributed Systems, 2020, vol. 32, no.7, pp. 1552-1564.
[5] Y. Wang and B. Kantarci. 2020. A novel reputation-aware client selection scheme for federated learning within mobile environments. IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2020, pp. 1-6.
[6] T. Nishio and R. Yonetani. 2019. Client selection for federated learning with heterogeneous resources in mobile edge. IEEE international conference on communications (ICC), 2019, pp. 1-7.
[7] O. Shahid, S. Pouriyeh, R. M. Parizi, Q. Z. Sheng, G. Srivastava, and L. Zhao. 2021. Communication efficiency in federated learning: Achievements and challenges. arXiv preprint - arXiv:2107.10996, 2021.
[8] W. Luping, W. Wei, and L. I. Bo. CMFL: Mitigating communication overhead for federated learning. IEEE 39th Int. Conf on Distributed Computing Systems (ICDCS), 2019, pp. 954-964.
[9] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu. 2018. A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges. IEEE Communications Surveys & Tutorials, 2018, vol. 21, no.1, pp. 393-430.
[10] E. Seo, D. Niyato, and E. Elmroth. 2021. Auction-based Federated Learning using Software-defined Networking for resource efficiency. 17th IEEE International Conference on Network and Service Management (CNSM), 2021, pp. 42-48.
[11] P. Tam, S. Math, and S. Kim. 2021. Efficient resource slicing scheme for optimizing federated learning communications in software-defined IoT networks. Journal of Internet Computing and Services, 2021, vol. 22, no. 5, pp. 27-33.
[12] J. Mills, J. HU, and G. Min. 2019. Communication-efficient federated learning for wireless edge intelligence in IoT. *IEEE Internet of Things Journal*, 2019, vol. 7, no. 7, pp. 5986-5994.
[13] V. Balasubramanian, M. Aloqaily, M. Reisslein, and A. Scaglione. 2021. Intelligent resource management at the edge for ubiquitous IoT: an SDN-based federated learning approach. IEEE Network, 2021, vol. 35, no.5, pp. 114-121.
[14] X. Ma, L. Liao, Z. Li, R.X. Lai, and M. Zhang. 2022. Applying Federated Learning in Software-Defined Networks: A Survey. Symmetry 2022, 14, 195. https://doi.org/10.3390/sym14020195
[15] GNS3 software, freely available at https://www.gns3.com/
[16] OpenDaylight web site - https://www.opendaylight.org/
[17] J. Medved, R. Varga, A. Tkacik, and K. Gray. 2014. OpenDaylight: Towards a model-driven SDN controller architecture. IEEE WoWMoM, 2014, pp. 1–6.
[18] D. J. Beutel et al. 2020. “Flower: A Friendly Federated Learning Research Framework,”. arXiv:2007.14390v5, 2020, available at https://flower.dev/
[19] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. Int. Conf. on Artificial Intelligence and Statistics (AISTATS), 2017, pp. 20-22.
[20] A. Krizhevsky, V. Nair, and G. Hinton. 2005. The Cifar-10 Dataset. Online, 2005, available at http://www.cs.toronto.edu/~kriz/cifar.html
[21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520.
[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. 2017. Densely Connected Convolutional Networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261-2269.
[23] Y. G. Kim and C. J. Wu. 2021. AutoFL: Enabling heterogeneity-aware energy efficient federated learning. MICRO 2021 - 54th Annual IEEE/ACM International Symposium on Microarchitecture, 2021, pp. 183-198.