Topology Self-optimization for Anti-tracking Network via Nodes Distributed Computing

Changbo Tian[1,2], Yongzheng Zhang[1,2], and Tao Yin[1,2](✉)

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
{tianchangbo,zhangyongzheng,yintao}@iie.ac.cn

[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract. Anti-tracking network aims to protect the privacy of network users' identities and communication relationship. The research of P2P-based anti-tracking network has attracted more and more attentions because of its decentralization, scalability, and widespread distribution. But, P2P-based anti-tracking network still faces the attacks on network structure which can destroy the usability of anti-tracking network effectively. So, a secure and resilient network structure is an important prerequisite to maintain the stability and security of anti-tracking network. In this paper, we propose a topology self-optimization method for anti-tracking network via nodes distributed computing. Based on convex-polytope topology (CPT), our proposal achieves topology self-optimization by each node optimizing its local topology in optimum structure. Through the collaboration of all nodes in network, the whole network topology will evolve into the optimum structure. Our experimental results show that the topology self-optimization method improves the network robustness and resilience of anti-tracking network when confronting to the dynamic network environment.

Keywords: Topology self-optimization · Distributed computing · Node collaboration · Network optimization · Anti-tracking network

1 Introduction

Anti-tracking network [1–4] provides secure, anonymous communication for network users to protect the privacy of their network identities and communication relationships. As a large, scalable and stable network system, the design and implementation of anti-tracking network has faced huge challenges. On the one

Supported by the National Key Research and Development Program of China under Grant No. 2019YFB1005203, the National Natural Science Foundation of China under Grant No. U1736218.

H. Gao and X. Wang (Eds.): CollaborateCom 2021, LNICST 406, pp. 405–419, 2021.
https://doi.org/10.1007/978-3-030-92635-9_24

hand, anti-tracking network is an open network and allows each node join or exit from network freely. Then, malicious nodes can infiltrate in the network and measure network topology and network scale [5–8]. On the other hand, the management and optimization of network topology is essential for the dynamic changed topology. In general, the network optimization is implemented by the network controller which has the global view of the network. But the frequent communication between the controller and network can be easily monitored and traced by the adversary [9–12]. So, a stable, resilient and self-optimizing network structure is the foundation for anti-tracking network to provide secure and reliable communication [13].

Recent researchers [14–17] have developed many approaches to improve the robustness and security of anti-tracking network effectively. From the present researches [18–20] it seems, the topology optimization methods have attracted more and more attentions because an optimum network structure can bring about a vast improvement in the performance of network communication and anti-destroy ability. However, anti-tracking network is an P2P-based network which allows each node joins or leaves the network freely. In general, network optimization is achieved by network controller which is vulnerable to network tracing and monitoring. Some approaches [21–25] have been proposed to achieve network self-optimization, but most of them is focused on the resource allocation, path selection optimization and so on. In the dynamic network environment, network structure oriented self-optimization method is important to improve the security and resilience of anti-tracking network.

To address this problem, we propose a topology self-optimization method for anti-tracking network. Our proposal is based on convex-polytope topology (CPT) [26] and network self-optimization algorithm to improve the security and resilience of anti-tracking network. We make three key contributions in this paper as follows:

- We apply convex-polytope topology in the construction of anti-tracking network topology. Anti-tracking network based on CPT has better robustness.
- We define an optimal topology model based on CPT to achieve the topology optimization of anti-trakcing network.
- We propose a topology self-optimization method based on CPT, which improves the security and resilience of anti-tracking network.

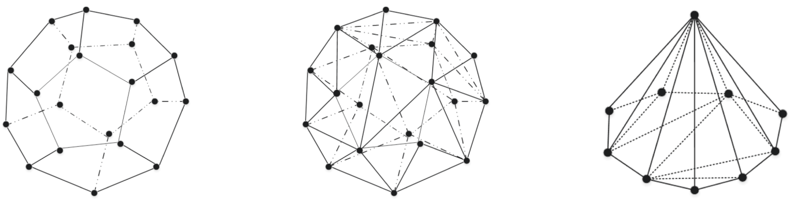
2 Related Works

Network optimization is to improve the performance of anti-tracking network, and the stable, reliable network structure is the prerequisite of efficient communication. Aimed to self-optimization of network topology, Auvinen [27] proposes a topology management algorithm based on neural network which does not predetermine favorable values of the characteristics of the peers. The decision whether to connect to a certain peer is done by a neural network, which is trained with an evolutionary algorithm. Tian [22] proposes smart topology construction method (STon) to provide the self-management and self-optimization of topology for anti-tracking network. By deploying the neural network on each node of the anti-tracking network, each node can collect its local network state and calculate the network state parameters by the neural network to decide the link state with other nodes. With the collaboration of all nodes in the network, the network can achieve the self-management and self-optimization of its own topology. Liu [21] proposes an adaptive overlay topology optimization (AOTO) technique. AOTO is scalable and completely distributed in the sense that it does not require global knowledge of the whole overlay network when each node is optimizing the organization of its logical neighbors. Sun [23] presents THash, a simple scheme that implements a distributed and effective network optimization for DHT systems. THash uses standard DHT put/get semantics and utilizes a triple hash method to guide the DHT clients to choose their sharing peers in proper domains. Liang [24] presents the optimization formulations, and proposes a set of heuristic algorithms for the construction and dynamic management of the multiple sub-stream trees for practical implementation which can significantly improve the delay performance of existing P2P streaming systems. Jelasity [25] proposes a generic protocol for constructing and maintaining a large class of topologies. In the proposed framework, a topology is defined with the help of a ranking function. The nodes participating in the protocol can use this ranking function to order any set of other nodes according to preference for choosing them as a neighboring node. Liao [28] presents a trust-based topology management protocol, which aims to promote the fairness and service quality of P2P system by integrating a trust model into its topology management.

3 Introduction to Convex-Polytope Topology

3.1 Basic Properties

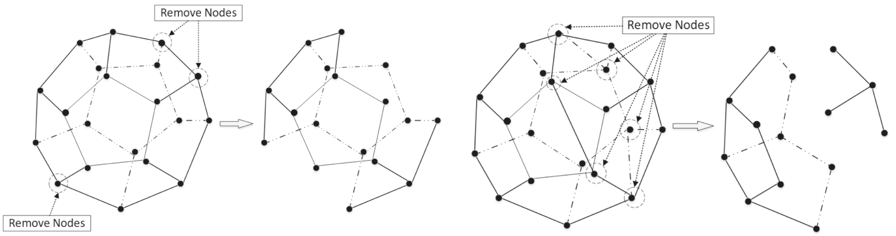
Convex-polytope Topology (CPT) [26] is a structured topology in which all nodes are constructed into a logical structure of convex-polytope as illustrated in Fig. 1. One of the advantages of CPT is the elimination of cutvertex because any two nodes in CPT have at least two non-overlapping paths. In this case, some nodes are removed from CPT, CPT can still keep the convex-polytope structure except the nodes in ring connection.



(a) CPT in sparse structure. (b) CPT in dense structure. (c) CPT with central point.

Fig. 1. CPT with different structures.

As illustrated in Fig. 2(a), some nodes are removed from CPT, CPT still keep the convex-polytope structure. But in Fig. 2(b), the nodes in ring connection are removed, CPT is split into two parts. But in the practical application, it is a small probability event that the removed nodes happened to be in the ring connection. So, when some nodes are removed, the timely recovery of CPT will keep the robustness and invulnerability of network.



(a) CPT keeps the convex-polytope structure when some nodes are removed.

(b) CPT breaks into two parts when the nodes in ring connection are removed.

Fig. 2. The influence of node removal on the structure of CPT.

However, the connection structure and connection density have a big influence on the performance of CPT. As illustrated in Fig. 1, CPT in sparse topology is susceptible to the network churn, and the network structure is more vulnerable to the disconnection of nodes. CPT in dense topology has better robustness, but key nodes with high degree may appear to become the potential threats. As illustrated in Fig. 1(c), CPT has a central node which has connections with all other nodes. CPT with central node is unstable because the disconnection of central node would cause big damage in the structure of CPT.

3.2 The Optimum Structure of CPT

We consider the optimum structure of CPT based on which network has better robustness and invulnerability. Obviously, network density has a big influence on network connectivity and communication efficiency. The dense topology performs better than the sparse topology in network robustness. But in the extreme case illustrated in Fig. 1(c), the uneven distribution of node degree results in the unstable network structure in which some nodes have very high degree, but other nodes have very low degree. Then, the nodes with high degree play the very important roles in the network and the attack to such key nodes would severely disrupt the network structure, even partition the network.

On the basis of above considerations, we define the optimum structure of CPT as that CPT has the maximum network connectivity while conforming to the convex-polytope structure, and the degree of each node is close to the average of all nodes' degrees. CPT with maximum connectivity(CPT_M) has a special property that each surface of CPT is triangle. If CPT_M has n nodes, the number of edges is $l = 3 \times (n-2)$. So, for CPT_M with n nodes, the number of edges is fixed. Then, we can calculate the average degree $\overline{d}$ of CPT_M as shown in Eq. 1. When N approaches infinity, $\overline{d}$ approximately equals to 6. So, the average degree $\overline{d}$ can be set as the baseline for each node to measure and adjust its local topology.

$$\overline{d} = \lim_{N \to \infty} \frac{2 \times L}{N} = 6 - \lim_{N \to \infty} \frac{12}{N} = 6 \tag{1}$$

Formally, for CPT with n nodes, its optimum structure can be defined as shown in Eq. 2, in which N_v denotes the number of nodes, N_e denotes the number of edges, $Degree(v_i)$ denotes the degree of node v_i.

$$CPT_{optimum} = \{N_v = n, N_e = 3 \times (n-2), Degree(v_i) \to \overline{d}\}(1 \leq i \leq n) \tag{2}$$

4 Topology Self-optimization

The goal of topology self-optimization is to maintain the network topology in the optimum structure of CPT. Topology self-optimization is achieved by nodes' distributed computing. At first, each node calculates the optimum local topology ($T_{optimum}$) according to the situation of its current local topology and $T_{optimum}$ is an optimization objective for each node to adjust its local topology. But the final optimization plan is decided by the collaboration between the current node and its neighboring nodes and make sure the optimized local topology is beneficial to both sides.

Before that, we put some notations used in the following discussion in Table 1 to help readers refer to them conveniently.

Table 1. Notations

Notation	Description
CPT	Convex-polytope topology
$CPT_{optimum}$	The optimum structure of CPT
$T_{optimum}$	The optimum local topology of each node
$T_{original}$	The original local topology of each node
O	The criterion of local topology
$\overline{d}$	The average node degree of $CPT_{optimum}$
d_i	The node degree of node v_i
N_l	The node number of the local topology
r_d	The disconnection request between two nodes
r_c	The connection request between two nodes

4.1 Calculation of Optimum Local Topology

As we have mentioned above, $T_{optimum}$ is the optimization objective for each node to optimize local topology, and $T_{optimum}$ also needs to conform to the property of $CPT_{optimum}$. So, the evaluation standards of $T_{optimum}$ can be concluded as: (1) the distribution of nodes' degree, and (2) the difference between the degree of each node with the average degree $\overline{d}$ in $CPT_{optimum}$.

Here, we define the local topology of node v_i as the topology constructed by node v_i and its neighboring nodes. Then, the criterion of $T_{optimum}$ can be calculated as shown in Eq. 3, in which N_l denotes the node number of the local topology, d_i denotes the node degree of node v_i in the local topology and $\overline{d}$ denotes the degree baseline which has been discussed in Sect. 3.2.

$$O = \frac{\sum_{i=1}^{n}(d_i - \bar{d}))^2}{N_l} \tag{3}$$

Equation 3 computes the variance of all nodes' degree in local topology with the average node degree $\overline{d}$. The deviation of nodes' degree from $\overline{d}$ is lower, the node degree is more close to the average node degree of network. So, $T_{optimum}$ has the minimum value of O. Each node changes its local connection status and calculates O to assess the changed local topology until find the $T_{optimum}$.

In order to keep the network connectivity of $CPT_{optimum}$unchanged, if one node breaks the link with its neighboring node, it has to instruct the two relevant neighboring nodes to build new connection. As illustrated in Fig. 3, for example, node v_0 breaks the link with node v_4, because two surfaces share the same link (v_0, v_4), the two neighboring nodes v_1 and v_6 need to build new connection with each other. Likewise, node v_0 breaks the link with v_3, then its two relevant neighboring nodes v_2 and v_5 build new connection with each other. In this way, some nodes reduce their degree and the others increase their degree to achieve the equilibrium of network connectivity.

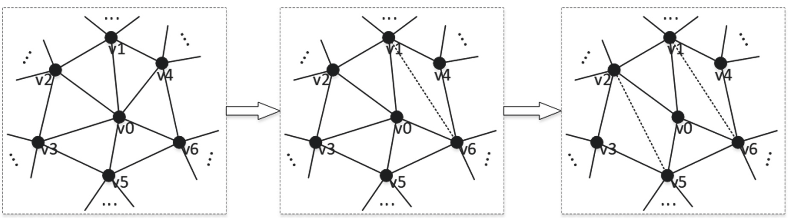


Fig. 3. The adjustment of local topology of node v_0.

The calculation of $T_{optimum}$ is an optimum result search algorithm from all the possibilities that each node adjusts its local topology. In the adjustment of local topology, the parameters in Eq. 3, such as $d_i(1 \le i \le m)$ and N_l, also need to be adjusted according to the changed local topology. Algorithm 1 shows the pseudocode of the calculation of $T_{optimum}$ which is implemented by recursive algorithm, the detailed workflow of Algorithm 1 is concluded as follows:

(1) Node v_0 first calculates the criterion O_0 of its original local topology $T_{original}$, and stores it in an array L.
(2) Node v_0 successively disconnects with one of its neighboring nodes v_i to generate a new local topology T_i. Node v_0 calculates the criterion O_i of each new local topology T_i and stores them in the array L.
(3) Based on each T_i, node v_0 recursively executes the step (2) to calculate the criteria of all the new changed topology until node v_0 has no neighboring nodes. All the calculated criteria are stored in the array L.
(4) Find the minimum value O_{min} of criterion in array L, and the local topology related with O_{min} is the optimum local topology for node v_0.

Algorithm 1. Calculation Algorithm of Optimum Local Topology

Input: M: the original connection matrix, C_n: the neighboring nodes set, v_0: current node
Output: T: the connection matrix of $T_{optimum}$

function $Calculation(M, C_n, L)$ ▷ Recursive algorithm to search all the possibilities that node v_0 changes its local topology
    if $C_n.size() > 0$ **then**
        for v in C_n **do**
            v_0 disconnects with v
            $v_i, v_j \leftarrow Surface(v_0, v)$ ▷ Get the two neighboring nodes in the same surface of node v_0 and v
            v_i connects with v_j
            $M' \leftarrow Update(M)$ ▷ Get the changed topology
            ${o_t}' = O(M')$ ▷ Calculate O of the changed topology
            $L.append({o_t}', M')$
            ${C_n}' = C_n.remove(v)$
            $Calculation(M', {C_n}', L)$
        end for
    else
        return
    end if
end function

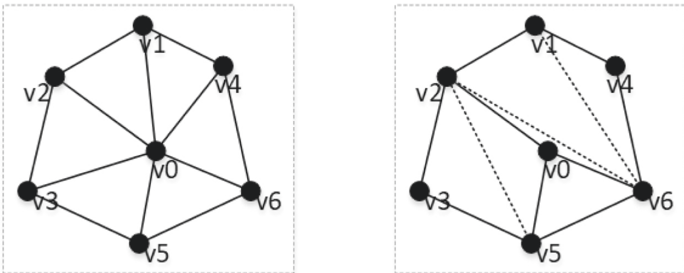
function $Main(M, C_n, L)$
    $o = O(M)$ ▷ Calculate O of the original topology of node v_0
    $L.append(o, M)$
    $Calculation(M, C_n, L)$
    $(o_t, M_t) = L.min()$ ▷ Get the min O and its related local topology
    return M_t
end function

According to Algorithm 1, each node searches its $T_{optimum}$ through breaking links. So, the degree of each node in $T_{optimum}$ will not exceed its degree in $T_{original}$. $T_{optimum}$ is just an optimization suggestion for each node to optimize its local topology. The final adjustment of local topology may not completely conform to the structure of $T_{optimum}$ because any adjustment of topology should be confirmed by the relevant nodes.

4.2 Topology Self-optimization via Nodes' Collaboration

The optimum local topology provides the useful information for each node to optimize its local topology. But the optimization of each node's local topology can not totally depend on the optimum local topology. The effect of topology adjustment on other nodes also need to be taken into consideration. So, the self-optimization of topology is achieved by nodes' collaboration.

Each adjustment of topology involves four nodes, two nodes executing disconnection operation and the other two nodes executing connection operation. So, each adjustment of topology should be confirmed by the relevant four nodes.



(a) The original topology of node v_0.

(b) The optimum topology of node v_0.

Fig. 4. The original topology of node v_0 and its calculated optimum topology.

Assume the original topology of node v_0 is as shown in Fig. 4(a), and it calculates the optimum topology as shown in Fig. 4(b). To adjust its local topology from $T_{original}$ to $T_{optimum}$, node v_0 needs the following operations: (1) the disconnection of v_0 and v_4, the connection of v_1 and v_6; (2) the disconnection of v_0 and v_1, the connection of v_2 and v_6; (3) the disconnection of v_0 and v_3, the connection of v_2 and v_5. For convenience of discussion, we use r_d and r_c to denote the disconnection request and connection request respectively.

Take the example of the collaboration of node v_0, v_3, v_2 and v_5, the detailed process of these nodes' collaboration in the adjustment of local toppology can be described as follows:

(1) Node v_0 sends r_d to v_3 for disconnection, and sends r_c to v_2 and v_5 to instruct them to build new connection. The criteria of $T_{original}$ and $T_{optimum}$ of node v_0 are denoted by O_0 and O'_0 respectively.

(2) After node v_3, v_2, v_5 receive r_c or r_d, they respectively calculate their criteria O'_3, O'_2 , O'_5 of their local topologies changed according to the request. The criteria of the original topology of node v_3, v_2, v_5 are denoted by O_3, O_2, O_5 respectively. If $O'_3 \leq O_3$, $O'_2 \leq O_2$, $O'_5 \leq O_5$, node v_3, v_2, v_5 agree with the topology adjustment requested by node v_0. Otherwise, go to Step (3).

(3) For node $v_k(k \in \{2, 3, 5\})$ and $O'_k > O_k$, v_k calculates the difference D_k between O_k and O'_k. Node v_0 calculates the difference D_0 between O_0 and O'_0. If $D_k < D_0$, node v_k agrees with the topology adjustment. Otherwise, go to step (4).

(4) If at least three nodes satisfy the rules shown in step (2) and (3), the topology adjustment has to be implemented. Otherwise, the local topology involes these four nodes stays the same.

In each adjustment of local topology, the relevant nodes can arrive at consensus or not that is decided by effect of changed topology on each node. If the topology adjustment evolves the local topology of relevant nodes to better structure, of course they should agree with the topology adjustment. If some nodes get worse topology structure, they compare the effect of topology adjustment and the node with bigger effect has the decision to adjust topology or not. At last, the local topology has to be adjusted when at least three nodes arrive at consensus.

According to the above adjustment process, each node adjusts its local topology according to its $T_{optimum}$ only when the relevant nodes arrive at consensus. So, the final optimized local topology of each node may not completely accord with its $T_{optimum}$ because some neighboring nodes may not arrive at consensus in topology adjustment.

All nodes adjust their local topology according to their $T_{optimum}$, then the network topology gradually evolves to optimum structure. In case that each node frequently requests for the topology adjustment to affect the performance of network communication, we set the topology stability parameter S as shown in Eq. 4, in which n denotes the node number in the local topology, and O_i denotes the criterion of node v_i' local topology.

$$S = \frac{\sum_1^n (O_r)}{n} (1 \leq r \leq n) \tag{4}$$

The topology stability parameter S is the average value of the topology criterion O of all nodes in the local topology. Each node v_i calculates the parameter S_i in its local topology. Then, the topology stability condition of node v_i can be set as $S_i \leq s_{max}$, in which s_{max} is the upper limit parameter for each node to adjust the sensitivity of the local topology optimization. The parameter s_{max} is smaller, the node implements the local topology optimization more frequently.

5 Performance Evaluation

In this section, the performance of anti-tracking network based on our proposal is evaluated through computer simulations. The simulation computer has a 12-Core

4 GHz CPU and 64 GB RAM. We first evaluate the effectiveness of our proposal (Topology Self-optimization, TS). Then, we compare TS with other two network topology optimization methods: neural network based network optimization method (NN) [22] and distributed hash table based network optimization method (DHT) [23] in network resilience.

5.1 Evaluation of Network Optimization

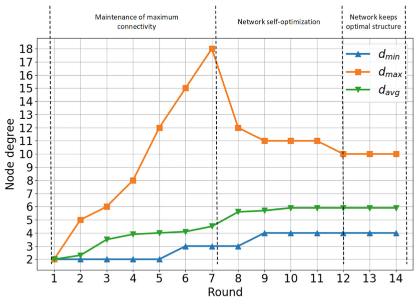
To evaluate the effectiveness of our proposal, we use ring topology and centralized topology seperately illustrated in Fig. 5(a) and Fig. 5(b) to construct a network with 1000 nodes, and deploy the self-optimization algorithm on this network. We use d_{min}, d_{max} and d_{avg} to denote the minimum node degree, maximum node degree and average node degree of all nodes respectively. We define one round of network self-optimization as that all nodes finish the optimization of its local topology. Then we calculate the above criteria in each round of network self-optimization to analyze the effectiveness of our proposal.

(a) Ring topology. (b) Centralized topology.

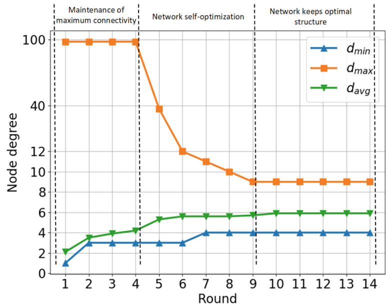
Fig. 5. Two topologies for performance evaluation of network self-optimization.

As shown in Fig. 6(a), the network is constructed in ring topology originally. The node degree of each node is 2. Before each node begins to optimize its local topology, it has to maintain its local topology in maximum connectivity. So, d_{max} increases sharply because the network needs to reach the maximum connectivity at first. Inevitably, the degree of some nodes will get bigger. After all nodes reach the maximum connectivity of their local topology, they begin to optimize their local topology. Then, d_{max} decreases until the network reaches the optimal structure. At last, d_{max} keeps nearly 10, d_{min} keeps nearly 4, and d_{avg} keeps nearly 6 which accords with the property of CPT shown in Eq. 1.

As shown in Fig. 6(b), the network is constructed in centralized topology in which one node has connections with all other nodes. So, the center node has a very high degree. At first, each node needs to maintain its local topology in maximum connectivity, d_{max} keeps unchanged for a few rounds. After the network reaches the maximum connectivity, d_{max} decreases sharply and keeps nearly 9 at last. The change of d_{max} shows the effectiveness of network self-optimization.



(a) The change of node degree in the self-optimization process of network based on ring topology.



(b) The change of node degree in the self-optimization process of network based on centralized topology.

Fig. 6. The change of β values of CPTs, NN, THash in random-p removal and top-p removal.

In order to present the effectiveness of network self-optimization intuitively, we calculate the node degree distribution after network self-optimization in ring topology and centralized topology respectively. D_r denotes the node degree distribution generated in ring topology, D_c denotes the node degree distribution generated in centralized topology. As illustrated in Fig. 7, the output topology of network self-optimization in both ring topology and centralized topology is almost the same. More than 80% of the nodes have the degree in the interval $[5, 8]$ which proves that the distribution of node degree is approximately close to uniform distribution. So, our proposal is effective to optimize the network into optimal structure.

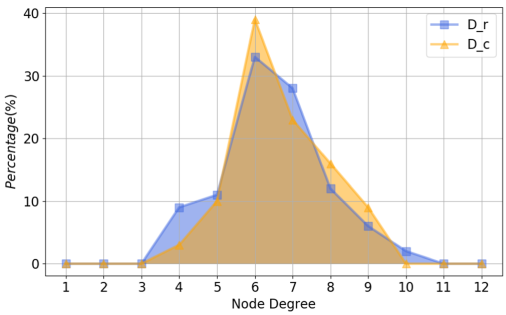


Fig. 7. The node degree distribution after network self-optimization in different topologies.

5.2 Evaluation of Network Resilience

To evaluate the resilience of anti-tracking network based on our proposal, we compare our proposal (CPTs) with neural network based network (NN) [22] and distributed hash tables based network (THash) [23] in the same scenario. NN achieves the self-optimization of network topology by the neural network algorithm depolyed in each node. THash implements a distributed and effective network optimization for DHT systems.

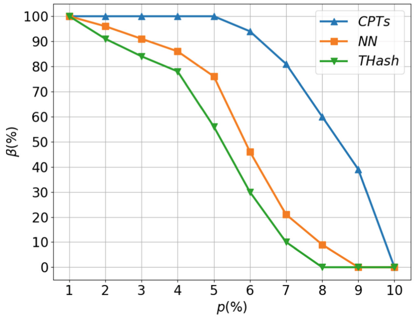
We seperately simulate three networks with 2000 nodes according to CPTs, NN and THash. Through removing p percent of nodes from the three networks each time, we use the node number of maximum connected graph to measure the network resilience when confronted to dynamic network scenario. We use Eq. 5 to quantify the performance of network resilience. $G(p)$ denotes the subgraph after p percent of nodes is removed from the original network, $MCS(G(p))$ denotes the maximum connected subgraph of $G(p)$, $Num(g)$ denotes the node number of a graph g, N_G denotes the node number of the graph G. The metric β measures the maximum connectivity of the network after some nodes are removed from the network. The β is higher, the network resilience is better.

$$\beta = \frac{Num(MCS(G(p)))}{N_G} \tag{5}$$

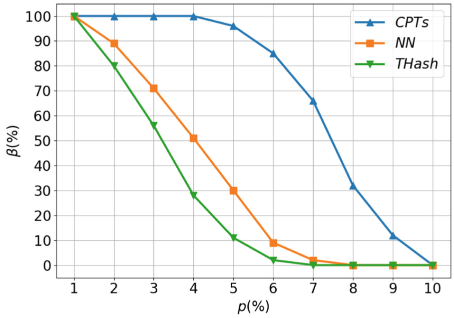
In the experiments, we use two different ways to remove nodes from network:

- **Random-p Removal:** In each round of nodes removal, we remove p percent of nodes from the network randomly.
- **Top-p Removal:** In each round of nodes removal, we remove p percent of nodes with the highest degree.

As shown in Fig. 8(a), β of CPTs decreases slowly which means CPTs has better network resilience in random-p removal than NN and THash. For top-p removal shown in Fig. 8(b), β of CPTs still decreases slowly, but β of NN and THash decreases sharply because top-p removal has bigger damage to network structure. But in both node removal methods, CPTs keeps good performance in network resilience. As we have mentioned above, the optimal structure of CPT has the maximum network connectivity conforming to the convex-polytope structure, and uniform distribution of node degree. So, it is not too much difference between top-p removal and random-p removal in the network with balanced distribution of node degree. Consider the ideal situation, if the node degree of all nodes in the network with TS is close to 6 (the average node degree of $CPT_{optimum}$), the removal of any nodes has the same effect on the topology structure.



(a) Random-p Removal.



(b) Top-p Removal.

Fig. 8. The change of β values of CPTs, NN, THash in random-p removal and top-p removal.

6 Conclusion

In this paper, we propose a topology self-optimization method for anti-tracking network via nodes distributed computing. Our proposal applies convex-polytope topology (CPT) in the construction of anit-tracking network. Based on CPT, we achieve the topology self-optimization for anti-tracking network. We also define an optimum structure of CPT in which network has maximum network connectivity and balanced distribution of node degree. Each node optimizes its local topology, then the whole network evolves into optimum structure of CPT through the collaboration of all nodes. Each node first calculates its optimum local topology according to its local topology situation. Then, each node negotiates with its neighboring nodes to adjust its local topology according to the calculated optimum local topology. When the relevant nodes arrive at consensus, the local topology will be adjusted according to the optimum local topology. Or, they will keep the local topology unchanged to make sure each adjustment of local topology is beneficial to all the relevant nodes.

In the experiments, we evaluate the network optimization and network resilience of our proposal. The experimental results show that our proposal has a good performance in network optimization. The network based on our proposal can achieve topology self-optimization effectively. Compared with the current network optimization methods, our proposal has better network resilience when confronting to dynamic network environment.

Acknowledgements. The authors would like to thank the anonymous reviewers for their insightful comments and suggestions on this paper. This work was supported in part by the National Key Research and Development Program of China under Grant No. 2019YFB1005203, the National Natural Science Foundation of China under Grant No. U1736218.

References

1. Hoang, N.P., Kintis, P., Antonakakis, M., Polychronakis, M.: An empirical study of the I2P anonymity network and its censorship resistance. In: Proceedings of the Internet Measurement Conference 2018, pp. 379–392 (2018)
2. Nambiar, A., Wright, M.: Salsa: a structured approach to large-scale anonymity. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 17–26 (2006)
3. Tian, C., Zhang, Y., Yin, T., Tuo, Y., Ge, R.: Achieving dynamic communication path for anti-tracking network. In: 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE (2019)
4. Kim, S., Han, J., Ha, J., Kim, T., Han, D.: SGX-TOR: a secure and practical tor anonymity network with SGX enclaves. IEEE/ACM Trans. Netw. **26**(5), 2174–2187 (2018)
5. Bauer, K., McCoy, D., Grunwald, D., Kohno, T., Sicker, D.: Low-resource routing attacks against tor. In: Proceedings of the 2007 ACM workshop on Privacy in electronic society, pp. 11–20 (2007)
6. Wright, M.K., Adler, M., Levine, B.N., Shields, C.: The predecessor attack: an analysis of a threat to anonymous communications systems. ACM Trans. Inf. Syst. Secur. (TISSEC) **7**(4), 489–522 (2004)
7. Kang, B.B., et al.: Towards complete node enumeration in a peer-to-peer botnet. In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, pp. 23–34 (2009)
8. Loesing, K., Murdoch, S.J., Dingledine, R.: A case study on measuring statistical data in the Tor anonymity network. In: Sion, R., et al. (eds.) FC 2010. LNCS, vol. 6054, pp. 203–215. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14992-4_19
9. Murdoch, S.J., Danezis, G.: Low-cost traffic analysis of tor. In: 2005 IEEE Symposium on Security and Privacy (S&P 2005), pp. 183–195. IEEE (2005)
10. Chen, T., Cui, W., Chan-Tin, E.: Measuring Tor relay popularity. In: Chen, S., Choo, K.-K.R., Fu, X., Lou, W., Mohaisen, A. (eds.) SecureComm 2019, Part I. LNICST, vol. 304, pp. 386–405. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37228-6_19
11. Das, D., Meiser, S., Mohammadi, E., Kate, A.: Anonymity trilemma: strong anonymity, low bandwidth overhead, low latency-choose two. In: 2018 IEEE Symposium on Security and Privacy (SP), pp. 108–126. IEEE (2018)
12. Cociglio, M., Fioccola, G., Marchetto, G., Sapio, A., Sisto, R.: Multipoint passive monitoring in packet networks. IEEE/ACM Trans. Netw. **27**(6), 2377–2390 (2019)
13. Shirazi, F., Diaz, C., Wright, J.: Towards measuring resilience in anonymous communication networks. In: Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society, pp. 95–99 (2015)
14. Chen, M., Liew, S.C., Shao, Z., Kai, C.: Markov approximation for combinatorial network optimization. IEEE Trans. Inf. Theory **59**(10), 6301–6327 (2013)
15. Alon, N., Awerbuch, B., Azar, Y., Buchbinder, N., Naor, J.: A general approach to online network optimization problems. ACM Trans. Algorithms (TALG) **2**(4), 640–660 (2006)
16. Chen, J., Touati, C., Zhu, Q.: A dynamic game approach to strategic design of secure and resilient infrastructure network. IEEE Trans. Inf. Forensics Secur. **15**, 462–474 (2019)

17. Shafigh, A.S., et al.: A framework for dynamic network architecture and topology optimization. IEEE/ACM Trans. Netw. **24**(2), 717–730 (2015)
18. Kang, S.: Research on anonymous network topology analysis. In: 2015 International Conference on Automation, Mechanical Control and Computational Engineering, pp. 1416–1421, Atlantis Press (2015)
19. Bouchoucha, T., Chuah, C.-N., Ding, Z.: Topology inference of unknown networks based on robust virtual coordinate systems. IEEE/ACM Trans. Netw. **27**(1), 405–418 (2019)
20. Jansen, R., Johnson, A.: Safely measuring Tor. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1553–1567 (2016)
21. Liu, Y., Zhuang, Z., Xiao, L., Ni, L.M.: AOTO: adaptive overlay topology optimization in unstructured P2P systems. In: GLOBECOM 2003. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489), vol. 7, pp. 4186–4190. IEEE (2003)
22. Tian, C., Zhang, Y.Z., Yin, T., Tuo, Y., Ge, R.: A smart topology construction method for anti-tracking network based on the neural network. In: Wang, X., Gao, H., Iqbal, M., Min, G. (eds.) CollaborateCom 2019. LNICST, vol. 292, pp. 439–454. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30146-0_31
23. Sun, Y., Richard Yang, Y., Zhang, X., Guo, Y., Li, J., Salamatian, K.: THash: a practical network optimization scheme for DHT-based P2P applications. IEEE J. Sel. Areas Commun. **31**(9), 379–390 (2013)
24. Liang, C., Liu, Y., Ross, K.W.: Topology optimization in multi-tree based P2P streaming system. In: 2009 21st IEEE International Conference on Tools with Artificial Intelligence, pp. 806–813. IEEE (2009)
25. Jelasity, M., Babaoglu, O.: T-man: Gossip-based overlay topology management. In: Brueckner, S.A., Di Marzo Serugendo, G., Hales, D., Zambonelli, F. (eds.) ESOA 2005. LNCS (LNAI), vol. 3910, pp. 1–15. Springer, Heidelberg (2006). https://doi.org/10.1007/11734697_1
26. Tian, C., Zhang, Y., Yin, T.: Modeling of anti-tracking network based on convex-polytope topology. In: Krzhizhanovskaya, V.V., et al. (eds.) ICCS 2020, Part II. LNCS, vol. 12138, pp. 425–438. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50417-5_32
27. Auvinen, A., Keltanen, Y., Vapa, M.: Topology management in unstructured P2P networks using neural networks. In: 2007 IEEE Congress on Evolutionary Computation, pp. 2358–2365. IEEE (2007)
28. Liao, Z., Liu, S., Yang, G., Zhou, J.: TTMP: a trust-based topology management protocol for unstructured P2P systems. In: He, X., Hua, E., Lin, Y., Liu, X. (eds.) Computer, Informatics, Cybernetics and Applications. LNEE, vol. 107, pp. 271–281. Springer, Dordrecht (2012). https://doi.org/10.1007/978-94-007-1839-5_29