A Simulated Annealing Strategy for Reliable Controller Placement in Software Defined Networks

Sagarika Mohanty
Dept. of CSE
National Institute Of Technology
Rourkela, Odisha, India
sagarikam_23@yahoo.com

Khushboo Kanodia
Dept. of CSE
National Institute Of Technology
Rourkela, Odisha, India
kkanodia2307@gmail.com

Bibhudatta Sahoo
Dept. of CSE
National Institute Of Technology
Rourkela, Odisha, India
bdsahu@nitrkl.ac.in

Kuldeep Kurroliya
Dept. of CSE
National Institute Of Technology
Rourkela, Odisha, India
kuldeep.nitrkl2304@gmail.com

Abstract— Software defined networking (SDN) empower the service providers and enterprisers to respond quickly to changing business requirements. Due to the logically centralized but physically distributed control architecture the resilience of the control plane is a major challenging issue. Failure of a controller needs reassignment of switches to other working controllers with enough capacity. Due to this, the propagation latency may increase which will lead to the increase cost of the network. In this paper, a heuristic technique called simulated annealing strategy for reliable controller placement (SASRCP) is proposed which reduces the propagation latency and minimizes the routing cost of the network. This algorithm is compared with a greedy (GRE) and a random algorithm (RA). The evaluation results confirms that SASRCP performs better than GRE and RA.

Keywords— Software Defined Network; Reliability; Simulated Annealing Strategy for Reliable Controller Placement (SASRCP); Propagation Latency

I. INTRODUCTION

Software Defined Network (SDN) is the next generation network architecture. In today's world with huge network traffic, emergence of IOT, cloud, fog and big data applications, the need arises for efficient, faster and more resilient architecture. Placing a single controller is a bottleneck considering scalability and reliability. Placement of multiple controllers called as controller placement problem (CPP) is a major challenging issue in a wide area network as addressed by many researchers. They have considered various performance metrics such as latency, cost, resiliency, load balancing and energy.

Due to controller failure, the network elements are disconnected. These disconnected data plane elements must be allocated to other working controller with enough capacity to maintain the network performance. This will increase the latency and in turn results in the increase cost of the network. The proposed method address this issue of controller failure while considering certain constraints.

The contributions are as follows:

- Two new formulations for reliable controller placement is proposed.
- To find the initial state random method is adopted for initial placement of controllers.
- Switches are assigned to the controller according to the shortest path matrix.
- A simulated annealing strategy for reliable controller placement (SASRCP) is proposed.
- A greedy algorithm (GRE) is proposed considering the weight (node degree) of the node as a metric.
- SASRCP is compared with GRE and a random algorithm (RA).
- The objective is to reduce the routing cost of the network considering the demands of switches and controllers capacity.
- Experimental results verified SASRCP performs better compared to GRE and RA.

The remainder part is arranged as follows. Section II describes some related papers available in this area. Problem statement is discussed in section III. The next part elaborates different algorithms. Results and analysis are discussed in section V. Section VI draws the conclusion.

II. RELATED WORK

Heller et al. [1] introduced the controller placement problem by considering the average and worst case latency. Bari et al. [2] proposed a strategy that adapts with changing network conditions. Beheshti et al. [3] considered a single controller architecture and proposed a resiliency parameter for connecting the switches with the controller. In [4] authors proposed a reliability-aware controller placement which shows a tradeoff between reliability and latencies. Sallahi et al. [5] proposed a model which reduces the cost of the network considering certain constraints. In [6] Hock et al. considered the resilience issue and showed that quality and resilience are often difficult to achieve at a time and for that trade-offs is required. A placement strategy for improving survivability of control plane was presented in [7] where they considered three points:

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

Table 1. Input Parameters and Variables

$G(N,E)$	The network topology
$N = S' \cup C'$	Nodes in the network
$E = \{e_1, e_2 ...\}$	Communication links
$S' = \{s'_1, s'_2, ...s'_n\}$	OpenFlow Switches
$K' = \{k'_1, k'_2, ..., k'_p\}$	Controllers set
P	Set of controller positions
$\rho_{s'_i}$	load of switch s'_i
$\sigma_{k'_j}$	Controller capacity k'_j
$d_{s'_i k'_j}$	Propagation latency of switch s'_i and controller k'_j
P_f	Failure probability of controller
$\gamma_{k'_j} = 1$	Binary variable if controller is active at location k'_j
$\vartheta^e_{s'_i k'_j l} = 1$	Binary variable if switch $s'_i \in S'$ is linked to controller at location $k'_j \in K'$ at level-l

connectivity, capacity and recovery. The focus of Ros et al. [8] is to model reliable southbound interfaces between nodes and controllers. However, they have not considered the demand of the switches and controller capacity. In [9] Vizarreta et al. presented techniques for placement of controllers that describes the optimal working and backup paths for controllers. According to them by providing a resilient routing principle they are able to reduce the latency of the control plane, while protecting the network for a single link and node failure. Killi et al.[10] proposed a strategy which plans ahead for controller failure. In [11] Tanha et al. proposed a technique considering controller failure at different resilience levels. In the capacitated next controller placement problem [12] the authors have considered the capacity and reliability of controllers. In [13] the authors formulated the problem based on the clique concept of graph theory. Jalili et al.[14] presented a model for reliable placement of controllers. They have considered the flow setup time and inter controller latency. Huang et al. [15] used the queuing model and formulated the problem as an optimization problem which discuss the control plane utilization and response time.

III. PROBLEM STATEMENT

The objective is to minimize the routing cost with and without failure.

The objective functions are stated as follows for two scenarios:

$$\min \sum_{s'_i \in S} \sum_{k'_j \in K} \rho_{s'_i} d_{s'_i k'_j} P_f \vartheta^e_{s'_i k'_j l} \qquad (1)$$

$$\min \sum_{s'_i \in S} \sum_{k'_j \in K} \rho_{s'_i} d_{s'_i k'_j} (1 - P_f) \vartheta^e_{s'_i k'_j l} + \sum_{s'_i \in S} \sum_{k'_j \in K} \rho_{s'_{il}} d_{s'_i k'_j} P_f \vartheta^e_{s'_i k'_j l}$$

(2)

Constraints:

$$\sum_{k'_j \in K'} \gamma_{k'_j} = p \qquad (3)$$

$$\vartheta_{s'_i k'_j} \le \gamma_{k'_j} \qquad \forall s'_i \in S', \forall k'_j \in K' \qquad (4)$$

$$\sum_{k'_j \in K'} \gamma_{k'_j} = 1 \qquad (5)$$

$$\sum_{k'_j \in K'} \vartheta_{s'_i k'_j} = 1 \qquad \forall s'_i \in S' \qquad (6)$$

$$\sum_{e \in E} \sum_{s'_i \in S'} \rho_{s'_i} \vartheta^e_{s'_i k'_j} \le \sigma_{k'_j} \gamma_{k'_j} \qquad \forall k'_j \in K'$$

(7)

Equation (1) computes the routing cost either with or without failure. In $\vartheta^e_{s'_i k'_j l} = 1$, switch $s'_i \in S'$ is linked to controller at location $k'_j \in K'$ at level-l. Level-l indicates either 0 or 1. A failure probability P_f is assumed to be either 0 or 1. P_f as 0 means no failure and switches are attached to their primary controller and P_f as 1 means switches are attached to the rest available controller in the network. In failure scenario, the calculation of propagation latency will be the sum of the propagation latency from switches to their primary controller and the latency from primary controller to the other controller.

$$d_{s_i'k_j'} = d_{s_i'k_j'} + d_{k_j'k_b'} \qquad s_i' \in S \wedge k_j', k_b' \in K' \quad (7)$$

For scenario 2, a constant failure probability is considered. Equation (2) computes the routing cost both with and without failure. Constraint (3) states that total p controllers are placed in the network Constraint (4) limits a switch from allocating to a position if the controller is not deployed at that position. The total load of the switches will be less than the controller's capacity as listed in constraint (7). Constraint (5) and (6) are binary variables. It can be either 0 or 1.

IV. ALGORITHM FORMULATION

The following points play a significant role in the simulated annealing strategy for reliable controller placement problem (SASRCP).

A. Finding the Initial Configuration/state

The initial placement solution Fs represents the configuration by placing p controllers in a n nodes network. Then the switches will be connected to any one controller. In the initial configuration, the switches are connected to their nearest controller considering distance as a metric.

Random placement of controllers (RPC) randomly finds the controller locations. Here, the input is the graph $G(N,E)$, where $|N| = n$. The outputs are the set of controller locations P where $|P| = p$. The nodes are selected randomly as the controller without following any heuristics. To place the p controllers in P potential sites in random placement, each position has a uniform probability.

Algorithm 1**.** Random placement of controllers (RPC)

```
Input: $N, p$
Output: $P$
for $i = 1$ to $p$ do
    $P \leftarrow \phi$
    randomly select node $n_i \in N$
    $P \leftarrow P \cup n_i$
    $N \leftarrow N - n_i$
end for
```

Table 2.Notations

Notation	Description
Fs, Fs'	Initial feasible/placement solution and next neighboring solution
T_{ini}	Initial temperature
T_{fnl}	Final temperature
m	Number of iterations
β	Cooling factor
$f(Fs)$	Evaluate objective function of Fs

After randomly placing controllers at the potential controller locations P, the switches will be assigned to its closest feasible controller. For that the minimum distance L_d for each switch is calculated.

$$L_d = \min d_{s_i'k_j'} \quad (8)$$

Algorithm 2. $initial\_state(spm, S, P)$

```
Input: $spm, S', P$
Output: $Fs$
for each $s_i' \in S'$ do
    calculate $L_d$ for each switch $s_i'$ using $spm$
    assign switch $s_i'$ to its potential controller $k_j'$
end for
Determine $Fs \leftarrow$ controller set $P$ and a set of switches assigned to them
```

Algorithm 3. $neighbouring\_state(Fs)$

```
Input: $Fs, N, p, P$
Output: $Fs'$
for $i = 1$ to $p$ do
    randomly select a node $p_j \in P$
    $P \leftarrow P - p_j$
end for
for $i = 1$ to $n$ do
    randomly select a node $n_i \in N$
    $N \leftarrow N - n_i$
    $P \leftarrow P \cup n_i$
end for
Determine $Fs' \leftarrow$ controller set $P$ and a set of switches assigned to them
```

B. Neighboring State

Neighboring state/solution Fs' can be generated after doing some perturbations on the initial state Fs. To create a new possible configuration, reassigns all switches of a failed controller to the rest controllers. The idle active controller will now act as a switch node.

C. Annealing Schedule

Annealing schedule denotes the number of iterations at each temperature and temperature range. It starts with an initial high temperature which is an integer T_{ini} and decreased by a cooling factor β. The stopping criteria is the final temperature T_{fnl}. In SASRCP algorithm, a fixed number of iterations m at each temperature is taken. After every m iterations, temperature decreases by a factor $t = \beta t$, where $t \in [0,1]$.

D. Acceptance Function

The probability of choosing a new solution depends on both the temperature as well as the difference in the objective function. As the schedule proceeds, along with the decrease in temperature the probability of accepting a bad move decreases. The difference of the objective function between the neighboring solution Fs' and the current state Fs is calculated. Then the solution will be selected depending upon Metropolis condition. If the neighboring configuration yields better result then it will be selected. Otherwise, a random number r will be chosen from $[0,1]$. If $r < \exp^{(-\delta C/t)}$ then the neighboring solution will be the current one. Here, t is a control parameter.

E. SASRCP

In each iteration, SASRCP calculates the new state Fs' from the neighborhood of the initial state Fs in step 6 Then evaluation of the objective function value is done in step 8. The two placement solutions are compared to select the placement solution for the next iteration depending on the acceptance function. The cooling of temperature is done by a cooling factor $t = \beta t$ in step 18.

F. Greedy Algorithm (GRE)

The algorithm followed a greedy approach to determine a set of a potential location for deploying controllers. Here, the weight of a node is considered as a metric. The weight of a node represents the flow generated by it. The links available to a particular node is taken as its node degree d_n. The node having greater node degree value is considered as the candidate position for deploying controllers. Then the assignment of switches to the nearest controller and calculation of objective function is done as before.

Algorithm 4. SASRCP

Input: $Fs, T_{fnl}, T_{ini}, m, \beta$

Output: $f(Fs), Fs$: Final Controller Placement

```
1.  Let $t = T_{ini}$
2.  $Fs \leftarrow initial\_state(spm, S, P)$
3.  $f(Fs) \leftarrow evalute\_objective(Fs)$
4.  while $(t > T_{fnl})$
5.      for $(i = 0; i < m; i++)$
6.          $Fs' \leftarrow neighbouring\_state(spm, S, P)$
7.          $f(Fs') \leftarrow evalute\_objective(Fs')$
8.          $\delta C = f(Fs') - f(Fs)$
9.          if $\delta C \le 0$ then
10.             $Fs \leftarrow Fs'$
11.         else
12.             $r \leftarrow$ random variable from $[0,1]$
13.             if $r < \exp^{(-\delta C/t)}$ then
14.                 $Fs \leftarrow Fs', f(Fs) \leftarrow f(Fs')$
15.             end if
16.         end if
17.     end for
18.     $t = \beta t$
19. end while
20.
21. return $Fs, f(Fs)$
```

Algorithm 5. GRE

Input: $G(N,E), spm$

Output: $Fs, f(Fs)$

```
$P \leftarrow \phi$
for every node $n_i \in N$ do
    $d_n = \deg ree(n_i)$
end for
$d_n = sort(d_n)$
for $i = 1$ to $p$ do
    $P \leftarrow P \cup d_n[1]$
```

```
    $d_n \leftarrow d_n - d_n[1]$
end for
$Fs \leftarrow initial\_state(spm, S, P)$
$f(Fs) \leftarrow evalute\_objective(Fs)$
return $Fs, f(Fs)$
```

V. RESULT AND ANALYSIS

Table 3. Parameters used in SASRCP algorithm

SA Parameters	Description	Value
T_{ini}	Initial temperature	10
T_{fnl}	Final temperature	0.001
m	Number of iterations	500
β	Cooling factor	0.9

For simulation, various parameters are considered such as initial temperature, cooling factor and stopping criterion. The initial temperature is set to $10^0 C$ to start the algorithm. It is observed that at 0.9 the cooling ratio β is giving better result so this is considered as the cooling factor and used a maximum of 500 iterations as stopping criterion for each temperature. The final temperature is taken as 0.001. The simulation is conducted in PYTHON to validate the algorithms and for finding the solutions. Three networks are taken from the topology zoo for the analysis: Agis (25 nodes), Geant (37 nodes) and Surfnet (50 nodes). The analysis of results is based on propagation latency. To obtain propagation latency, the distance is calculated by using the Haversine formula with the latitude and longitude information of nodes.

Here, inter controller latency is taken into consideration in both failure condition and in normal situation (no failure). The aim is to minimize the propagation latency so that the routing cost will also get reduced. In scenario 1, the routing cost is calculated using equation 1. The failure probability P_f is assumed to be either 0 or 1. $P_f = 0$ indicates no failure and $P_f = 1$ means one failure and the switches will be reassigned to the rest controllers present in the network. In scenario 2, the routing cost is calculated using equation 2. Here, a constant failure probability $P_f = 0.1$ is considered.

In the network with the increasing number of controllers, the latency (both switch to controller and inter controller) decreases. This reduces the routing cost. In SASRCP we assign switches to the controller according to the shortest path matrix both in failure and no failure condition. This reduces the propagation latency.

Fig.1. shows the impact of propagation latency on the increasing number of controllers. From the results it is observed that with increase in controller number, propagation latency decreases both in failure free and failure condition. SASRCP is giving better results compared to GRE and RA.

Fig.2. shows the impact of propagation latency on the increasing number of controllers with a constant failure probability $P_f = 0.1$. From the results it is observed that with increase in controller number, propagation latency decreases and SASRCP performs better compared to the other two.

VI. CONCLUSION

In this paper, two new formulations for the reliable controller placement problem are proposed which minimizes the routing cost of the network while considering the demands of switches and controllers capacity. A random method is adopted to find out the location of the controllers for the initial configuration. Then assignment of switches to the controllers is done based on the shortest path matrix.

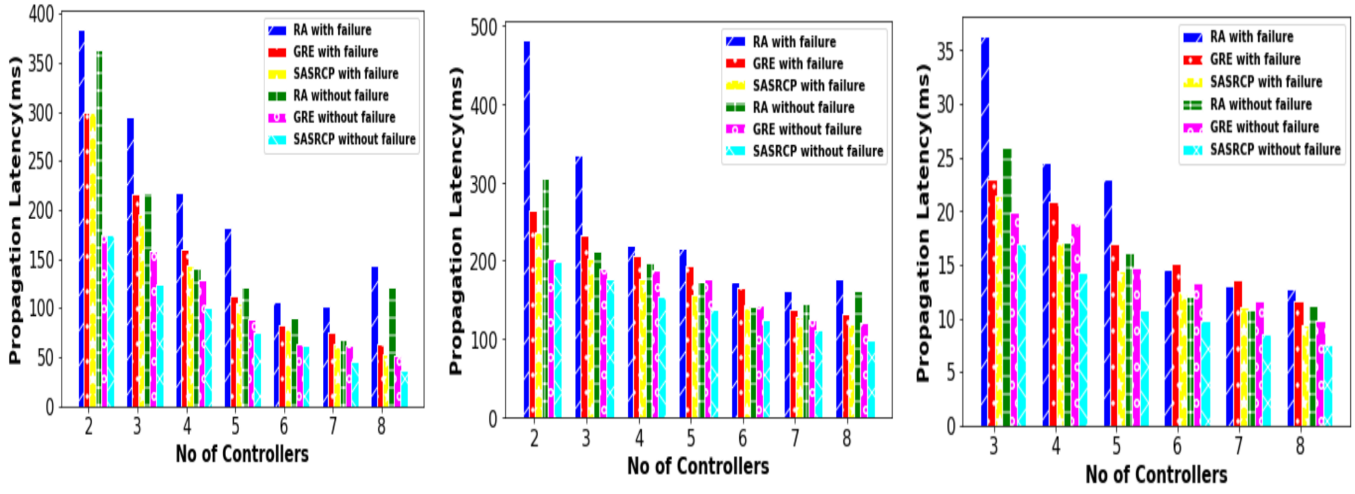

Fig. 1. Scenario 1. $P_f = \{0,1\}$ **for Agis, Geant and Surfnet networks**

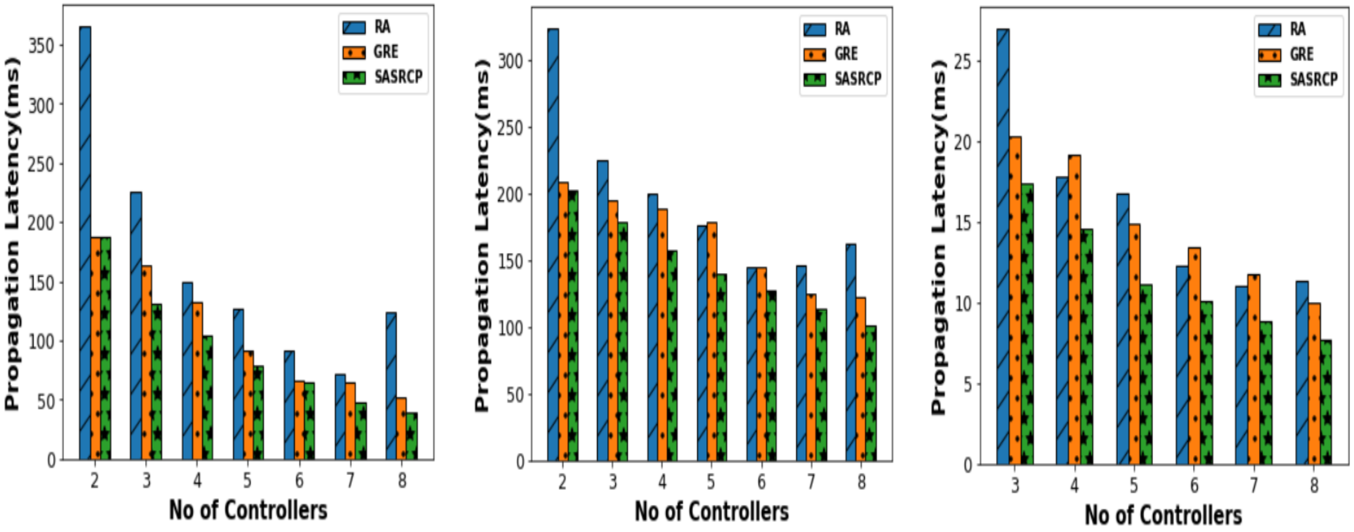

Fig. 2. Scenario 2 with $P_f = 0.1$ for Agis, Geant and Surfnet networks

The simulated annealing strategy for reliable controller placement algorithm (SASRCP) is applied on the three network topologies. From the results it is found that SASRCP is giving better performance while comparing it with GRE and random algorithm (RA). In the future work some other reassignment and load balancing techniques will be considered on some other network topologies.

REFERENCES

[1] Heller, R. Sherwood, and N. McKeown, “The controller placement problem”. *In Proc. HotSDN'12*, pp.7-12,2012.

[2] 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”. *In International Conference on Network and Service Management*, pp 188-May 2013.

[3] Beheshti, Neda, and Ying Zhang. "Fast failover for control traffic in software-defined networks." In *2012 IEEE Global Communications Conference (GLOBECOM)*, pp. 2665-2670. IEEE, 2012.

[4] Y. N. Hu, W. D. Wang, X. Y. Gong, X. R. Que, and S. D. Cheng, “Reliability-aware Controller Placement for Software-Defined Networks”. *in Proc. IFIP/IEEE Int. Symp. IM*, pp.675, 2013.

[5] S. Afrim and St-Hilaire M., “Optimal Model for the Controller Placement Problem in Software Defined Networks*”, IEEE Communications Letters*,Vol.19,No.1,January 2015.

[6] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran-Gia, “Pareto-Optimal Resilient Controller Placement in SDN-based Core Networks,” *in ITC*, Shanghai, China, 2013.

[7] Tanha, Maryam, Dawood Sajjadi, and Jianping Pan. "Enduring node failures through resilient controller placement for software defined networks." In *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1-7. IEEE, 2016.

[8] Müller, Lucas F., Rodrigo R. Oliveira, Marcelo C. Luizelli, Luciano P. Gaspary, and Marinho P. Barcellos. "Survivor: An enhanced controller placement strategy for improving SDN survivability." In *2014 IEEE Global Communications Conference*, pp. 1909-1915. IEEE, 2014.

[9] Ros, F.J.,Ruiz, P.M.,”On Reliable Placements in Software-Defined Networks”, *Computer Communications, Volume 77, Elsevie*r, 1 March 2016,pp.41-51.

[10] Vizarreta, Petra, Carmen Mas Machuca, and Wolfgang Kellerer. "Controller placement strategies for a resilient SDN control plane." In *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pp. 253-259. IEEE, 2016.

[11] Killi, Bala Prakasa Rao, and Seela Veerabhadreswara Rao. "Controller placement with planning for failures in software defined networks." In *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1-6. IEEE, 2016.

[12] Killi, Bala Prakasa Rao, and Seela Veerabhadreswara Rao. "Capacitated next controller placement in software defined networks." *IEEE Transactions on Network and Service Management* 14, no. 3 (2017): 514-527.

[13] Tanha, Maryam, Dawood Sajjadi, Rukhsana Ruby, and Jianping Pan. "Capacity-aware and delay-guaranteed resilient controller placement for software-defined WANs." *IEEE Transactions on Network and Service Management* 15, no. 3 (2018): 991-1005.

[14] Jalili, Ahmad, Manijeh Keshtgari, and Reza Akbari. "A new framework for reliable control placement in software-defined networks based on multi-criteria clustering approach." *Soft Computing*: 1-20,2019.

[15] Huang, Victoria, Gang Chen, Qiang Fu, and Elliott Wen. "Optimizing Controller Placement for Software-Defined Networks." In 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 224-232. IEEE, 2019.