OPTIMIZED ADAPTIVE LOAD BALANCING METHOD IN SDN NETWORKS USING THE ADAPTIVE ANT COLONY APPROACH

Vadym Shchur, *PhD Student*
NTUU "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine; e-mail: vadimmshchur@gmail.com
Ahmad Ali AlZubi, *Dr.Sc., Prof.*
King Saud University, Riyadh, Saudi Arabia

https://doi.org/10.23939/istcmtm2023.04.

Abstract. In modern software-defined networks, providing efficient load balancing is a crucial task for optimal resource utilization and ensuring stable quality of service. To achieve these goals, in this paper, we propose a new innovative load-balancing method for SDN networks based on an anticolonial approach with dynamic parameter settings.

This proposed method demonstrates high efficiency in the face of variable network dynamics and diverse node loads. Its main advantage is the ability to adapt to changing load and traffic conditions in real-time. The algorithm continuously analyses the load on the nodes and dynamically adjusts the weighting factors to ensure optimal traffic distribution.

The proposed method stands out due to its ability to effectively maintain load balance under a variety of calls and loads, making it a powerful tool for ensuring reliability and performance in networks.

Key words: Software-defined networking, load balancing, SDN networks, adaptive anticolonial method, dynamic parameter adjustment.

1. Introduction

The growth of data and the increasing demands on network performance in today's world are challenging engineers and researchers to improve the performance of computer networks. From responding to peak loads to ensuring data transfer speeds and minimizing latency, efficient management of network resources is becoming critical.

Software-defined networking (SDN) represents a new approach to networking, where network control and management are separated from the physical infrastructure. This architecture allows for faster deployment of new services, reduced maintenance costs, and improved scalability and flexibility.

However, the introduction of SDN networks introduces new challenges, in particular, those related to load balancing. Ensuring an even distribution of traffic between network nodes is becoming a key task to maintain a stable data transfer rate and prevent the overloading of individual network components.

To solve this problem, there are many methods and approaches, including load balancing, using backup routes, adaptive algorithms, and others. However, most of the existing approaches have their limitations, especially when it comes to adapting to changes in the network and load.

2. Software-defined networks

The integration of SDN networks leads to new challenges, in particular those related to load balancing. Ensuring an even distribution of traffic between network nodes is becoming a key task to maintain a stable data rate and prevent the overloading of individual network components. To solve this problem, many methods and approaches have been proposed, such as load balancing, the use of backup routes, adaptive algorithms, etc.

Despite significant advances in the field of load balancing in software-defined networks (SDNs), most existing methods have certain limitations and shortcomings. An overview of the most common shortcomings will help to understand the need for improvement and the development of new approaches.

2.1. Static parameters

Many modern load-balancing methods, such as Round Robin described by the authors in [4], use static parameters. This leads to insufficient adaptation to changes in the network, as such parameters remain unchanged for a long time. As a result, the flexibility of load balancing is limited, which can lead to suboptimal results.

2.2. Failure to consider load

In [5], the authors consider the "Least Connections" algorithm but do not take into account the difference in load between nodes, so they can distribute traffic without considering the current load on the nodes. This leads to the overloading of some nodes and underutilization of others, which can degrade system performance.

2.3 Lack of flexibility

The "Weighted Round Robin" algorithm proposed by the authors [6] has limited flexibility in choosing routes, and may not provide sufficient optimization of load balancing and congestion avoidance. The limited ability to maneuver routes limits the optimization potential.

2.4. Computational cost

Some sophisticated methods, such as "Adaptive Load Balancing" described in [7], may require significant computational resources to perform calculations and make decisions. High computational complexity can affect the performance of the network and make it less efficient.

2.5. Vulnerability to attacks

"Random Load Balancing" proposed by the authors in [8] may be vulnerable to attacks. The main reason for this vulnerability is that the method randomly distributes traffic between nodes without taking into account their current load. Also, due to the randomness of traffic distribution, the Random Load Balancing method can lead to uneven load distribution between nodes. Certain nodes may be overloaded, while others may be underutilized. This can create an opportunity for attacks, such as DDoS, that target specific nodes.

Therefore, to overcome these limitations and shortcomings, it is important to develop new load-balancing methods that are flexible, adaptive, and meet the requirements of modern SDN networks.

3. Goal

The main purpose of this paper is to develop a new load-balancing method for software-defined networks based on an Ant Colony approach with dynamic parameter tuning, as well as to achieve optimal traffic distribution and improve network performance under conditions of variable load and network dynamics.

4. Materials and methods

4.1 The "Adaptive Ant Colony" method

The Adaptive Ant Colony method is one of the methods of load balancing in software-defined networks (SDN), which is based on the ant algorithm with adaptation elements. In this method, ant agents model the behavior of ants in nature, which interact with each other and their environment using pheromones. Pheromones are used to determine the choice of optimal routes for traffic transmission in the network.

One of the main advantages of the Adaptive Ant Colony method is its ability to adapt to changes in the network and load. To achieve this adaptability, additional parameters are introduced that regulate the weights of pheromones and their evaporation rates. These parameters were automatically adjusted depending on the network status and traffic volume.

The detailed algorithm describes the Adaptive Ant Colony method, namely the calculation of updated pheromone weights and evaporation rates, which ensures the balance of routes for traffic.

A formula for updating the pheromone weight:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (1)$$

The formula for updating the evaporation rate:

$$\rho_{ij}(t+1) = (1-\alpha)\,\rho_{ij}(t) + \alpha\cdot\frac{f_{ij}(t)}{\max_{i,j} f_{ij}(t)}, \quad (2)$$

Where $\tau_{ij}(t)$ pheromone weight on the edge (i,j) at the time point t, ρ is the pheromone evaporation rate; $\Delta\tau_{ij}(t)$ increase in pheromone on the rib (i,j) at the time point t; α is the coefficient of influence of traffic intensity on the evaporation rate update; $f_{ij}(t)$ traffic intensity on the edge (i,j) at the time point t.

The Adaptive Ant Colony algorithm has its limitations and drawbacks that may affect its effectiveness in certain conditions. Among them is the sensitivity to the initial values of the parameters, which can lead to inadequate behavior and limited convergence of the algorithm. The high computational complexity of the algorithm can also create obstacles, especially on large networks or limited resources [9]. In this case, the algorithm can become vulnerable to getting stuck on certain edges or nodes, which leads to suboptimal solutions.

In addition, the dependence on traffic intensity can limit the adaptability of the algorithm, as well as limit its ability to balance pheromone regulation. The algorithm may also not always be able to effectively adapt to sudden dynamic changes in the network or load, which can lead to delays in response. In data-constrained environments, the algorithm may also produce inaccurate solutions because its decisions are based on statistical data.

4.2. The "Enhanced Adaptive Ant Colony" method

The Enhanced Adaptive Ant Colony method is one of the most advanced strategies for load balancing in SDN. In [9], the authors proposed this method as a way to improve the adaptability and efficiency of the Adaptive Ant Colony algorithm.

The main feature of the Enhanced Adaptive Ant Colony is its ability to dynamically adapt to changes in the network and load. To achieve this adaptability, the authors introduce additional parameters that regulate the weights of pheromones and their evaporation rates. These parameters can be automatically adjusted depending on the network status and traffic volume [9]. This allows the algorithm to respond effectively to changes and optimally distribute the load between network nodes.

To achieve this, additional parameters are introduced that regulate the weights of pheromones and their evaporation rate. These parameters are automatically adjusted depending on the network status and traffic volume. The formulas for calculating the updated pheromone weights and evaporation rates are as follows:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (3)$$

$$\tau_{ij}(t) = (1/L_k(t)) * Q, \quad (4)$$

here $L_k(t)$ is the length of the route k at the moment t, and Q is a constant that determines the amount of pheromone released by an ant.

The improvement introduced by Enhanced Adaptive Ant Colony is a more precise adaptive parameter control. This allows the algorithm to adapt to changes more efficiently, providing better load balancing, lower computational costs, and higher reliability than previous methods.

The Enhanced Adaptive Ant Colony method does indeed make improvements to the Adaptive Ant Colony algorithm [10], but it also has its limitations and drawbacks. One of the most important drawbacks is the high computational complexity of the method. This can lead to a significant increase in computational resources and time to execute the algorithm, especially in branched networks with a large number of nodes and various routes.

An additional disadvantage may be an insufficient response to sudden changes in the network. Since the adaptation is based on historical data and statistics, the algorithm may not effectively deal with situations that were not foreseen in the previous data.

In addition, an Enhanced Adaptive Ant Colony may be vulnerable to attacks similar to the Adaptive Ant Colony method [9]. For example, attacks that send artificial traffic can distort statistics and lead to incorrect load distribution.

In general, Enhanced Adaptive Ant Colony is the first step towards improving the adaptability and efficiency of load balancing in software-defined networks, but its limitations and drawbacks should be considered when further developing and improving load balancing algorithms.

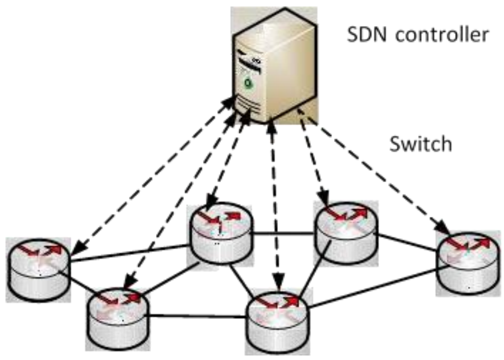
5. Proposed Balancing Method

The concept of SDN represents a new approach to networking, where network management and control are decoupled from the physical infrastructure. This architecture contributes to faster deployment of new services, lower maintenance costs, and improved scalability and flexibility.

A distinctive feature of SDN is that the network is organized and managed at the software level with the help of virtual switches and a central SDN controller [1,2] (Fig.).

There are various solutions for load balancing in SDNs, but some limitations remain relevant, which requires the development of more efficient approaches. Among these solutions, we focus on the Enhanced Adaptive Ant Colony algorithm, which serves as the basis for our proposed improvements.

The Enhanced Adaptive Ant Colony algorithm is based on the concept of Ant Colony Optimization (ACO) [3]. In this method, a colony of virtual ants moves around the network, depositing and emitting pheromones for communication. This guides subsequent ants in choosing routes with a higher concentration of pheromones, which contributes to load balancing to some extent. However, the basic algorithm has its drawbacks, such as static parameter values, limited flexibility in route selection, and sensitivity to changes in the network. The adaption to network conditions aims to improve the accuracy of load balancing, respond effectively to changes in network load, and provide a more flexible solution for modern SDN environments.



Structure of the SDN network

The proposed method for load balancing in software-defined networks (SDNs) is based on the Ant Colon approach and considers the dynamic parameter adjustment. The main principles and stages of the method, as well as the formulas for each stage, are described step by step below.

The anticolonial approach is based on the interaction of agents that mimic the behavior of ants in finding paths to food sources. The key idea is the ability of agents to exchange information about the quality of current paths and use this information to choose the optimal route [5].

To address the limitations of the basic algorithm, we developed optimization steps that allow the algorithm to adapt to changing network conditions. Let's take a closer look at each of these steps, starting with its initialization.

Step 1: Initialization

In this step, we set up the algorithm and prepare it for execution.

1. Let's define N as the number of network nodes.

2. Initialize pheromone levels on possible routes: $\tau_{ij} = \tau_0$, where i and j are network nodes, τ_{ij} is the pheromone level between nodes i and j, τ_0 is the initial pheromone level.

3. Deploy agents on the source nodes of the network where the load is outbound. Each agent identifies its current node and the assigned node.

Step 2: Dynamically adjust the settings

To adapt to changes and achieve the most efficient load balancing, we use dynamic parameter settings.

1. Calculate the current network load.
2. Determine the coefficients for the parameter weights α and β:

$$\alpha = \alpha_{\text{base}} \times \frac{\text{load\_factor}}{\text{max\_load}}, \beta = \beta_{\text{base}} \times \frac{\text{max\_load}}{\text{load\_factor}}$$

where α_{base} and β_{base} are the initial values of the parameters, and max_load is the maximum possible load degree.

Step 3: Enhanced Route Selection

To improve the convergence and accuracy of balancing, we use improved route selection with "elite routes".

1. When the agent chooses a route, we use the probability of route selection:

$$P_{ij} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{_{k}\ \tau_{ik}^{\alpha} \cdot \eta_{ik}^{\beta}}.$$

2. After selecting a route, update the pheromone level on it with an additional increase:

$$\Delta\tau_{ij} = \frac{\text{load\_factor}}{\text{max\_load}} \times \tau_{ij}.$$

Step 4: Adaptive Pheromone Update

Instead of a static evaporation coefficient, we use an adaptive coefficient that depends on the degree of load:

$$\rho = \rho_{\text{base}} + \ \text{load\_factor} - 1\ \times \rho_{\text{base}}.$$

It permits them to respond more effectively to changes in load and maintain the load balance at an optimal level.

Step 5: Dynamically adjust the settings

To ensure adaptation to changes in the network and the most efficient load balancing, we use dynamic adjustment of the settings (Step 2).

1. Calculate the current network load.
2. Determine the coefficients for the parameter weights α and β:

$$\alpha = \alpha_{\text{base}} \times \frac{\text{load\_factor}}{\text{max\_load}},$$

$$\beta = \beta_{\text{base}} \times \frac{\text{max\_load}}{\text{load\_factor}},$$

Step 6: Dynamically adjust the settings

We apply dynamic parameter adjustment to ensure that the algorithm is adaptive to changes in the network.

1. Determine the adaptability coefficient for pheromones:

$$\phi = \frac{\text{load\_factor}}{\text{max\_load}}.$$

2. Adaptively change the evaporation parameter:

$$\rho = \rho_{\text{base}} + \ \phi - 1\ \times \rho_{\text{base}}.$$

3. Change the values of the parameters α and β:

$$\alpha = \alpha_{\text{base}} \times \phi\ ,$$

$$\beta = \beta_{\text{base}} \times\ 1 - \phi\ .$$

Step 7: Finding the best solution

After several iterations of the algorithm, each agent can choose several possible routes. From these, we choose the best one, considering both pheromones and node weights.

1. Determine the score for each agent route:

$$\text{evaluation} = \frac{\text{pheromone\_level} \times \text{node\_weight}}{\text{path\_length}}.$$

2. Choose the route with the highest score as the best one.

The use of dynamic parameter settings permits to adaptation of the algorithm parameters in real-time. Instead of static values, the algorithm parameters now automatically change depending on changes in the network and load. This allows the algorithm to respond effectively to changes, maintaining optimal load distribution even under changing conditions.

The second important improvement concerns the enhancement of the route selection process for virtual ants. Paths now take into account more factors, including the current load of nodes and paths. So the ants choose the best routes based on the current circumstances, ensuring an even distribution of traffic and avoiding congestion.

Increased adaptability is the third key aspect of the optimized algorithm. The introduction of dynamic parameter tuning, combined with an improved route selection process, made the algorithm more adaptive to dynamic changes in the network. It can effectively adapt to load fluctuations and changes in the network structure.

These improvements together contribute to better load balancing in SDN networks. By providing accurate traffic distribution, preventing individual nodes from being overloaded, and ensuring high network performance, the optimized Enhanced Adaptive Ant Colony algorithm becomes a powerful tool for maintaining a stable and efficient network with a programmable structure.

6. Conclusions

Based on the Ant Colony algorithm, the proposed method applies the ideas of adaptability and dynamic parameter tuning to achieve optimal load balancing. It enables the network to adapt to changes in load and ensure even distribution of data among nodes. The adaptability of the algorithm allows you to maintain a stable level of performance even in the face of dynamic load changes.

The load-balancing process involves several steps, such as initialization, dynamic parameter tuning, improved route selection, adaptive pheromone updating, and finding the best solution. Each of these steps performs an important function in ensuring effective load balancing.

The Enhanced Adaptive Ant Colony method realizes the next step towards ensuring the resilience and optimal performance of SDN networks. Considering the dynamics of modern networks and the growing demands on their performance, it provides an opportunity to effectively solve the load problem and ensure reliable and fast data transmission.

References

[1] A. Garcia-Saavedra, P. Serrano, A. Banchs, X. Costa- Perez, "Machine learning for network automation: Over- view, architecture, and applications," IEEE Communica- tions Magazine, vol. 56, no. 3, pp. 11-17, 2018. DOI: 10.1109/MCOM.2018.1700980.

[2] L. Tang, S. Han, H. Zhang, and M. Gerla, "Resilient SDN traffic engineering: A survey," IEEE Communications Surveys & Tutorials, vol. 18, no. 4, pp. 2682-2706, 2016. DOI: 10.1109/COMST.2016.2581599.

[3] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 26, no. 1, pp. 29-41, 1996. DOI: 10.1109/3477.484436.

[4] M.saied, S. Saha, I. Sayem. (2022). A Comparative Study on Load Balancing Techniques in Software Defined Net- works, January 2022, DOI:10.35444/IJANA.2022.13401

[5] Ghorbani, S., Sama, M. R. B., & Abolhasani, M. (2015). An efficient load balancing algorithm for software-defined networking. In Proceedings of the 2015 IEEE/ACM 8th Inter- national Conference on Utility and Cloud Computing

[6] (pp. 80-85). https://www.researchgate.net/publication/ 329144198_ Efficient_load_balancing_algorithm_in_cloud_ computing.

[7] Chen, C. C., & Chen, M. (2015). A weighted round-robin algorithm for software-defined networking. In Proceed- ings of the 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (pp. 138-143). https://www.springerprofessional.de/en/weighted-round- robin-load-balancing-algorithm-for-software- defin/16887408

[8] Bari, M. F., Boutaba, R., Esteves, R., Granville, L. Z., Podlesny, M., Rabbani, G., & Zhang, Q. (2013). Data cen- ter network virtualization: A survey. IEEE Communica- tions Surveys & Tutorials, 15(3), 1614-1634 https://www.academia.edu/26829535/Data_Center_Netwo rk_Virtualization_A_Survey

[9] Y. Tao et al, A Mobile Service Robot Global Path Plan- ning Method Based on Ant Colony Optimization and Fuzzy Control, Appl. Sci. 2021, 11(8), 3605; https://doi.org/10.3390/app11083605

[10] R. Mehmood, F. Ahmed, "Enhanced dynamic ant colony load balancing algorithm for SDN," Journal of Ambient Intelligence and Humanized Computing, vol. 11, pp. 5907-5921, 2020. DOI: 10.1007/s12652-020-01906-4.