Evaluation of Joint Controller Placement for Latency and Reliability-Aware Control Plane

Kurdman Abdulrahman Rasol
Department of Computer Architecture
Universitat Politècnica de Catalunya (UPC)
Barcelona, Catalonia, Spain
kurdman.rasol@upc.edu

Jordi Domingo-Pascual
Department of Computer Architecture
Universitat Politècnica de Catalunya (UPC)
Barcelona, Catalonia, Spain
jordi.domingo@ac.upc.edu

Abstract—**The separation of the forwarding and control planes in software-defined networking provides flexibility for network management. The Controller Placement Problem (CPP) is an important issue affecting network performance.**

This paper presents an evaluation of the Joint Latency and Reliability-aware Controller Placement (LRCP) optimization model. LRCP provides network administrators with flexible choices to simultaneously achieve a trade-off between the switch-to-controller latency and the controller-to-controller latency, including the reliability aspect using alternative backup paths.

Control plane latency (CPL) is used as the evaluation metric and it is defined as the sum of average switch-to-controller latency and the average inter-controller latency. For each optimal placement in the network, the control plane latency using the real latencies of the real network topology is computed.

Results from the control plane latency metric show how the number and location of controllers influence the reliability of the network. In the event of a single link failure, the real CPL for LRCP placements is computed and assesses how good the LRCP placements are. The CPL metric is used to compare with other models using latency and reliability metrics.

Index Terms—**Software Defined Networking, Control Plane Latency, Multiple Controller Placements, Single-Link Failure, Control Plane Reliability.**

I. INTRODUCTION

The fundamental idea behind Software Defined Networking (SDN) architectures is that it allows programmability to configure the network by splitting the control plane (the control logic) from the data plane (the forwarding plane) and simplifying the functions of the control plane. SDN facilitates a logically centralized control plane that enables the administrators to build more simple, customizable, programmable, and manageable networks [1], [2].

The data plane forwards data packets according to decisions made by the controllers. The control plane is accountable for generating packet forwarding rules for the network devices such as switches and routers. Both data and control packets share the same networking infrastructure. The controller placement is an essential design decision that influences the overall network performance in the SDN network [3].

As shown in Figure 1, the control plane comprises two types of communication. The first one is the control traffic exchanged between switches and their assigned controller, that is, the controller-to-switch traffic (CS). The second type of communication is the control traffic exchanged among controllers (CC). Controller-to-controller (CC) traffic is needed to synchronize the shared data structures and guarantee a consistent global view of the network, and it maintains a centralized view of the network state [4], [5].

The East/Westbound Application Programming Interface (API) is used to communicate between controllers (CC) [6]. On the other hand, the Southbound API (i.e., OpenFlow) is used to communicate between controllers and switches (CS) [7].

In order to solve the controller placement problem, most of the related literature only considered the latency of the communication between controllers and switches. In contrast, the (CC) latency is not sufficiently considered within the evaluation of control plane performance. Defining a balance between these two types of control traffic (CS and CC) may influence the overall performance of the network.

LRCP addresses the reliability aspect by considering the effect of a single link failure [8]. This paper proposes the Control Plane Latency (CPL) metric to evaluate controller placements provided by LRCP and assesses how good the LRCP placements are in a real controller deployment.

The structure of the paper is organized as follows. Section II describes the state of the art of the controller placement problem in software-defined networks. Section III presents a summary of the proposal we aim to evaluate. Section IV presents the performance evaluation. Finally, Section V describes the main discussion and conclusion of the achieved results.

II. RELATED WORK

Previous studies discussed a variety of concerns related to the Controller Placement Problem (CPP) in Software Defined Networking (SDN). Extensive surveys may be found in [9] and [10].

The Controller Placement Problem (CPP) was first presented by Heller et al. [11]. The purpose of the study was to determine the minimum number of controllers needed and where each controller should be placed to provide low latencies between the switches and their corresponding controllers. The performance of the CPP is evaluated on the Internet2

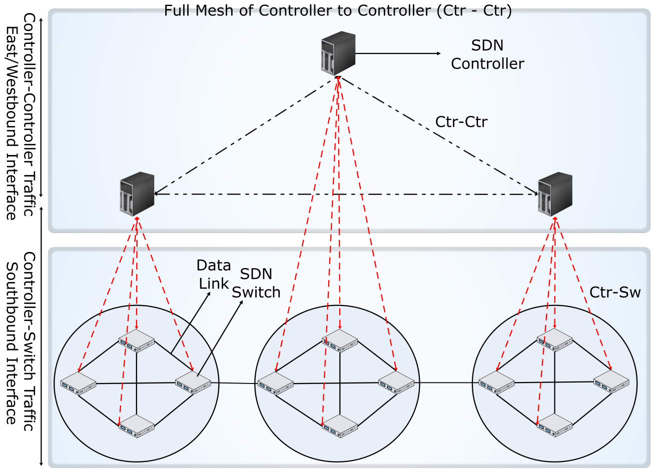

Fig. 1. SDN Control Plane Architecture

OS3E real network topology [12]. In addition, the K-means algorithm is used to minimize the average latency between switches and their assigned controllers [13]. Another research is the K-center algorithm to minimize the maximum latency between controllers and switches [14].

The authors of the Joint Placement Latency Optimization (LCP) [15] defined an accumulated latency to solve CPP, which takes into consideration both the latency between controllers and switches and inter-controller communication latency. The optimization objective is to minimize the accumulated latency and the optimal number of controllers.

Reliability is also a crucial concern for controller placements. Hu et al. implement a metric known as the expected percentage of control path loss for a failed network component; this is used to characterize the reliability of SDN control networks. They present a heuristic *l-w-* greedy algorithm to evaluate the trade-off between reliability and latency. The aim of this optimization is to minimize the expected percentage of control path loss. In this case, the reliability metric is defined as the expected percentage of control path loss, where the control path loss is the number of broken control paths due to network failures [16], [17].

Yuqi Fan et al. propose a Latency-Aware Reliable Controller Placement Algorithm (LARC) [18]. The aim of this algorithm is to minimize the average latency between all the switches and their corresponding controllers when a single-link failure occurs. The latency of each path includes the primary path latency and an average of the corresponding possible backup paths under a single-link failure condition.

Yuqi Fan et al. propose an efficient Reliability-aware Controller Placement (RCP) algorithm [19] for multiple controller placements. The RCP algorithm splits the network into multiple subnetworks and places a controller in each subnetwork. A local search algorithm determines the controller locations and maps the relationship between switches and controllers. The simulation results show that the proposed RCP algorithm can effectively reduce the latency of the primary and backup paths.

AK Singh et al. propose a Varna-based optimization (VBO) [20] for a reliable CPP to ensure that it minimizes the total average latency. Results show that the proposed VBO algorithm gives better performance compared to other efficient heuristic algorithms such as particle swarm optimization (PSO) [21], teacher learning-based optimization (TLBO) [22], Jaya algorithms [23] based solution for RCPP.

The Joint Latency and Reliability-aware Controller Placement (LRCP) is proposed in [8]. The optimization model for controller placement considers both latencies between switches to their controllers and the latency between controllers at the same time. This is referred to as joint latency. Furthermore, the authors consider the reliability impact of a single failure. This is referred to as reliability-aware.

In this paper, the Control Plane Latency (CPL) metric is used to evaluate Joint Latency and Reliability-aware Controller Placement (LRCP) optimization.

III. DESCRIPTION OF THE LRCP OPTIMIZATION

LRCP is an extension of the models CPP [11], which presents the basic controller placement problem, and LCP [15], which defines the joint latency as an optimization function. Furthermore, it includes the reliability aspect following the LARC model for a single-link failure [18].

LRCP is defined to reduce the accumulated latency by integrating the two sub-objectives of reliability and joint latency. Latency is approximated by the distance between the nodes. The optimization of LRCP is formulated as a mixed-integer linear programming (MILP) under both latency and reliability constraints. It must be noted that the LRCP optimization is done off-line and thence its complexity is not a drawback.

A summary of the original proposal is presented in this section.

Reliability-Latency (RL) is defined as the weighted sum of the primary and the backup path latencies (1). The primary path latency (PPL) between two nodes is based on the shortest path routing algorithm. The backup path latency (BPL) is the average latency of all the possible alternative paths when a single-link failure in the primary path occurs. To do the computation of the average backup path latency, a single-link along the primary path is removed from the network and a new shortest path between the nodes is calculated. The Reliability-Latency between a pair of nodes is defined as:

$$\mathrm{RL} = \theta * \mathrm{PPL} + (1 - \theta) * \mathrm{BPL} \tag{1}$$

Parameter θ assigns the weight to the real latency (PPL) between two nodes and the additional average latency in case of a failure (BPL).

The controller to switch latency (CS) is the average latency between a controller and all its associated switches. The average latency of the inter-controller communications (CC) is the average latency of all possible paths between pairs of controllers. The joint latency is defined as the weighted sum of the average controller to switch latency (CS) and the average inter-controller latency (CC). This is referred to as joint latency (2), where λ is the weight between the two latencies. Both CS and CC use the Reliability-Latency (RL) in order to include the reliability factor. CS and CC are functions of RL. The joint latency is the optimization function and it is defined as:

$$\text{joint latency} = \lambda * \text{CS(RL)} + (1 - \lambda) * \text{CC(RL)} \tag{2}$$

Both parameters of the LRCP optimization (λ and θ) play a key role in controller placements. The optimal placement may vary depending on either λ or θ. A particular placement is defined by the location of the controllers and the set of switches associated with each controller.

Once the control plane application is decided, λ is used to find a trade-off for the performance of the control plane application. For instance, when λ is close to one it means that the control plane application needs very fast communication between switches and their controller, and the inter-controller latency is not a key factor for the performance. The optimal performance of the control plane depends on a balance between the latencies CS and CC.

Parameter θ allows finding better controller placements considering the probability of link failures. A greater probability of link failure may need lower values of θ, so that the alternative backup paths have a larger weight in the placement decision. The case when $\theta = 1$ considers the primary path latency (PPL) only by ignoring backup path latency (BPL). We may see θ as a parameter to find preventive placements that may be affected less than the original one by single-link failures. It is interesting and useful to make an assessment of the control plane performance of these preventive placements.

The major functions and details of the LRCP optimization model and the significant results can be found in [8].

IV. PERFORMANCE EVALUATION

As LRCP is evaluated on the Internet2 OS3E network topology, the performance evaluation is done using the same topology. OS3E contains 34 nodes and 42 edges. The Python interface of the GUROBI optimization is used to solve the Mixed Integer Linear Programming (MILP) model and to validate the results presented in LRCP.

The goal of the evaluation is to assess the goodness of the optimal placements provided by LRCP in a real deployment. For each value of λ and θ, LRCP gives the optimal placement (location of the controllers and the set of switches associated with each controller). When a particular placement is deployed in the network, the important metric is the overall control plane latency. Thus, the proposed metric for the evaluation is the Control Plane Latency (CPL), defined as the sum of the average switch to controller latency (CS) and the average inter-controller latency (CC).

For each optimal placement found, CPL is computed as:

$$CPL = \text{CS} + \text{CC} \tag{3}$$

Control plane latency (CPL) is computed using the actual latencies of the real network topology. This corresponds to the primary path latency (PPL) when no failures occur, and it is taken as the reference value.

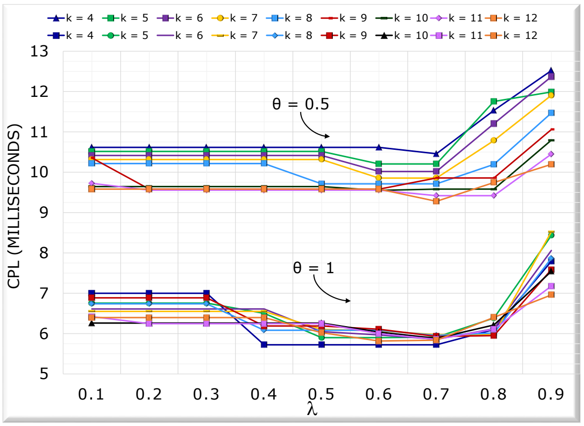

Fig. 2. Control plane latency (CPL) without a single-link failure

In order to include a degree of reliability, in case a link failure event occurs, the preventive placement considers this event and the average backup path latency (BPL) is added in the optimization. This provides a slightly larger value of the control plane latency (CPL) value with respect to the average of the primary path latency only (PPT). For the evaluation, two cases are differentiated. The first one does not include any link failure and it intends to assess the real Control Plane Latency (CPL) for the LRCP preventive placements which include a weight for the BPL (that is θ less than 1). The second case computes the real CPL when a single-link failure occurs and assesses how good are the LRCP preventive placements which include the case of a single-link failure. Intuitively, when a single-link failure occurs, an LRCP preventive placement obtained for θ less than 1 should behave better than the one obtained for PPL only ($\theta = 1$). The goal is to quantify their performance.

A. *Control Plane Latency Evaluation Without Link Failure*

Figure 2 illustrates the impact of the number of controllers (k) on the control plane latency (CPL). The case in which $\theta = 1$ with different values of λ corresponds to the average primary path latency (PPL). For $\theta < 1$, the placement considers the average backup path latency (BPL) and implies slightly different controller locations and associated switches.

As expected, it provides a higher Control Plane Latency, because it does not correspond with the optimal placement when no failures occur. The case with $\theta = 0.5$, shown in Figure 2 indicates that for for five controllers (k = 5) and $\lambda = 0.5$, the average control plane latency for $\theta = 1$ is 5.9 ms while considering the cost of a single-link failure with $\theta = 0.5$ the preventive placement gives an average control plane latency of 10.5 ms. This is an increase of about 78%. Unless there is a high probability of link failures, higher values for θ should be used; otherwise, the sought protection is counterproductive.

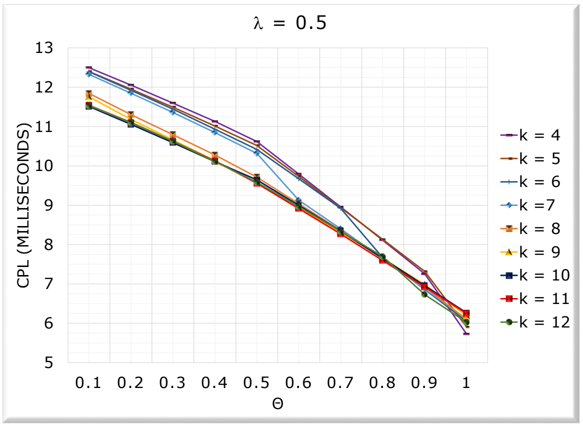

Fig. 3. Control plane latency (CPL) and $\lambda = 0.5$

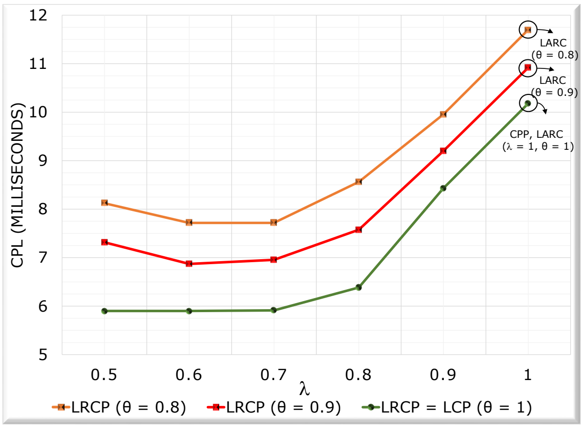

Fig. 4. Control plane latency (CPL) for five controllers: LRCP, LARC, LCP, and CPP

Useful values for θ are those close to 1. Figure 3 shows the relationship between the control plane latency and the number of controllers by varying the weights of θ when $\lambda = 0.5$. It is interesting to note that the control plane latency grows linearly by increasing the weight of backup path latency (decreasing values of θ) since the average backup path latency (BPL) provides a larger value of the control plane latency with respect to the value for the average primary path latency (PPT).

When five controllers are deployed ($k = 5$) and $\lambda = 0.5$, the control plane latency decreases from 8.1 ms to 5.9 ms when θ ranges between 0.8 and 1. This means an increase of 38% in the control plane latency with respect to the primary paths if there is no failure. For $\lambda = 0.6$ (not shown in the figure), the control plane latency is 7.7 ms with respect to 5.9 ms; that is an increment of 31%.

It can be seen from the Figure 3 that giving more weight to the backup path (lower values of θ) produces preventive placements with an increase in the control plane latency. As a conclusion, values of less than $\theta = 0.8$ are not useful because the CPL increases more than 30% if there are no failures.

In order to compare LRCP evaluation with previous proposals and as a validation of the study, Figure 4 presents the average control plane latency for the case of five controllers ($k = 5$). For $\theta = 1$, using different values of λ, the result corresponds with the basic joint placement latency optimization (LCP) [15] and LRCP [8].

The LCP optimization model minimizes the joint latency and the number of network controllers providing placements with a balance between CS and CC latencies. However, the LCP optimization model does not consider the probability of a link failure.

In particular, the case when $\lambda = 1$ corresponds to the original optimization of the Controller Placement Problem (CPP), and for $\theta = 1$, it corresponds with the Latency-Aware Reliable Controller placement algorithm (LARC).

The CPP places the controllers to minimize the latency between switches and controllers only without considering the failure scenario [11]. The LARC algorithm aims to minimize the accumulated latency by integrating the two sub-objectives: the average of the primary path latency and the backup path latency into one objective between switches and controllers when a single-link failure occurs [18].

As a last example of the cost associated with using preventive placements, when five controllers are placed and the value of λ is set to 0.5, it can be observed in Figure 4 that the control plane latency with $\theta = 1$ is 5.9 ms whereas it is 7.3 ms in case of $\theta = 0.9$. This means an increase of 24%. Depending on the probability of link failures, this extra latency might be tolerated.

B. Control Plane Latency Evaluation for Single-Link Failure (SLF)

In this section, the Control Plane Latency (CPL) is computed for LRCP placements considering the effect of single-link failure. A link failure may affect CS paths, CC paths, both, for one controller or for several controllers at the same time. To compute the average CPL, for each placement (location of the controllers and the set of assigned switches to each controller), one link i is removed and CPL(i) is computed. The final value of the CPL metric is the average of CPL(i) for all the links being removed one at a time.

Figure 5 presents the control plane latency with a single-link failure for five controllers ($k = 5$). We compare the results shown in Figures 4 and 5. For the placements including the primary latency (PPL) only ($\theta = 1$), the average control plane latency for $\lambda = 0.6$ is 5.9 ms (see Fig. 4) and 6.4 ms in the case of a single- link failure (see Fig. 5). This is an increase of about 10%. For $\theta = 0.8$ and $\lambda = 0.6$ the CPL is 7.7 ms when there are no failures (Figure 4) and 6.4 ms with a single-link failure. This means a relative decrease of about 14%.

This confirms that for low link failure probabilities using values of $0.8 < \theta < 1$, provide reliable placements with a reasonable increase of about 10% in the CPL respecting the reference values (when no links fail). But this increase is

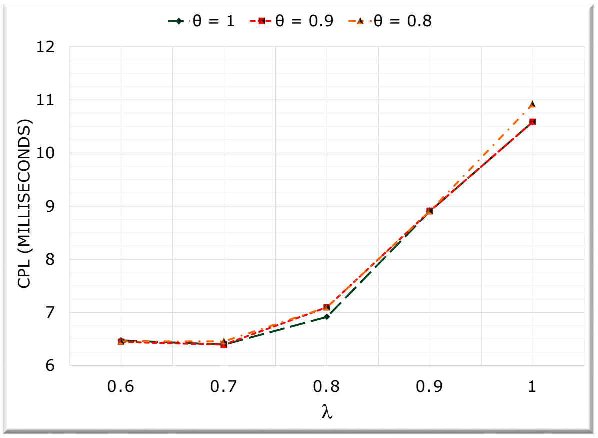

Fig. 5. Control plane latency (CPL) with a single-link failure when $k = 5$

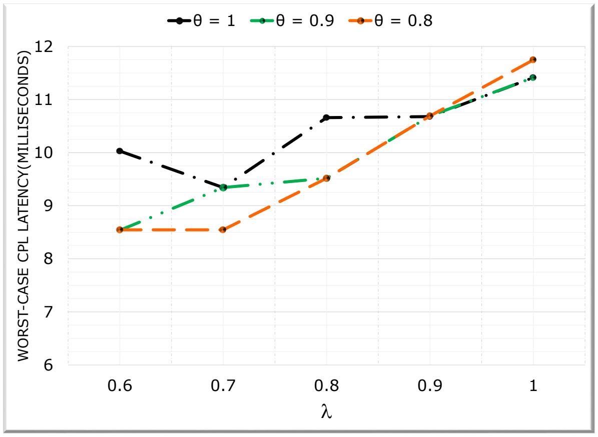

Fig. 6. Worst-case CPL latency with single-link failure when $k = 5$

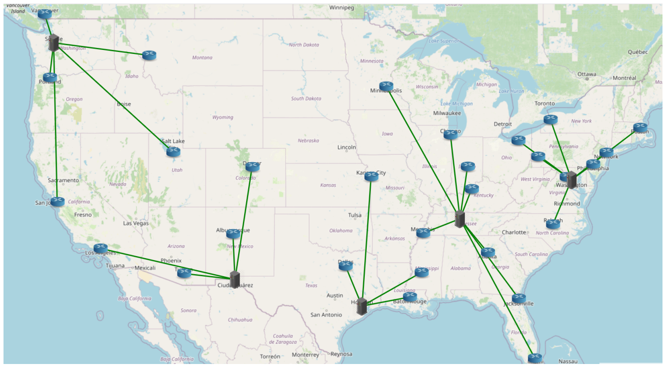

Fig. 7. Optimal controller locations and sets of switches associated with each controller ($\theta = 1$, $\lambda = 1$)

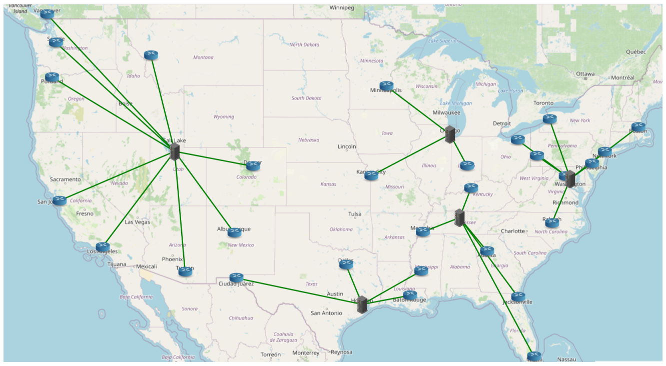

Fig. 8. Optimal controller locations and sets of switches associated with each controller ($\theta = 0.9$, $\lambda = 0.9$)

compensated when a link fails with a relative decrease of about 14%.

As mentioned before, in the case of a single-link failure, the computed CPL is an average latency. Then, it is relevant to investigate the worst-case when a link fails. Figure 6 presents the worst-case control plane latency when a single-link failure occurs. For instance, for $k = 5$ when θ ranges from 0.8 to 1 and, the control plane latency for $\lambda = 0.8$ ranges from 9.5 ms to 10.6 ms, and for $\lambda = 0.6$ ranges from 8.5 ms to 10.0 ms. That is, for $\lambda = 0.8$ the worst-case CPL increases by 12% and for $\lambda = 0.6$ it increases by 17%.

It is worth mentioning that two nodes have exactly one link in the Internet2 OS3E network: Miami and Vancouver. These two switches could get disconnected from their associated controller if the corresponding link fails and there is no possible backup path.

Table I, II and III show the details of further results. The same colour means identical placements (controller locations and sets of switches associated with each controller) and, thus, the same control plane latency (CPL). As an example, Figure 7 and 8 show how different values of θ and λ influence the placement of controllers that manage the control plane, and the distribution of all switches assigned to each controller. In LRCP, the parameter λ is used to seek a kind of balance between CS and CC latencies to optimize the performance of a given control plane application. Parameter θ is used to look for a preventive placement in case a single-link failure occurs. The table contains results for values of θ between 0.8 and 1 because, from the previous evaluation results, these are the useful values for a real deployment. For a given placement, fixed by the combination of λ and θ, CPL is the average control plane latency during a normal operation (without failures), while CPL(sf) is the average control plane latency when a single-link failure occurs. The CPL-CPL(sf) is the relative increase of CPL (sf) with respect to CPL. As it may be observed in Table I, preventive placements with $\theta = 0.8$ imply slightly higher values of CPL while smaller differences for CPL(sf). The table also includes the worst-case value of CPL and its variance for each placement. It is up to the network manager to choose the appropriate values of the parameters and use a preventive placement instead of the one without considering backup paths.

			θ = 1		
Weights of λ	λ = 0.6	λ = 0.7	λ = 0.8	λ = 0.9	λ = 1
CPL(sf)	6.47	6.39	6.92	8.90	10.58
CPL	5.90	5.91	6.39	8.43	10.18
CPL-CPL(sf)	0.57	0.48	0.53	0.47	0.40
CPL-CPL(sf)	10%	8%	8%	6%	4%
Min CPL(sf)	5.90	5.91	6.39	8.43	10.18
wc CPL(sf)	10.03	9.34	10.66	10.68	11.41
ratio wc CPL / min CPL	70%	58%	67%	27%	12%
Var CPL	1.03	0.80	0.91	0.53	0.34

TABLE I

COMPARISON OF THE CONTROL PLANE LATENCY WITH AND WITHOUT SLF FOR 5 CONTROLLERS WHEN θ = 1

			θ = 0.9		
Weights of λ	λ = 0.6	λ = 0.7	λ = 0.8	λ = 0.9	λ = 1
CPL(sf)	6.44	6.39	7.10	8.90	10.58
CPL	6.02	5.91	6.58	8.44	10.18
CPL-CPL(sf)	0.43	0.48	0.51	0.46	0.40
CPL-CPL(sf)	7%	8%	8%	5%	4%
Min CPL(sf)	6.02	5.91	6.58	8.44	10.18
wc CPL(sf)	8.54	9.34	9.52	10.69	11.41
ratio wc CPL / min CPL	42%	58%	45%	27%	12%
Var CPL	0.65	0.80	0.81	0.53	0.34

TABLE II

COMPARISON OF THE CONTROL PLANE LATENCY WITH AND WITHOUT SLF FOR 5 CONTROLLERS WHEN θ = 0.9

			θ = 0.8		
Weights of λ	λ = 0.6	λ = 0.7	λ = 0.8	λ = 0.9	λ = 1
CPL(sf)	6.45	6.45	7.10	8.90	10.92
CPL	6.02	6.02	6.58	8.44	10.51
CPL-CPL(sf)	0.07	0.07	0.08	0.05	0.04
CPL-CPL(sf)	43%	43%	51%	46%	40%
Min CPL(sf)	6.02	6.02	6.58	8.44	10.51
wc CPL(sf)	8.55	8.55	9.52	10.69	11.75
ratio wc CPL / min CPL	42%	42%	45%	27%	12%
Var CPL	0.65	0.65	0.81	0.53	0.34

TABLE III

COMPARISON OF THE CONTROL PLANE LATENCY WITH AND WITHOUT SLF FOR 5 CONTROLLERS WHEN θ = 0.8

V. CONCLUSION

In this paper, the average control plane latency (CPL) metric is used to evaluate the Joint Latency and Reliability-aware Controller Placement (LRCP) optimization model. CPL is defined as the sum of the average switch to controller latency (CS) and the average inter-controller latency (CC). LRCP provides optimal placements considering at the same time a kind of balance between CS and CC latencies, and a reliability tradeoff with preventive placements in case a single-link failure occurs. Parameters λ and θ are used, respectively, for each goal. Preventive placements consider the corresponding backup paths.

The main contribution of the paper is the assessment of the goodness of LRCP preventive placements in a real deployment. In order to quantify the evaluation, the CPL metric is used. From the evaluation results presented in LRCP [8], the reference values selected for the evaluation are five controllers (k = 5) and values of λ from 0.6 to 1.

From the results obtained, we may conclude that preventive placements for values of $\theta < 0.8$ are not advisable because they introduce too much extra latency in the control plane in a normal operation without failures. On the other hand, placements with $\theta >= 0.8$ give a good tradeoff between the added latency while in normal operation and when a single-link failure occurs.

In summary, for low link failure probabilities using values $0.8 < \theta < 1$ provide reliable preventive placements with a reasonable increase of the CPL respect the reference values (when no links fail), and this increase is compensated when a link fails with a relative decrease of CPL.

VI. ACKNOWLEDGEMENTS

This publication is part of the Spanish I+D+i project TRAINER-A (ref. PID2020-118011GB-C21), funded by MCIN/AEI/10.13039/501100011033. This work has been also partially funded by the Spanish Ministry of Economy and Competitiveness, under contract TEC 2017-90034-C2-1-R (ALLIANCE).

REFERENCES

[1] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," in Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, Jan. 2015, doi: 10.1109/JPROC.2014.2371999.

[2] Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turletti. A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Communications Surveys and Tutorials, Volume: 16, Issue: 3, Page(s): 1617 – 1634, Third Quarter 2014.

[3] Singh, A. K., and Srivastava, S. (2018). A survey and classification of controller placement problem in SDN. International Journal of Network Management, 28(3). https://doi.org/10.1002/nem.2018

[4] Zhang, T., Bianco, A., and Giaccone, P. (2017). The role of inter-controller traffic in SDN controllers placement. 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2016, 87–92. https://doi.org/10.1109/NFV-SDN.2016.7919481

[5] Oktian, Y. E., Lee, S. G., Lee, H. J., and Lam, J. H. (2017). Distributed SDN controller system: A survey on design choice. Computer Networks, 121, 100–111. https://doi.org/10.1016/j.comnet.2017.04.038

[6] Latif, Z., Sharif, K., Li, F., Karim, M. M., Biswas, S., and Wang, Y. (2020). A comprehensive survey of interface protocols for software defined networks. Journal of Network and Computer Applications, 156, 1–30. https://doi.org/10.1016/j.jnca.2020.102563

[7] Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2015). OpenFlow: Enabling Innovation in Campus NetworksSoftware-defined networking: A comprehensive survey. Proceedings of the IEEE, 103(1), 14–76. Retrieved from http://ccr.sigcomm.org/online/files/p69-v38n2n-mckeown.pdf

[8] K. A. R. Rasol and J. Domingo-Pascual, "Joint Latency and Reliability-Aware Controller Placement," 2021 International Conference on Information Networking (ICOIN), 2021, pp. 197-202, doi:10.1109/ICOIN50884.2021.9333864.

[9] J. Lu, Z. Zhang, T. Hu, P. Yi and J. Lan, "A Survey of Controller Placement Problem in Software-Defined Networking," in IEEE Access, vol. 7, pp. 24290-24307, 2019, doi: 10.1109/ACCESS.2019.2893283.

[10] T. Hu, Z. Guo, P. Yi, T. Baker and J. Lan, "Multi-controller Based Software-Defined Networking: A Survey," in IEEE Access, vol. 6, pp. 15980-15996, 2018, doi: 10.1109/ACCESS.2018.2814738.

[11] B. Heller, R. Sherwood, N. McKeown, The controller placement problem, in Proceedings of the First Workshop on Hot Topics in Software Defined Networks (ACM, Helsinki, Finland, 2012), pp. 7–12.

[12] F. Yeung, "Internet 2: scaling up the backbone for R&D," in IEEE Internet Computing, vol. 1, no. 2, pp. 36-37, March-April 1997, doi:10.1109/4236.601096.

[13] G. Wang, Y. Zhao, J. Huang, Q. Duan and J. Li, ”A K-means-based network partition algorithm for controller placement in software defined network,” 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, 2016, pp. 1-6, doi: 10.1109/ICC.2016.7511441.

[14] K. S. Sahoo, B. Sahoo, R. Dash and M. Tiwary, ”Solving Multi-controller Placement Problem in Software Defined Network,” 2016 International Conference on Information Technology (ICIT), Bhubaneswar, 2016, pp. 188-192, doi: 10.1109/ICIT.2016.047.

[15] K. A. Rasol and J. Domingo-Pascual, ”Joint Placement Latency Optimization of the Control Plane,” 2020 International Symposium on Networks, Computers and Communications (ISNCC), Montreal, QC, 2020, pp. 1-6, doi: 10.1109/ISNCC49221.2020.9297271.

[16] Y. Hu, W. Wang, X. Gong, X. Que and S. Cheng, ”On reliability-optimized controller placement for Software-Defined Networks,” in China Communications, vol. 11, no. 2, pp. 38-54, Feb 2014, doi: 10.1109/CC.2014.6821736.

[17] Y. Hu, W. Wendong, X. Gong, X. Que and C. Shiduan, ”Reliability-aware controller placement for Software-Defined Networks,” 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), Ghent, 2013, pp. 672-675.

[18] Fan Y., Xia Y., Liang W., Zhang X. (2018) Latency-Aware Reliable Controller Placements in SDNs. In: Chen Q., Meng W., Zhao L. (eds) Communications and Networking. ChinaCom 2016. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 210. Springer, Cham.

[19] Y. Fan, T. Ouyang, Reliability-aware controller placements in software-defined networks, in The 21st IEEE International Conference on High-Performance Computing and Communications, HPCC 2019, Zhangjiajie, China, 2019.

[20] Singh, AK, Maurya, S, Kumar, N, Srivastava, S. Heuristic approaches for the reliable SDN controller placement problem. Trans Emerging Tel Tech. 2020; 31:e3761. https://doi.org/10.1002/ett.3761

[21] Gao C, Wang H, Zhu F, Zhai L, Yi S. A particle swarm optimization algorithm for controller placement problem in software defined network. In: Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2015); 2015; Zhangjiajie, China.

[22] Singh AK, Kumar N, Srivastava S. PSO and TLBO based reliable placement of controllers in SDN. Int J Comput Netw Inf Secur. 2019;11(2):36-42.

[23] Rao RV. Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int J Ind Eng Comput. 2016;7(1):19-34.