A Greedy Approach for Controller Placement in Software-Defined Networks for Multiple Controllers

Stavroula Lalou
Department of Digital Systems
University of Piraeus
Piraeus, Greece
slalou@unipi.gr

Georgios Spathoulas
Dept. of Inform. Sec. and Comm. Techn.
NTNU
Gjøvik, Norway
georgios.spathoulas@ntnu.no

Sokratis Katsikas
Dept. of Inform. Sec. and Comm. Techn.
NTNU
Gjøvik, Norway
sokratis.katsikas@ntnu.no

Authors

Stavroula Lalou holds a Diploma of Cultural Technology and Communication from Aegean University of Mytilene since 2007, a MSc in Computer Science from University of Staffordshire of UK since 2011 and she is a full time PhD Student Department of Digital Systems of University of Piraeus since December of 2016.

Dr. Georgios Spathoulas holds a Diploma of Electrical and Computer Engineering from Aristotle University of Thessaloniki since 2002, a MSc in Computer Science from University of Edinburgh since 2005 and a PhD from the Department of Digital Systems of University of Piraeus since 2013. He is a member of Laboratory Teaching Staff of the Department of Computer Science and Biomedical Informatics of University of Thessaly since 2014 and he teaches in both undergraduate and postgraduate study programs of the Department. He is also collaborating, as a post doctoral researcher, with the Critical Infrastructures Security and Resilience group in NTNU CCIS. His research interests are related to network security, privacy preserving techniques and blockchain technology. He is the co-author of more than 40 publications in peer reviewed journals and conference proceedings. He has also served as Program Committee member in international conferences and he has taken part in both national and international research programs.

Sokratis K. Katsikas is the Director of the Norwegian Center for Cybersecurity in Critical Sectors and Professor with the Department of Information Security and Communication Technology, Norwegian University of Science and Technology. He is also Professor Emeritus of the Department of Digital Systems, University of Piraeus, Greece. He received the PhD in Computer Engineering from the University of Patras, Greece, the MSc in Electrical and Computer Engineering from the University of Massachusetts at Amherst, USA, and the Dipl. Eng. Degree in Electrical Engineering from the University of Patras, Greece. In 2019 he has awarded a Doctorate Honoris Causa by the Dept. of Production and Management Engineering of the Democritus University of Thrace, Greece. In 2021 he was ranked 7th in the security professionals category of the IFSEC Global influencers in security and fire list. In the past, among others, he has been the Rector of the Open University of Cyprus; Rector and Vice Rector of the University of the Aegean, Greece; General Secretary of Telecommunications and Posts of the Hellenic Government; Chair of the National Council of Education of Greece; member of the Board of the Hellenic Authority for the Security and Privacy of Communications; and member of the Board of the Hellenic Authority for the Quality and Accreditation of Higher Education. He has authored or co-authored more than 300 journal papers, book chapters and conference proceedings papers. He is serving on the editorial board of several scientific journals, he has co-authored/edited 46 books and conference proceedings, and he has served on/chaired the technical programme committee of more than 800 international scientific conferences. He chairs the steering committee of the ESORICS conferences and he is the Editor-in-Chief of the International Journal of Information Security (Springer).

Aims and Contributions of our paper

- The Controller Placement Problem (CPP) addresses the strategic positioning of SDN controllers within a network, crucial for efficient network management. It impacts network performance in latency, reliability, scalability, and resource usage. SDN architecture, separating control and data planes, enhances scalability and programmability compared to traditional architectures. Determining the optimal number and placement of controllers is a key challenge, with network latency being a primary performance factor. The study proposes a heuristic greedy algorithm to minimize end-to-end latency and reduce maximum latency between controllers and switches, aiming to mitigate controller queuing delay. Ultimately, deploying controllers in SDN-wide networks seeks to minimize maximum latency between controllers and switches.

The contribution of this paper is:

- An effective solution that address the challenges associated with placing controllers optimally in SDN environments with multiple controllers.
- An implementation with greedy algorithm that finds the location that is closest to the most switches that do not have assigned controllers. Place a controller in that location. Assigns each switch within the controller’s coverage to that controller. Calculates the latency between each switch and its assigned controller
- A heuristic -greedy algorithm designed to address the Controller Placement Problem. This algorithm aims to minimize end-to-end latency and reduce maximum latency between controllers and switches, with the ultimate goal of mitigating controller queuing delays. By proposing a specific algorithmic approach, the study contributes to the practical implementation of controller placement strategies within SDN environments.

The Controller Placement Problem (CPP)

- The CPP in Software-Defined Networking (SDN) involves strategically deploying controllers within the network, impacting various performance metrics like availability, fault tolerance, and convergence time. SDN, with its separated control and data planes, offers solutions to network challenges, making CPP a critical concern for optimizing network performance.
- The need to deploy multiple controllers in Software-Defined Networking (SDN) arises from the limitations of a single centralized controller, which struggles with the increasing demands of growing networks and applications. Relying on a single controller can create a bottleneck and a potential point of failure, negatively impacting overall network performance. Therefore, a multi-controller approach is recommended for large-scale SDNs to ensure the scalability of the control plane.
- Effectively placing multiple controllers is complex but essential to optimize network scalability and minimize latency, especially in larger networks. Two common multi-controller architectures are flat and hierarchical. Deploying multiple controllers in large-scale SDN environments aims to reduce latency, distribute the workload, and improve various performance metrics related to controller placement. This approach is particularly necessary for Wide Area Networks (WANs), as opposed to small networks like data centers where a single controller may suffice. Multiple controllers help maintain scalability and reliability in extensive SDN setups.

Our Proposal

- Our research focuses on latency which is one of the most often used performance indicators. Transmission, propagation, queuing, and processing delay make up the total latency. We evaluate latency between switch to controller latency (also known as controller-node latency) and controller-controller latency.
- The proposed algorithm is a greedy algorithm for placing controllers near switches to minimize the latency between controllers and nodes. The algorithm calculates the latency between each controller and its assigned nodes. The latency is calculated as the Euclidean distance between the controller and node, plus the transmission delay. The algorithm assigns each node to the controller with the lowest latency.
- The algorithm starts by initializing a costs, where each element costs[i][j] represents the cost of connecting switch i to controller j. The costs are calculated by computing the Euclidean distance between the switch and the controller.
- The algorithm then initializes an assigned array, where each element assigned[i] is a list of nodes assigned to controller i. The unassigned nodes are stored in a list called unassigned.
- The algorithm then enters a loop, where it repeatedly selects an unassigned node and assigns it to the closest available controller. The closest controller is determined by finding the controller with the smallest cost to the node's switch. If no controller is available, the node is skipped.
- The algorithm continues until all nodes have been assigned to a controller. The final result is a list of lists, where each inner list contains the nodes assigned to a specific controller.

Our Proposal

- We use the controller-node latency and controller-controller (propagation, queuing, and processing delay) as a crucial performance parameter. We have implemented a heuristic approach based on greedy algorithm. The basic idea of this approach is to obtain the minimum number of controllers which minimizes inter nodes distances to obtain acceptable latency from nodes to their assigned controller and also between controllers. Greedy algorithm uses the Euclidean distance between nodes and controllers as the cost function to determine the best immediate solution. In the context of the controller placement problem, the greedy algorithm calculates the Euclidean distance between each unassigned node and each available controller, and assigns the node to the controller with the smallest distance. Then, the distance between switches and controllers is calculated and the nodes are allocated to the closest controller.
- If the controller does not have the capacity to handle the node, the algorithm searches for the following nearest controller and carry out the same operation. This process will continue until an controller found and allocate rest of the node to the controller. The greedy solution provide by the algorithm fails when no controller can accommodate the required capacity, that case is considered as the worse case.
- A greedy heuristic algorithm is a type of algorithm that makes the locally optimal choice at each stage with the hope of finding a global optimum. It is a simple and fast algorithm. In the context of the CPP in multiple software-defined networking, a greedy heuristic algorithm can be used to find a solution that minimizes the latency between controllers and switches.

Our Proposal

- Here is the steps of the proposed approach for the CPP in SDN:
- Initialize a list of available locations for controllers.
- While there are still switches without assigned controllers:
  a. Find the location that is closest to the most switches that do not have assigned controllers.
  b. Place a controller in that location.
  c. Assign each switch within the controller's coverage to that controller.
- Calculate the latency between each switch and its as- signed controller.
- In our approach, switches is a list of (x, y) coordinates representing the locations of the switches, controllers is a list of (x, y) coordinates representing the locations of the controllers, and nodes is a list of nodes to be assigned to controllers. Each node has a switch attribute indicating which switch it is connected to.
- The greedy controller placement function first calculates the Euclidean distance between each switch and controller and stores the distances in the costs matrix. It then initializes a list assigned to keep track of which nodes have been assigned to which controllers.

Our Proposal

- The algorithm initializes a list of potential controller locations and iterates through unassigned switches, assigning each to the controller with the lowest cost in terms of distance. It permutes the list of unassigned nodes and assigns each node to the nearest controller, updating the assigned list accordingly.
- It creates a list of available locations for controllers and a vector of Location structs, each containing the index, distance to the nearest unassigned switch, and a vector of unassigned switches within the controller’s coverage.
- The algorithm enters a loop that continues until all switches have assigned controllers.
- In each iteration, it identifies the location closest to the most unassigned switches, places a controller there, and assigns each switch within that controller’s coverage area.
- Locations are sorted based on their distance and the number of unassigned switches within their coverage. Controllers are placed at the top locations, and switches within the coverage radius are assigned to these controllers.
- The algorithm calculates the latency between each switch and its assigned controller, updating a list of controller locations and a modified switch distances list with these latencies.
- The function continues this process until all switches are assigned to controllers. A vector of Controller structs, where each Controller struct contains the index of the controller and the number of assigned switches.
- The function returns a list of controller locations and a modified switch distances list that contains the latency between each switch and its assigned controller. The function works by iteratively placing controllers in the location that is closest to the most unassigned switches, and then assigning all switches within the controller’s coverage to that controller. The function continues to place controllers until all switches have been assigned to a controller.

Performance evaluation- Experimental Setup

- To evaluate the performance of the proposed mechanism we conducted a network simulation. We evaluated the performance of our system in terms of latency and transmission delay. These factors are crucial in Software-Defined Networking (SDN) due to the frequent communication between switches and controllers. The Controller Placement Problem in SDN often focuses on minimizing both transmission delay and controller processing delay. Transmission delay in CPP is similar to the facility location problem, where the goal is to find the best location to place the controllers to reduce the distance between the switches and the controllers. This is an important factor in ensuring efficient communication and reducing latency in SDN networks.
- A simulation has been conducted to assess the performance of the proposed scheme. The system on which the simulation was executed was based on an VM with Ubuntu 22.04 OS, 16 GB of memory and OpenFlow Switches. We emulate the performance using Mininet and Ryu controller [22] component-based software defined networking framework.

Performance evaluation- Experimental Setup

- In our experiments, we created a network of 6 controllers, 8 nodes (switches). We calculated the latency between con- trollers to nodes and between controllers.
- The latency between two controllers is calculated as the Euclidean distance between the two controllers and the trans- mission delay because the signal has to travel from one controller to the other, and then back to the first controller. The transmission delay is also counted for the round-trip time. The Euclidean distance between two controllers increases as the controllers are placed further apart, so the latency between two controllers will also increase as they are placed further apart.
- The nodes are placed at positions (3, 3), (4, 4), (3, 4), (3, 5), (4, 3), (4, 4), (5, 3), and (5, 4). The Euclidean distance between between Controller 1 and Node 1 is math.sqrt((0 - 3) ** 2 + (5 - 3) ** 2) = 3.61 units, so the latency between Controller 1 and Node 1 is 3.61.
- Controllers are placed at positions (0, 5), (10, 5), (20, 5), (30, 5), (40, 5), and (50, 5). The Euclidean distance between two adjacent controllers is 10 units, so the latency between two adjacent controllers is 10 + 2 * transmission delay.

Performance evaluation Results

TABLE I
CONTROLLER-NODE TRANSMISSION DELAY IN SECONDS

Controllers	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
Controller 1	0.21	0.42	0.44	0.42	0.43	0.44	0.46	0.43
Controller 2	0.48	0.21	0.33	0.33	0.30	0.33	0.36	0.34
Controller 3	0.59	0.43	0.21	0.22	0.22	0.21	0.22	0.22
Controller 4	0.57	0.46	0.32	0.21	0.21	0.22	0.22	0.21
Controller 5	0.53	0.41	0.32	0.33	0.21	0.22	0.22	0.21
Controller 6	0.63	0.48	0.36	0.35	0.34	0.21	0.22	0.21

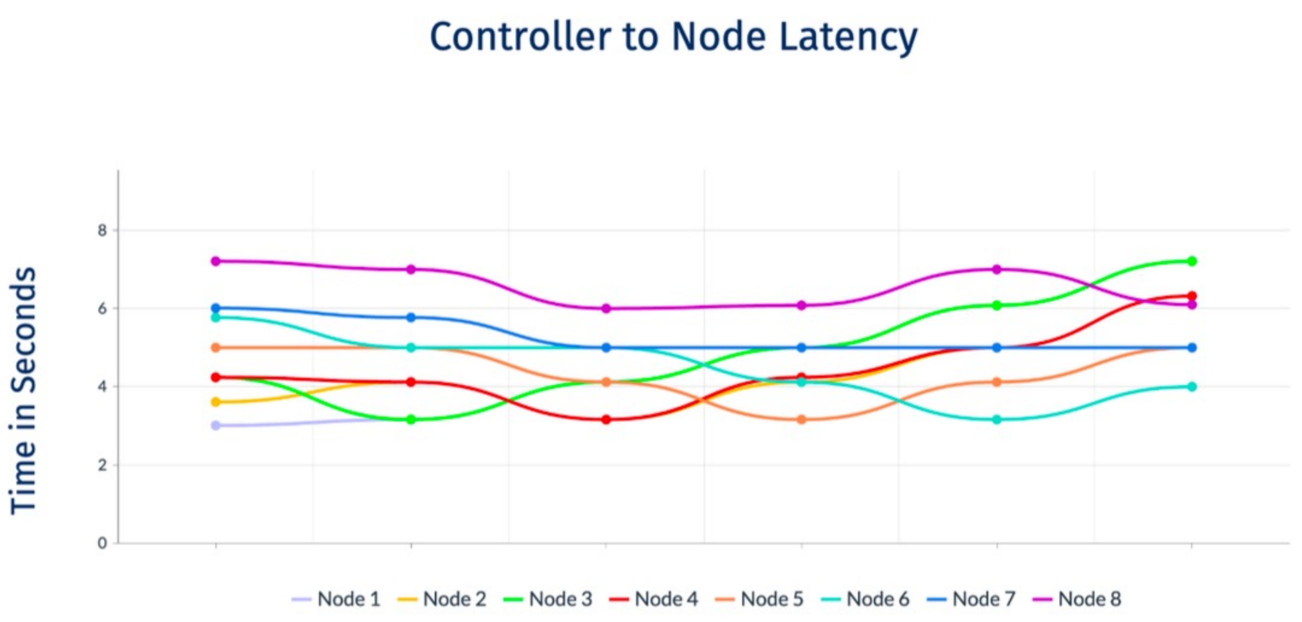
TABLE II
AVERAGE TRANSMISSION DELAY PER CONTROLLER

Controllers	Average Transmission Delay per Controller
	Average Latency
Controller 1	0.122 Seconds
Controller 2	0.118 Seconds
Controller 3	0.133 Seconds
Controller 4	0.132 Seconds
Controller 5	0.124 Seconds
Controller 6	0.115 Seconds

- Table I shows the latency between each controller and the nodes assigned to it. The first controller has a latency of 0.21 seconds for the first node and 0.42 seconds for the second node.
- The Table II calculates the average controller latency. The delay is calculated as the sum of the transmission delay.
- In our proposed algorithm, the total transmission latency between controllers to nodes is 10.067 seconds, it is the sum of all the latencies between each controller and its assigned nodes as described in Table I. While the transmission delay between controllers is 1.414 seconds and lower than the controller- controller latencies. The transmission latency as shown in Table I includes the time it takes for the switch to receive the data from the sender, process it, and send it to the receiver. The controller-controller latency is shown in Table II. According to the results in a SDN network with multiple controllers the latency numbers are low.

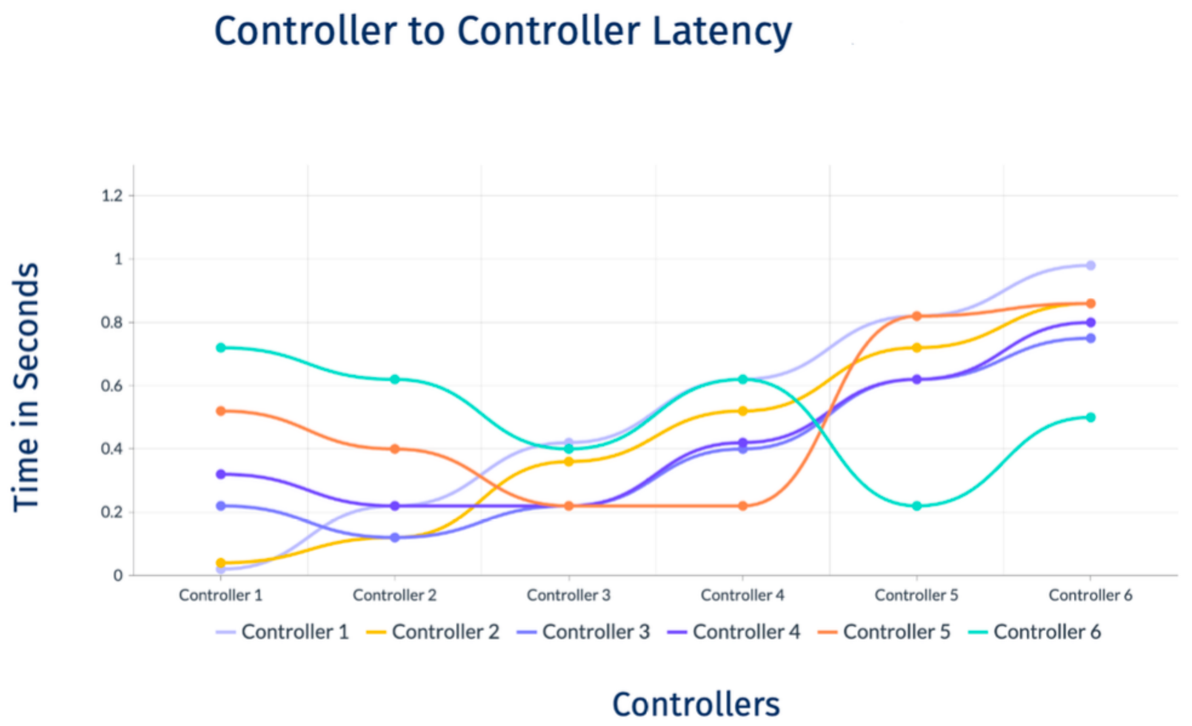
Performance evaluation Results

- Controller-to-node latency is the time it takes for a message to travel from a controller to a node in a network. Transmission delay is the time it takes for a message to be transmitted over the physical link between the controller and the node.
- The Figure depicts the latency between each controller and its assigned nodes. The latency is calculated in seconds.



Performance evaluation Results

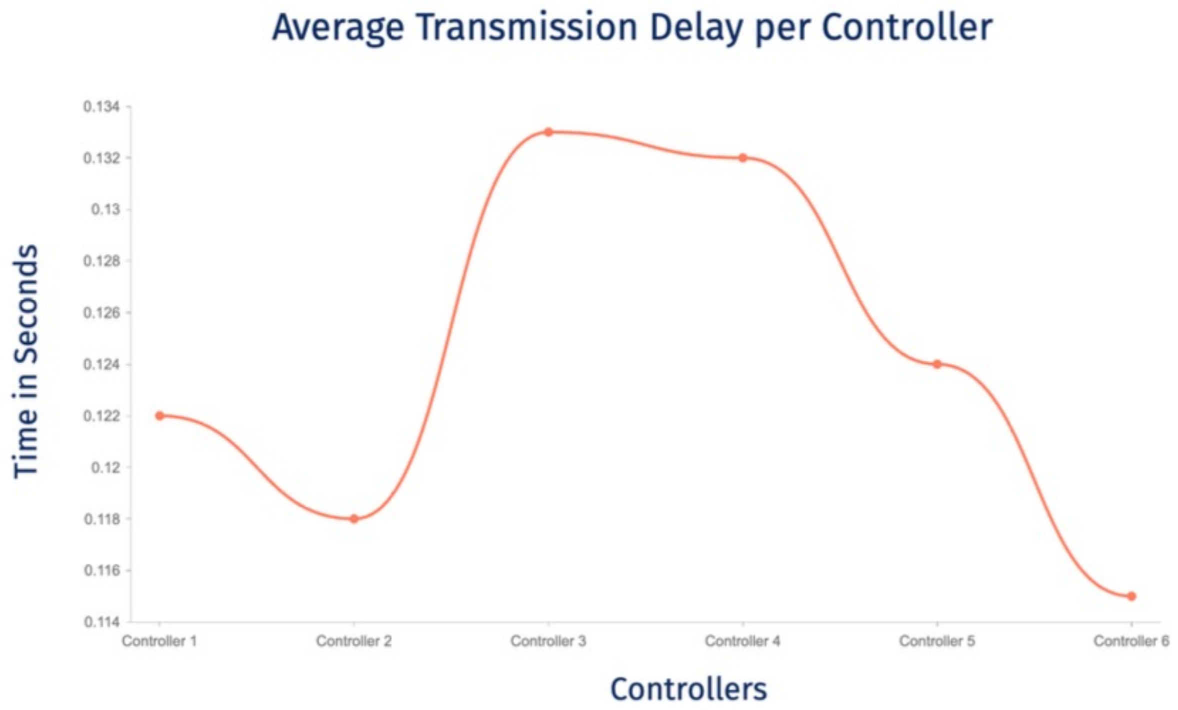
- The Controller-Controller latencies shows the latency between each pair of controllers. The latencies are relatively small and the switch is able to transmit data quickly between them. The controller-controller latencies is represented in Figure is the time it takes for the switch to process the data. These latencies are lower than the total transmission latencies because the switch only needs to process the data once, not for each controller sending a message to another controller.



Performance evaluation Results

- The latency numbers represent the time it takes for a controller to send a message to another controller as depicted in following Figure.
- The latency numbers are calculated as the average of the time it takes for the sender to send the message and the time it takes for the receiver to receive the message.
- The latencies are relatively small and the switch is able to transmit data quickly between them. These latencies are lower than the total transmission latencies because the switch only needs to process the data once, not for each controller sending a message to another controller.
- Transmission delay per controller is illustrated in Figure. The transmission delay per controller is obtaining by dividing the total transmission delay by the number of controllers. The transmission delay per controller is calculated by taking the average of the latencies for each controller, which can be done by summing up the latencies for each controller and dividing by the number of nodes.

The total transmission latency between controllers and nodes is higher than the controller-controller latencies because the switch needs to perform additional processing tasks. It includes the time it takes for the switch to receive the data from the sender, process it, and send it to the receiver.



Conclusion

- The idea of CPP in SDN is to adapt the facility location problem concepts to find the best location to place the con- trollers in the network, in order to reduce the transmission delay and improve the overall performance of the network. This implementation presents a greedy algorithm designed to optimize controller placement within a network, with the aim of minimizing latency between controllers to nodes and controllers to controllers and transmission delay.
- Experimental results demonstrate the efficiency of the implementation, achieving near-optimal solutions within the CPP framework. The observed latencies between controllers and nodes, as well as between controllers themselves, remain low, affirming the success of the controller placement strategy. Furthermore, our proposal minimizes transmission delay between controllers, which is critical to ensuring high network performance and efficiency. According to our results, our proposal ensures high network performance, scalability, and efficiency by minimizing transmission delay between controllers and between controllers and switches.
- Future work could investigate the impact of different network topologies on controller placement. This could provide insights into the algorithm’s performance in different network configurations.

References

[1] B. Heller, R. Sherwood and N. McKeown, ”The controller placement problem”, Proc. of the First Workshop on Hot Topics in Software Defined Networks, HotSDN ’12, ACM, New York, USA, pp. 7–12, 2012, doi:http://doi.acm.org/10.1145/2342441.2342444.

[2] G. Sch ̈utz and J.A. Martins, ”A comprehensive approach for optimizing controller placement in Software-Defined Networks”, pp. 199, 2020, doi: https://doi.org/10.1016/j.comcom.2020.05.008.

[3] G. Wang, Y. Zhao, J. Huang, Q. Duan, J. Li, ”A K-means-based network partition algorithm for controller placement in software defined network”, IEEE International Conference on Communications, Kuala Lumpur, pp. 1–6, 2016.

[4] L. Zhu, R. Chai and Q. Chen, ”Control plane delay minimization based SDN controller placement scheme”, Proc. 9th International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, pp. 1–6, 2017.

[5] G. Yao, J. Bi, Y. Li and L. Guo, ”On the capacitated controller placement problem in software defined networks”, IEEE Commun. Lett. 18, pp. 1339–1342, 2014.

[6] M.T.I. ul Huque, W. Si, G. Jourjon and V. Gramoli, ”Large-scale dynamic controller placement”, IEEE Trans. Netw. Serv. Manag. 14, pp. 63–76, 2017.

[7] M. Tanha, D. Sajjadi, R. Ruby and J. Pan, ”Capacity-aware and delay- guaranteed resilient controller placement for software-defined WANs” IEEE Trans. Netw. Serv. Manag. 15, pp. 991–1005, 2018.

[8] B.Zhang, X. Wang and M. Huang, ”Multi-objective optimization controller placement problem in internet-oriented software defined network”, Comput. Commun. 123, pp. 24–35, 2018.

[9] G. Wang, Y. Zhao, J. Huang, and Y. Wu, ”An effective approach to controller placement in software defined wide area networks,” IEEE Trans. Netw. Serv. Manag.,pp. 344-355, 2017.

[10] Mamushiane L, Mwangama J and Lysko AA. ”Controller placement optimization for Software Defined Wide Area Networks (SDWAN)”, pp. 45-66, 2021.

[11] Rasol KA and Domingo-Pascual J. ”Evaluation of joint controller placement for latency and reliability-aware control plane”, Eighth In- ternational Conference on Software Defined Systems (SDS) IEEE, pp. 1-7, 2021.

References

[12] Eugen Borcoci, Radu Badea and Serban Georgica Obreja, and Marius Vochin. ”On multi-controller placement optimization in software defined networking-based wans”, (ICN 2015), pp. 273, 2015.
[13] Fan Z, et al. ”A multi-controller placement strategy based on delay and reliability optimization in SDN”. Proc. 28th Wireless and Optical Communications Conference (WOCC), IEEE, 2019.
[14] Fan Y, Xia Y, Liang W and Zhang X. ”Latency-aware reliable controller placements in SDNs”, Proc. Communications and Networking: 11th EAI International Conference, (ChinaCom 2016 Chongqing), pp. 152-162, China, September 24–26, 2016.
[15] David Hock, Matthias Hartmann, Steffen Gebert, Michael Jarschel, Thomas Zinner, and Phuoc Tran-Gia. ”Pareto-optimal resilient controller placement in sdn-based core networks”, Teletraffic Congress (ITC), 25th International IEEE, pp. 1–9, 2013.
[16] Chen W, Chen C, Jiang X and Liu L. ”Multi-controller placement towards SDN based on Louvain heuristic algorithm”, IEEE Access, 2018.
[17] Liao J, Sun H, Wang J, Qi Q, Li K and Li T. ”Density cluster based approach for controller placement problem in large-scale software defined networkings”, 2017.
[18] Wang G, Zhao Y, Huang J and Wu Y. ”An effective approach to controller placement in software defined wide area networks”, IEEE Trans Netw Serv Manag, pp. 344-355, 2017.
[19] Heller B, Sherwood R and McKeown N. ”The controller placement problem”, ACM SIGCOMM Comput Commun Rev.,pp. 473-478, 2012.
[20] Hu T, Guo Z, Yi P, Baker T and Lan J. ”Multi-controller based software- defined networking: a survey”, IEEE Access, 2018.
[21] Dhar M, Debnath A, Bhattacharyya BK, Debbarma MK and Debbarma S. ”A comprehensive study of different objectives and solutions of controller placement problem in software-defined networks”, Trans Emerg Telecommun Technol., pp. 33, 2022.
[22] ”Ryu omponent-based software”, [Online]. Available from: ”https://ryu- sdn.org/”