International Journal of
Science and Engineering Investigations
vol. 8, issue 95, December 2019
Received on December 22, 2019
ISSN: 2251-8843

A Multi-Adaptive Controller Placement Algorithm in Software-Defined Networks

Hossein Akbari[1], Zahra Shabani[2], Abolfazl Toroghi Haghighat[3]
[1]Computer Engineering Department, Qazvin Branch, Islamic Azad University, Qazvin, Iran
[2]Computer Engineering Department, Ferdowsi University, Mashhad, Iran
[3]Faculty of Computer and Information Technology, Qazvin Branch, Islamic Azad University, Qazvin, Iran
([1]hossein_akbari11@yahoo.com, [2]Zahra.shabani@mai.um.ac.ir, [3]haghighat@qiau.ac.ir)

Abstract-In order to simplify the management of traditional networks, the software-defined networks (SDN) were proposed as a promising pattern which provide the programmability for the network configuration by separating two levels of control plane and data plane. By using and expanding SDN, the controller placement is considered as an important task in the software-defined networks. In the meantime, the allocation of switches to controllers plays a very important role in the quality of service (QoS). Afterwards, these criteria were used in the analytic hierarchical process (AHP) technique to fulfill the multi-criteria allocation of the switch to the controller, controller to controller as well as find the shortest. Eventually, the genetic algorithm of the controller placement was combined with the analytic hierarchical process technique to solve such a special problem of controller placement. The results represented that the allocative efficiency of this approach can reduce the link load balancing problem. In another word, the maximum link utilization of the route control was reduced.

Keywords- *Software-Defined Networks (SDN), Controller Placement, Genetic Algorithm, Analytic Hierarchical Process, Large-Scale Networks*

I. INTRODUCTION

Software-defined networks appear as an encouraging pattern that separates the control plane from the data plane. It has the capability to control the network centrally through the software. Using a single controller is inefficient for controlling the traffic of large networks. Therefore, having multiple controllers in large-scale networks is a necessity. The placement of multiple controllers using an optimal method is a research challenge in the software-defined networks. The problem of controller placement includes: The minimum number of controllers and the positions of these controllers are required in a network. Many researchers have proposed some solutions to the controller placement problem, which is considered as an NP-hard problem. Generally, the solutions have been considered regarding different factors (such as the propagation delay between switches and controllers, and controllers) and the constraints (including the capacity of controllers and switches).

Some papers on the allocation of switches to the controllers have been inefficient. As a matter of fact, at the place of finding the controllers, it is usually assumed that each switch is allocated to the nearest controller based on the propagation delay [1, 4, 6]. Moreover, none of the papers, which have considered the load and capacity of the controllers, have attended to the flow of the path and the number of loads applied per link [16, 10]. It is obvious that the load amount can increase the occurrence probability and the probability of connected links and nodes breakage.

In this paper, the controller placement problem is examined in such a way to find a suitable position for the controller. Two important aspects of this problem have also been considered as searching in the search space for the placement: Considering the specific placement in a space that the algorithm encounters, two mechanisms have been considered to find the best state of allocating the switches to the controllers. Considering the importance of allocating the controllers to the switches as well as the routes which are used for the connection, our purpose is to examine the problem of deploying the controller in the software- defined network not only as a position but also as an allocation problem.

In this regard, three fundamental criteria are very effective in this allocation problem. These parameters include: The propagation delay of the switch to the controller, the hop count between the switches and controllers, and the link utilization in the path have been considered. Since many papers only consider the propagation delay and this parameter has effects on the availability time and the convergence in the large networks, therefore the propagation delay is considered as one of the most important parameters. The second important criterion is the hop count, which is defined as the hop count between the switches and the controllers. In actual fact, any stored hop represents a good estimation of the distance between the switch and the controller. Using the link (link utilization) has been considered as the third metric. This metric is related to the quality of the determined routes to examine the performance of the allocated routes.

In order to use these criteria for allocating the switches to the controllers as well as controlling the routes for each pair of switches and controllers, the multi-criteria allocation method called the Analytic Hierarchical Process (AHP) technique had been used. The Analytic Hierarchical Process technique is a smart selection to apply these criteria because it considers flexibility to prioritize the criteria and select the weight in accordance with the importance of criteria for finding a solution [13]. Furthermore, using this technique, the effects of important criteria are effectively applied to deploying the controllers. This method has been intended by considering the controller placement problem, and it is an appropriate method not only for the allocation of the switches to the controllers but also for the controlled routes. Some techniques introduced in [13] and [14] have been considered for this allocation.

For solving the controller placement problem, it is required to use a very efficient algorithm. The genetic algorithm is a powerful algorithm that is compatible with the controller placement problem. The genetic algorithm is considered as the basic algorithm, and other methods and algorithms, such as the Analytic Hierarchical Process-based allocation algorithm, are called with the main algorithm whenever required.

For evaluating the performance of this method, different evaluation experiments are performed. First, the performance of the analytic hierarchical technique based algorithm has been analyzed considering different weights. The objective of this evaluation is to analyze the importance of different factors that have effects on the allocation of switches to the controllers. [23] Afterwards, two important indices have been used to evaluate our proposed model, called “link loading indicator” and “maximum link utilization controller placement” or “LL (1)” and “MLUCP” in short, in order to demonstrate that the proposed model has a high efficiency in finding a problem and it presents the solutions with better quality. The MLUCP and LL (l) criteria are important because they consider the load balancing in the links [15, 16]. The lower these criteria, the better the link load. Several papers attempt to reduce these criteria on the networks and data centers [15, 17]. We used this metric to evaluate the quality of the solution and represented that the proposed method has a better load balancing than other methods. For further information regarding this research, the effect of three parameters (propagation delay, hop count, link utilization) on optimizing the controller position and presenting the optimal routes to allocate the switches to the controllers are considered in solving the controller placement problem. Therefore, the AHP-based algorithm has been introduced to perform a multi-criteria allocation. Here, we discuss how other criteria, in addition to the propagation delay, can be effective on the controller placement problem and initiate the allocation process as a new important problem in the controller placement problem. Furthermore, we propose a controller genetic placement algorithm (CGPA) to solve the controller placement allocation problem.

In this regard, three fundamental criteria are very effective in this allocation problem. These parameters include: The propagation delay of the switch to the controller, the hop count between the switches and controllers, and the link utilization in the path have been considered. Since many papers only consider the propagation delay and this parameter has effects on the availability time and the convergence in the large networks, therefore the propagation delay is considered as one of the most important parameters. The second important criterion is the hop count, which is defined as the hop count between the switches and the controllers. In actual fact, any stored hop represents a good estimation of the distance between the switch and the controller. Using the link (link utilization) has been considered as the third metric. This metric is related to the quality of the determined routes to examine the performance of the allocated routes. It should be mentioned that the link utilization parameter is only used for traffic control.

In order to use these criteria for allocating the switches to the controllers as well as controlling the routes for each pair of switches and controllers, the multi-criteria allocation method called the Analytic Hierarchical Process (AHP) technique had been used. The Analytic Hierarchical Process technique is a smart selection to apply these criteria because it considers flexibility to prioritize the criteria and select the weight in accordance with the importance of criteria for finding a solution [13]. Furthermore, using this technique, the effects of important criteria are effectively applied to deploying the controllers. This method has been intended by considering the controller placement problem, and it is an appropriate method not only for the allocation of the switches to the controllers but also for the controlled routes. Some techniques introduced in [13] and [14] have been considered for this allocation.

For solving the controller placement problem, it is required to use a very efficient algorithm. The genetic algorithm is a powerful algorithm that is compatible with the controller placement problem. The genetic algorithm is considered as the basic algorithm, and other methods and algorithms, such as the Analytic Hierarchical Process-based allocation algorithm, are called with the main algorithm whenever required.

II. Related Works

The controller placement problem has been discussed in a couple of papers. This problem is comparable with facility location problem in many aspects. Many works address the controller placement problem in SDN, but slightly different objectives. The works [7-15] target fault tolerance, whereas [16-18] aim at balancing the loads on the controllers. In [3, 4, 5], they analyze the controller placement problem as K-median or K-center problems, and their objective is to minimize the average or the worst-case. Considering the topology, the minimum cost is considered for positioning, and the controller placement problem requires finding the number and position of the required controllers. It expresses the minimum costs considering the number of controllers [6], a delay in the relation between the switch and the controller [4], controller convergence time [13], the combination of more than one criterion [1, 7, 8]. Additionally, many papers present controller placement as a loading problem [16, 10]. In order to perform this, they determine a specific capacity for each controller with the load balancing between different controllers. If a load of any controller is greater than the threshold, the switches will retransfer to other controllers [7,11,12].

Yu et al. [18] extended the problem considering the limited capacity of the network elements. They discussed a controller positioning problem with the purpose of reducing the worst latency of the controlled routes, which decreased the load limitation satisfaction of the controllers. While their attempts at the load balancing on the controllers, the link load balancing, and the assignment routes have not been considered in this paper. Ahmadi et al. [1] A Multiple Non-Combinatorial Sorting Genetic Algorithm (MHNSGA) state that the mean deviation from MHNSGA and Pareto optimal set is 0.8%. With over 14 million MHNSGA search space, it is 20 times faster than POCO Framework.

Some papers present the controller placement problem as a multi-objective combinatorial optimization (MOCO) problem and some important proposed objectives [1, 7, 8]. In [8], other important objectives have been proposed in a network, which plays a very significant role in deciding where the controllers are deployed. These objectives have consisted of the delay between each switch and the determined controller, the delay between each pair of the controller, and the load balancer among the controllers. Jalili et al [23] Presented a multi-objective genetic algorithm-based solution for the controller placement problem [7]. They proposed that NSGA-II-based heuristics have been presented to solve the controller placement problem. In [19], the authors presented a specialized heuristics for optimizing similar objectives which are called Pareto capacitated K-medias (PCKMs). They presented PCKM by analyzing a set of optimization objectives and the recursive solutions that the conflicts between them may be examined. New multi-objective algorithms and the controller positions have been evaluated based on these objectives. The fundamental challenge of these multi-objective models is the lack of analysis for the assignment routes and its effective factors. Some researchers have intended the load which is enforced on the controllers by the switches when solving the controller placement problem [10, 11, 20]. They determine the capacity of each controller, and if the controller is loaded, i.e., when the loading to some controllers exceeds a determined threshold, they will reapply it. A new load balancing system, called Load Balancing problem for Devolved Controllers (LBDC), is used for the authorized controllers in the data centers [10]. In this approach, each controller can locally control the traffic in a part of the switch. When the unbalanced traffic loading occurs, some of them send a part of their monitored works to other controllers to keep the workload in the dynamic form. Borcoci et al. [18] A multi-criteria optimization algorithm can be presented with regard to hypotheses without tests. Kseniti et al. [21] propose a bargaining game approach that represents a better compromise than a single goal. Jalili et al. [23] Multiple criteria analysis of controller placement problem states that the mean deviation from MHNSGA and Pareto optimal set is 0.8%. With over 14 million MHNSGA search space, it is 20 times faster than POCO Framework.

III. PROPOSED SOLUTION

In this section, considering the significance of propagation latency, hop count and link utilization for assigning the switches to the controllers, an efficient algorithm called the analytic hierarchy process (AHP) had been proposed to analyze these criteria based on different priorities which are considered by the network administrator. Moreover, a controller placement genetic algorithm (CPGA) is introduced to solve this type of controller placement problem. The implementation process and the related problems are presented in the following.

The propagation delay between i and j is obtained through the delay between two switches. The number of links traversed to reach from the origin to the destination has been considered as the number of steps i and j. Also, during the allocation, how many times a link is used has been considered as the link utilization. [23]

A. AllocationMethod

Here, the required information and the background have been explained for the AHP allocation process. Afterward, our proposed algorithm has been presented.

1) Shortest Routes

Because of the significance of the path in the allocation process, we are supposed to find all the shortest path with the same cost between two desired nodes. Figure (1) puts the procedure Dijikstra_s_d in the path matrix of Figure (1) for calculating the shortest route sp (s, d) between the source node s and the destination node d along with the cost of the shortest distance. [23]

Cost	AssignPathMat

Figure 1. A PathMat entry

Each time a placement is created, either in the initialization process or searching in a neighborhood, it should be evaluated by the objective functions. The evaluation method depends on two factors; how the controllers of the placement are distributed in the network and which switches are assigned to each controller. Hence, the method by which a controller is selected to be assigned to a switch is a crucial factor. Suppose that a placement p=[(p1, p2, . . , pk)] Is considered at this step. Given some objective functions in the controller placement problem, evaluation of this placement depends on how switches are assigned to controllers in P.

As discussed, the three criteria propagation delay (PD), hop count (HC), and link utilization (LU) are considered for assigning switches to controllers in this paper. Therefore, the assignment of a controller to a switch is a multi-adaptive problem. The AHP technique is an efficient approach to this kind of problem. For a typical switch s, the goal is assigning a controller among the candidates [p1, p2, . . , pk]. If s and one of the controllers in P are the same locations, this controller is assigned to others. Otherwise, the next steps should be executed.

TABLE I. THE COMPARISON METHOD OF THE CRITERIA

Value	*The preference situation of criterion i with respect to criterion j*	*Explanation*
1	Equally Preferred	Criterion i has no priority to Criterion j.
3	Moderately Preferred	Criterion i is slightly more important than Criterion j.
5	Strongly Preferred	Criterion i is more important than Criterion j.
7	Very Strongly Preferred	Criterion i is much more important than Criterion j
9	Extremely Preferred	Criterion i is extremely more important than Criterion j
2-4-6-8	Intermediate	Intermediate value; for example 8 indicates the more importance than 7 and less importance than 9 for criterion

TABLE II. MUTUAL COMPARISON OF THE THERE CRITERIA

	Propagation Delay	*Hop Count*	*Link Utilization*	*Weights*
Propagation Delay	1	B	C	w_1
Hop Count	1/b	1	D	w_2
Link Utilization	1/c	1/d	1	w_3

B. Mutual Comparison of Criteria

In this step, a criterion weight w_i is assigned to each criterion i, (i=1,2,3). Now, the construction procedures of these weights are explained. The method [13, 14] showed in Table 1 is applied for mutual comparison of the criteria.

The initial weights for criteria are determined by the information provided in Table 1. A 3-by-4 Table 2 is generated by the decision-maker to capture the mutual comparison results of the criteria. Afterward, the geometric mean of each row is calculated. Finally, the three obtained weights are normalized and written in the Weights column. The final phase is to yield a set of weights whose sum equals to 1 as $w = w_1, w_2, w_3$.

C. Mutual Comparison of Candidates

After the weight determination of the criteria, in the next step, a pairwise comparison of the candidates, (i.e. $p1, p2, .., pk$), should be performed based on each criterion. Therefore, it is required to do three kinds of comparison between candidates, with respect to propagation delay, hop count, and link utilization, respectively.[23].

D. Comparison method

Algorithm 1 shows the evaluation method, which is common for the three comparisons. The inputs are two values a and b which are going to be compared together with two values e_{min} and e_{max} as their lower and upper bounds, respectively. (i.e., a,b ε [e_{min}, e_{max}])[14,18].

Algorithm 1- Criterion Comparison Procedure

```
Input: a, b, emax, emin
if a==b then w= 1; return  end if
   p ← (emax- emin)/9;
   if a<b then
     for i=1 to 9 do
          if (i-1)*p < b-a  ≤  i*p then
               w ← i; return
         end if
     end for
   end if
   if a>b then
     for i=1 to 9 do
      if (i-1)*p < a-b  ≤  i*p then
              w ← 1/i; return
         end if
     end for
   end if
end if
Output: w
```

Table 3 demonstrates such a comparison for the first criterion, (i.e. propagation delay). The first k columns create a k-by-k matrix. The last column is weights obtained by normalization of the geometric means of the rows similar to Table 2. Algorithm 1 shows how this comparison is made. Here, d_{min}, d_{max} denote the minimum distance between node s and all the candidates, (i.e.P_i, i=1,…,k), respectively.

For each pair of candidates (p_i, p_j), a and b represent the shortest distances (based on propagation delay) from s to pi and pj, respectively. w_D (i,j) is calculated as the comparison value for this pair of candidates through Algorithm 1. For this, the values a, b, d_{min}, d_{max} are sent as the arguments of Algorithm 1. These values are the numbers in the i^{th} a row of the k-by-k matrix in Table 3. Afterward, the geometric mean of each row is calculated, and after normalization, they are written as the "Weights" column in Table 3. The output of this mutual comparison is the weight vector ($w_{1,pd}, w_{2,pd}, \ldots, w_{k,pd}$).[23].

TABLE III. MUTUAL COMPARISON OF THE CONTROLLER CANDIDATES BASED PROPAGATION DELAY

	P_1	P_1	….	P_1	*Weights*
P_1	1	$a_{1,2}$	….	$a_{1,k}$	$w_{1,pd}$
P_2	$1/a_{1,2}$	1	….	$a_{2,k}$	$w_{2,pd}$
…	…	…	….	…	…
P_k	$1/a_{i,k}$	$1/a_{2,k}$	….	\	$w_{k,pd}$

TABLE IV. THE RESULT TABLE OF MUTUAL COMPARISON BETWEEN CANDIDATES

	Propagation Delay	*Hop Count*	*Link Utilization*
P_1	$w_{1,pd}$	$w_{1,hc}$	$w_{1,lu}$
P_2	$w_{2,pd}$	$w_{2,hc}$	$w_{2,lu}$
…	….	….	…
P_k	$w_{k,pd}$	$w_{k,hc}$	$w_{k,lu}$

Algorithm 2- Delay Prefer Determination Procedure

```
Input: InfoMat, P, LUMat, s , n, k
WD ← [];
dmin← minimum delay between s and P;
dmax← maximum delay between s and P;
for i=1 to k do
    for j=i to k do
        a ← InfoMat(s ,P(i)).Delay;
        b ← InfoMat(s ,P(j)).Delay;
        WD(i,j) ← CriComparison( a,b, dmin, dmax); //Algorithm3
        WD(j,i) ← 1/(WD(i,j));
    end for
end for
for i=1 to k do
    wd(i) ← the geometric mean of row i;
end for
Normalization of the weight vector wd;
Output: weight vector wd
```

The process at the stage of Hop Count is similar to Algorithm2. However, here, h_{min} and h_{max} denote the minimum hop count between node s and all the candidates, (i.e. $pi, i = 1,..,k$) in P, respectively. For each pair of candidates (pi, pj), a and b represent the shortest distances (based on hop count) from s to pi and pj, respectively. The result of this comparison is $(w_{1,hc}, w_{2,hc}, \ldots, w_{k,hc})$.

First, for each pair (s,pi), all the shortest paths based on propagation delay and hop count are determined from the InfoMat. Then, the link utilization of each path is calculated as the summation of link utilization of all the links of the path (from LUMat). Finally, the average value of these link utilizations is calculated and regarded as the average link utilization of paths from s to pi. For each pair of candidates (pi, pj), a and b represent the average link utilization from s to pi and pj, respectively. Wlu (I, j) is calculated as the comparison value for this pair of candidates through Algorithm 1. Afterward, the geometric mean of each row is calculated, and these weights are normalized. The output of this process is the weight vector $(w_{1,h1}, w_{2,h1}, \ldots, w_{k,h1})$. [23].

Therefore, for a typical node s, all the candidates $(pi, i = 1,..,k)$ are mutually compared and three weight vectors $(w_{1,pd}, w_{2,pd}, \ldots, w_{k,pd})$, $(w_{1,hc}, w_{2,hc}, \ldots, w_{k,hc})$ and, $(w_{1,lu}, w_{2,lu}, \ldots, w_{k,lu})$ are made as the weights of the k candidates based on propagation delay, hop count, and link utilization, respectively. This result is depicted in Table 4 and the following matrix:

$$WC = \begin{bmatrix} w_{1,pd} & w_{1,hc} & w_{1,lu} \\ w_{2,pd} & w_{2,hc} & w_{2,lu} \\ w_{k,pd} & w_{k,hc} & w_{k,lu} \end{bmatrix}$$

IV. MAIN ALGORITHM

Finally, the Algorithm 3 is used to do the assignment of switches to controllers based on the three mentioned criteria. Algorithm3 demonstrates how the controllers in a placement $p = [P_1, P_2, \ldots, P_k]$ are assigned to all switches.

The score of each candidate $pi, i = 1,..,k$, is obtained as the sum of the products of the priority of each criterion by the priority of the candidates based on the criterion. Therefore:

$$Sc(P(i)) = w_1 * w_{i,pd} + w_2 * w_{i,hc} + w_3 * w_{i,hc} \quad i=1,\ldots k \qquad (1)$$

After obtaining the score of all candidates based on Equation (1), the candidate which is getting the maximum score is selected to be assigned to the switch s. The selected candidate is called d.

Algorithm 3-AHP-Based Assignment

```
Input: InfoMat, P, LUMat, DelayAdjMat,
    HobAdjMat, n, k
    LUMat ← Initialize(LUMat);
    w=(w1,w2,w3) ← Mutual comparison of the criteria //Table 1&2
    AssignVector ← [],  AssignPathMat ← [];
    for each node s in V do
        if s ϵ p   then
            d=AssignVector(s) ← s;  path← [s];
             Add path to AssignPathMat;
        else
            WC=(wd, wh,wlu) ← Mutual comparison of candidates
            d=AssignVector(s)
             PathMat ← []; EvalMat ← [];
             Add all shortest paths from s to d into PathMat
             EvalMat ←Evaluation of PathMat
                 WA=(wAd, wAh,wAlu) ← Mutual comparison of candidates
for paths
        end if
      Add path to AssignPathMat;
      LUMat ← UpdateULMat( LUMat, path );
    end for
Output: AssignVector, AssignPathMat, LUMat
```

Table 5 shows this matrix and the candidate's paths. Like what is described for mutual comparison of controller candidates, the path candidates: Path 1 to Path X are mutually compared with respect to the three criteria. For example, for comparing Path 1 with Path 2 with respect to propagation delay, a=D1 and b=D2, and d_{min} and d_{max} as the minimum and maximum values in the first column of the EvalMat matrix, respectively. The weight matrix $w_A = (w_{Ad}, w_{Ah}, w_{Alu})$ as the output of these comparisons. Hence, regarding to Table 5:

$$d_{min} = \min\{ D_1, D_2, \ldots, D_x \}, \; d_{max} = \max\{ D_1, D_2, \ldots, D_x \} \qquad (2)$$

TABLE V. THE X-BY-3 MATRIX EVALMAT

	Delay	*Hop Count*	*Link Utilization*
Path1	D_1	H_1	l_1
Path2	D_2	H_2	L_2
….	…….	…	…
Path$X$	D_x	H_x	L_x

A. *Positioning Method: Controller Placement Genetic Algorithm*

To find the location of controllers in software-defined networks, a genetic algorithm for solving the problem of controller placement based on the AHP method is introduced.

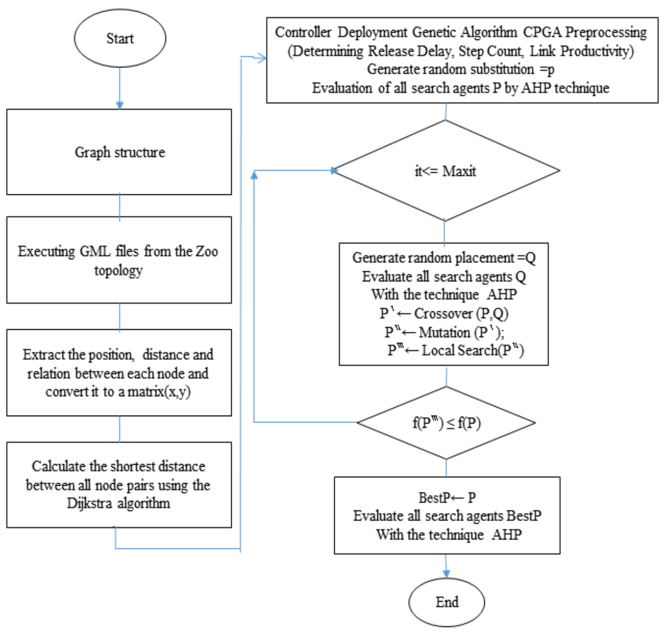

Figure 2. flowchart of the proposed methodology

The problem of controller placement is an NP-hard problem, including the location of controllers and the minimum number of controllers needed in a network. In general, the user can define different criteria for controlling the placement of the controller in a network. Some critical aspects of controller troubleshooting include:

- Minimizing the delay between each node and the controller.
- Double selection and comparison of controllers with the least delay.
- Capacity of controller

The network is represented a graph G (E, V), where V represents a set of nodes, or E represents links or edges. In addition, the distance matrix D contains the shortest path delay between each node pair where D (j, i) represents the delay between node i and j. Before starting to solve the problem, the number of controllers required is specified by k.

$$\pi_0^{\text{imbalance}}(p) = \max n_p^0 - \min n_p^0$$

$$\min \text{Delay}(V, C_V) + \min \sum \left(C_i, C_j\right) \quad (3)$$

In a large-scale network, finding a good location is the best use of the network connection between switches [2]. Rapid response and reliable connectivity between switches and controllers is a key point for SDN networks [3].

1) *Server Capacity Restriction*

Due to resource constraints such as processors, memory, and access bandwidth, a server can only manage a limited number of switches. On the other hand, overload controllers may reduce SDN performance [2]. It's difficult to get a Balance controller, so we use imbalance. Whereas from n_p^8, the number of p controllers is assigned to 8 nodes.

2) *Communication between controller*

In multi-controller SDNs, each switch is controlled by a specific controller. If one controller wants to send messages to the switch controlled by another controller, the controllers must communicate with each other [3]. Therefore, communication between controllers affects end-to-end communication performance.

Where c_v is the controller on the placement P, which is assigned to switch v, and Delay (Cv, v) denotes the shortest distance (based on the propagation delay) between switch v and controller Cv. The objective function, maximum node to controller latency, decide on whether a solution is accepted or not. Hence, it indirectly influences on how location is made.

Inputs G (E, V), k, maxIt, wicmax, DelayAdjMat, HobAdjMat, which represent the topology graph, the number of controllers, the maximum number of iteration as termination criteria, the permutation counter, delay adjacency matrix and hop adjacency matrix, respectively. It should be noted that in various experiments, k is assigned values of 1 to 5 and maxIt values of 3 to 5.

First, the preprocessing for gathering information used in evaluating solutions. Then, using a random process, a temporary initial population is created. These solutions should be evaluated later. For this purpose, for each deployment, Fig. 2 is used to assign switches to controllers in the respective location. After completing the process, the objective function equation (3) is used to evaluate these solutions. Fig. 3 uses the operators of the main genetic algorithm: Crossover and mutation. In Algorithm 4, the first placement is randomly generated in P and is considered as the best solution called BestP. Note that as soon as the controllers are deployed through the algorithm operators, the AHP process is described to assign all switches to this set of controllers and determine the controller path. Then, based on the objective function, equation (3) is evaluated.

The estimation process is performed using a random process to create an initial population. The main loop consists of these operations: A random Q is generated and the Crossover execution is generated with P. Here, a random intersection is considered where the child is obtained from two random parents P and Q. The mutation in child P is obtained by randomly substituting one of its controllers and P. Then a local search is performed in the range of mutated neighbors (P'').

Algorithm 4—Controller Placement Genetic Algorithm

```
1.   Input: G=(V,E), k, MaxIt, wicmax, DelayAdjMat, HobAdjMat
2 .   n← |V|
3 .  Preprocess stage
3 .  P← Generate a random placement
4.  Evaluate all search agents (AHP+Objective) //Call AHP-based Algorithm
5.   BestP ← P, it←1, wic←0;
6.     while it ≤ MaxIt do
7.              Q ← Generate a random placement
8 .              Evaluate all search agents (AHP+Objective) //Call AHP- based Algorithm
9.           P` ← Crossover (P,Q);
10:          P`` ← Mutation (P`);
11:          P``` ← Local Search(P``);
12:             if f(P```) ≤ f(P) then
13:                P← P```;
14:                if f(P) ≤ f(BestP) then
15.                   BestP← P;
16.                end if (line 14)
17.        else
18 .               wic ← wic+1;
19.         end if (line 12)
20.        if wic==wicmax then
21.               P← Mutation(BestP);
22.               wic←0;
23.        end if (line 20)
24.        it← it+1;
25.     Evaluate all search agents (AHP+Objective) //Call AHP-based Algorithm
26.    end while
27:  Output: BestP
```

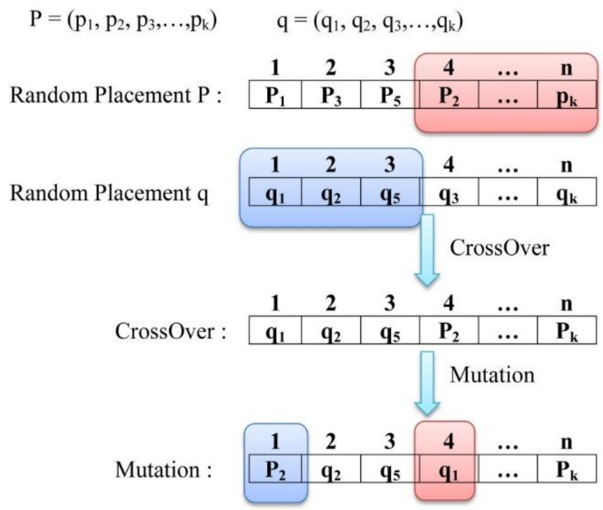

Figure 3. state chart of Genetic algorithm

V. Simulation Results

This section of the thesis describes the results of the evaluation of our Algorithm. The CPGA is used, implemented in two different ways with different weights. This indicates that due to the allocation of AHP to the CPGA, it leads to substitutions that are not only justifiable in terms of delay, but also correlate the link load. Zoo topology [22], which has been considered in several papers as the basic topology for solving the control problem, is used in our evaluation method. The AHP weight vectors and evaluations used in evaluation are listed in [23].

After the implementation of Algorithm 4, the total propagation delay and the total number of steps (respectively the sum of the propagation delays and the number of steps per node assigned to the controller) and the maximum link utilization (as the maximum link utilization of each node assigned to the controller). Given is calculated for the resulting allocation. This method is repeated 50 times and the mean values recorded. Finally, the numbers are divided by the maximum corresponding values and written on each axis. Hence, all values for the three criteria have the same time interval [0, 1].

A. Assignment analysis

In this section, the evaluation method is carried out through experiments, which are described in some planned programs. Various evaluation experiments are performed to evaluate the performance of our proposed method. In this scenario shows that the proposed model offers better quality solutions, we use two important indicators to evaluate the model. Each set of solutions is used on the basis of two indicators: "Link Load Indicator" as well as "maximum link utilization of Control Paths" or "LL (1)" and "MLUCP", respectively. [15] For this purpose, the CPGA used to solve Equation (2) is implemented in two different methods with different weights. This indicates that due to the allocation of AHP to the CPGA, it leads to substitutions that are not only justifiable in terms of delay, but also correlate the link load. All experiments are performed on a 4 GHz Intel Core i5 processor with 6 GB of RAM. Solution approaches are implemented using MATLAB 2019a. The Zoo topology, which has been considered in several papers as the basic topology for solving the control problem, is used in our evaluation method. The AHP weight vectors used in our evaluation are listed in Table 6 [23] In addition, we assume that the control input current of each node is 100kpps. This assumption is based on a reference [18].

TABLE VI. Different scenarios for AHP allocation evaluation

	$b = 9$, $c = 9$, $d = 1$	$b = 1/9$, $c = 1$, $d = 9$	$b = 1$, $c = 1/9$, $d = 1/9$	$b = 1$, $c = 1$, $d = 1$	$b = 1$, $c = 5$, $d = 5$
$k = 2$	CS1	CS2	CS3	CS4	CS5
$k = 3$	CS6	CS7	CS8	CS9	CS10
$k = 4$	CS11	CS12	CS13	CS14	CS15
$k = 5$	CS16	CS17	CS18	CS19	CS20

In summary, one can see that strictly focusing on one metric will substantially sacrifice two other criteria. However, the last two sets of parameter settings here work well. Again, focusing equally on both propagation delay and hop count far more than link utilization (the last weight vector), has resulted in the best assignment and offers a better trade-off for a minimum of costs.[23]

B. Analysis of placement

In this section, CPGA is used to solve Equation (2). We show that due to the allocation of AHP to CPGA, it leads to placements that are not only justifiable in terms of release timing, but also consider the link load balance. As mentioned before, the loading problem, especially the load control paths, is an important aspect of the controller issue and should be considered. It is important to note that if all switches to controllers are overused via multiple overlaps (shared links) or some links are over-assigned, then if the delay between switches and controls Increase, affects queue length and processor delay in the corresponding paths. Morever, if the breakdown for the shared links is complete, several assignments must be made between the switches and the controllers that include this link. Note that relocation is an expensive process. If the number of switches that need to be re-verified is high, both network management and network maintenance errors are expensive.

Therefore, link load balancing provides advanced failover and bandwidth management for SDN to assure continuous operation of the network in the event that one or more links between switches and controllers become unavailable or slow to respond.

The MLUCP and LL (l) metrics are important because they consider the load balancing on the links [15]. The lower the value of these metrics, the better the load on the links is done. We have used these metrics to evaluate the quality of our solution and we have shown that the proposed method has a better load balancing than other methods. The following link load indicator is used to evaluate the quality of placements in terms of link load balancing. Given a typical placement P, the following quantity is used for load of each link l in the network topology:

$$LL(l) = \frac{(n-1)}{m} * 2^{n-1} \quad (4)$$

Where n denotes how many times this link, l, is used in all paths between controllers and their allocated switches in the assignment imposed by placement P and m shows the total number of links (including the repeated ones) used in the paths. Based on this formula, increasing the number of link usage will be punished exponentially. The quantity obtained by (3) is used as an indicator for the quality of the assignment based on link load balancing.

Here, for the first scenario, the algorithm CPGA is run on the Equation (3) in two ways. First, the problem considers the minimum propagation delay as the only criterion for the assignment process. Second, The AHP assignment with weight setting b=1, c=5, and d=5 is taken into account. To this end, simulations are done using internet2, with 34 nodes, as our network topology and the results denote which assignment is better regarding link load balancing. [23]

MLUCP is another important metric to evaluate the quality of achieved placements with respect to assigned control paths. The MLUCP denotes the highest link utilization among all control paths from switches to their assigned controllers. This metric is stated in Equation (5).

$$MLUCP(P) = \max_{i,j \in V} DP_{ij} \quad (5)$$

Where P represents placement of controllers and as the control load of switch i imposed by placement P. In other words, given a placement P as a set of controllers, the AHP-based approach determines the controller assigned to each switch i and also the control path. Each control path has a link utilization and denotes the highest link utilization among these paths. Therefore, for each AHP weight of Table 6, after each running of the CPGA on the problem (3), the achieved solution is evaluated based on (5). For this weight, the algorithm is run for 50 times and the average values for the indicator (5) is captured as the performance metric value for this weight.

According to Table 7, AW1 and AW4 emphasize strictly on propagation delay, AW2 and AW5 impose the same preference for propagation delay and hop count but much less emphasis on link utilization, and AW3 and AW6 consider the same priority for the three criteria (propagation delay, hop count, and link utilization)

TABLE VII. DIFFERENT WEIGHTS FOR AHP-BASED ALLOCATION

Notation	*Topology*	*Weight vectors*
AW1	Abilene	b=9,c=9,d=1
AW2	Abilene	b=1,c=5,d=5
AW3	Abilene	b=1,c=1,d=1
AW4	Internet2	b=9,c=9,d=1
AW5	Internet2	b=1,c=5,d=5
AW6	Internet2	b=1,c=1,d=1

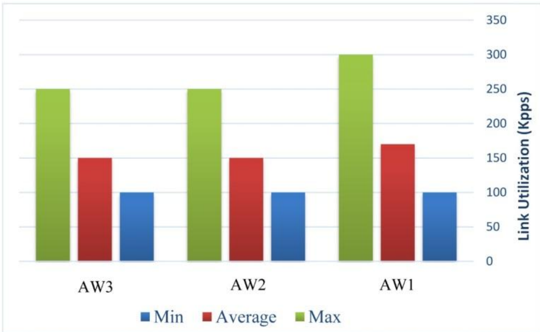

Figure 4. Link Utilization for Sample AW1 to AW3

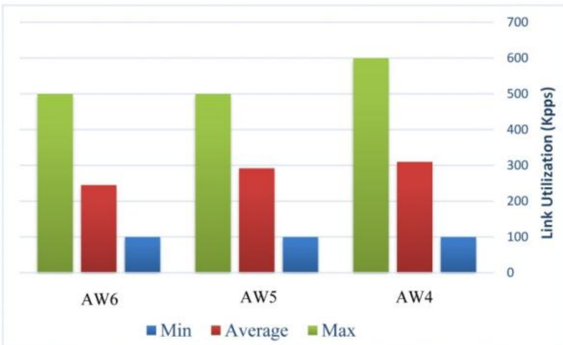

Figure 5. Link Utilization for Sample AW4 to AW6

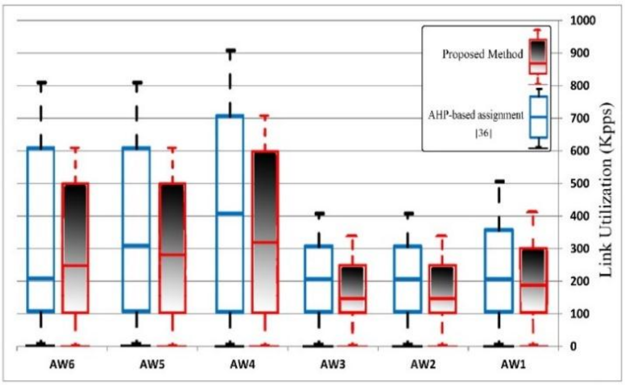

Figure 6. comparison of the link utilization

Figure 4 to Figure 6 shows the results. Figure 4 shows the link utilization for the AW1 to AW3 sample. We show the minimum, maximum and mean values for these three samples obtained in Abilene topology with 11 nodes. Also, Figure 5 shows the link utilization for the sample AW4 to AW6 with minimum, maximum and mean values in the graph with 34 nodes. Furthermore, Fig 6 It shows that the proposed method is more efficient than Mr. Jalili's method [23].

For example, for Abilene topology, the weight of AW2 and AW3 is almost similar and the performance is better than the weight of AW1. For the AW1 weight, we see that the maximum link utilization is 500 kpps. Similarly, for Internet2 topology, the weight of AW5 and AW6 is about the same and performs better than the weight of AW4. However, the weight of the AW6 is higher than the AW5, because its average is lower. When the number of nodes and links is high, the weights show their properties better. We mentioned the distribution of traffic control on the link.

It is seen that for both topologies, AW1 and AW4 are constantly expanding the use of links in the other two topologies, which means that traffic across links across the network is highly unbalanced, while in the network Internet 2, the variance reaches a peak of 23%.

In general, from the results, AHP-based allocation shows better performance than delay-based allocation. Since this value averages values for all the criteria considered, it can be considered as an agreement between the values of the minimization function and the goal of load balancing.

VI. CONCLUSION AND FUTURE WORK

In this paper, a new problem with controller placement is called "assigning a switch to controller". In order to evaluate the performance of this process, three important criteria are considered such as propagation delay, hop count and link utilization (link usage). These metrics are used to analyze the optimality of controller locations and also to allocate the controller to the switch. In addition, since a multi-criteria decision needs to be made to switch to the controller allocation problem, we have chosen AHP to solve the problem. In addition, a controller placement genetic algorithm, which is used to improve the proposed method, is proposed to solve the problem of location-allocation in the controller placement problem. CPGA implementation demonstrates the effectiveness of our proposed approach in various cases that examines the optimal location of the controllers, the optimal assignment of the switch to the controller, the delay between the controllers and the balance between them. The results showed that the proposed multi-criteria allocation method is better than the delay-based allocation in terms of link utilization. This may not only be the impetus for moving to the problem of controller placement but also the problem of controller placement related to allocation and routing.

As future work directions, further studies can be conducted to address the Mobility-based Controller Deployment Problem. To achieve better results, studying Software-defined networks for the next generation of networks, such as 5G, where user mobility affects network performance. Thus the transfer of mobility from controllers is an open research issue.

REFERENCES

[1] A. Jalili, M. Keshtgari, R. Akbari, “Optimal controller placement in large scale software defined networks based on modified NSGA-II,” Applied Intelligence, pp. 1-15, 2017.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] M. Tanha, D. Sajjadi, R. Ruby, J. Pan, “Capacity-aware and Delayguaranteed Resilient Controller Placement for Software-Defined WANs.” IEEE Transactions on Network and Service Management, 2018.K. Elissa, “Title of paper if known,” unpublished.

[3] B. Heller, R. Sherwood, N. McKeown, “The controller placement problem,” In Proceedings of the first workshop on hot topics in software defined networks, ACM, pp. 7-12, 2012.

[4] G. Wang, Y. Zhao, J. Huang, Q. Duan, J. Li, “A K-means-based network partition algorithm for controller placement in software defined network,” In Communications (ICC), 2016 IEEE International Conference on IEEE, pp. 1-6, 2016].

[5] J. Zhao, H. Qu, J. Zhao, Z. Luan, Y. Guo, “Towards controller placement problem for software-defined network using affinity propagation,” Electronics Letters, 2017.

[6] A. Sallahi, M. St-Hilaire, “Expansion Model for the Controller Placement Problem in Software Defined Networks,” IEEE Communications Letters, vol. 21, no. 2, pp. 274-277, 2017.

[7] A. Jalili, V. Ahmadi, M. Keshtgari, M. Kazemi, “Controller placement in software-defined WAN using multi objective genetic algorithm,” In Knowledge-Based Engineering and Innovation (KBEI), 2015 2nd International Conference on IEEE, pp. 656-662, 2015.

[8] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel, M. Hoffmann, “Heuristic approaches to the controller placement problem in large scale SDN networks,” IEEE Transactions on Network and Service Management, vol. 12, no,. 1, pp. 4-17, 2015.

[9] K. S. K. Liyanage, M. Ma, P. H. J. Chong, “Controller placement optimization in hierarchical distributed software defined vehicular networks.” Computer Networks, 135, 226-239, 2018.

[10] X. Gao, L. Kong, W. Li, W. Liang, Y. Chen, G. Chen, “Traffic load balancing schemes for devolved controllers in mega data centers”. IEEE Transactions on Parallel and Distributed Systems, 28(2), 572- 585, 2017.

[11] J. Liao, H. Sun, J. Wang, Q. Qi, K. Li, T. Li, “Density cluster based approach for controller placement problem in large-scale software

[12] defined networkings,” Computer Networks, no. 112, pp. 24-35, 2017.

[13] Y. Hu, T. Luo, W. Wang, C. Deng, "On the load balanced controller placement problem in Software defined networks," In Computer and Communications (ICCC), 2016 2nd IEEE International Conference on, pp. 2430-2434, 2016.

[14] T. L. Saaty, L. G. Vargas, "Uncertainty and rank order in the analytic hierarchy process," European Journal of Operational Research, vol.32 , no. 1, pp. 107-117, 1987.

[15] T. L. Saaty, "Decision making with the analytic hierarchy process,"International journal of services sciences, vol. 1, no. 1, pp. 83-98,2008.

[16] F. P. Tso, D. P. Pezaros, "Improving data center network utilization using near-optimal traffic engineering." IEEE transactions on parallel and distributed systems, 24(6), 1139-1148, 2013.

[17] L. Liu, H. Mei, B. Xie, "Towards a multi-QoS human-centric cloud computing load balance resource allocation method." The Journal of Supercomputing, 72(7), 2488-2501, 2016.

[18] H. P. Jiang, D. Chuck, W. M. Chen, "Energy-aware data center networks." Journal of Network and Computer Applications, 68, 80- 89, 2016.

[19] G. Yao, J. Bi, Y. Li, L. Guo, "On the capacitated controller placement problem in software defined networks," IEEE Communications Letters, vol. 18, no. 8, pp. 1339-1342, 2014.

[20] S. Lange, S. Gebert, J. Spoerhase, P. Rygielski, T. Zinner, S. Kounev, P. Tran-Gia, "Specialized heuristics for the controller placement problem in large scale SDN networks." In 27th International IEEE Teletraffic Congress (ITC 27), pp. 210-218, 2015.

[21]]. C. A. Wang, B. Hu, S. Chen, D. Li, B. Liu, "A Switch Migrationbased Decision-making Scheme for Balancing Load in SDN," IEEE Access, 2017.

[22] T. Wang, F. Liu, J. Guo, H. Xu, "Dynamic SDN controller assignment in data center networks: stable matching with transfers," In Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on, pp. 1-9, 2016.

[23]] .S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, M. Roughan, "The internet topology zoo." IEEE Journal on Selected Areas in Communications, 29(9), 1765-1775, 2011.

[24] A.Jalili , M.Keshtgari , R.Akbari , R.Javidan, "Multi Criteria Analysis of Controller Placement Problem in Software Defined Networks." Computer Communications, 2018.

Hossein Akbari received the B.S. and M.S. degrees in computer engineering (2015 and 2019, respectively). He is a member of Ooj Lab, and His main research interests include Software Defined Networks, Network Function Virtualization, network security and cloud computing. He completed this work when he was an M.S student of Islamic Azad University,Qazvin, Iran.

Zahra Shabani is currently a master student at Ferdowsi University – Mashhad,Iran. She is a Member of National Genius Community of Iran and his current research areas are Software Defined Networks, Network Function Virtualization and cloud computing and Robotic.

Abolfazl Toroghi Haghighat received the B.S., M.S. degrees in electronic engineering (1993 and 1996, respectively) from Electrical Department Sharif University of Technology, Tehran,Iran. He received the Ph.D. degree in computer engineering (2003) from Amirkabir University of Technology, Tehran, Iran. His research interests include high speed computer networks, wireless and mobile networks (MANET and Sensor Networks), distributed systems and distributed operating system and modeling, simulation and performance evaluation of computer systems. He has published several papers in international conferences and journals. He is currently a faculty member at the Electrical, Computer and IT Department,Qazvin Islamic Azad University, Qazvin, Iran.

How to Cite this Article:

Akbari, H., Shabani, Z. & Haghighat, A. T. (2019) A Multi-Adaptive Controller Placement Algorithm in Software-Defined Networks. International Journal of Science and Engineering Investigations (IJSEI), 8(95), 97-106. http://www.ijsei.com/papers/ijsei-89519-14.pdf