2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

Resource Allocation Optimization with Guaranteed Security Level for Anti-traceable Access

Bo Jin[1,2], Hao Jin[2◇], Yashen Wang[2], Yangyang Li[2],
Xi Luo[1], Xiaoqing Gong[1], Xiaoyan Yin[1*]

[1]School of Information Science and Technology, Northwest University, Xi'an, China 710127.
[2]National Engineering Laboratory for Risk Perception and Prevention (RPP), China Academy of Electronics and Information Technology of CETC, Beijing, China 100041.
[◇]co-first author [*]Corresponding author
Email: jinbo1@stumail.nwu.edu.cn, jh_cetc@163.com, yswang@bit.edu.cn, liyangyang@cetc.com.cn, luoxi@stumail.nwu.edu.cn, gxq@nwu.edu.cn, yinxy@nwu.edu.cn

***Abstract*—Internet visitors can be well protected by hiding their traces. The purpose of traditional traceability is to obtain the user's IP address and physical location through some technical means. However, the essence of anti-traceability is to hide information browsing and search intention through trace hiding technology. Inspired by the PPM (Probability Packet Marking) strategy, an effective method of anti-tracing is to prevent the attacker from reconstructing the user's access path in a VPN network. We investigate a systematic study of anti-tracing to capture the relationship between resource allocation and security level guarantee in this paper. Furthermore, we propose a novel resource allocation and link scheduling algorithm with guaranteed security level. Numerical results verify that our proposed resource allocation solution can achieve good performance in terms of anti-tracing and security level guarantee.**

I. Introduction

Nowadays, Internet has been configured with traceability technology to prevent DDos attacks. The traceability technology is to reconstruct user's access path by collecting routing logs and analyzing communication packets, so as to achieve the purpose of tracing the IP addresses of users. This kind of defense can prevent DDos attacks to a certain extent, but increasingly, websites are profiting from users' personal information that can be traced back to them. For most normal users, this kind of behavior violates their personal privacy. It is necessary to protect the personal privacy of users with appropriate anti-tracing technology, especially for some government departments, they will often visit some websites with sensitive topics, and websites are likely to steal the personal information of users, at this time the consequences of information disclosure would be unimaginable.

Compared with the anti-tracing technology, more researches is working on the traceability attack. The main objective of a traceability attack is to reconstruct the access path of the attack target to obtain users' IP addresses. Therefore, building an path that is difficult for an attacker to reconstruct will be one of the important means to resist tracing attacks.

In recent years, the methods of tracing attacks include mainly Link Detection, Logging Technique, ICMP (Internet Control Message Protocol) message and PPM (Probability Packet Marking) strategy. Both Link Detection [1] and Logging Technique [2] require the support of network relay devices, while ICMP Message and PPM [3] [4] [5] [6] require insert specific fields into the IP header for marking. Among them, PPM algorithm [7] is the most popular algorithm of traceability attack. Savage et al. first proposed the PPM algorithm and analyzed its complexity. However, the complexity of these algorithms are too high in the calculation of complex access paths and have a high false positives ratio. To analyze individual packets, Alex et al. [4] designed an algorithm based on hash to track users' packets, while Hasmukh et al. [8] used a lightweight grouping tag algorithm to improve accuracy. At the same time, traceability technology based on package content analysis has also become a hot topic. Zhu et al. [9] analyzed the data source based on the SIR model, while Xie et al. [10] used causal analysis and random roaming to trace the source.

For PPM and other traceability algorithms, the longer the access path is, the higher the difficulty of tracing [3]. Correspondingly, the longer the access path is, the higher the time delay will be. However, blindly increasing the length of access path will prolong the delay indefinitely. Therefore, the tradeoff between delay and security level guarantee is indispensable.

In this paper, we establish a secure access path to defend against tracing attack. Links and the length of access path can be scheduled dynamically based on different security level requirements of users. Furthemore, the length of the access path can be accurately calculated by calculating the activation times of links. We can provide the recommended access path length for security level requirements. Combined with practical scenarios and theoretical derivation, we propose a minimum delay scheduling strategy that meets the constraints of users' security level requirements, and derive it as a delay optimization problem with security level constraints. Finallly, an approximate algorithm is proposed to solve the optimization problem.

We conduct simulations to verify the effectiveness of our proposed algorithm. With our resource allocation scheme, both system performance and security level can be guaranteed, and finally users' traces are hidden, and anti-tracing is realized.

978-0-7381-3199-3/20/$31.00 ©2020 IEEE
DOI 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00140

The rest of our paper is organized as follows. We introduce the system model in details in Section 2. In Section 3, we describe the resource allocation framework and problem formulation. In Section 4, we propose a solution based on the approximate algorithm. Simulation results are presented in Section 5. We survey the related works in Section 6. Finally, we conclude this paper in Section 7.

II. System Model

We consider a multi-hop VPN network $G(S, L)$ to model resource allocation for anti-traceable access, where S, which consist of domestic servers and foreign servers, is the set of VPN servers, L is the set of edges between VPN servers. The traffic originated by source node s_e and terminated at destination node d_e corresponds to session e, $e \in E$, where E is the set of sessions. Furthermore, each session has a specific security level requirement. Based on requirement for the security level, every session selects the appropriate path to meet its security level need and minimize transmission delay.

Communication Model. Every edge in the graph G is expressed as $l(i,j)$, $i \in S$, $j \in S$, and $j \in N_i$, where N_i is the set of neighbor nodes of VPN *i*. We adopt the time slot based link scheduling, and set one time slot as the minimum time for a packet is transmitted successfully over a link. The time slots imply a time sequence, link scheduling thus is equivalent to time slot allocation among sessions. We assume that all sessions are generated at slot 0, and complete the transmission of all packets at slot T. We use $n^e_{l(i,j)}(t)$ to represent that link $l(i,j)$ is activated or not by session e at time slot t, where $n^e_{l(i,j)}(t)$ is a binary variable, if the link $l(i,j)$ is activated by session e at slot t, $n^e_{l(i,j)}(t) = 1$, otherwise, $n^e_{l(i,j)}(t) = 0$. Mathematically, we have

$$n^e_{l(i,j)}(t) = \begin{cases} 1 & \textit{l(i,j)} \text{ is activated in slot } t \\ & \text{for session } e \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Based on Eq. (1), we can get the single-in constraint, then

$$\sum_{e\in E}\sum_{k\in N_i} n^e_{l(k,i)}(t) \le 1, i \in S, t \in N_i \quad (2)$$

Eq. (2) can be interpreted as the number of links activated in all up-links of node *i* should be less than or equal to 1 at any slot. In this way, we have the single-out constraint as:

$$\sum_{e\in E}\sum_{j\in N_i} n^e_{l(i,j)}(t) \le 1, i \in S, t \in N_i \quad (3)$$

For transmission scheduling, we assume that each session is unicast and the node is full-duplex, which means that node *i* can receive and send data at same time, but the number of links activated cannot exceed two at any slot. Thus, the full-duplex constraint is expressed as:

$$\sum_{e\in E} n^e_{l(i,j)}(t) + \sum_{e\in E} n^e_{l(k,i)}(t) \le 2, i \in S, \{j,k\} \in N_i, t \in T \quad (4)$$

We combine Eqs. (2), (3) and (4) to get the communication constraints of nodes, we have

$$\sum_{e\in E}\sum_{j\in N_i} n^e_{l(i,j)}(t) + \sum_{e\in E}\sum_{k\in N_i} n^e_{l(k,i)}(t) \le 2, (i \in S, t \in T) \quad (5)$$

Link Activation Model. We denote the scheduling frequency of link $l(i,j)$ as $f^e_{l(i,j)}$ for session e. Thus, we have

$$f^e_{l(i,j)} = \sum_{t\in T} n^e_{l(i,j)}, (i \in S, j \in N_i) \quad (6)$$

Then, the activation frequency over link $l(i,j)$ for all sessions during $[0,T]$ can be expressed as:

$$f_{l(i,j)} = \sum_{e\in E} f^e_{l(i,j)}, (i \in S, j \in N_i) \quad (7)$$

We use a binary variable $z^e_{l(i,j)}$ to represent the state of the link $l(i,j)$ activated or not by session *e*. If the number of times that link $l(i,j)$ has been activated equals 0, the link state is 0, otherwise, the state is 1. We have

$$z^e_{l(i,j)} = \begin{cases} 1 & f^e_{l(i,j)} \ge 1 \\ 0 & otherwise \end{cases} \quad (8)$$

Similarly, the state of link $l(i,j)$ which is activated or not by all sessions can be expressed as:

$$z_{l(i,j)} = \begin{cases} 1 & \sum_{e\in E} z^e_{l(i,j)} \ge 1 \\ 0 & otherwise \end{cases} \quad (9)$$

Throughput Analysis. We assume that the packet generation process for session *e* follow a Poisson distribution with parameter λ^e, and one packet of size D is generated at a time. We assume that only one packet is transmitted each time the link is activated. Thus, we can get the minimum throughput of link $l(i,j)$ as follows:

$$\mu^{min}_{l(i,j)} = D z_{l(i,j)}, i \in S, j \in N_i \quad (10)$$

Then, we define $W_{l(i,j)}$ to represent the bandwidth of link $l(i,j)$, (0,*T*]. Therefore, the maximum throughput over link $l(i,j)$ during $[0,T]$ can be expressed as:

$$\mu^{max}_{l(i,j)} = T \cdot W_{l(i,j)} \quad (11)$$

Eq. (10) is the minimum amount of transmission required to complete the transmission of session, and Formula (11) is the maximum amount of transmission that the link can load. Combined with (10) and (11), we can get the constraint of throughput:

$$\mu^{min}_{l(i,j)} \le \mu_{l(i,j)} \le \mu^{max}_{l(i,j)} \quad (12)$$

and the throughput $\mu_{l(i,j)}$ can be expressed by the following formula:

$$\mu_{l(i,j)} = f_{l(i,j)} \cdot W_{l(i,j)} \quad (13)$$

Delay Analysis. First, we analyze delay for a single-hop network, and then extend it to the multi-hop VPN network. We assume that s_e is the source node of session e and node j is its next hop, and the system time taken to transmit the packet over link $l(i,j)$ includes waiting time and sending time. Let $d^e_{l(se,j)}$ denote as the system time, $d^e_l(s_e,j) = w^e_l(s_e,j) + t^e_l(s_e,j)$, where $w^e_l(s_e,j)$ is the waiting time which equals to the sum of the packet generation interval and waiting time in the sending queue, and $t^e_l(s_e,j)$ is the transmission delay. As mentioned earlier, it is an unicast system, which means that the sending node only sends data to one certain receiving node at a specific time slot. Suppose packets on all nodes are scheduled in a First Come First Served (FCFS) mode, every VPN generates and then transmits packets with an M/M/1 queue. The expected waiting time can be calculated as follows:

$$w^e_l(s_e,j) = (\frac{\rho_{l(s_e,j)}}{1-\rho_{l(s_e,j)}})^2 \tag{14}$$

where $\rho_{l(s_e,j)}$ is the load ratio of $l(s_e,j)$, which equals to the ratio of traffic to bandwidth of link $l(s_e,j)$, i.e., $\rho_{l(s_e,j)} = f_{l(s_e,j)}/\mu_{l(s_e,j)}$. The higher the load ratio, the busier of system, and the longer of expected waiting time. The sending time can be computed as follows:

$$t^e_l(s_e,j) = \frac{\lambda_{l(s_e,j)}D}{\mu_{l(s_e,j)}} \tag{15}$$

In this way, the delay from the source node s_e to the next node j can be expressed as:

$$d^e_l(s_e,j) = (\frac{\rho_{l_{(s_e,j)}}}{1-\rho_{l_{(s_e,j)}}})^2 + \frac{\lambda_{l_{(s_e,j)}}D}{\mu_{l_{(s_e,j)}}} \tag{16}$$

We extend the single-hop delay analysis to the multi-hop network. Suppose that the session e has experienced h^e hops from the source node to the destination reception. Since the system is not fully loaded, the delay at each hop is mainly the time caused by packet transmission. Therefore, the delay of session e can be obtained as follows:

$$d^e = \sum_{h^e}(w^e_{l_{(i,j)}} + t^e_{l_{(i,j)}}) = \sum_{h^e}((\frac{\rho_{l_{(i,j)}}}{1-\rho_{l_{(i,j)}}})^2 + \frac{\lambda_{l_{(i,j)}}D}{\mu_{l_{(i,j)}}}) \tag{17}$$

Thus, the average delay of all sessions can be expressed as:

$$d^{ave} = \frac{\sum_{e\in E} d^e}{|E|} \tag{18}$$

III. Resource Allocation Framework

A. Problem Formulation

Taking the developed system model and the aforementioned constraints into account, we are interested in optimizing resource allocation to minimize the average delay over all links with guaranteed security level. Thus, out target problem can be formulated as follows:

$$\begin{aligned} OPT \quad & min\ d^{ave} \\ s.t. \quad & \text{Communication constraint: (5)} \\ & \text{Throughput constraint: (12)} \\ & \text{Hop requirement } h^e \end{aligned} \tag{19}$$

where d^{ave} is the average delay, which can be obtained based on Eq. (18). Then, we apply an equivalent transformation to it:

$$\begin{aligned} d^{ave} &= \frac{1}{|E|}\sum_{e\in E} d^e \\ &= \frac{1}{|E|}\sum_{e\in E}\sum_{h^e}(w^e_{l(i,j)} + t^e_{l(i,j)}) \\ &= \frac{1}{|E|}\sum_{e\in E}\sum_{h^e}((\frac{\rho_{l(i,j)}}{1-\rho_{l(i,j)}})^2 + \frac{D\lambda_{l(i,j)}}{\mu_{l(i,j)}}) \\ &= \frac{1}{|E|}\sum_{i\in S}\sum_{j\in T_i}((\frac{z_{l(i,j)}}{W_{l(i,j)}-1})^2 + \frac{D\lambda_{l(i,j)}}{W_{l(i,j)}f_{l(i,j)}}) \end{aligned} \tag{20}$$

where the packet generation rate of the session λ^e, the link bandwidth $W_{l(i,j)}$ and the packet size D are constants, while the activation frequency of the link $f_{l(i,j)}$ and hops of the access path h^e depend on the result of resource allocation. Obviously, it is a mixed-integer nonlinear programming problem. Generally, resource allocation in multi-hop networks is an NP-hard problem. An feasible solution in the next section will be proposed to deal with this problem.

B. Determine h^e

To guarantee security level, we start the security analysis with traditional traceability algorithms. The main idea of the traditional algorithm, PPM (Probabilistic Packet Marking) [7], is that the packets are marked with a certain probability and then each intermediate node receives the marked packet to reconstruct the path of the attack target.

Complexity Analysis. The algorithm complexity of PPM depends on the network scale. Furthermore, the packet size and transmission rate are different in different networks. Generally, we take the reconstruction time as the performance parameter of the algorithm. Reconstruction time is defined as the number of token packets an attacker needs to reconstruct the access path. We assume that the attacker marks the packet with a probability of p, and the length of the access path from the target node to the attacker is h. In this way, the probability that the target receives a marked packet can be expressed as:

$$P = p(1-p)^{h-1} \tag{21}$$

Based on Eq. (21), the expected number of marked packets required for path reconstruction, whose length is h, can be calculated as:

$$E(h) \geq \frac{\beta}{\gamma p(1-p)^{h-1}} \tag{22}$$

where β and γ are constants, β is related to link quality, and γ is related to network size. Based on Eq. (22), the reconstruction time can be computed as:

$$RT(h) = \frac{1}{p(1-p)^{h-1}} \tag{23}$$

From Eq. (23), we can know that the complexity is exponentially related to the length of the access path.

Security Analysis. As described above, the complexity of path reconstruction increases exponentially with the increase of path length. Inspired by the relationship between reconstruction complexity and path length, we will establish a function between user security requirement and path length.

Firstly, we quantify the security requirements of users. For simplicity, we rank the security requirement as security level. Secondly, we try to find a functional relationship between the security level and the access path length. Since an exponential relationship exists between the traceability difficulty and path length, we use the logarithmic relationship to establish the functional relationship between security level and path length as follows:

$$h^e = \lceil \alpha \ln(1+g^e) \rceil, h^e \geq h^{min}, e \in E \tag{24}$$

where h^e is the length of access path of session e, i.e., the number of hops, g^e is the security level of session e, and α is a parameter related to system parameters. Similarly, we can get the expression of security level:

$$g^e = \lceil e^{\frac{h^e}{\alpha}} - 1 \rceil, h^e \geq h^{min}, e \in E \tag{25}$$

The relationship between security level and hop requirement is shown in Fig. 1. The three curves in the figure is corresponding to $\alpha = 0.8$, $\alpha = 0.6$, and $\alpha = 0.3$, respectively. As shown in Fig. 1, the relationship between security level and number of hops follows a logarithmic function, which can avoid infinite increase of hops. At first, the number of hops required increases with the increase of the level. Gradually, the growth trend slows down. Finally, the number of hops reach a steady-state. Generally, larger α results in bigger number of hops.

IV. Resource Allocation with Guaranteed Security Level

Resource allocation has been formulated as as a mixed-integer nonlinear programming problem with multiple constraints, as shown as Eq. (19). Here, we will propose an feasible solution for the target problem.

A. Approximate Algorithm

The nonlinear part of OPT mainly comes from the optimization objective and transmission rate constraint. We use the approximation algorithm to transform the nonlinear part of the optimization objective into the linear part with controllable error. Firstly, we define a new function $m(x) = \frac{1}{x}$, and replace the objective function with $d^{ave} = \frac{1}{|E|}\sum_{i\in S}\sum_{j\in N_i}((\frac{z_{l(i,j)}}{W_{l(i,j)}-1})^2 + \frac{D\lambda_{l(i,j)}}{W_{l(i,j)}} \cdot m(f_{l(i,j)}))$. According

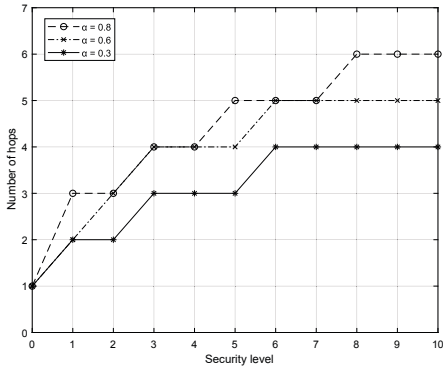

Fig. 1. The number of hops vs. security level.

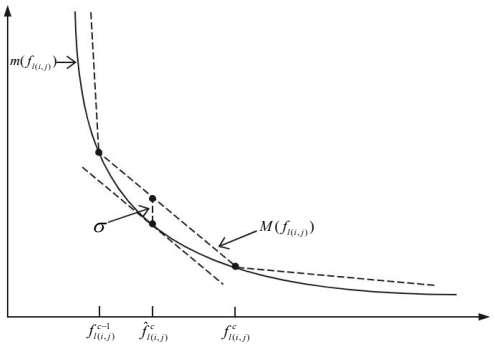

Fig. 2. The Piece-wise Approximation.

to Eqs. (6)-(12), $f_{l(i,j)} \leq T$ and $f_{l(i,j)} \geq \frac{\lambda_{l(i,j)}}{\mu_{l(i,j)}}$, where $\mu_{l(i,j)}$ is also a nonlinear parameter, it is constrained by Eq. (12). So we relax the left half of Eq. (12).

$$f_{l(i,j)} \geq \frac{\lambda_{l(i,j)}}{\mu_{l(i,j)}} \geq \frac{\lambda^{min}}{\mu^{max}_{l(i,j)}} = \frac{\lambda^{min}}{f_{l(i,j)} \cdot w_{l(i,j)}} = \sqrt{\frac{\lambda^{min}}{w_{l(i,j)}}} \tag{26}$$

Based on Eq. (26), we get $f_{l(i,j)}$ is in the range of $[\sqrt{\frac{\lambda^{min}}{w_{l(i,j)}}}, T]$. We can also get $\frac{\partial m(x)}{\partial x^2} = \frac{2}{x^3} > 0$. Thus, $m(x)$ is a convex function. It allows us to use piece-wise linearization technique to approximate $m(f_{l(i,j)})$. Here, our goal is to replace curve $m(x)$ with a minimum set of line segments, while ensuring that each segment does not deviate from curve $m(x)$ more than the given error σ. Suppose that C segments are the minimum number of segments required to represent the curve $m(x)$, and $f^0_{l(i,j)}, f^1_{l(i,j)}, \cdots, f^C_{l(i,j)}$ are the value of the endpoints of these segments on the X-axis. Since $f_{l(i,j)}$ is in the range of $[\sqrt{\frac{\lambda^{min}}{w_{l(i,j)}}}, T]$, We have $f^0_{l(i,j)} = \sqrt{\frac{\lambda^{min}}{w_{l(i,j)}}}$ and $f^C_{l(i,j)} = T$.

To find the line segments $f^0_{l(i,j)}, f^1_{l(i,j)}, \cdots, f^C_{l(i,j)}$, we start at the first point $f^0_{l(i,j)}$ and calculate the slope of the first line segment $q^1_{l(i,j)}$, and make sure that the error of this line segment does not exceed σ from the original curve. Given the starting point $f^0_{l(i,j)}$ and slope of the line segment $q^1_{l(i,j)}$,

we can calculate where the line segment intersects point $f^1_{l(i,j)}$ with the original curve. And then we take the intersection point $f^1_{l(i,j)}$ as the starting point for the next segment, and we repeat the process until we cover all the feasible regions of $f_{l(i,j)}$. Let $M^c(f_{l(i,j)})$ and $q^c_{l(i,j)}$ denote as the cth linear segment and its slope, we have

$$M^c(f_{l(i,j)}) = q^c_{l(i,j)} \cdot (f^c_{l(i,j)} - f^{c-1}_{l(i,j)}) + m(f^{c-1}_{l(i,j)}) \quad (27)$$

$$q^c_{l(i,j)} = \frac{m(f^c_{l(i,j)}) - m(f^{c-1}_{l(i,j)})}{f^c_{l(i,j)} - f^{c-1}_{l(i,j)}} \quad (28)$$

As shown in Fig. 2, we assume that $(\hat{f}^c_{l(i,j)}, M^c(\hat{f}_{l(i,j)}))$ is the point with the maximum error between the line segment and the original curve within the range $[f^{c-1}_{l(i,j)}, f^c_{l(i,j)}]$, and the following relation can be obtained:

$$M^c(\hat{f}^c_{l(i,j)}) - m^c(\hat{f}^c_{l(i,j)}) = \sigma \quad (29)$$

Since $(\hat{f}^c_{l(i,j)}, M^c(\hat{f}_{l(i,j)}))$ is a point on line segment $M^c(f_{l(i,j)})$, we can get the following equation:

$$q^c_{l(i,j)} = \frac{\partial m^c(\hat{f}^c_{l(i,j)})}{\partial f_{l(i,j)}} \quad (30)$$

According to Eqs. (29) and (30), we can compute the slope, and then we can plug in Eqs. (27) and (28) to get to the intersection. Notice that when $m(f^{c-1}_{l(i,j)}) \leq \sigma$, Eqs.(29) and (30) have no solutions. In this case, we set $f^c_{l(i,j)} = T$ as the end point and connect $(f^{c-1}_{l(i,j)}, m(f^{c-1}_{l(i,j)}))$ and $(T, m(T))$ as the last line segment. Therefore, for a given error σ, we can use Algorithm 1 to compute a set of intersection points $f^0_{l(i,j)}, f^1_{l(i,j)}, \cdots, f^C_{l(i,j)}$ and a set of slopes $q^1_{l(i,j)}, q^2_{l(i,j)}, \cdots, q^C_{l(i,j)}$ to represent the original curve.

Algorithm 1 Piecewise Linearization.

Initialization: c = 1 and $f^{c-1}_{l(i,j)} = \sqrt{\frac{\lambda^{min}}{w_{l(i,j)}}}$.
while $f^{c-1}_{l(i,j)} < T$ and $m(f^{c-1}_{l(i,j)}) > \sigma$ **do**
  Calculate slop $q^c_{l(i,j)}$ based on Eq. (29).
  Calculate intersection point $f^c_{l(i,j)}$ based on Eqs.(26)-(27).
  c = c + 1.
end while
if $f^{c-1}_{l(i,j)} \geq T$ **then**
  C = c - 1, $f^C_{l(i,j)} = T$ and recalculate the slope $q^C_{l(i,j)}$ based on Eq. (29)
else if $m(f^{c-1}_{l(i,j)}) \leq \sigma$ **then**
  C = c, $f^C_{l(i,j)} = T$ and calculate the slope $q^C_{l(i,j)}$ based on Eq. (29).
end if

Let $M(f_{l(i,j)})$ denote as the concatenated linear segments obtained from Algorithm 1. Then, we replace the optimization objective d^{ave} with a linear optimization function d^{ave}_L:

$$\begin{aligned} min \quad & d^{ave}_L \\ s.t. \quad & d^{ave}_L = \frac{1}{|E|} \sum_{i \in S} \sum_{j \in T_i} ((\frac{z_{l(i,j)}}{W_{l(i,j)} - 1})^2 + \frac{D\lambda_{l(i,j)}}{W_{l(i,j)}} M(f_{l(i,j)})) \\ & M^c(f_{l(i,j)}) = q^c_{l(i,j)} \cdot (f^c_{l(i,j)} - f^{c-1}_{l(i,j)}) + m(f^{c-1}_{l(i,j)}) \\ & M^c(\hat{f}^c_{l(i,j)}) - m^c(\hat{f}^c_{l(i,j)}) \leq \sigma \\ & (c = 1, 2, \cdots, C, f_{l(i,j)} \in [\sqrt{\frac{\lambda^{min}}{w_{l(i,j)}}}, T]) \end{aligned} \quad (31)$$

In this way, we have a new linear optimization problem to replace the original target problem. The linear optimization problem OPT-L is expressed as follows:

$$\begin{aligned} \textit{OPT-L} \quad & min\ d^{ave}_L \\ s.t. \quad & \text{Communication constraint: (5)} \\ & \text{Throughput constraint: (12)} \\ & \text{Hop requirement: (24)} \\ & \text{Approximation algorithm constraint: (31)} \end{aligned} \quad (32)$$

OPT-L is currently a mixed integer linear programming (MILP) problem, and we can use commercial solvers (e.g. CPLEX) to solve it effectively.

B. Toy Example

The scheduling process of *OPT-L* algorithm is shown as Fig. 3. There are 4 nodes, 2 sessions and 6 links. While link 1, 2 belong to the set of links between domestic servers, link 3, 4, 5, and 6 are international links. We assume that each session needs to go through at least 2 hops and transmits 2 packages. The source node of session 1 is *A*, the corresponding destination node is *D*, and the scheduled link is indicated by the solid arrow. The source node of session 2 is *B*, the corresponding destination node is *C*, and the scheduled link is indicated by the dotted arrow.

- $t = 0$, node *A* and node *B* generate packages 1, 2 and 3, 4, respectively.
- $t = 1$, the algorithm activates link 3 and link 4 to transfer package 1 and 3 simultaneously. Because of the minimum hops requirement, links 5 and 6 will not be activated. Although the delay over link 1 is low, there are two packages already in the waiting queue of nodes *A* and *B*, and the resulting long waiting time causes link 1 to be abandoned by A. Similarly, link 1 is also discard by B.
- $t = 2$, the algorithm activates link 1 and 2, and transmit package 1 to *D*, package 2 to *B*, package 3 to *C*, and package 4 to *A*. Since the link is full-duplex, bidirectional transmission is allowed. Because the queue length of all nodes is 1, but the delay over link 1 and 2 is low, links 1 and 2 are thus activated. At this point, packet 1 and 3 arrive at the destination node, and went through 2 hops.
- $t = 3$, the algorithm activates link 3 and 4 to transfer packages 3 and 4 to *C* and *D*, respectively, and transmits successfully all packets of session 1 and 2.

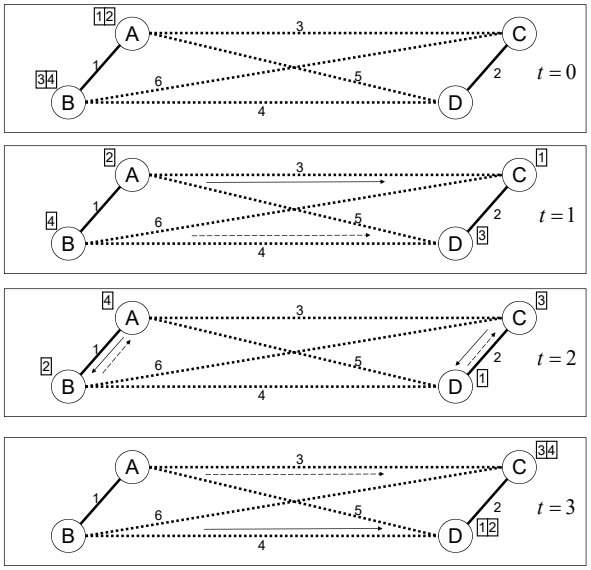

Fig. 3. An example of link scheduling.

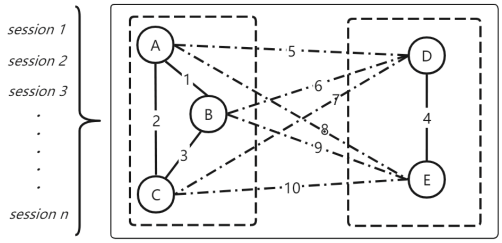

Fig. 4. The simulation topology.

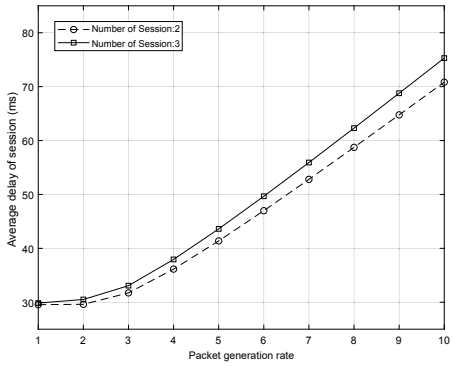

Fig. 5. Impact of packet generation rate on average delay.

C. Error Analysis

We replace the original problem with *OPT-L* and analyze the error boundary as follows:

$$OPT\text{-}L - OPT^* \leq \varepsilon \tag{33}$$

Proof: Assuming that OPT^* is the optimal solution of the original problem, the result is d^{ave*}. Since OPT^* satisfies all constraints of Eq. (32), we can construct a feasible solution *OPT-L*, whose $f_{l(i,j)}$ is the same as OPT^*. Accordingly, the difference between d^* and d_L^{ave} are calculated as follows:

$$\begin{aligned} d_L^{ave} - d^* &= \frac{1}{|E|}\sum_{e\in E}\sum_{h^e}[M(f_{l(i,j)}) - m(f_{l(i,j)})] \\ &\leq \frac{1}{|E|}\sum_{e\in E}\sum_{h^e}\sigma \end{aligned} \tag{34}$$

Then, we set $\frac{1}{|E|}\sum_{e\in E}\sum_{h^e}\sigma = \varepsilon$ to get Eq. (33). In this way, for the given error ε, we can calculate the linear error σ to get a set of line segments instead of the original curve, and then turn the problem into a MILP problem to solve.

V. Simulation

In this section, we evaluate the performance of the proposed resource allocation algorithm. For evaluation, the set of nodes is composed of 3 domestic VPNs and 2 overseas VPNs, as shown in Fig. 4, and the source node and destination node of a session can be any node. In this set of simulations, the bandwidth is chosen from [5M, 30M], and the delay over a specific link is inversely proportional to the distance between nodes.

Impact of Packet Generation Rate on Average Delay. We first analyze the influence of packet generation rate on the average delay, and the experimental results are shown in Fig. 5. We consider two cases where the number of sessions is 2 and 3, respectively. Generally, the delay of simultaneous transmission of three sessions is larger than that of two sessions. We assume that the hop requirement is 2 for all sessions, under packet generation rate $lamda$, which is chosen from [1, 10], the delay ascend slowly when λ is not larger than 2, and then the delay almost linear increases with λ. This is because the waiting time is negligible under light load, however, the delay raises rapidly when the traffic goes up.

After analyzing the impact of packet generation rate on average delay, we try to explore the link scheduling under different resource conditions. We assume that two sessions need to be served simultaneously, and the hop requirement of these two sessions are 2 and 3, respectively.

Link Scheduling with Insufficient Resources. The result of link scheduling under resource shortage are shown in Fig. 6. Here, the packet generation rate of both sessions is 6. There are 6 available links in the network, link 1, 4, 5 are domestic links and link 2, 3 and 6 are international links. As shown in Fig. 6(a), session 1 has activated the link for 12 times in total. Since the hop requirement is 2 and the packet generation rate is 6, the scheduling algorithm meets the minimum hop requirement of session 1. Similarly, Fig. 6(b) verifies that the scheduling algorithm can meet the minimum hop requirement of session 2. Combining Fig. 6(a) and Fig. 6(b), we can know that the scheduling algorithm assigns the links to session 1 and 2 evenly, session 1 uses link 1, 4, 5 and session 2 uses link 1, 2, 3, 6. As shown in Fig. 6(c), activation times of most

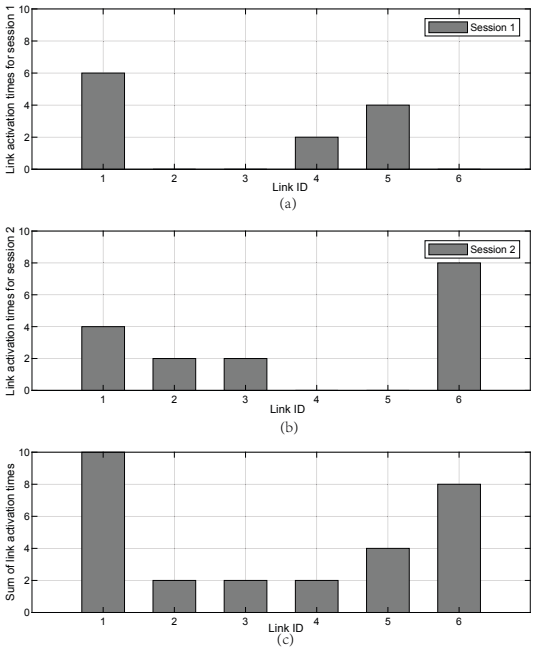

Fig. 6. Link Scheduling with Insufficient Resources.

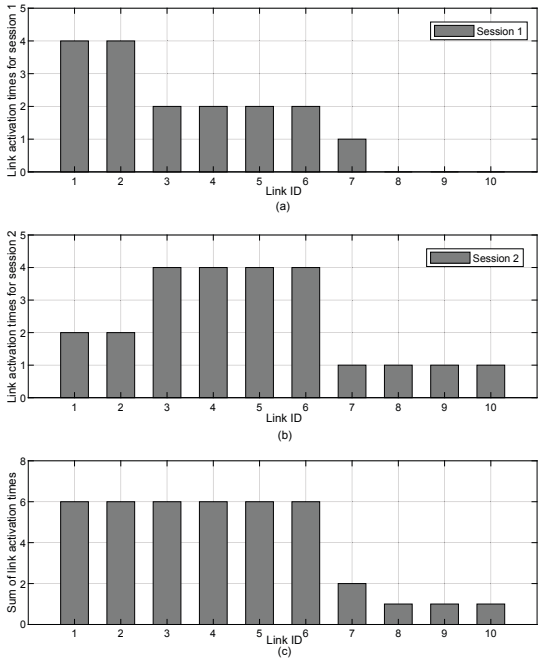

Fig. 7. Link Scheduling with Sufficient Resources.

links are similar, but links 2, 3 and 6 are significantly less than the others. This is because 2, 3 and 6 are international links, and their delay is higher than that of domestic links. To minimize the delay, the scheduling algorithm will activate domestic links. The activation times of link 1 are significantly higher than that of other links, because there are only three domestic links in total, and the domestic links are relatively deficient. In this case, Link 1 become the bottleneck link of system and be excessively scheduled.

Link Scheduling with Sufficient Resources. The result of link scheduling with sufficient resource are shown in Fig. 7. Here, the packet generation rate of both sessions is 8. There are 10 available links in the network, link 1, 2, 3, 4, 5, 6 are domestic links, and link 7, 8, 9 and 10 are international links. Similar results can be obtained, as shown in Fig. 7(a) and Fig. 7(b), while links are activated 17 times in session 1, and links are activated 16 times in session 2. Both sessions meet the minimum hop constraint. There is no bottleneck link in the network. The number of times the two types of links (domestic/international) are scheduled is basically the same. Combined Fig. 6 and Fig. 7, the proposed algorithm schedules low latency links with high priority when links with low delay are available, i.e., under sufficient resource. Furthermore, the experimental results show that our algorithm can efficiently balance load for links of the same kind, and improve the system performance.

Hop Count Comparison. Comparison of the minimum hop requirement and actual hops is shown as Fig. 8. As mentioned earlier, the minimum hop requirement increases with guaranteed security level. We can konw that both sessions

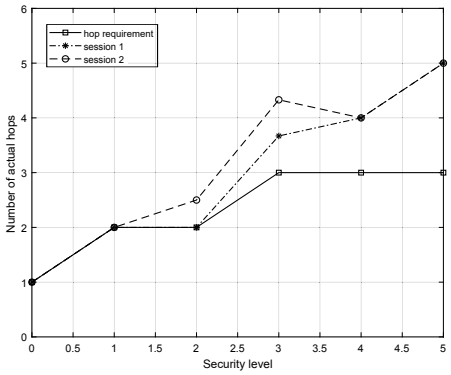

Fig. 8. Hop count comparison.

satisfy the minimum hop constraint from the simulation result of the proposed algorithm. Furthermore, the actual number of hops experienced by sessions is slightly higher than the minimum hop requirement. The reason is, to achieve the optimization goal of minimizing the delay, it is necessary to bypass the bottleneck link from time to time, so the number of hops is higher than needed.

VI. Related Work

In recent years, many works have been conducted on traceablity and anti-traceablity, which share common principle. Existing research can generally be classified into two categories.

Traceablity based on PPM and route log. The traceablity technology is mainly based on probabilistic packet marking

algorithm and routing log design. In [3], the algorithm complexity of probabilistic packet marking algorithm was analyzed, and a backtracking algorithm based on packet marking is proposed. In [4], an IP traceability algorithm based on hash was designed to track IP packets sent from a single source network. In [5], the issue of maintaining ISP network privacy in a tag-based backtracking solution was investigated. In [11], an IP backtracking algorithm was designed based on the OpenFlow Controllers. In [12], Shi et al. analyzed the network security based on SD-VPN network and OpenFlow protocol. In [8], an improved PPM algorithm was proposed to promote accuracy. For large-scale access, the DDoS attack was analyzed in [13] and [14], and the prevention and mitigation plan were put forward. Tracing using tagged packet headers is a means of tracing a single packet or a small amount of access, the authentication marking scheme of [15] has high precision but requires hardware support. In [6], the packet header was combined with signature technology to mark the source of the packet. In [16], Belenky et al. proposed the DPM algorithm to trace through a small number of packets. Recording routing logs and designing a reasonable traceability scheme is another research direction of traceability. In [1], routing information was recorded by setting up a path-aware history recorder and tracing the route when necessary. In [2], Yang et al. proposed a hybrid traceability scheme combining packet marking and routing log.

Traceablity based on package content analysis. In an uncontrolled network system, marking and logging may not be possible. The main solution is to trace the data packet content. In [17], Jiang et al. traced the source by analyzing the network topology and propagation scheme. In [9], Zhu et al. analyzed the data sources based on the SIR model. The network structure was reconstructed based on social network in [18] and [19]. In [18], Hao et al. proposed a mining framework. Traceablity based on reasoning mechanism was present in [19]. A multi-community cloud cooperation model was proposed for the security of community cloud in [20]. In [10], Xie et al. used causal analysis and random roaming to trace the source. In [21], Shah et al. constructed the maximum likelihood tree to trace the source. In [22], Luo et al. analyzes the infection spread in large networks.

VII. Conclusion

In this study, we explore the resource allocation problem for secure transmission in multi-hop VPN networks. Taking link type, transmission constraint and queue model into consideration, we develop a mathematical model to find the relationship between the number of hops and security level requirement. Further, we can confirm that the security level can be guaranteed by resource allocation aiming at minimizing delay. Finally, the formulated objective function is transformed into linear integer programming by piecewise linearization technique. Through extensive simulation results, we verified that our proposed algorithm can allocate resource efficiently, balance load effectively and guarantee the security level.

VIII. Acknowledgment

This research is supported in part by National Natural Science Foundation of China under Grant 61872295, the New Generation of Artificial Intelligence Special Action Project under Grant AI20191125008 and Shaanxi Natural Science Foundation under Grant 2020JM-416.

References

[1] N. Handigol, B. Heller, V. Jeyakumar, D. MaziÍĺres, and N. McKeown, "I Know What Your Packet Did Last Hop: Using Packet Histories to Troubleshoot Networks," in *NSDI'14*, 2014, pp. 71–85.

[2] M.-H. Yang and M.-C. Yang, "RIHT: A Novel Hybrid IP Traceback Scheme," in *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, 2012, pp. 789–797.

[3] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical Network Support for IP Traceback," in *SIGCOMM*, 2000, pp. 295–306.

[4] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer, "Hash-Based IP Traceback," in *SIGCOMM*, 2001, pp. 3–14.

[5] L. Su, D. M. Divakaran, and V. Thing, "Privacy Preserving IP Traceback," in *IEEE 4th International Conference on Identity, Security, and Behavior Analysis*, 2018.

[6] X. Song and A. Perrig, "Advanced and Authenticated Marking Schemes for IP Traceback," in *INFOCOM*, 2001, pp. 878–886.

[7] S. Kaul, R. Yates, and M. Gruteser, "Real-Time Status: How Often Should One Update?" in *INFOCOM*, 2012, pp. 2731–2735.

[8] H. Patel and D. C. Jinwala, "LPM: A lightweight authenticated packet marking approach for IP traceback," in *Computer Networks*, 2018, pp. 41–50.

[9] K. Zhu and L. Ying, "Information Source Detection in the SIR Model: A Sample-Path-Based Approach," in *IEEE/ACM TRANSACTIONS ON NETWORKING*, 2016, pp. 408–421.

[10] Y. Xie, V. Sekar, D. A. Maltz, M. K. Reiter, and H. Zhang, "Worm Origin Identification Using Random Moonwalks," in *IEEE Symposium on Security and Privacy*, 2005.

[11] N. Yoshiura and H. Yano, "IP Traceback method by OpenFlow," in *ICSIM*, 2020, pp. 194–198.

[12] L. Shi, F. Wang, and C.-H. Lung, "Improvement of Security and Scalability for IoT Network Using SD-VPN," in *IEEE NOMS*, 2018.

[13] H. Dai, Y. Wang, J. Fan, and B. Liu, "Mitigate DDoS Attacks in NDN by Interest Traceback," in *INFOCOM*, 2013, pp. 381–386.

[14] R. Saxena and S. Dey, "DDoS attack prevention using collaborative approach for cloud computing," in *Cluster Computing*, 2020, pp. 1329–1344.

[15] T. Baba and S. Matsuda, "Tracing Network Attacks to Their Sources," in *IEEE INTERNET COMPUTING*, 2002, pp. 20–26.

[16] A. Belenky and N. Ansari, "IP Traceback With Deterministic Packet Marking," in *IEEE COMMUNICATIONS LETTERS*, 2003, pp. 162–164.

[17] J. Jiang, S. Wen, S. Yu, Y. Xiang, and W. Zhou, "Identifying Propagation Sources in Networks: State-of-the-Art and Comparative Studies," in *IEEE COMMUNICATIONS SURVEYS TUTORIALS*, 2017, pp. 465–481.

[18] F. Hao and D.-S. Park, "cSketch: A Novel Framework for Capturing Cliques from Big Graph," in *The Journal of Supercomputing*, 2018, pp. 1202–1214.

[19] F. Hao, G. Min, M. Lin, C. Luo, and L. Yang, "MobiFuzzyTrust: An Efficient Fuzzy Trust Inference Mechanism in Mobile Social Networks," in *IEEE Transactions on Parallel and Distributed Systems*, 2014, pp. 2944–2955.

[20] F. Hao, G. Min, J. Chen, F. Wang, M. Lin, C. Luo, and L. Yang, "An Optimized Computational Model for Multi-Community-Cloud Social Collaborative Computing," in *IEEE Transactions on Services Computing*, 2014, pp. 346–358.

[21] D. Shah and T. Zaman, "Rumor Centrality: A Universal Source Detector," in *SIGMETRICS*, 2012, pp. 199–210.

[22] W. Luo, W. Tay, and M. Leng, "Identifying Infection Sources and Regions in Large Networks," in *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, 2013, pp. 2850–2865.