

# Controller Placement Algorithms in Software Defined Network - A Review of Trends and Challenges

Si-Kee Yoon<sup>1</sup>, ZIA Khalib<sup>1</sup>, N Yaakob<sup>1,2</sup>, and A Amir<sup>1,3</sup>

<sup>1</sup>School of Computer and Communication Engineering, University Malaysia Perlis, Pauh Putra, 02600 Arau, Perlis

<sup>2</sup>School of Computer Science and IT, RMIT University, Melbourne, VIC 30000, Australia

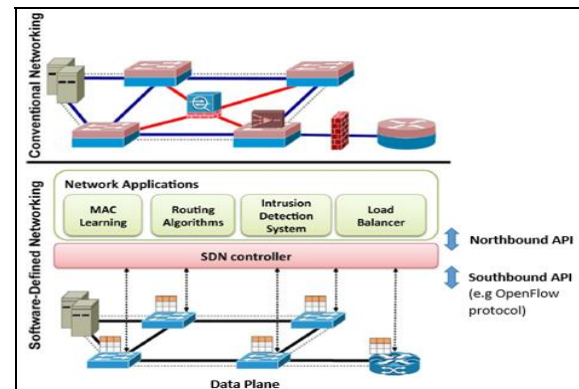
<sup>3</sup>School of Information Technology, Monash University, Clayton, VIC 3800, Australia

**Abstract.** Traditional network architectures are complex to manage, comparatively static, rigid and difficult to make changes for new innovation. The proprietary devices in such architectures are based on manual configuration which are unwieldy and error-prone. Software Defined Network (SDN) which is described as a new network paradigm that decouple the control plane from data plane are capable to solve today's network issues and improve the network performance. Nevertheless, among so many challenges and research opportunity in SDN, Controller Placement Problem (CPP) is said to be the most important issues which can directly affect the overall network performance. Thus far, the issue regarding the CPP and its challenge has not been completely reviewed and discussed properly in any other papers. In this paper, we provide a comprehensive review on several optimized controller placement problem algorithms in SDN. This paper also highlights some limitations of the reviewed methods and also emphasizes on suitable approach to address the aforementioned problems.

## 1. Introduction

Software Defined Network (SDN) is an emerging network architecture which decoupled the control plane from the data plane, make forwarding decisions based on flow rather than destination based [1]. SDN help speed up the innovation process, lower the complexity and the cost on hardware implementation with its open sources features. Accordingly, network management become flexible and programmability compared to those conventional networking. SDN is currently highly implemented in many applications [1], such as data center ( e.g. Google, Facebook etc.), cloud computing environment, Internet-of-thing (IoT) and wireless mesh networks (WSN). Fig. 1 shows the different of conventional and SDN architectures. Referring to the figure, SDN is made up of 3 layers which are data plane, control plane and application layer.

In recent years, Controller Placement Problem (CPP) is one of the hot topics in SDN research fields which can be denoted as NP-hard or known as facility location problem which normally regards the quantity and location of controllers in network [2]. According to the statements in [3-5], network performance may be affected due to inefficient controller placement strategy. Example consequences under poor controller selection such as long recovery time after failures.



**Fig. 1.** Conventional networking versus Software Defined Network architecture.

The SDN control plane can be either in centralized or distributed [3]. Nevertheless, SDN deployment with a physically centralized controller has several limitations which may lead to a single point of failure for the entire network. To address these constraints, recent researches have supported implementation of multiple (distributed) SDN controllers. The remainder of this paper is structured as follows. In Section II, we discuss some of the important related works for controller placement in SDN and also provide performance comparisons between those works. Conclusion is pointed out in Section III.

\* Corresponding author: [zahereel@unimap.edu.my](mailto:zahereel@unimap.edu.my)

## 2. Related Work

Several categories of controller placement approaches found in literature can be shown in Fig. 2. The following subsections describe each of the category in details.

### 2.1 Static Controller Placement

The mathematical formulation (1) is generally used to search for best placement of  $k$  controllers and  $n$ - node graphs, with the total number of placements:

$$\binom{n}{k} = \frac{n!}{(n-k)!k!} \quad (1)$$

The first research on CPP was done by Heller et al. [2] which highlighted the importance to minimize average and worst case propagation latency in between the switch-to-controller as shows in formulation (2) and (3).

Average latency:

$$L_{avg}(S') = \frac{1}{n} \sum_{v \in V} \min_{(s \in S')} d(v, s) \quad (2)$$

Worst case (maximum) latency:

$$L_{wc}(S') = \max_{(v \in V)} \min_{(s \in S')} d(v, s) \quad (3)$$

However, this approach is a topology-based that is only limited to a certain topology and does not consider the capacity of controllers constraints, make it not really applicable in real network. The Capacitated Control Placement Problem (CCPP) in [4] is considering the additional load of controller. Cervello-Pastor et al. in [5] propose  $k$ -Criteria strategy is achieve best minimum number of controllers as compared to methods in [2]. The modified approach in [6] performs a single run time which is faster for clustering and more efficient in large-scale network compared to the iteration-based of capacitated  $k$ -Center algorithm in [5].

The works presented in [7-8] are natural-inspired metaheuristic optimization approaches. It is similar to human action in decisions making to solve CPP. Moreover, multiple controllers of hierarchical architecture [9-10] are designed to solve the single point of failure and computational complexity problem in large scale network. On the other hand, there are researches emphasis on the importance of reliability among switch and controller in CPP and several new metrics are proposed as stated in [11-12]. Others heuristic method are also proposed to solve CPP in [13-15]. Some methods in [16-17] use mathematical approaches to cater CPP. A new Integer Linear Programming (ILP) formulation is generated in [16] to find the minimum number of active controllers. In [17], a mathematical model named Failure Foresight Capacitated Controller Placement (FFCCP) presented in closest assignment when controller failures occurred.

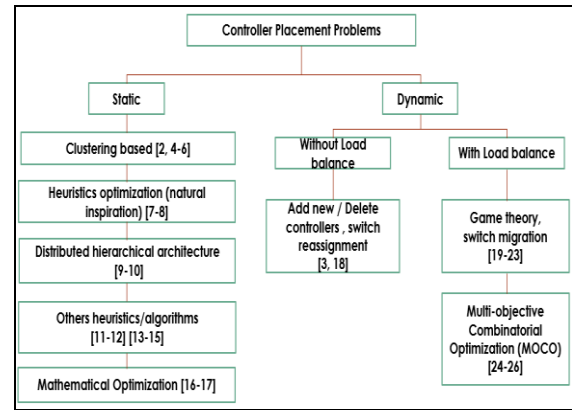


Fig. 2. Taxonomy of controller placement problem.

### 2.2 Dynamic Controller Placement

The previous section discussed all related works of static switch-to-controller mapping. This section will discuss, dynamic controller placement methods [3, 18-23] including those multi-objectives approaches that feasible in dynamic changes of large network as stated in [24-26].

Initially, dynamic controller provision problem (DCPP) is solved by simply dynamically changing the controller number and their locations as done in [3, 18] without load adaptation mechanism. This involves a dynamic adjustment of active controllers based on network conditions and instant switch-to-controller reassignment in case of controller failures. Dixi et al. [18] introduce “Elasticon” to dynamically alter controller resource pool and migrating switches to neighbour controller. Non-zero game theorem approach in [20] which performing load adaptation algorithms added on top of previous work ElastiCon for master controller reflection. The switch migration method as implemented in [19, 21-23] are some other enhanced DCPP approaches with load adaptation.

Furthermore, the Multi-objective Combinatorial Optimization (MOCO) approaches in [24-26] allow trade-off of multiple metrics to achieve overall optimization. Thus, users can decide a more suitable results based on their demands. Pareto-based Optimal Controller placement (POCO) is an exhaustive search actual optimal Pareto frontier which is generally used as reference value. POCO was then extended by using several heuristic algorithms to obtain approximated Pareto optimal placement to compute faster with acceptable accuracy in large network.

Table 1 shows the performance summary of the several methods discussed in this paper which is in static controller placement methods [2, 4-17]. Table 2 shows the performance summary of dynamic controller placement methods in [3, 18-26]. There are many heuristic algorithms proposed by considering various objective functions to solve controller placement problem.

**Table 1.** Performance comparison on static controller placement approaches.

Ref.	Techniques	Advantages	Disadvantages	Performance metrics / Objective Function									
				Flow setup time	Inter-controller latency	Switch-to-controller latency	Capacity controller	Inter-controller synchronization	Load balance	Reliability	Throughput		
[2]	1. k-Center (max Lavg) 2. k-Median (min Lavg) (2012) 3. Add k-Critical (2014)	- Consider the distance between switch-to-controller - Trade-off number of controllers with latency minimization  - Consider network physical characteristic and load adaptation	- Number of controllers needed depends on the topology - High memory and time consumed for large networks			✓							
[5]			- Shortest path distance for nodes-to-controller is not the best decision - The average results achieved, but not the optimal one.			✓				✓			
[4]	Capacitated k-Center strategy (2014)	- Capacity of controllers increased	- The average results achieved, but not the optimal one.			✓		✓			✓		
[6]	Capacitated Density Cluster-Based CP (2017)	- Faster clustering - Can be extended for different objective solutions	- Partition cluster size can't as large as possible (capacity based)		✓	✓		✓				✓	(Controller-loss switch)
[7]	Network Clustering PSO (NCPSO) (2015)	- Overcome weakness of solely used of PSO in discrete problem. - Controller's load optimized	Not specified			✓		✓			✓		
[8]	PSO Capacitated Global Latency (2015)	- Better propagation latency, convergence time is achieved - Cognitive learning phases	- Controller numbers are given know in advance (not practical in real network)		✓	✓		✓					
[9]	Distributed hierarchical Architecture (2015)	- Computational complexity reduced (load distribution) - Flexibility enhanced (functionality distribution) - To ensure load balancing below the given threshold	- Flow setup time and response time raised, higher complexity, more layering and clustering area needed as network size increased						✓			✓	✓
[10]	Hierarchical controller load balancing (2016)		- 2 layer structure may not enough - Frequently (1s) controller's load checking less efficiency						✓		✓		
[11]	Expected (%) of control path loss (2014)	- Trade-off between reliability and latency for solving RCP.	- Given the topology network - Random generated failure probabilities			✓							✓

[12]	(Extended) Minimum five nine reliability for FTCP (2016)	- The operational probability of network components, controller's capacity, and different reliability levels are considered	- Minimum number of controllers achieved are limited to certain topology	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		
[13]	Partition fairness minimization (2016)	- Network connectivity maintained even in failure cases - Fairly distribution switch-controller network	- Topology limited optimization result	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		
[14]	QoS guaranteed CPP (2015)	- QoS constraint (response time) never been concerned in previous work	- High controller overhead and response time increased - The result is topology based	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		
[15]	Multiple mapping with resilience constraints (2017)	- Better fairness and flexible load distribution - More stable towards dynamic traffic fluctuation	- Fixed and pre-defined controller placement even in dynamic traffic - Small network scale tested	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		
[16]	New resilience ILP formulation (2016)	- Trade-off on latencies and maximum inter-controller load difference is considered	- Maximal latency increased if average latency minimized (nearest controller selection)	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
[17]	Failure Foresight CCPP (2016)	- First work on worst-case latency minimization in between switches and their backup controllers	- Tested in small-scale network (at most 12 controllers)	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		

**Table 2.** Performance comparison on dynamic controller placement approaches.

Ref.	Techniques	Advantages	Disadvantages	Performance metrics / Objective Function												
				Flow setup time	Inter-controller latency	Switch-to-controller latency	Capacity controller	Inter-controller synchronization	Load balance	Reliability	Throughput					
[3]	Greedy Knapsack and Simulated Annealing (2013)	- Power saved as inactive controller in sleep stage - Flow setup time reduced - Low communication overhead	- Long convergence time of DCP-SA even better near optimal result - Network instability possibility	✓	✓					✓						
[18]	ElastiCon (2013)	- Minimal influence on response time	- Possibility new load imbalance - Many RTT consumed before it start	✓			✓									
[19]	Crash-tolerant dynamic switch migration (2014)	- Resource utilization is enhanced - Response time reduced - Controller failover supported	- Scalability need improved - Response time increased as controller clustering needed	✓									✓			✓
[20]	Non-Zero-Sum Game (2014)	- Simple, low complexity, minimum packet drops & delay - Topology independent, real time and utilization maximized	- Inconsistency possibility of each controller's routing table updating - Possible new load imbalance of local (closer) migration			✓								✓		

[21]	Flow-based dynamic switch migration (2015)	- New metric : weight of switch nodes (importance / throughput) Controller load release faster	- Small and medium network switches not consider	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[22]	Control mechanism distributed hopping algorithm (2017)	- No controllers added in migration - Lower average flow setup time - Better average network utilization	- Not enough just consider control load and the locality for choosing a target controller	✓	✓	✓	✓	✓	✓	(communication overhead)	✓	✓	✓
[23]	Switch migration-based decision-making (2017)	- Migration efficiency first considered	- Small scale network implementation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[24]	Pareto capacitated k-Medoids (PCKM) (2015)	- Applicability in large network - Capacity controller included in the switch-controller assignment	- No single best optimal solution.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[25]	Teaching-learning based optimizer (2016)	- Faster computation times - Less memory (search space)	- Only estimated optimal results are obtained not exact reading	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[26]	Multi-Criteria Decision Algorithm (2014)	- Trade-off multiple objectives before making decision - Failure cases are considered	- Number of controllers are given - No implemented in large and real network	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

### 3. Discussion

SDN controller placement problem (CPP) can be categorized as static and dynamic methods. There are various performance metrics are discussed in most of the reviewed paper. In general, existing proposed methods are normally have a prior knowledge of the controller numbers to find the suitable placement in order to optimize the network performance. In another way, the optimal controller numbers are usually obtained on the condition as the specific network topology to be implemented are given in advance. Furthermore, the mapping decision based on dynamic traffic condition can be categorized into three types. First is single mapping, then second is multiple mapping where one switch assigns to multiple controllers in the case of any controller failure can distribute the loads to other controllers. Lastly, the third type of mapping decision is dynamic switch assignment to the controller that is periodically triggered by measuring value to avoid the congestion happen.

### 4. Conclusion

In conclusion, partition and fairness minimization heuristics give a fairer distribution of switch-to-controller allocation, but the results are limited to certain topologies. This

paper discussed most performance metrics as compared to others static controller placement approaches. On the other hand, the flow-based dynamic switch migration algorithm is designed to migrate the boundary switches to its lightweight neighbour controller when there is flow burst. This paper performs most parameters among all others dynamic controller placement approaches. However, this topology only adopted for small and medium network size. Of all the reviewed methods, none have considered topology-aware controller placement based on network conditions that changes automatically. Majority research works just experimented their controller placement designs under particular kinds of network topologies environment. Most of the papers simply assume one controller failures at a time to make the simulation easier. For future works, variety network congestion scenarios should be included in controller placement designed such as multiple controllers, nodes or links are possible to fail at the same time. This is important to increase the accuracy and feasibility of the controller placement algorithm proposed. In addition, packet loss rate as one of the important QoS parameter to evaluate the network performance is also less being concerned in most of the current SDN controller placement.

The reported research in this paper is financially funded by the Ministry of Higher Education (MoHE) Malaysia through the Fundamental Research Grant Scheme (FRGS) and University Malaysia Perlis (UniMAP).

## References

- [1] T. Bakhshi, "State of the Art and Recent Research Advances in Software Defined Networking," *Wirel. Commun. Mob. Comput.*, **2017**, 35 (2017)
- [2] B. Heller, R. Sherwood, and N. McKeown, "The Controller Placement Problem," *SIGCOMM Comput. Commun. Rev.*, **42**, 6 (2012)
- [3] M. F. Bari, A. R. Roy, S. R. Chowdhury, Q. Zhang, M. F. Zhani, R. Ahmed, and R. Boutaba, "Dynamic Controller Provisioning in Software Defined Networks," *Proceedings of the 9th International Conference on Network and Service Management (CNSM)*, 8 (2013)
- [4] G. Yao, J. Bi, Y. Li, and L. Guo, "On the Capacitated Controller Placement Problem in Software Defined Networks," *IEEE Commun. Lett.*, **18**, 4 (2014)
- [5] Y. Jim and C. Cervell, "On the controller placement for designing a distributed SDN control layer," *IFIP Netw. Conf.*, 9 (2014)
- [6] J. Liao, H. Sun, J. Wang, Q. Qi, K. Li, and T. Li, "Density cluster based approach for controller placement problem in large-scale" *Comput. Networks*, **112**, 12 (2017)
- [7] S. Liu, H. Wang, S. Yi, and F. Zhu, "NCPSO : A Solution of the Controller Placement Problem in Software Defined Networks," *Springer International Publishing*, 13 (2015)
- [8] C. Gao, H. Wang, and F. Zhu, "A Particle Swarm Optimization Algorithm for Controller Placement Problem in Software Defined Network," *Springer International Publishing*, 11 (2015)
- [9] P. D. Bhole and D. D. Puri, "Distributed Hierarchical Control Plane of Software Defined Networking," *Int. Conf. CICN*, 7 (2015)
- [10] G. G. and F. A. H. Selvi, "Cooperative Load Balancing for Hierarchical SDN Controllers," *IEEE 17th Int. Conf. HPSR*, 6 (2016)
- [11] H. U. Yannan, W. Wendong, G. Xiangyang, Q. U. E. Xirong, and C. Shiduan, "On Reliability-optimized Controller Placement for Software-Defined Networks," *China Commun.*, **11**, 17 (2014)
- [12] F. J. Ros and P. M. Ruiz, "On reliable controller placements in Software-Defined Networks," *Comput. Commun.*, **77**, 11 (2016)
- [13] D. M. F. Mattos, O. C. M. B. Duarte, and G. Pujolle, "A Resilient Distributed Controller for Software Defined Networking," *IEEE ICC*, 6 (2016)
- [14] T. Y. Cheng, M. Wang, and X. Jia, "QoS-Guaranteed Controller Placement in SDN," *IEEE GLOBECOM*, 6 (2015)
- [15] V. Sridharan, M. Gurusamy, and T. Truong-huu, "On Multiple Controller Mapping in Software Defined Networks with Resilience Constraints," *IEEE Commun. Lett.*, 4 (2017)
- [16] N. Perrot and T. Reynaudt, "Optimal Placement of Controllers in a Resilient SDN Architecture," *12th Int. Conf. DRCN*, 7 (2016)
- [17] B. Prakasa, R. Killi, and S. V. Rao, "Optimal Model for Failure Foresight Capacitated Controller Placement in Software-Defined Networks," *IEEE Commun. Lett.*, **20**, 4 (2016)
- [18] A. Dixit, F. Hao, S. Mukherjee, T. V Lakshman, and R. Kompella, "Towards an Elastic Distributed SDN Controller," *SIGCOMM Comput. Commun. Rev.*, **43**, 6 (2013)
- [19] C. Liang, R. Kawashima, and H. Matsuo, "Scalable and Crash-Tolerant Load Balancing Based on Switch Migration for Multiple OpenFlow Controllers," *Second Int. Symp. Comput. Netw.*, 7 (2014)
- [20] H. K. Rath, V. Revoori, S. M. Nadaf, and A. Simha, "Optimal Controller Placement in Software Defined Networks ( SDN ) using a Non-Zero-Sum Game," *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 6 (2014)
- [21] L. Yao, P. Hong, W. Zhang, J. Li and D. Ni, "Controller Placement and Flow based Dynamic Management Problem towards SDN," *IEEE ICCW*, 6 (2015)
- [22] G. Cheng, H. Chen, H. Hu, and Z. Wang, "Toward a Scalable SDN Control Mechanism via Switch Migration," *China Commun.*, **14**, 14 (2017)
- [23] C. A. N. Wang, B. O. Hu, S. Chen, D. Li, and B. I. N. Liu, "A Switch Migration-Based Decision-Making Scheme for Balancing Load in SDN," *IEEE Access*, **5**, 8 (2017)
- [24] S. Lange, S. Gebert, J. Spoerhase, P. Rygielski, T. Zinner, S. Kounev, and P. Tran-Gia, "Specialized Heuristics for the Controller Placement Problem in Large Scale SDN Networks," *27th Int. Teletraffic Congr.*, 9 (2015)
- [25] V. Ahmadi, A. Jalili, and M. Khorramzadeh, "Multi-Objective Controller Placement Problem: issues and solution by heuristics," *IJCSIS*, **14**, 7 (2016)
- [26] E. Borcoci, R. Badea, S. G. Obreja, and M. Vochin, "On Multi-controller Placement Optimization in Software Defined Networking - based WANs," *ICN 2015: Fourteenth Int. Conf. Networks*, 6 (2015)