2020 IEEE 4th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC 2020)

A Multi-controller deployment method based on PSO algorithm in SDN environment

Yi Li, Wenwen Sun, Shaopeng Guan*
School of Information and Electronic Engineering, Shandong Technology and Business University
Yantai, China
13270753237@163.com, s15956165569@163.com, konexgsp@gmail.com

***Abstract*—With the development of emerging information technologies, the underlying network architecture is transforming into Software Defined Networks (SDN). In order to improve the reliability and resilience of the network, the Software Defined Networks is usually managed by multiple controllers. How to effectively deploy multiple controllers is a key issue in SDN. In this paper, the PSO (particle swarm optimization) algorithm is adopt for controller deployment of SDN, and a new SDN multi-controller deployment method is designed based on PSO using dynamic parameters. The algorithm abstracts the multi-controller deployment scheme into particles. By improving the global search capability and convergence accuracy of the PSO algorithm, the switch-controller average transmission delay is optimized to achieve effective deployment of multiple controllers in a continuous two-dimensional space. Experiments are conducted to compare the algorithm before and after the improvement. The results show that the scheme is effective and the average transmission delay of the switch-controller is reduced by 10% after the algorithm is improved.**

Keywords*—*software defined networks; controller; deployment; particle swarm optimization

I. INTRODUCTION

With the explosive development of information technology as cloud computing, Internet of Things and big data, higher demands are placed on the network to process massive amounts of data, such as lower packet loss rate, millisecond delay, and more harsh security level. Traditional network architecture is difficult to meet these requirements, so a new type of network architecture, SDN (software defined networks), was emerged.

SDN is a new type of network architecture proposed by Stanford University's clean slate research group [1]. It separates the coupled data forwarding function and control function in the traditional network, and sets up independent control layer and data layer. The control layer controls data forwarding of the underlying SDN switch by sending flow tables. Compared with traditional networks, the simplified control plane design improves the efficiency of network management while providing flexibility and scalability features. These advantages of SDN have led to the research and attention of many scholars, which are increasingly being combined with emerging technologies such as the Internet of Things, cloud computing, and big data. It can be seen that the application of SDN to emerging technologies that cannot be met by traditional network architecture is a very hot topic.

In SDN, the control layer occupies a crucial position. To ensure the robustness of the control layer, SDN usually adopts a logical centralized multi-controller deployment mode, and the number of controllers determines the cost of the network and the average transmission delays. As early as 2012, Heller B et al. proposed this problem that is named as CPP (Controller Placement Problem), and pointed out that the CPP is a multi-objective optimization problem. Its optimal solution cannot be found in limited computer resources [2]. Many scholars have studied and explored CPP. In literature [3], J. X. Liao, et al proposed a density-based controller deployment scheme. The scheme divides the network into several sub-networks, which can improve the time consumption and transmission delay to some extent, but the effect is not satisfactory. in literature [4], S. Lange, et al discussed the possibility of applying meta-heuristic algorithms in large-scale SDN networks, and proposes a controller deployment scheme based on the Pareto optimal solution, which can provide operators controller deployment optimal solution based on Pareto according to specific performance indicators. However, this solution has the disadvantages of a long deployment time and high resource consumption. In literature [5], U. Huque M. T. I, et al introduced a CPP solution based on dynamic flow management algorithm. The solution introduces a dynamic load balancing strategy, which can realize rapid deployment of controllers in large-scale SDN networks, but does not consider the reliability of deployment. In literature [6], Z. Zhang, et al, according to the enhanced learning algorithm, put forward a Q-placement multi-controller deployment algorithm to improve the convergence efficiency of the algorithm.

Considering the unreachability of CPP optimal solution, the heuristic algorithm is applied. PSO (Particle Swarm Optimization) is a heuristic group intelligence algorithm [7], which abstracts the solution of the problem into particles and finds the optimal solution to the problem in the search domain by iteration. In literature [8], C. Gao, et al applied PSO to CPP and proposed global delay indicators. In literature [9], A. K. Singh, et al proposed a reliable multi-controller deployment model, applying the teaching and learning algorithm to the problem solving, and comparing it with the PSO algorithm. However, the new CPP index proposed by literature [8] [9] both weakened the influence

978-1-7281-4390-3/20/$31.00 ©2020 IEEE

of the switch-controller average transmission delay index. In view of the shortcomings of PSO as poor global search ability and low convergence accuracy, we introduce a dynamic parameter strategy and designs an algorithm based on PSO using dynamic parameters. Then the algorithm is applied to the solution of CPP to derive the optimal controller deployment scheme in SDN network.

II. PROBLEM DESCRIPTION

A. Controller Plcement Problem

For a given SDN topology, it is important to consider the number of placed controllers, the location of the controllers, and the mapping relationship between the controllers and the switches. It is named as the controller placement problem (CPP). In the SDN topology, there are three transmission delays between the nodes: the transmission delay between switch and switch, the transmission delay between the switch and the controller, and the transmission delay between controller and controller. Among them, the transmission delay between the switch and the controller in the actual network is the main indicator to measure the quality of the deployment scheme. Since the CPP can be modeled as a multi-objective optimization problem, we select the average delay and maximum delay between the switch and the controller as the deployment indicators, then give the mapping relationship between the controller and the switch and the location of each controller. The CPP is an NP-Hard problem and it is impossible to find an optimal solution in a finite resource [2].

B. Model Establishment

The SDN consists of a set of controllers and switches. Each switch is connected to only one controller. The controllers are interconnected and the switches are interconnected by a controller. By referring to controllers and switches as physical nodes and abstracting them into points, the links between physical nodes are abstracted as edges, and the network can be modeled as an undirected graph: $G=(M,E)$. Where M is a collection of all physical nodes in the network topology, and E is a collection of all edges in the network topology. Assuming that the number of switches is N and the number of controllers is K, the entire network is divided into K classes, and the switches in each class belong to one controller [2]. Then the set of switches can be expressed as:

$$V=\left\{v_1,v_2,...,v_i\right\}\ i=1,2,...,N. \tag{1}$$

v_i indicates the i-th switch.

The set of controllers can be expressed as:

$$C=\left\{c_1,c_2,...,c_j\right\}\ j=1,2,...,K. \tag{2}$$

c_j indicates the j-th controller.

Assume that the set of switches controlled by the j-th controller is CV_j, The controller to which the i-th switch belongs is represented by $C(v_i)$, then:

$$CV_j=\left\{v_i\in V:C\left(v_i\right)=c_j\right\}. \tag{3}$$

$$C\left(v_i\right)=\min d\left(v_i,c_j\right),\ j=1,2,...K. \tag{4}$$

Where $d\left(v_i,c_j\right)$ represents the distance between the controller and the switch, and min represents the minimum function.

Naming the set of controllers to which each switch belongs is $Label$, then:

$$Label=\left\{C\left(v_1\right),C\left(v_2\right),...,C\left(v_i\right)\right\},\ i=1,2,...,N. \tag{5}$$

Set the average transmission delay between the switch and its mapping controller as $Delay$, then the formula is as shown in equation (6):

$$Delay=\frac{1}{N}\sum_{v_i\in V}d\left(v_i,C\left(v_i\right)\right),\ i=1,2,...,N. \tag{6}$$

Set the maximum transmission delay between the switch and its mapping controller to $MaxDelay$. The formula is as shown in equation (7):

$$MaxDelay=\max d\left(v_i,C\left(v_i\right)\right),i=1,2,...,N. \tag{7}$$

Where max represents the maximum function.

Based on the above definitions, we use the average transmission delay between the switch and the controller as the objective function, and design a new controller deployment algorithm to finally obtain the location of the controllers and the mapping relationship between the controllers and the switches.

III. ALGORITHM DESIGNING

A. Particle Swarm Optimization

Eberhart and Kennedy proposed the PSO algorithm in 1995[7]. It is a natural population-based stochastic optimization algorithm. The PSO algorithm searches for the optimal solution by iteration through a population called particles. The iterative process of each particle is mainly affected by two factors. One is the individual history optimal and the other is the global optimal. Since it emerges, the PSO algorithm has been successfully applied in many fields. In this paper, we propose to apply the PSO algorithm to the solution of CPP.

For CPP, the following initial definition is given as follow [7]:

1) Position vector of controller coordinates

It consists of the two-dimensional coordinates of each controller, that is, if k controllers need to be deployed, the position vector of the controller coordinates is as shown in equation (8):

$$X_i = \left(a_1^i, b_1^i, a_2^i, b_2^i, ..., a_k^i, b_k^i\right). \quad (8)$$

Where X_i is the position vector of the controller coordinates of the i-th deployment scheme, a_k^i and b_k^i are the abscissa and ordinate of the k-th controller of the i-th deployment scheme.

2) Speed vector of controller coordinates

For the position vector of the controller coordinates, the velocity vector needs to be set for each dimension, and the velocity vector is related to the positional movement of the particles. Then the velocity vector of the controller coordinates is as shown in equation (9):

$$V_i = \left(v_1^i, v_2^i, ..., v_{2k-1}^i, v_{2k}^i,\right). \quad (9)$$

Where, V_i is the speed vector of the controller coordinates of the i-th deployment scenario, and k is the number of deployed controllers.

3) Speed update formula

The speed update for each deployment scenario is shown in Equation (10):

$$V_{id}^{t+1} = \omega V_{id}^t + c_1 rand\left[0,1\right]\left(P_{id} - X_{id}^t\right) + c_2 rand\left[0,1\right]\left(P_{gd} - X_{id}^t\right). \quad (10)$$

The superscript indicates the t-th iteration. ω is the inertia factor. c_1 and c_2 are the acceleration constants, the former is the individual learning factor of each particle, the latter is the social learning factor of each particle. $rand\left[0,1\right]$ is the random number in the interval $\left[0,1\right]$. P_{id} is the d-th dimension of the individual optimal solution of the i-th deployment scenario. P_{gd} represents the d-dimension of the global optimal solution. X_{id}^t represents the d-th dimension of the position vector of the controller coordinates of the current deployment scenario.

4) Location update formula

$$X_{id}^{t+1} = X_{id}^t + V_{id}^{t+1}. \quad (11)$$

Where X_{id}^{t+1} and X_{id}^t represent the position vector of the controller coordinates of the t-th and (t+1)-th iterations of the i-th deployment scheme respectively. V_{id}^{t+1} represents the speed vector of the controller coordinates of the (t+1)-th iteration of the i-th deployment scheme.

Based on the above definition, the PSO algorithm can be applied to the CPP solution. To solve the problem, only the number of initial random deployment schemes, the number of iterations, and the ω, c_1, c_2 constants ought to be set . However, the PSO algorithm with static parameters has the disadvantages of being easy to fall into local optimum and poor convergence at the end. Therefore, we introduce dynamic parameter strategy.

B. Dynamic Parameter Strategy

In the process of solving CPP, all the possible solutions in the search space are needed in the early stage, and the latter needs to quickly converge to the optimal value. The three key parameters of the PSO algorithm ω, c_1, c_2 are static constants. Although the algorithm settings are simplified, they cannot meet the different requirements of the pre- and post-CPP. In this paper, we design an iterative-based dynamic parameter strategy to overcome the shortages.

When setting a large value of the inertia factor ω in the PSO algorithm, the speed vector of the controller coordinates will inherit the speed vector of the controller coordinates of the previous generation deployment scheme, which is not conducive to the fast convergence of the algorithm later. But it can show more random nature, which is needed to enhance the search breadth in the early stage of the algorithm. The dynamic inertia factor $d\omega$ is as shown in equation (12):

$$d\omega = \omega_{\max} - \left(\omega_{\max} - \omega_{\min}\right) \times \left({}^{t}\!/_{T}\right). \quad (12)$$

In equation (12), $\omega_{\min}$ and $\omega_{\max}$ are the minimum and maximum values of the preset inertia factor respectively. T is the number of algorithm iterations, and t is the number of algorithm runs.

The individual learning factor c_1 and the social learning factor c_2 in the PSO algorithm, respectively, affect the degree to which each deployment scenario is influenced by the individual historical best and the global optimal deployment scenario. In order to enhance the randomness of the algorithm in the early stage, it is necessary to improve the influence of the individual history optimal while weakening the global optimal impact.

Therefore, as the algorithm running, c_1 should be smaller and smaller, and c_2 should gradually increase. The dynamic individual learning factor dc_1 and the dynamic social learning factor dc_2 are as shown in equations (13) and (14), respectively:

$$dc_1 = c_1 \times \left(t/T \right). \tag{13}$$

$$dc_2 = c_2 \times \left(1 - t/T \right). \tag{14}$$

In equation (14), c_1 and c_2 are the maximum values of the preset individual learning factors and social learning factors, respectively.

C. Improved PSO

The dynamic parameter strategy is introduced and applied to the CPP solution. The flow chart of the improved PSO algorithm is shown in Fig. 1.

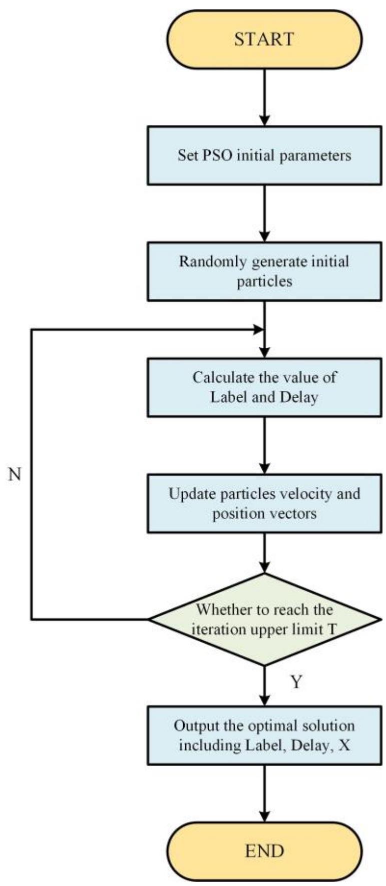

Fig. 1. DPPSO.

In the flowchart, the initial set parameters include n, T, $\omega_{\min}$, $\omega_{\max}$, c_1, c_2 and $G = (M, E)$. The values of Label and Delay are derived from equations (5) and (6), respectively. The velocity and position vector of the particle are calculated by the formula (10) and (11) from the dynamic parameter strategy. When the iteration upper limit T is reached, the Label value and the controller coordinate X of the optimal solution are output.

IV. Simulation and Evaluation

In order to verify the performance of DPPSO, we conduct experiments on the real public network Savvis. Assume that the algorithm randomly generates 100 initial controller deployment scenarios, and the number of iterations is set to 60. By changing the number of controllers, the switch-controller average transmission delay index is recorded. At the same time, in order to verify the improvement effect of the algorithm, we also compare the results of PSO and improved PSO algorithms. We use MATLAB language to implement the algorithm. The development tool is MATLAB R2018b, the development environment is windows 10 professional (64 bit), intel(R) Core (TM) i5-7300HQ CPU @ 2.50GHz, 16.00G memory.

The average transmission delay of between the switch and the controller is an important indicator in the CPP. We conduct experiments on the Savvis topology and record the average transmission delay of between the switch and the controller under different controller numbers (2-5 controllers). The result is shown in Fig. 2.

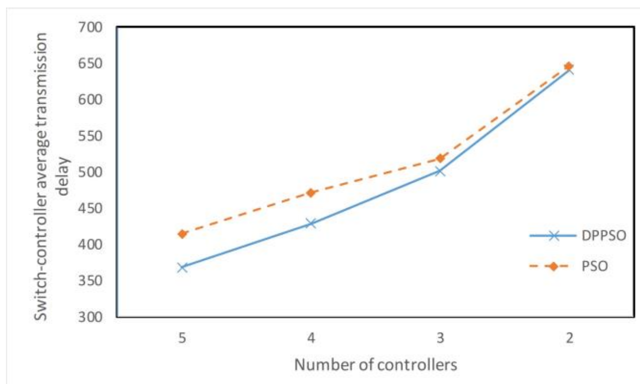

Fig. 2. Switch-controller average transmission delay.

As can be seen from Fig. 2, when the number of controllers is small, the average transmission delay between the switch and the controller is relatively high, which is caused by an increase in the average controller area of the controller. In addition, the deployment scheme obtained by the improved PSO algorithm is significantly better than the PSO in this indicator, and the gap between the two becomes larger as the number of controllers increases. Therefore, the improved PSO algorithm will perform better in large-scale actual networks that need to deploy a larger number of controllers.

The improved PSO algorithm is also better than the PSO algorithm on the average transmission delay between the switch and the controller, which indicates that the search breadth of the algorithm is guaranteed in the early stage. Then in order to verify the convergence accuracy of the algorithm later, the MATLAB deployment result graph under the four controllers is shown in Fig. 3.

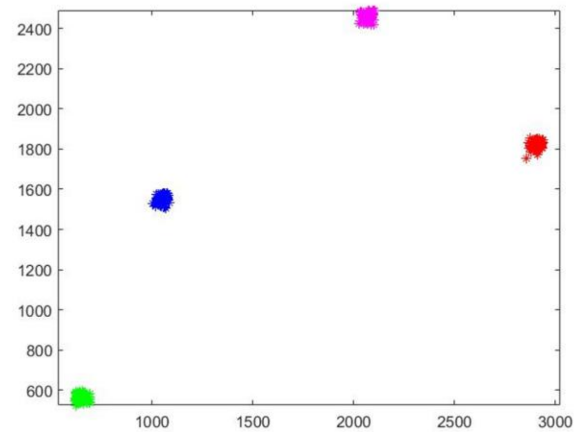

Fig. 3. MATLAB deployment results.

It can be seen from Fig. 3 that 100 particles are closely concentrated, which indicates that all the deployment schemes in the later stage of the algorithm converge rapidly toward the global optimal deployment scheme. Therefore, the improved PSO algorithm has good convergence characteristics.

V. Conclusion

With the rapid development of cloud computing, Internet of Things, big data and other information technologies, the SDN architecture has entered people's field of vision. In order to ensure the stable operation of large-scale SDN networks, distributed deployment of multiple controllers is an urgent problem to be solved. We propose to adopt the PSO algorithm to multi-controller deployment of SDN. In view of the shortcomings of PSO algorithm as poor global search ability and slow convergence in the later stage, we introduce dynamic parameter strategy and improve PSO algorithm. The algorithm has better search breadth and convergence accuracy, and finally obtains the lowest transmission delay scheme between the switch and the controller in a continuous two-dimensional space by iterative method. At last, the experiment is conducted in the real open network topology Savvis, and results are compared with the PSO algorithm. The results show that the improved PSO algorithm solves the problem of poor convergence accuracy in the later stage, and it is superior in the average transmission delay index between the switch and the controller.

References

[1] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, et al, “Software-defined networking: A comprehensive survey,” Proceedings of the IEEE, vol. 103(1), 2015, pp.14-79.

[2] B. Heller, R. Sherwood and N. McKeown, “The controller placement problem,” *SIGCOMM Comput. Commun. Rev.*, vol. 42(4), Sep. 2012.

[3] J. X. Liao, H. F. Sun, J. Y. Wang, et al, “Density cluster based approach for controller placement problem in large-scale software defined networkings,” *Comput. Netw.*, vol. 112, Jan. 2017, pp. 24-35.

[4] S. Lange, S. Gebert, T. Zinner, et al, “Heuristic approaches to the controller placement problem in large scale SDN networks,” IEEE Trans. Netw. Service Manage, vol. 12(1), Mar. 2015, pp.4-17.

[5] U. Huque M. T. I , W. Si , G. Jourjon , et al, “Large-Scale Dynamic Controller Placement,” IEEE Transactions on Network and Service Management, vol. 14(1), 2017, pp.63-76.

[6] Z. Zhang, L. Ma, K. Leung, et al, “Q-Placement: Reinforcement-Learning-Based Service Placement in Software-Defined Networks,” Proc. 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2018, pp. 1527-1532.

[7] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” Proc. The Sixth International Symposium on Micro Machine and Human Science, IEEE, 2002, pp. 39-435.

[8] C. Gao, W. Hua, F. Zhu, et al. “A Particle Swarm Optimization Algorithm for Controller Placement Problem in Software Defined Network,” Proc. Lecture Notes in Computer Science, Springer, Cham, 2015, pp. 44-54.

[9] A. K. Singh, N. Kumar and S. Srivastava, “PSO and TLBO based Reliable Placement of Controllers in SDN,” I.J. Computer Network and Information Security, vol. 2, 2019, pp. 36-42.