CPP-Based Cooperative Defense Against DoS Attacks in Future Non-terrestrial Networks

Zhaori Cong(✉), Zhilong Zhang, and Danpu Liu

Beijing Laboratory of Advanced Information Network,
Beijing Key Laboratory of Network System Architecture and Convergence,
Beijing University of Posts and Telecommunications,
Beijing 100876, People's Republic of China
694660532@qq.com, {zhangzhilong,dpliu}@bupt.edu.cn

Abstract. In future non-terrestrial networks, satellites with sufficient computing resources are expected to serve as base stations or access points, which changes the structure of traditional satellite networks into a more flexible environment. For such satellite nodes, denial of service (DoS) attacks may become a potential secure threat that should be prevented. Existing researches mainly focus on the defense against DoS attacks on ground nodes and have no consideration of the attacks on future satellites. Moreover, the problem of reducing the access delay when a satellite is under DoS attack has not been addressed. In this paper, we study the DoS attack defense strategy in non-terrestrial networks. By adopting client puzzle protocol (CPP) and load balancing, we propose a cooperative defense strategy where multiple auxiliary nodes are used to help the attacked node to process the intensive attack requests. An access delay minimization problem to optimize the selection of auxiliary nodes, puzzle difficulty as well as traffic offload ratios is then formulated based on queuing theory and solved. Simulation results show that the proposed scheme not only improve the network's anti-attack capability, but also achieves desirable performance in average access delay.

Keywords: Client puzzle · DoS Attack · Non-terrestrial networks · Delay

1 Introduction

Non-terrestrial networks (NTN) has been included in 3GPP's 5G standards due to its wide coverage, strong disaster resistance, and low vulnerability to physical attacks on the ground. The perspective of the future 6G network also includes the NTN architecture, where satellites not only work as relay nodes, but also can be base stations and capable of performing simple calculations. However, non-ground nodes are easy to become targets of DoS attacks due to the limited

H. Gao et al. (Eds.): ChinaCom 2020, LNICST 352, pp. 256–267, 2021.
https://doi.org/10.1007/978-3-030-67720-6_18

computing capability. Therefore, the defense against DoS attacks in the weak authentication process[1] is essential to be concerned.

DoS attack refers to attackers' repeated and intensive requests within a short time, which prevent legal users from being normally served. At present, the main form of DoS attacks is resource-consuming attack. By consuming network bandwidth, computing capacity or other useful resources, attackers will make the systems overwhelmed and paralyzed. Traditional methods of defense against DoS attacks include strengthening the stability of the operating system [1], using firewalls to resist [2,3], bandwidth limitation and QoS guarantee [4], as well as traffic load balancing [5]. These methods are effective in preventing DoS attacks with small traffic and simple structure. However, with the increase of network bandwidth and the use of distributed technologies for DoS attacks, it's difficult for the traditional defense methods to meet the needs. Therefore, the client puzzle protocol (CPP) [6,7] has been proposed to address this issue. In this design, a client has to calculate and obtain the answer of a specified puzzle with a certain degree of difficulty before sending out a request, thus the frequency of requests made by attackers is significantly reduced. There have been some existing studies using CPP to resist DoS attacks on different types of system resources. For example, in [8], the authors propose a light encryption mechanism suitable for the IoT environment. In [9], a lightweight cache reliability problem is addressed. To sum up, the recent research on CPP-based defense mainly focus on designing the function form of the puzzle to make the overall access process more secure.

As in future non-ground networks, satellite nodes may assume the function of authentication. Therefore, some new problems need to be considered. On the one hand, the computing capacity of satellite nodes is limited. The ability to withstand attacks is not as good as ground nodes. On the other hand, the propagation delay of satellite network is large. Weak authentication may increase the access delay of normal users and affect the normal communication. Therefore, security is not the only factor to be considered, the optimization of user's access delay also needs to be addressed.

Motivated by above analysis, we design a weak access authentication mechanism for NTN based on CPP and load balancing. When a DoS attack occurs, multiple nodes around the attacked node are selected to act as auxiliary nodes. The attacked node makes puzzle based on the number of requests it receives, and distributes the solution it receives again to the auxiliary nodes. The auxiliary nodes help verify the correctness of the solution. Thus the nodes anti-attack capability is greatly improved. Furthermore, we establish the delay model based on queuing theory. The model is solved by taking the minimum access delay as the optimization goal, and the optimized selection of auxiliary nodes, puzzle difficulty as well as traffic offload ratios are obtained. Finally, simulation results show that our proposed scheme can effectively reduce user's access delay under DoS attack and improve the ability to resist attacks.

[1] Weak authentication is usually performed before strong authentication (i.e., identity authentication).

The rest of this paper is organized as follows. Section 2 describes the proposed CPP-based cooperative weak authentication scheme and formulates an access delay minimization problem. Section 3 provides a solution to this optimization problem. Section 4 simulates the algorithm and verifies the optimization of the algorithm proposed in this paper to the delay. Section 5 summarizes this paper.

2 System Model and Problem Formulation

In future NTN, satellites can serve as base stations or access points. If extra computing resources are deployed on the satellites, they can perform user authorization and other calculations. However, these satellites may be targeted by DoS attackers as well.

As shown in Fig. 1, the NTN includes satellites nodes, ground nodes, normal users and DoS attackers who control a lot of zombie hosts. We assume that each satellite will use CPP against DoS attacks. When the attacked satellite receives a large number of access requests from both attackers and legitimate users, it will collect the resource information of its surrounding satellite nodes. Through these information, the attacked node will determine the difficulty of puzzle and the optimal auxiliary nodes to cooperatively defend the DoS attacks. As a result, when the attacked node receives the puzzle's solutions returned by the attackers and legitimate users, it will distributes these solutions to the auxiliary nodes. After the auxiliary nodes complete the verification of the solutions, the whole weak authentication process ends.

According to above design of cooperative defense among satellite nodes, it is clear that the single satellite nodes ability against DoS attack will be greatly improved with the introduction of multiple auxiliary nodes. However, solving a puzzle will greatly increase the access delay of the legitimate users, especially in the NTN scenario where the propagation delay between the satellite and the ground is very large. In this case, how to improve the legitimate users delay performance through the optimization on the difficulty of the puzzle, as well as the selection of auxiliary nodes and traffic allocation among them becomes an important issue, which will be addressed in the following sections.

2.1 Client Puzzle and Load Balancing Model

Let $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$ denote the set of all nodes that can assist to balance the traffic load, including satellites and ground stations. When the attacked node n_0 detects an abnormal number of access requests, it selects the auxiliary nodes and determines the traffic offloading ratio for each node. Let k_i denote node selection indicator, which is given by

$$k_i = \begin{cases} 0, \text{node } n_i \text{ is not selected as an auxiliary node} \\ 1, \text{node } n_i \text{ is selected as an auxiliary node} \end{cases} \tag{1}$$

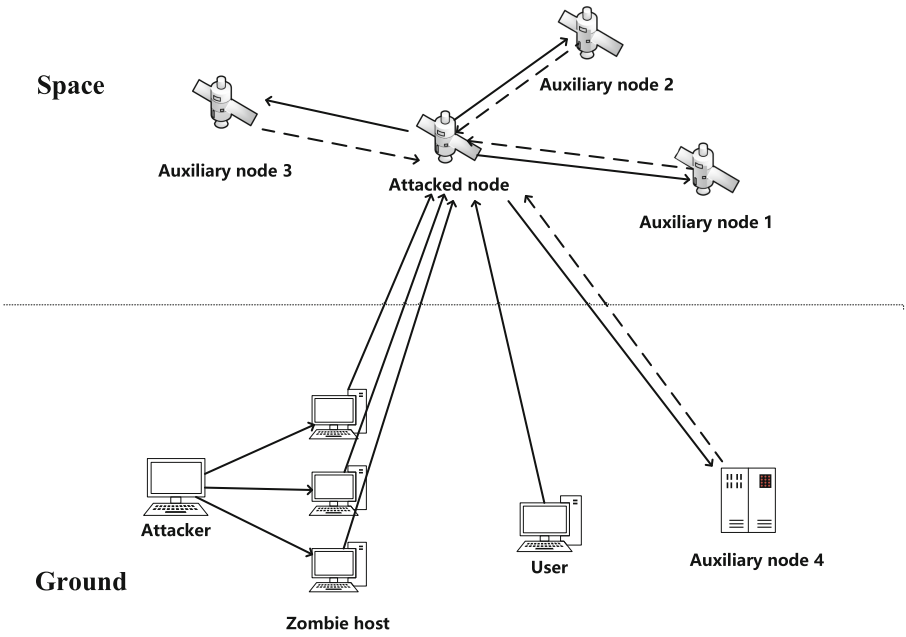

Fig. 1. System model of defense against DoS attack in NTN.

Moreover, let a_i be the offloading ratio of node i and formulate a ratio set $\mathcal{A} = \{a_0, a_1, ..., a_N\}$, which is constrained by

$$\sum_{i=0}^{N} k_i a_i = 1, i \in \{0, 1, 2, ..., N\} \tag{2}$$

Then, node n_0 generates a puzzle with difficulty j and sends it to the users and attackers, where

$$j \in \{0, 1, ..., P\} \tag{3}$$

Both legitimate users and attackers need to consume their own resources to solve the puzzle and send the solutions to node n_0. These solutions are then offloaded to the nodes in the auxiliary network for verification according to the established traffic allocation scheme.

2.2 Delay Model Based on Queuing Theory

Assume that the solutions of users and attackers arrive at node n_0 according to Poisson distribution. Due to the use of CPP for defense, different computing time for solving a puzzle leads to the changing of total average arrival rate, which is determined by difficulty j and denoted as $\lambda_0(j)$. Therefore, the average arrival rate of the traffic offloaded to node i is given by:

$$\lambda_{0i}(j) = a_i \lambda_0(j) \tag{4}$$

Let $f(j)$ denote the computational resource consumed by the puzzle with difficulty j. The resource consumption has an exponential relationship with difficulty, which is given by $f(j) = K_1 2^{K_2 j}$ [10]. Assume that the time for attackers to process the puzzle can be expressed as:

$$t_{pa}(j) = \frac{f(j)}{C_a} \tag{5}$$

where C_a (cycles per second) is the computing capability of attackers.

Similarly, the time for users to process the puzzle can be expressed as:

$$t_{pu}(j) = \frac{f(j)}{C_u} \tag{6}$$

where C_u is the computing capability of a typical user.

For both legitimate users and attackers, calculating the puzzle will result in a lower request arrival rate. But compared with the users' relatively long request interval, the time to calculate the puzzle can be ignored. Therefore, the arrival rate of users' solutions is set to a constant λ_u. On the other hand, the attackers' initial frequency of sending requests is quiet high, so computing puzzle can effectively delay making solutions and reduce the arrival rate of solutions. The larger the difficulty of puzzle it is, the more time attackers takes to compute, and the more the attack frequency decreases. However, the access delay for the legitimate user will also be larger with the increase of puzzle difficulty.

Given that the average time between two requests from the same attacker is approximately the time to process a puzzle $t_{pa}(j)$, the total average arrival rate at a given puzzle difficulty j is:

$$\lambda_0(j) = \lambda_u + \frac{1}{t_{pa}(j)} \tag{7}$$

Assume that the maximal number of solutions verified by satellite node n_i per unit time is μ_i, which is determined by the computing capability of the auxiliary nodes. Further assume that the maximal number of solutions transmitted from n_0 to n_i per unit time is μ_i', which is determined by the bandwidth resources.

The delay for solution verification on the attacked node is different from that on the auxiliary nodes. When a solution is verified on the attacked node, only the processing delay needs to be considered. When a solution is verified on an auxiliary node, the total delay is composed by three parts: the transmission delay between the attacked node and the auxiliary node, the processing delay, and the propagation delay in space transmission. In this paper, we use M/M/1 to model transmission delay and processing delay. Because the above two delays are independent of each other, we use two M/M/1 in series for modeling [11], which is shown in Fig. 2.

Therefore, the average authentication delay of each request is given by:

$$T_i(j) = \begin{cases} \frac{1}{\mu_i - \lambda_{0i}(j)} + \frac{1}{\mu_i' - \lambda_{0i}(j)} + t_{pi}, & \text{on auxiliary nodes} \\ \frac{1}{\mu_i - \lambda_{0i}(j)}, & \text{on attacked node} \end{cases} \tag{8}$$

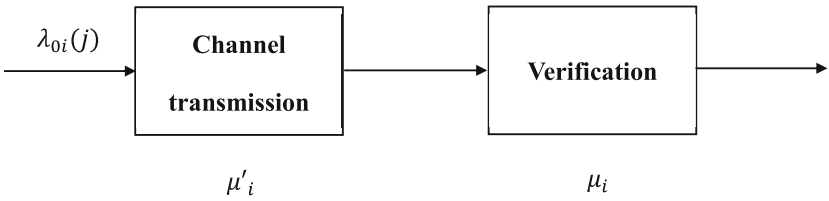

Fig. 2. A queuing model of attacked nodes to auxiliary nodes.

where the first line represents the average authentication delay of requests which are processed on auxiliary nodes. t_{pi} means propagation delay which depends on the distance between node 0 and node i. The second line represents the authentication delay of requests which are processed on the attacked node.

The probability that a typical user's access request is processed at n_i is a_i, so the average access delay of the users in the system can be expressed by:

$$T = t_{pu}(j) + \sum_{i=0}^{N} k_i a_i T_i(j) \tag{9}$$

To minimize the above delay, an optimization problem is formulated according to the Eq. (3) to (9):

$$\begin{aligned} \min_{j,k_i,a_i} \quad & t_{pu}(j) + \sum_{i=0}^{N} k_i a_i T_i(j) \\ \text{s.t.} \quad & \lambda_{0i}(j) < \mu_i \\ & \lambda_{0i}(j) < \mu'_i \\ & j \in \{0, 1, ..., P\} \\ & \sum_{i=0}^{N} k_i a_i = 1 \end{aligned} \tag{10}$$

3 Problem Solving

Since problem (10) has 3 variables with interaction effects, it is difficult to obtain the optimal solution directly. To simplify the problem, we first traverse the puzzle difficulty j. For a given difficulty j, the auxiliary node selection scheme and the corresponding traffic allocation scheme can be obtained relatively easily. Therefore, two algorithms are designed for selecting auxiliary node and traffic allocation, respectively. Finally, compare the users' access delays of all difficulties to get the best solution.

3.1 Auxiliary Node Selection

For the construction of the auxiliary node network, it is obvious that the larger the number of auxiliary nodes included, the stronger the network's ability to

Algorithm 1. Auxiliary node selection

Input: n_i, μ_i, μ_i', j, $\lambda_0(j)$, t_{pi}
1: $T_i = \frac{1}{\mu_i} + \frac{1}{\mu_i'} + t_{pi}$
2: Sort all nodes in an increasing order according to T_i, and obtain an updated set $\{n_1', n_2', ..., n_N'\}$.
3: **for** $M = 1 : N$ **do**
4: **if** $\mu_0 + \sum_{m=1}^{M} \min\{\mu_m, \mu_m'\} < \lambda_0(j)$ **then**
5: Add n_M' to $\mathcal{N}_j'$.
6: **end if**
7: **end for**
Output: $\mathcal{N}_j'$

defend against attacks. Therefore, it is desirable to select as many auxiliary nodes as possible. However, the complexity of the following optimization algorithm increases exponentially with the number of auxiliary nodes. Since the computing resources of the satellite nodes are relatively scarce, it is difficult to support the implementation of algorithms with excessive complexity. In addition, for the actual situation, there is generally a difference in transmission cost between the attacked node and different satellite nodes. Different auxiliary nodes have different processing capabilities. Choosing different auxiliary nodes will also affect the access delay of normal users. Therefore, we have to determine whether a satellite is suitable to serve as an auxiliary node according to the resource information of neighbor satellite nodes and ground nodes.

When the difficulty j is fixed, the arrival rate of access requests $\lambda_0(j)$ is also a fixed value. Through Eq. (8), by setting $\lambda_{0i}(j)$ to 0, we can obtain a lower bound of an auxiliary node's authentication delay, which is given by

$$T_i = \frac{1}{\mu_i} + \frac{1}{\mu_i'} + t_{pi} \tag{11}$$

We sort all satellite nodes in an increasing order according to the above boundary value, and add satellite nodes successively to set $\mathcal{N}_j'$ until Eq. (12) are satisfied.

$$\mu_0 + \sum_{m=1}^{M} \min\{\mu_m, \mu_m'\} > \lambda_0(j) \tag{12}$$

where μ_0 and μ_m represent the average number of requests verified by n_0 and node in $\mathcal{N}_j'$ per unit time. μ_m' represents the average number of requests transmitted between n_0 to n_m' per unit time. M is the number of the finally selected nodes. The auxiliary node selection scheme is summarized in Algorithm 1.

Algorithm 2. Defense algorithm based on CPP and load balancing

Input: n_i, μ_i, μ_i', $\lambda_0(j)$, t_{pi}, $f(j)$, C_u

1: **for** $j = 0 : P$ **do**
2: Use algorithm 1 to select the auxiliary nodes $\mathcal{N}_j' = \{n_1', n_2', ..., n_M'\}$.
3: Calculate the minimum of $T_j = t_{pu}(j) + \sum_{m=0}^{M} a_{mj}' T_i(j)$.
4: Get the traffic allocation scheme $\mathcal{A}_j' = \{a_{1j}', a_{2j}', ..., a_{Mj}'\}$.
5: **end for**
6: $T = \min\{T_j\}$
7: Find the corresponding auxiliary nodes $\mathcal{N}'$, traffic allocation schemes $\mathcal{A}'$ and difficult of puzzle j'.

Output: T, $\mathcal{A}'$, j', $\mathcal{N}'$

3.2 Traffic Allocation

After we have determined auxiliary nodes $\mathcal{N}_j' = \{n_1', n_2', ..., n_M'\}$, the optimization problem becomes:

$$\begin{aligned} \min_{j, a_m'} \quad & t_{pu}(j) + \sum_{m=0}^{M} a_m' T_i(j) \\ \text{s.t.} \quad & \lambda_{0m}(j) < \mu_m \\ & \lambda_{0m}(j) < \mu_m' \\ & j \in \{0, 1, ..., P\} \\ & \sum_{m=0}^{M} a_m' = 1 \end{aligned} \tag{13}$$

where a_m' represents the traffic allocation scheme for both auxiliary nodes and attacked node, $m \in \{0, 1, 2, ..., M\}$.

With regard to the problem (13), when j is determined, the optimization problem is simplified to a single variable problem. The complexity of the optimization solution is greatly reduced.

The expansion form of the objective function T (taking the solution verified on the auxiliary node as an example) is:

$$T = t_{pu}(j) + \sum_{m=0}^{M} a_m' \left[\frac{1}{\mu_m - \lambda_{om}(j)} + \frac{1}{\mu_m' - \lambda_{om}(j)} + t_{pi}(j) \right] \tag{14}$$

When j is fixed, the variables $t_{pu}(j)$ and $\lambda_{0m}(j)$ are also constants. T is a function about the multivariate variable a_m', and the constraints of the independent variable a_m' are:

$$\lambda_{0m}(j) < \mu_i \tag{15}$$

which can be transformed into a linear constraint.

$$a_m' < \frac{\mu_m}{\lambda_0(j)} \tag{16}$$

Table 1. Parameter setting

Parameters	Values
μ_m, $m = 0, 1, ..., 6$	70, 75, 200, 82, 66, 65, 10
μ'_m, $m = 0, 1, ..., 6$	null, 120, 200, 150, 150, 60, 50
User's access request arrival rate λ_u	20

The entire optimization problem (13) is simplified to a linear programming problem. It is relatively easy to find the minimum value of the objective function. By comparing the minimum of each difficulty j, we can get the parameters including the difficulty of the puzzle j, auxiliary node $\mathcal{N}'$ and traffic allocation scheme $\mathcal{A}'$.

The whole solving process is summarized in Algorithm 2. First we traversal all the difficulties. For a given difficulty, we can get the node selection scheme through Algorithm 1. Then the whole problem is simplified and the minimum value of the objective function can be found relatively easy. Finally, by comparing the minimum of each difficulty j, we can get the parameters we need.

For NTN, some nodes such as low-orbit satellite nodes are mobile. The period of parameters change caused by movement is much longer than the period of access authentication. Therefore, when the network parameters change, the algorithm must be performed once again.

4 Simulation Results and Analysis

In this section, we verify the effectiveness of the scheme proposed in this paper with two baselines including non-cooperative CPP-based defense and cooperative CPP-based defense with fixed puzzle difficulty.

We use matlab to perform performance comparative simulation of user access delays under different conditions. A total of 66 satellites of the Iridium system and ground nodes are selected. The parameters and motions of the satellites are imported through STK. According to the obtained trajectory data, satellites capable of communicating within a certain period of time are selected to construct a satellite network. Some important parameters are listed in Table 1. The first value of μ'_m is set to null, because it represents the transmission rate of n_0 to n_0. According to the paper [12], the average propagation delay of the satellite node to the ground node of the Iridium system is 15 ms, and the propagation delay between satellites is about 3 ms to 8 ms.

Figure 3 is the comparison result of the cooperative CPP-based defense and the non-cooperative CPP-based defense. It can be seen that under the same attack intensity, the minimum users' access delay of the cooperative CPP-based defense is about 48 ms. However, the non-cooperative CPP-based defense is about 100 ms. In addition, the difficulty corresponding to the minimum delay of the cooperative CPP-based defense is 6, and the non-cooperative CPP-based

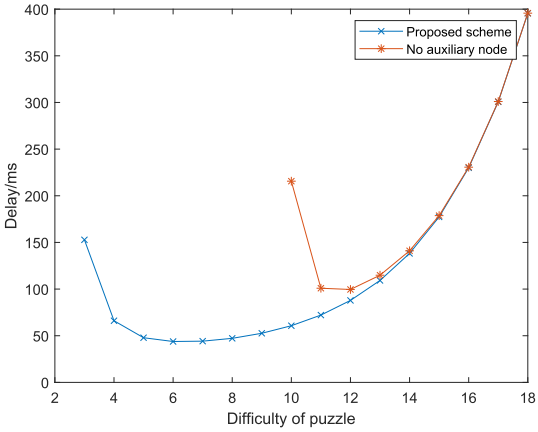

Fig. 3. Comparison of the access delay between the proposed scheme and single node network under the same attack intensity (number of nodes is 6 and number of attacks per second is 10000).

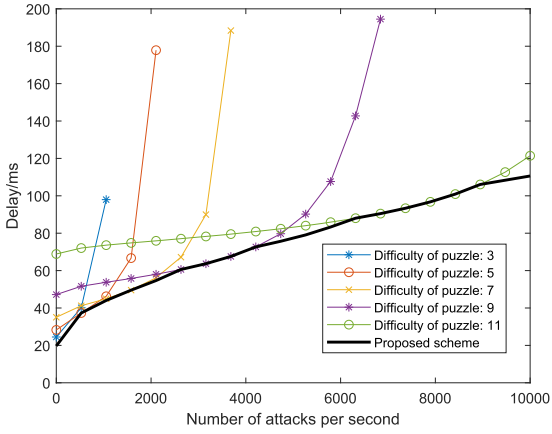

Fig. 4. Comparison of the access delay between the proposed scheme and the fixed puzzle difficulty method under different attack intensities (number of nodes is 6).

defense is 12. Lower difficulty can save the satellite nodes' available computing resources on the basis of delay optimization.

Figure 4 shows the comparison of the access delay between the proposed scheme and fixed puzzle difficulty scheme under the same number of auxiliary nodes. It can be seen from Fig. 4, the users' access delay of the proposed scheme has always been the lowest. In the case of small preset difficulty, the users' access delay will rapidly go up with the increase of attack intensity. Small difficulty may even cause a crash due to insufficient computing resources of the satellites. If the preset difficulty is too large, the satellite nodes' available computing resources are

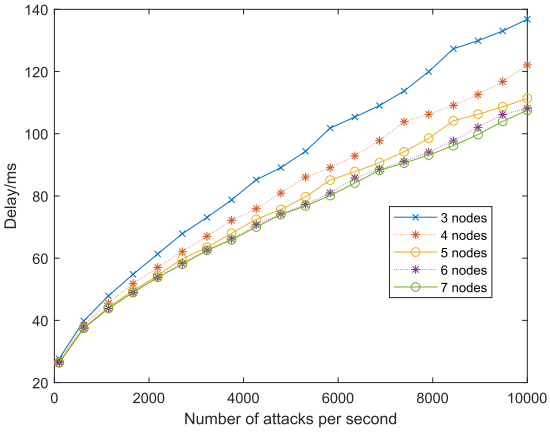

Fig. 5. Comparison of access delays of different numbers of auxiliary nodes using the proposed scheme under different attack intensities.

wasted when there is no attack or the attack intensity is small. Meanwhile, large difficulty will increase the users' access delay. Therefore, it can be judged that dynamically selecting the difficulty of the puzzle can better reduce the access delay and allocate the available computing resources more reasonably.

Figure 5 is a comparison of the users' access delays versus the numbers of auxiliary node. From the simulation result, it is clear that under the same condition, the larger the number of auxiliary nodes, the smaller the users' access delay. This shows that for the auxiliary network, the more nodes, the stronger the attack resistance. However, when the number of auxiliary nodes increases to a certain extent, the reduction in users' access delay becomes small. The reason is that at this time, the available computing resources of the entire network have far exceeded the resource consumption for processing requests. In this case, the main factor affecting the access delay of users is not available computing resources any more, but transmission delay and other factors. In addition, if the nodes with less available resources are used as auxiliary nodes, the access delay of the legitimate user may be increased. Therefore, using the algorithm to select suitable auxiliary nodes can obtain the optimal user's access delay.

5 Conclusion

In this paper, we focus on the potential DoS attacks in future non-terrestrial networks where satellites can perform access authentications. First, the types of DoS attacks and related defense measures is introduced. Then, we propose a model for resisting DoS attacks through the CPP and load balancing technology. After that an algorithm is designed to defend against DoS attacks, including auxiliary node selection algorithm and traffic allocation algorithm to minimize the access delay of legitimate users. Finally, the simulation results verify the effectiveness of

our proposed scheme for reducing user's access delay and computation pressure of satellite nodes.

Acknowledgment. This work is supported by the Open Project of A Laboratory under Grant No. 2017XXAQ08 and 2016XXAQ09, the National Natural Science Foundation of China under Grant No. 61971069 and 61801051.

References

1. Chowdhury, N.R., Negi, N., Chakrabortty, A.: A new cyber-secure countermeasure for LTI systems under DoS attacks. In: 2019 27th Mediterranean Conference on Control and Automation (MED), pp. 304–309 (2019)
2. Afianti, F., Wirawan, Suryani, T.: Lightweight and DoS resistant multiuser authentication in wireless sensor networks for smart grid environments. IEEE Access **7**, 67107–67122 (2019)
3. Echevarria, J.J., Garaizar, P., Legarda, J.: An experimental study on the applicability of SYN cookies to networked constrained devices. Softw. Pract. Exper. **48**(3), 740–749 (2018)
4. Abirami, K., Harini, N., Vaidhyesh, P.S., Kumar, P.: Analysis of web workload on QoS to assist capacity. In: Pandian, D., Fernando, X., Baig, Z., Shi, F. (eds.) ISMAC 2018. LNCVB, vol. 30, pp. 573–582. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-00665-5_57
5. Tan, X., Li, H., Wang, L., Xu, Z.: Global orchestration of cooperative defense against DDoS attacks for MEC. In: IEEE Wireless Communications and Networking Conference (WCNC) 2019, pp. 1–6 (2019)
6. Michalas, A., Komninos, N., Prasad, N.R., Oleshchuk, V.A.: New client puzzle approach for DoS resistance in ad hoc networks. In: IEEE International Conference on Information Theory and Information Security 2010, pp. 568–573 (2010)
7. Gupta, D., Saia, J., Young, M.: Proof of work without all the work. In: Proceedings of the 19th International Conference on Distributed Computing and Networking, pp. 1–10 (2018)
8. De Almeida, M.P., de Sousa Júnior, R.T., García Villalba, L.J., Kim, T.-H.: New DoS defense method based on strong designated verifier signatures. Sensors **18**(9), 2813 (2018)
9. Almashaqbeh, G., Kelley, K., Bishop, A., Cappos, J.: CAPnet: a defense against cache accounting attacks on content distribution networks. In: IEEE Conference on Communications and Network Security (CNS) 2019, pp. 250–258 (2019)
10. Aura, T., Nikander, P., Leiwo, J.: DOS-resistant authentication with client puzzles. In: Christianson, B., Malcolm, J.A., Crispo, B., Roe, M. (eds.) Security Protocols 2000. LNCS, vol. 2133, pp. 170–177. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44810-1_22
11. Thomopoulos, N.T.: Fundamentals of Queuing Systems: Statistical Methods for Analyzing Queuing Models. Springer, Boston (2012). https://doi.org/10.1007/978-1-4614-3713-0
12. 3GPP: Study on New Radio (NR) to support non-terrestrial networks (Release 15). 3rd Generation Partnership Project (3GPP), Technical Specification Group Radio Access Network (TR) 38.811, version 15.2.0 (2019)