Article

On the Optimal Lawful Intercept Access Points Placement Problem in Hybrid Software-Defined Networks

Xiaosa Xu, Wen-Kang Jia *, Yi Wu and Xufang Wang

Fujian Provincial Engineering Technology Research Center of Photoelectric Sensing Application, Key Laboratory of OptoElectronic Science and Technology for Medicine of Ministry of Education, College of Photonic and Electronic Engineering, Fujian Normal University, Fuzhou 350007, China; xiaosaxu521@163.com (X.X.); wuyi@fjnu.edu.cn (Y.W.); fzwxf@fjnu.edu.cn (X.W.)

* Correspondence: wkjia@fjnu.edu.cn

Abstract: For the law enforcement agencies, lawful interception is still one of the main means to intercept a suspect or address most illegal actions. Due to its centralized management, however, it is easy to implement in traditional networks, but the cost is high. In view of this restriction, this paper aims to exploit software-defined network (SDN) technology to contribute to the next generation of intelligent lawful interception technology, i.e., to optimize the deployment of intercept access points (IAPs) in hybrid software-defined networks where both SDN nodes and non-SDN nodes exist simultaneously. In order to deploy IAPs, this paper puts forward an improved equal-cost multi-path shortest path algorithm and accordingly proposes three SDN interception models: T interception model, ECMP-T interception model and Fermat-point interception model. Considering the location relevance of all intercepted targets and the operation and maintenance cost of operators from the global perspective, by the way, we further propose a restrictive minimum vertex cover algorithm (RMVCA) in hybrid SDN. Implementing different SDN interception algorithms based RMVCA in real-world topologies, we can reasonably deploy the best intercept access point and intercept the whole hybrid SDN with the least SDN nodes, as well as significantly optimize the deployment efficiency of IAPs and improve the intercept link coverage in hybrid SDN, contributing to the implementation of lawful interception.

Keywords: lawful interception; hybrid SDN; intercept access point; minimum vertex cover

Citation: Xu, X.; Jia, W.-K.; Wu, Y.; Wang, X. On the Optimal Lawful Intercept Access Points Placement Problem in Hybrid Software-Defined Networks. *Sensors* **2021**, *21*, 428. <https://doi.org/10.3390/s21020428>

Received: 16 December 2020

Accepted: 4 January 2021

Published: 9 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

National security and social stability, in today's world, have been shaken by some security threats such as terrorist attacks, cybercrime and information warfare. For the law enforcement agencies (LEAs; L), therefore, lawful interception (LI) is still one of the main means to intercept a suspect or address these illegal actions at present. As we all know, lawful interception is a kind of data acquisition of communication network based on lawful authorization for the purpose of analysis or evidence collection. Thus, it allows the law enforcement agencies with court orders or other legitimate authorities to selectively eavesdrop on individual users. Most countries require those licensed telecom operators to provide legitimate interception gateways and nodes on their networks for communication interception. To deploy the gateways and nodes in legacy networking where traditional gateways or nodes rely on dedicated devices and backhaul links to intercept network traffic, however, leads to unimaginable cost. On the contrary, software-defined networking (SDN) [1], different from the traditional networking, can simplify the traditional network architecture [2] and thus enable efficient management and centralized control [3] for intercepting network traffic at an extremely low cost because of its property of software definition with OpenFlow protocol [4]. The deployment of SDNs, however, is not a one-step process, but a long process, namely, in the wake of the increasing deployment of SDNs [5], a situation where both SDN nodes and non-SDN (N-SDN) nodes exist simultaneously is

formed gradually. Therefore, it is of great significance to study how to design a brand-new network information lawful interception system architecture based on the software-defined network (SDN) technology and to discuss its challenges such as the deployment of intercept access point (IAPs), route selection of intercept, the minimum cost of intercept, the minimum number of intercept access points etc. in a hybrid SDN.

In this paper, we propose the deployment and optimization strategy of intercept access points, which includes single intercept access point selection, the shortest route optimization algorithm between three points, the minimum intercept traffic cost algorithm, and the restrictive minimum vertex cover algorithm.

The problem of single intercept access point selection is the shortest path problem that is to solve the shortest path between two given vertices in a weighted graph. At this time, the shortest path not only refers to the shortest path in the sense of pure distance, but also in the sense of economic distance, time and network. In this paper, the cost of shortest path between two points can refer to hop-count, traffic, transmission delay, transmission bandwidth, energy consumption etc. As is known to all, Dijkstra Algorithm [6] is the most typical single source shortest path algorithm, which is used to calculate the shortest path from one node to all nodes, and yet not all equal-cost multi-path shortest path. Meanwhile, Li [7] proposed an improved Dijkstra Algorithm that can find most of the shortest paths using the initial shortest path set through applying for concept of precursor node but cannot find all shortest paths. Moreover, a lot of related work with respect to the shortest path have been done by [8–14] in various fields.

In view of this, we develop an improved equal-cost multi-path shortest path algorithm (i.e., ECMP-Dijkstra) which can find all shortest paths between the source (S) and the destination (D), and accordingly put forward three SDN interception models based on ECMP-Dijkstra Algorithm in hybrid SDN. The three SDN interception models can be viewed as a cost-effective three-point shortest path algorithm with low time and space complexity, and thus can be used to deploy the best intercept access point reasonably in hybrid SDN.

The optimization of traffic engineering in hybrid SDN, like [15–17], is also one of our focuses. This study mainly concerns with the best transmission quality of intercepted data, the minimum cost of returning intercepted data to the interception center (i.e., LEA; L), the total traffic in global network, the transmission quality of traffic normally accepted by users when deploying intercept access points.

In reality, the deployment of intercept access points in the Internet does not simply corresponds to the micro perspective of a single data flow between three points. There is a very dynamic and complex traffic matrix [18] relationship and interactive influence among hundreds of millions of nodes in the large-scale Internet. A certain intercept access point (IAP; I) can meet the demand of traffic between S-D (from S to D) path, but there are also tens of millions of other traffic between intercept target node pairs, which may also flow through I node at the same time. Therefore, it is very important to select the deployment location of intercept access point, which must occupy the hub position, and greatly covers all intercepted traffic and must go through the critical path. For this reason, the location relevance of all intercepted targets and the operation and maintenance cost of operators must be taken into consideration from the global perspective, and thus the deployment problem of intercept access points is viewed as the minimum vertex cover problem (MVCP) that is NP-complete [19] to find its solution.

A lot of investigations have been done on MVCP in theory and applications for the last several decades [20–22]. Some parameterized algorithms about MVCP have been applied in biochemistry [23,24]. Moreover, the optimal approximation algorithm for MVCP have been proposed in [25–30]. Authors in [25–30] proposed the approximate optimization algorithm for MVCP by using the concept of degree.

Referring to their proposed algorithm, we develop a restrictive minimum vertex cover algorithm (RMVCA) in hybrid SDN networks to optimize the deployment efficiency of IAPs and to improve the link coverage of the whole interception system.

The ultimate aim of this paper is to contribute to the theory of lawful interception technology, the development of Internet and national security. In summary, the main contributions of this paper are as follows:

- To solve the problem of single intercept access point selection and routing between three points, we develop an improved equal-cost multi-path shortest path algorithm (i.e., ECMP-Dijkstra) and accordingly put forward three SDN interception models (e.g., T model, ECMP-T model and Fermat-point model) to deploy the best intercept access point reasonably in hybrid SDN, realizing the effective deployment of intercept access point in lawful interception system.
- Considering the location relevance of all intercepted targets and the operation and maintenance cost of operators of the whole interception system, we proposed a restrictive minimum vertex cover algorithm (RMVCA) to intercept the whole interception system with the least SDN nodes, optimize the deployment efficiency and improve the intercept link coverage for the whole interception system when deploying IAPs.
- Based RMVCA, we put forward three approaches PA, RA, and HA for experiments, and study and analyze the impact of different approaches on the efficiency of deploying intercept access points and on the intercept link coverage in hybrid SDN, to seek out the best RMVCA approach.
- We study and analyze the impact of different SDN interception models on various performance metrics of lawful interception system by using three real-world topologies, to seek out the best interception model.

In this paper, we first analyze various SDN interception models in hybrid software-defined networks and propose their algorithms, and then develop a restricted minimum vertex coverage algorithm from a global perspective. Extensive simulation results based on real-world network topology show that RMVCA can significantly improve network interception link coverage and deployment efficiency of IAPs of whole interception system, and that the performance metrics of the interception system are the best when Fermat-point interception model is adopted.

The remainder of this article is structured as follows. Section 2 surveys relevant work and Section 3 presents ECMP-Dijkstra Algorithm and SDN interception models. We propose the RMVCA in Section 4, followed by the performance evaluation of RMVCA and SDN interception models in Section 5. Then, Section 6 concludes the paper.

2. Related Works

Table 1 presents comparisons between our proposed and the related works according to different parameters.

Table 1. Comparisons of related works.

Lawful Interception	[31]	[32]	Our Proposed
SDN based	No	Yes	Yes
Cost	Very High	Low	Low
The shortest path algorithm	[6]	[7–14]	Our proposed
Time-Space complexity	Low	Medium	Medium-Low
The number of ECMP	One	The most	All
Minimum Vertex Cover	[33]	[20–30]	Our proposed
Time-Space complexity	Low	Medium	Medium-Low
Results	Very bad	Near-optimal	Near-optimal

2.1. Lawful Interception (LI) and Hybrid Software-Defined Networks (H-SDNs)

With the dramatic development of the Internet, an increasing number of people commit crimes on the Internet, and criminal activities are extremely rampant, which seriously affect people's security and national stability. Thus, lawful interception (LI) is still one of the momentous means for the law enforcement agencies (LEAs) to maintain national security, crack down on crime and prevent cybercrime. For interception system, an

intercept device is installed to intercept network traffic, and copies it back to LEAs, and then carries out identifying and analyzing by manual or machine. With the development of new network technology and the continuous increase of network traffic, it is a more and more common and difficult task to carry out lawful interception on the Internet [31] for helping tracking culprits and to understand the nature and behavior of current Internet traffic.

With the development of SDN technology, legacy Ether-net switches are gradually migrating to SDN, and this process is harmless [32]. Although the emerging SDN networks that provides programmability to networks can have an improvement in implementing traffic engineering (TE), management departments still hesitate to deploy SDN fully because of various reasons such as budget constraints, risk considerations as well as service level agreement (SLA) guarantees. This results in developing SDN network incrementally, i.e., to deploy the SDN network only through migrating fewer SDN switches in legacy network, thus, to form the hybrid SDN networks (H-SDNs). H-SDN network provides a coexistence and cooperation environment for N-SDN nodes and SDN nodes, which brings many benefits to traditional IP networks. For the near-optimal performance of traffic engineering, therefore, it is crucial to maximize the benefits of SDN with minimal SDN deployment. Therefore, it is imperative to deploy SDN intercept access point in a hybrid SDN (H-SDN) network where SDN nodes (routers) and legacy nodes coexist and operate in perfect harmony, realizing lawful interception. In H-SDN, the links between SDN nodes and between SDN nodes and N-SDN nodes can be intercepted (i.e., SDN links), and the links between N-SDN nodes cannot be intercepted (i.e., N-SDN links) due to the lack of special equipment and dedicated return link in hybrid SDN. In other words, in the interception system based SDN, the law enforcement agencies (LEAs) do not have to set up special equipment and a dedicated line in traditional IP networks, but can intercept traffic of links through SDN intercept access point to respond to requests from the interception center, which can greatly reduce the cost of traditional special equipment and leased lines. The interception system based SDN will be no longer restricted by the bandwidth of the intercepting dedicated equipment and link. By deploying intercept access point, the interception system will have a lot of redundant links or paths to be employed to return data flow, thus, to reduce or avoid the risk of single point failure or to further guarantee the multi-path routing method.

Therefore, the deployment of SDN intercept access point in interception system is helpful to perfect the route of intercepting traffic, to make full use of Internet bandwidth resources, to improve user's quality of service, and to further optimize the performance of the whole interception system.

2.2. Dijkstra Algorithm

The most classic single source shortest path algorithm is Dijkstra Algorithm [6], which was proposed in 1956 and became well-known three years later. Dijkstra Algorithm can calculate the shortest path from one node to all nodes, yet not all equal-cost multi-path shortest path.

Many modified algorithms based on Dijkstra Algorithm are proposed in [7–14]. An improved algorithm of Dijkstra Algorithm was proposed by Li [7]. Under the concept of precursor node, Li exploited the initial shortest path set calculated by Dijkstra Algorithm, to calculate most but not all of the shortest paths. The authors of [8] improved Dijkstra Algorithm for solving three issues, such as the ineffective mechanism to digraph. In addition, the work [9] proposed some modifications on Dijkstra Algorithm and made the number of iterations less than the number of the nodes. Work [10] proposed an optimized algorithm based on Dijkstra Algorithm to optimize logistics route for the supply chain. On the other hand, the study [11] modified Dijkstra Algorithm and the modified algorithm is very of efficiency for public transport route planning. Work [12] used Dijkstra Algorithm towards shortest path computation in navigation systems for making sensible decision and time saving decisions. By the way, the study [13] improved Dijkstra Algorithm to find the

maximum load path. Work [14] introduced an improved Dijkstra Algorithm for analyzing the property of 2D grid map and increased significantly the speed of Dijkstra Algorithm.

Referring to their proposed algorithms, we also improve Dijkstra Algorithm and propose an improved equal-cost multi-path shortest paths algorithm (ECMP-Dijkstra), which can calculate all equal-cost shortest paths from one node to all nodes, thus developing a cost-effective shortest path optimization algorithm between three points (i.e., S, D and L) with low time and space in hybrid SDN.

2.3. The Minimum Vertex Cover Problem (MVCP)

The traditional algorithm to solve the minimum vertex cover algorithm (MVCP) is 2-approximation [33]. This algorithm can find the set of vertex cover which is no more than twice of the optimal vertex cover, and the time complexity of the algorithm is $O(E+V)$. More importantly, the results obtained by this algorithm are different each time, and thus may be inaccurate and not approximate solution. However, this algorithm has its advantages: every time a vertex is selected, and all the edges connected by the vertex are deleted.

The authors in [20–22] made much contribution to MVCP in theory and applications. The authors in [23,24] proposed parameterized algorithms for MVCP, and applied them in biochemistry. Work [25] proposed an improved greedy algorithm for minimum vertex cover problem, and the algorithm used the concept of degree (i.e., the number of links connected by a node) to carry out an order of degree and to select the node with the largest degree to add to the minimum vertex cover set until the degree of all nodes is 0 (i.e., the vertexes in the minimum vertex cover set has covered all the edges). Thus, the result is a very excellent approximate solution. However, the process of judging the degree of the algorithm is too complicated. Authors in [26] presented a greedy heuristic algorithm for MVCP to offer better results on dense graphs. The study [27] presented a breadth first search approach, which can get the exact result of MVCP for grid graphs. Work [28] proposed a near-optimal algorithm named MAMA to optimize the unweighted MVCP, and MAMA can return near optimal result in quick-time. Authors in [29] proposed a NHGA for MVCP to yield near-optimal solutions. In [30], authors studied an ameliorated genetic algorithm for the partial VCP to skip the local optimum by powerful vertex and adaptive mutation. All of their algorithm are based on the concept of degree.

Combining with the advantages of the above algorithms, we proposed an ameliorated restrictive minimum vertex cover algorithm (RMVCA) in hybrid SDN using the concept of degree to significantly simplify the process of degree judgment and to yield near-optimal result, thus, in the whole interception system, realizing the optimization of the deployment efficiency of IAPs and the improvement of intercept link coverage.

3. ECMP-Dijkstra Algorithm and SDN Interception Models

3.1. ECMP-Dijkstra Algorithm

When deploying the best intercept access point in hybrid SDN, we have to calculate all equivalent shortest paths between two points and then select out the best route from all equal-cost shortest paths to choose the best node as IAP. The most typical single source shortest path algorithm is Dijkstra Algorithm [6]. Accordingly, an improved algorithm of Dijkstra Algorithm was proposed by Li [7]. Under the concept of precursor node, Li exploited the initial shortest path set calculated by Dijkstra Algorithm, to calculate most but not all of the shortest paths. In view of this, on the basis of Dijkstra Algorithm and Li's Algorithm, we propose an improved equal-cost multi-path shortest paths algorithm (ECMP-Dijkstra), which can calculate all equal-cost shortest paths from one node to all nodes. The notations used in the algorithms and in the following equations are listed in Table 2.

Table 2. Notations.

Notation	Meaning
N_{SDN}	the SDN nodes selected randomly from all nodes in H-SDN
S, D, L	the source and the destination and the interception center or the LEA
I	the set of the best intercept access points
$sp_{S,D}$ or sp_{S-D}	the shortest path from node S to node D
$N_{S,D}$	the set of nodes in the shortest path $sp_{S,D}$
SN	the set of SDN nodes
i	the SDN node or SDN devices
j	the index of the j -th element of a vector
$h(i)$	the set of hop-count, $h(i)$ denotes the hop-count or cost of node i
$numh(j)$	the set of costs, $numh(j)$ denotes the cost of the j -th element
$inh(j)$	$inh(j)$ denotes the node with the index of j
$minhops$	the minimum cost or hop-count
β	the maximum index
N	the number of nodes
$hops(i,j)$ or $hops_{i-j}$	the minimum hop-count or cost from node i to node j

The pseudo code of Dijkstra Algorithm is given in Algorithm 1. We input the source node s and an undirected graph $G(V, E)$ where V denotes the set of all nodes and E denotes the set of all edges. We explain Algorithm 1 that ∞ denotes an infinity and $sp_{s,i}$ denotes the shortest path from the source node s to node i . In lines 11–14, we get the minimum hop-count value $minhops$ and the corresponding node key. In lines 15–19, we remove node key from U and then add node key to S and add node key to the shortest path $sp_{s,i}$ to get the shortest path $sp_{s,key}$. Finally, we obtain the shortest path set SP from the source node s to all nodes in V .

Algorithm 1 Dijkstra Algorithm

Input: $s; G(V, E)$
Output: SP

```

1:  $S(s) = 0; U(i) = \infty, i \in V, i \neq s; SP = \emptyset$ 
2:  $SP \leftarrow SP \cup sp_{s,s}$ 
3: while  $U \neq \emptyset$  do
4:    $tsp = \emptyset; minhops = \infty; key = None$ 
5:   for edge  $e_{i,j}$  in  $E$  do // node  $i, j \in V, i \neq j$ 
6:     if  $hops(e_{i,j}) + S(j) \leq U(i)$  then
7:        $U(i) \leftarrow hops(e_{i,j}) + S(j)$ 
8:        $tsp(i) = j$ 
9:     end if
10:  end for
11:   $numu(k), inu(k) \leftarrow sort(U(i))$ 
12:   $\beta \leftarrow Num(numu(k))$ 
13:   $minhops = numu(\beta)$ 
14:   $U \leftarrow U - key$ 
15:   $S(key) = minhops$ 
16:   $S(key) = minhops$ 
17:  for shortest path  $sp_{0,s,i}$  in  $SP$  do
18:    if  $i == tsp(key)$  then
19:       $sp_{s,key} \leftarrow Merge(sp_{s,i}, key)$ 
20:       $SP \leftarrow SP \cup sp_{s,key}$ 
21:    end if
22:  end for
23: end while
24: return  $SP$ 

```

Based on Dijkstra Algorithm, we propose an improved equal-cost multi-path shortest paths algorithm (ECMP-Dijkstra) so as to calculate all equal-cost shortest paths from

the source node s to all nodes. Detailed pseudo code of ECMP-Dijkstra Algorithm is summarized in Algorithm 2. At beginning, we input the source node s and the shortest path set SP calculated by Dijkstra Algorithm, which contains only one shortest path from s to all nodes. In line 1, we use the shortest path set SP to calculate the minimum hop-count or cost set S from s to all nodes by the function $hops()$ and $S(i)$ denotes the minimum cost from node s to node i . In line 5, $rsp(i)$ denotes all equal-cost shortest paths from the source node s to the destination node i . We loop through the edge-set $E(e_{i,j})$ and judge whether the hop-count or cost from node s to node i (i.e., $S(i)$) plus the hop-count of $edge_{i,j}$ equals the hop-count from node s to node j (i.e., $S(j)$). If it does, then we add node j to all equal-cost shortest paths from node s to node i in lines 11–12, thus obtaining multiple shortest paths from node s to node j and adding them to the shortest path set SP in line 13. In lines 2–13, we exploit the precursor node and the initial shortest path set SP repeatedly, to add equal-cost shortest paths to SP and thus update SP constantly. In line 18, we delete the duplicate shortest path from SP using the function $DeleteDup()$. Thus, we update the shortest path set SP repeatedly until the number of shortest paths in SP does not increase.

Algorithm 2 ECMP-Dijkstra Algorithm

Input: $s; G(V,E); SP$
Output: SP

```

1:  $S \leftarrow hops(SP)$ 
2: repeat
3:    $nSP \leftarrow Num(SP)$ 
4:   for shortest path  $sp_{s,i}$  in  $SP$  do
5:      $rsp(i) = sp_{s,i}$ 
    //  $sp_{s,i}$  may contain more than one shortest path.
6:   end for
7:   for edge  $e_{i,j}$  in  $E$  do
    // node  $i, j \in V, i \neq j$ 
8:      $sp0_{i,j} \leftarrow Sp(e_{i,j})$ 
    // Convert  $e_{i,j}$  to shortest path  $sp0_{i,j}$ .
9:     if shortest path  $sp0_{i,j} \notin SP$  then
10:      if  $S(i) + hops(sp0_{i,j}) == S(j)$  then
11:        for shortest path  $sp'_{s,i}$  in  $rsp(i)$  do
12:           $sp'_{s,i} \leftarrow Merge(sp'_{s,i}, j)$ 
13:           $SP \leftarrow SP \cup sp'_{s,i}$ 
14:        end for
15:      end if
16:    end if
17:  end for
18:   $SP \leftarrow DeleteDup(SP)$ 
19:   $nSP' \leftarrow Num(SP)$ 
20: until  $nSP == nSP'$ 
  
```

We use three real-world topologies CRN, COST 239, NSFNet for simulation experiments, where China's 156 major railway nodes network (China Railway Network; CRN) [34] has 156 nodes and 226 links, Pan-European fiber-optic network (COST 239) [35] has 28 nodes and 41 links and T1 NSFNet network topology [36] has 14 nodes and 21 links. Under the three topologies, we compared ECMP-Dijkstra Algorithm with Dijkstra Algorithm and Li's Algorithm and the experimental results are shown in Figure 1 where TSP denotes the total number of shortest paths from one node to all nodes. Moreover, the higher the TSP, the better the intercept access points deployment may be. From the figures, we know that TSP of ECMP-Dijkstra Algorithm is higher than Dijkstra Algorithm and Li's Algorithm, thus, to deploy intercept access point reasonably.

Figure 1. The impact of three shortest path algorithm on TSP in three topologies. TSP. The higher, the better.

3.2. SDN Interception Models

For lawful interception in hybrid SDN, we first need to analyze how to intercept, that is, how to deploy intercept access point between the source (S), the destination (D) and the interception center (L). In this section, we will analyze various network interception models (i.e., the deployment strategies of IAP) in hybrid SDN. The deployment of intercept access point includes the single “IAP selection problem” in the shortest path S-D (i.e., the shortest path from S to D) and its derived “the shortest path algorithm problem between three points (i.e., S, D and L)”. The above two problems can be viewed as the same problem. Once the location of the intercept access point is determined, then the fourth point (IAP; I) can meet the service traffic between S, D and L. Under the condition that S-I, D-I, and L-I path are the shortest at the same time, the shortest path between three points can be solved to meet the needs of interception system.

We aim to solve the problem of selecting single intercept access point and routing between three-points, namely to deploy the best intercept access point in the shortest paths between S, D, and L. Analyzing interception models in hybrid SDN, we divide them into two interception models by the deployment location of intercept access point: legacy interception models and SDN interception models as shown in Figures 2 and 3.

Figure 2. Legacy interception models.

Figure 3. SDN interception models.

The legacy interception models include: S/I model, D/I model, and L/I model as shown in Figure 2a–c. As we all know, the interception service in legacy networks is limited by the deployment location of intercept access point due to the unimaginable cost of setting up special equipment and dedicated return link, to intercept network traffic. Thus, S or D or L is usually adopted as intercept access point I used to respond to the requirements of the interception center and to perform the traffic interception action in legacy network.

In this paper, we mainly study and analyze the SDN interception models, which includes T model, ECMP-T model and Fermat-point model as shown in Figure 3a–c. In view of the performance metrics of lawful interception system, the three SDN interception models are used to thoroughly study to find the optimal algorithm of deploying intercept access point.

Figure 3a shows T model: its name comes from the topology similar to the T-word. Under the concept of SDN networking in an undirected and weighted network $G(V, E)$, any SDN node on the shortest path S-D can be selected as intercept access point (I) under the premise of not affecting the existing shortest path arrangement of S-D (i.e., maintaining the existing end-to-end transmission quality). While only the node with the minimum hop-count (or cost) to the interception center (L) should be adopted as the best I-point to run the function to capture traffic transferred to the interception center.

Figure 3b shows ECMP-T model: based on the operation mode of T model, the path I-L must be the shortest path, but this shortest path S-I-D does not necessarily meet the optimal path. In fact, there may be more than one shortest path S-D, namely, the shortest path S-D is equal-cost multi-path (ECMP). Hence there may be a southward equal-cost shortest path in the T-word path theoretically, in which there is another intercept access point (I) and the hop-count (or cost) of I-L path is lower than the current one, so this interception model is called ECMP-T model that the nearest I-point from the interception center (L) is selected as the best intercept access point I among all the equivalent shortest paths between S and D.

Detailed pseudo code of T or ECMP-T model is presented in Algorithm 3. At the beginning of the algorithm, the set I used to store the best intercept access point is set to be empty in line 1. In lines 2–4, we calculate the shortest path $sp_{S,D}$ from S to D using Dijkstra or ECMP-Dijkstra Algorithm and then obtain the node-set $N_{S,D}$ in the shortest path $sp_{S,D}$, and next select out SDN nodes from the node-set $N_{S,D}$ to get the SDN node-set SN. If the SDN node-set SN is not empty, we traverse SDN nodes in SN and implement lines 6–16; otherwise, we fail to deploy intercept access point (IAP) between S, D and L, and thus save the wrong node combination of S, D and L in line 18. Line 7 calculates the lowest hop count (or cost) from SDN nodes to L and get the cost vector $h(i)$. In line 9, we sort the cost vector $h(i)$ by size of hop count in descending order and then get the sorted vector $numh(j)$ and the corresponding label vector $inh(j)$, where j denotes the subscript of j-th element of a vector. Line 11 takes the minimum hop-count value $minhops$ from the sorted vector $numh(j)$. Finally, in lines 13–14, we select the node with the minimum cost $minhops$ as the best intercept access point and then add the selected IAP $inh(j)$ to the set I.

Algorithm 3 T or ECMP-T Model

Input: N_{SDN} ; S; D; L

Output: I

```

1: I = ∅
2:  $sp_{S,D} \leftarrow \text{Dijkstra}(S,D)$  or  $\text{ECMP-Dijkstra}(S,D)$ 
3:  $N_{S,D} \leftarrow \text{Onodes}(sp_{S,D})$ 
4:  $SN \leftarrow \text{Select}(N_{S,D}, N_{SDN})$ 
5: if the SDN node-set  $SN \neq \emptyset$  then
6:   for node i in SN do
7:      $c(i) \leftarrow \text{hops}(\text{Dijkstra}(i,L))$  or
        $\text{hops}(\text{ECMP-Dijkstra}(i,L))$ 
8:   end for

```

```

9:  numh(j), inh(j) ← sort(h(i))
10:  β ← Num(SN)
11:  minhops ← numh(β)
12:  for key j in numh do
13:    if numh(j) = minhops then
14:      I ← I ∪ inh(j)
15:    end if
16:  end for
17: else
18:  SaveFail(S,D,L)
19: end if
20: return I

```

The only difference of pseudo code of T model and ECMP-T model is whether to use Dijkstra Algorithm or ECMP-Dijkstra Algorithm to calculate the shortest path.

Figure 3c shows Fermat-point model: In geometry, Fermat-point refers to the point with the smallest sum of the distances from the three vertices of the triangle. Accordingly, we extend it to the node with the smallest sum of the distances from the three nodes of S, D and L in SDN network, and at the same time with meeting the constraints of the shortest path of S-D, S-L and D-L between the three points. Theoretically, Fermat-point model is optimal.

Details of pseudo code of Fermat-point model are summarized in Algorithm 4. In lines 2–4, we calculate all equal-cost shortest paths of S-D, S-L, D-L using ECMP-Dijkstra Algorithm, and then obtain all node sets in the equal-cost shortest paths in lines 5–7, and next combine these node sets to get the node-set $N_{S,D,L}$ in line 8, and further select out SDN nodes from the node-set $N_{S,D,L}$ to get the SDN node-set SN . If the SDN node-set SN is not empty, we traverse SDN nodes in SN and implement lines 11–24; otherwise, we fail to deploy intercept access point (IAP) between S, D and L, thus to save the wrong node combination of S, D and L. Lines 12–14 calculate the lowest hop count (or cost) of i-S, i-D, i-L, and then add the results to the sum, to get the cost vector $h(i)$ in line 15. In lines 17–24, we sort the cost vector $h(i)$ by size of cost value in descending order and then take the minimum cost value $minhops$, and next select the node $inh(j)$ with the minimum cost $minhops$ as the best intercept access point and finally add the selected IAP $inh(j)$ to the set I .

Algorithm 4 Fermat-point Model

Input: N_{SDN} ; S; D; L
Output: I

```

1: I = ∅
2:  $sp_{S,D}, sp_{S,L}, sp_{D,L} \leftarrow \text{ECMP-Dijkstra}((S,D), (S,L), (D,L))$ 
3:  $N_{S,D}, N_{S,L}, N_{D,L} \leftarrow \text{Onodes}(sp_{S,D}, sp_{S,L}, sp_{D,L})$ 
4:  $N_{S,D,L} \leftarrow N_{S,D} \cup N_{S,L} \cup N_{D,L}$ 
5:  $SN \leftarrow \text{Select}(N_{S,D,L}, N_{SDN})$ 
6: if the SDN node-set  $SN \neq \emptyset$  then
7:   for node i in SN do
8:      $hs(i) \leftarrow \text{hops}(\text{ECMP-Dijkstra}(i,S))$ 
9:      $hd(i) \leftarrow \text{hops}(\text{ECMP-Dijkstra}(i,D))$ 
10:     $hl(i) \leftarrow \text{hops}(\text{ECMP-Dijkstra}(i,L))$ 
11:     $h(i) \leftarrow hs(i) + hd(i) + hl(i)$ 
12:   end for
13:  $numh(j), inh(j) \leftarrow \text{sort}(h(i))$ 
14:  β ← Num(SN)
15:  minhops ← numh(β)
16:  for key j in numh do
17:    if numh(j) = minhops then
18:      I ← I ∪ inh(j)

```

```

19:   end if
20: end for
21: else
22:   SaveFail(S,D,L)
23: end if
24: return I

```

We use sp_{i-j} to denote the shortest path from node i to node j , and $hops_{i-j}$ denotes the lowest hop-count or cost from node i to node j . We use ' $\rightarrow$ ' to denote that the next-node is N-SDN node and use ' $\Rightarrow$ ' to denote that the next-node is SDN node. Examples of three interception models are illustrated in Figure 4, where we select node 154, node 9, node 105 all marked by red as S, D and L respectively and select 30 nodes randomly in Figure 4 as SDN nodes which includes node $i \in \{4, 8, 11, 19, 23, 25, 31, 38, 49, 50, 58, 60, 65, 67, 77, 82, 89, 92, 100, 103, 117, 120, 121, 125, 128, 134, 140, 150, 152, 156\}$, to construct a hybrid SDN.

Figure 4. China's 156 major railway nodes network (China Railway Network; CRN).

We run T interception model: One shortest path from node 154 to node 9 is sp_{154-9} marked by pink in Figure 4 that is $154 \rightarrow 153 \Rightarrow 152 \rightarrow 146 \rightarrow 142 \rightarrow 136 \Rightarrow 134 \rightarrow 124 \Rightarrow 121 \Rightarrow 117 \rightarrow 94 \rightarrow 81 \Rightarrow 82 \rightarrow 74 \rightarrow 52 \Rightarrow 49 \rightarrow 32 \rightarrow 30 \Rightarrow 31 \Rightarrow 25 \rightarrow 9$, and $hops_{154-9} = 20$. Among all nodes in sp_{154-9} , node 117 that is an SDN node has the lowest hop count to node 105 due to $hops_{117-105} = 6$, and thus node 117 can be used as the best intercept access point I in T interception mode.

We run ECMP-T interception model: There are 22 equivalent shortest paths from node 154 to node 9, but we only show three shortest paths (i.e., sp_{154-9} contains $sp1_{154-9}$, $sp2_{154-9}$, and $sp3_{154-9}$) from node 154 to node 9 marked by pink, bright green, turquoise respectively in Figure 4. $sp1_{154-9}$ is $154 \rightarrow 153 \Rightarrow 152 \rightarrow 146 \rightarrow 142 \rightarrow 136 \Rightarrow 134 \rightarrow 124 \Rightarrow 121 \Rightarrow 117 \rightarrow 94 \rightarrow 81 \Rightarrow 82 \rightarrow 74 \rightarrow 52 \Rightarrow 49 \rightarrow 32 \rightarrow 30 \Rightarrow 31 \Rightarrow 25 \rightarrow 9$, and $sp2_{154-9}$ is $154 \rightarrow 153 \rightarrow 155 \rightarrow 144 \Rightarrow 140 \rightarrow 133 \Rightarrow 134 \rightarrow 124 \Rightarrow 121 \Rightarrow 117 \rightarrow 99 \rightarrow 97 \rightarrow 69 \rightarrow 68 \rightarrow 61 \Rightarrow 60 \rightarrow 56 \Rightarrow 23 \rightarrow 24 \Rightarrow 25 \rightarrow 9$, and $sp3_{154-9}$ is $154 \rightarrow 153 \rightarrow 155 \rightarrow 144 \Rightarrow 140 \rightarrow 133 \Rightarrow 134 \rightarrow 115 \rightarrow 113 \rightarrow 112 \Rightarrow 100 \rightarrow 101 \rightarrow 64 \rightarrow 63 \rightarrow 62 \rightarrow 59 \rightarrow 17 \rightarrow 16 \Rightarrow 19 \rightarrow 10 \rightarrow 9$, and $hops_{154-9} = 20$. Among all nodes in sp_{154-9} , node 100 that is SDN node in $sp3_{154-9}$ has the lowest hop count to node 105 due to $hops_{100-105} = 4$, and thus node 100 can be used as the best intercept access point (I) in ECMP-T interception mode. Apparently, $hops_{100-105} < hops_{117-105}$, namely, this I-point outperforms the one in the T model.

We run Fermat-point interception model: the node-set $N_{154-9-105}$ with no repeat is obtained by all sp_{154-9} , $sp_{154-105}$, sp_{9-105} (i.e., sp_{S-D} , sp_{S-L} , sp_{D-L}). Namely, $N_{154-9-105}$ contains all nodes of all the shortest paths from node 154 to node 9, from node 154 to node 105, and from node 9 to node 105. And then, the sum of hop-count from node 103 in $N_{154-9-105}$ to node 154, node 9, node 105 (i.e., $hops_{103-154,9,105}$) is the smallest and $hops_{103-154,9,105} = 23$. This means that node 103 that is SDN node in $N_{154-9-105}$ can be used as the best intercept access point I in Fermat-point interception mode.

We have solved the problem of single intercept access point deployment above, and then expand to deploy intercept access points in hybrid SDN.

Running different network interception models, we will study and analyze the influence on the best transmission quality of intercepted data (the minimum cost from intercept access point (I) to interception center (L); MILC), the total cost of running intercept operation in global network (TOC), and the quality of service of normal user's data stream (UQoS) with different proportion of SDN node. According to the proposed three models in Figure 3, MILC, TOC, UQoS are calculated in respectively in (1), (2) and (3), where N denotes the maximum node label or index, and any node can be selected as S, D and L in hybrid SDN topology, i.e., there are N^3 possibilities for node-combination of S, D and L. After the node-combination selection (S, D, L), the best intercept access point (I) can be got by the SDN interception models, then the hop count or cost of the shortest path S-I, D-I and L-I can be calculated by the function $hops(i,j)$, thus calculating MILC, TOC and UQoS.

4. Restricted Minimum Vertex Cover Algorithm

There is no exception that most network optimization deployment problems can be viewed as the minimum vertex cover problem (MVCP) in graph theory. In the process of migration of SDN technology for large-scale Internet, it may be faced with the situation of hybrid deployment of SDN nodes and non-SDN nodes (N-SDN). In this hybrid SDN, not all nodes have software-defined functions to play the role of intercept access point. Only some nodes with the function of software definition can respond to the requirements of the interception center and to run interception operation. Therefore, it is very critical to select the best deployment location of intercept access point. And IPA must occupy the position of the hub, greatly covering all traffic through the critical path, and under a certain proportion of threshold, it may not achieve 100% intercept link coverage. Therefore, the minimum vertex cover problem must be transformed into the restricted minimum vertex cover problem question (RMVCP).

$$MILC = \sum_{S=1}^N \sum_{D=1}^N \sum_{L=1}^N hops(L, I) \quad (1)$$

$$TOC = \sum_{S=1}^N \sum_{D=1}^N \sum_{L=1}^N hops(S, I) + hops(D, I) + hops(L, I) \quad (2)$$

$$UQoS = \sum_{S=1}^N \sum_{D=1}^N \sum_{L=1}^N hops(S, I) + hops(I, L) \quad (3)$$

Considering overall situation (e.g., the location relevance of all intercepted targets, the operation and maintenance cost of operators) from the whole interception system, we intend to develop a restricted minimum vertex cover algorithm (RMVCA) to achieve the best intercept link coverage of the whole network with the minimum number of intercept access points as well as optimize the efficiency of deployment when deploying intercept access points in the hybrid SDN.

RMVCP: given a network graph $G(V, E)$, where V denotes the set of all nodes, and E denotes the set of all links in the network. There exists non-SDN nodes and SDN nodes at the same time in the network where $V = S \cup N$, and S denotes the set of SDN nodes, and N denotes the set of non-SDN nodes. To find a P set ($P \subseteq S \subseteq V$), so that every link in the network is covered (intercepted) by at least an SDN node in the P set.

Figure 5 shows an example of solving RMVCP. In this hybrid SDN, SDN nodes (i.e., solid circle) set $S = \{1, 3, 8, 9, 11, 12, 13, 17, 19, 20, 21, 22, 25, 26, 27, 28\}$ and non-SDN nodes (i.e., light circle) set $N = \{2, 4, 5, 6, 7, 10, 14, 15, 16, 18, 23, 24\}$. Using RMVCA, the SDN nodes set $P = \{1, 8, 9, 11, 13, 20, 22, 25, 26, 27\}$ is recommended to be selected as the intercept access points set, but 7 links (marked as dotted lines) in the example failed to be covered due to the hybrid deployment of SDN and N-SDN nodes, and thus only about 80% of the links (marked as solid lines) are completely covered by 10 intercept access points.

Figure 5. Hybrid SDN covered by minimum vertexes.

RMVCA ensure the result a near-optimal solution or one of the approximate solutions, so as to meet the optimal solution of the deployment problem of intercept access points.

Based on the concept of degree, we, at a time, use greedy algorithm to select one approximate or equivalent optimal intercept access point to reduce the scale of the problem recursively, so as to obtain the minimum vertex approximation set covering all SDN links and achieve the best intercept link coverage with the minimum number of intercept access point.

Details of pseudo code of RMVCA are summarized in Algorithm 5. We input an undirected and weighted network cover set P is set to empty originally. In line 2–3, we get the set N of N-SDN nodes and the accordingly edge-set E_{N-SDN} of N-SDN nodes by the set N . Line 4 removes E_{N-SDN} from the edge-set E , to get the edge set E_{SDN} of SDN nodes. Lines 6–7 traverse each SDN node and calculate its degree $d(i)$. In line 9, we sort the degree vector $d(i)$ in the ascending order and get the sorted degree vector $numd(j)$ and the accordingly label vector $ind(j)$ where j denotes the index or subscript of j -th element. Line 11 selects the maximum degree $numd(\beta)$ from $numd(j)$ where β denotes the number of SDN nodes in the set S . In lines 12–18, we judge the degree of node and implement accordingly measures. If the maximum degree of node is not equal to zero in line 12, we first add the node $ind(\beta)$ where β denotes the subscript of β -th element to the minimum vertex cover set P in line 13 and then calculate the adjacent edge-set $\theta(ind(\beta))$ of node $ind(\beta)$ in line 14, and next remove the adjacent edge-set $\theta(ind(\beta))$ from the edge-set E in line 15, which leads to the degree reduction of each SDN node. Finally, we return line 5 to judge whether E_{SDN} is empty and then calculate the degree of each SDN node again. Otherwise, if the maximum degree of node is equal to zero, we break the loop and end the algorithm.

Algorithm 5 RMVCA (Restricted Minimum Vertex Cover Algorithm)**Input:** $G(V, E)$; S : the SDN node-set**Output:** P

```

1:  $P = \emptyset$ 
2:  $N \leftarrow V - S$ 
3:  $E_{N-SDN} \leftarrow \text{Edge}(N(i))$ 
4:  $E_{SDN} \leftarrow E - E_{N-SDN}$ 
5: while  $E_{SDN} \neq \emptyset$  do
6:   for node  $i$  in  $S$  do
7:      $d(i) \leftarrow i$ 
8:   end for
9:    $\text{numd}(j), \text{ind}(j) \leftarrow \text{sort}(d(i))$ 
10:   $\beta \leftarrow \text{Num}(S)$ 
11:   $\text{maxnumd} \leftarrow \text{numd}(\beta)$ 
12:  if  $\text{maxnumd} > 0$  then
13:     $P \leftarrow P \cup \text{ind}(\beta)$ 
14:     $\theta(\text{ind}(\beta)) \leftarrow \text{ind}(\beta)$ 
15:     $E_{SDN} \leftarrow E_{SDN} - \theta(\text{ind}(\beta))$ 
16:  else
17:    break
18:  end if
19: end while
20: return  $P$ 

```

Using RMVCA proposed above, we will study and analyze the influence of different SDN node proportion on the maximum intercept link coverage of the whole network (i.e., max-ILC) and the accordingly needed minimum number of SDN nodes in the P set for realizing the maximum intercept link coverage (i.e., numP), as well as the influence of RMVCA on the intercept link coverage (i.e., ILC) and the efficiency of deploying intercept access points in whole hybrid SDN.

5. Simulation and Results

5.1. Simulation Environment and Performance Metrics of Lawful Interception System

In our simulation, we choose three real-world backbone topologies CRN, COST 239, NSFNet to evaluate the performance of three SDN interception models. Under the three network topologies, we randomly select different number of nodes as SDN nodes to construct the hybrid SDN network and the weight of each link is set to 1 by default, and the source node (S), the destination node (D) and the interception center (L) are selected randomly and thus there are 3,796,416 (156^3), 21,952, 2,744 node combinations of S , D and L .

Under different proportion of SDN nodes, we will study and analyze the influence of different SDN interception models on the best transmission quality of intercepted data (the minimum cost from intercept access point (I) to the interception center (L); MILC), the total cost of running intercept operation in global network (TOC), and the quality of service of normal user's data stream (UQoS), the deployment efficiency of IAP (the total number of times to calculate the shortest path during the process of deploying IAP; TTC), and the total number of failures to deploy IEP (i.e., NFD).

According to the proposed three SDN interception models, MILC, MRLC, TOC, and UQoS are calculated respectively in (1), (2) and (3). Based RMVCA, we run different SDN interception models and calculate and count up MILC, TOC, UQoS, TTC and NFD of each node combination of S , D and L and then compare and analyze the results to evaluate the performance of three SDN interception models.

5.2. Benchmark Approach

In order to analyze the influence of RMVCA on the intercept link coverage of whole hybrid SDN and the efficiency of deploying intercept access points, we propose three approaches, proactive approach (PA), reactive approach (RA), hybrid approach (HA), and

then compare them by running three SDN interception models in real-world topology CRN. To show the effectiveness of HA, we compare it with the following baselines: PA and RA.

Experimental initialization: We randomly select some nodes as SDN nodes (i.e., given a hybrid SDN network topology), and then use RMVCA to calculate the minimum vertex cover set P required to achieve the maximum intercept link coverage in theory and the accordingly number N of SDN nodes in the P set. Additionally, the calculation amount of this initialization process is negligible compared with the one of the whole H-SDN.

Nodes selection: we traverse any node as S , D and L in topology CRN (i.e., there are 3,796,416 (156^3) possibilities for node-combination of S , D , L) and then the node combination of S , D and L is given for experiments.

Proactive approach (PA): when running SDN interception models to deploy intercept access point, we select the best intercept access point from the minimum vertex cover set P calculated by RMVCA. Details of pseudo code of PA in T or ECMP-T model are summarized in Algorithm A1 of Appendix A. The only difference of pseudo code of PA in T model and ECMP-T model is whether to use Dijkstra Algorithm or ECMP-Dijkstra Algorithm to calculate the shortest path.

Reactive approach (RA): according to the selected node combination of S , D and L , we run three interception models without exploiting RMVCA to deploy intercept access points.

Hybrid Approach (HA): running three SDN interception models to deploy intercept access point, we get the node-set $N_{S,D,L}$ where all nodes are selected from the shortest paths between S , D and L , and then obtain the node-set SP whose nodes also exist in the node-set P calculated by RMVCA. If the node-set SP is not empty, we preferentially select node from the SP set to deploy the best intercept access point; otherwise, we implement RA. Details of pseudo code of HA in T or ECMP-T model are summarized in Algorithm A2 of Appendix A.

When implementing PA or RA or HA, we count and calculate the frequency of the nodes selected as the best intercept access point, and then sort the nodes from largest to smallest based their frequency, and next select the first N nodes and calculate their intercept link coverage for studying and analyzing the impact of different approaches on the intercept link coverage (i.e., ILC) of the whole hybrid SDN. Additionally, we count the total times of calculating the shortest path (i.e., TTC) during the process of deploying intercept access points for studying and analyzing the impact of different approaches on the efficiency of deploying IAPs.

5.3. Results and Discussion

5.3.1. ILC

Using RMVCA, we study and analyze the influence of different numbers of N-SDN nodes on the maximum intercept link coverage (i.e., max-ILC) and the accordingly needed minimum number of SDN nodes in the P set for realizing the maximum intercept link coverage (i.e., numP). Moreover, we take the operator's operation and maintenance cost (i.e., the minimum number of SDN nodes) and network intercept link coverage into account comprehensively, so as to find the best proportion of SDN nodes from the experimental results.

Randomly selecting the number of N-SDN nodes (node $i \in (0,156)$) in CRN topology, we conducted 10,000 experiments in the same proportion of N-SDN nodes. Due to the different network topologies under the same SDN node proportion, the results of each experiment are different. The statistical results of 10,000 experiments are shown in Figures 6 and 7.

Figure 6. The influence of different numbers of N-SDN nodes on max-ILC.

Figure 7. The influence of the number of N-SDN nodes on numP.

Figure 6 shows the influence of different numbers of N-SDN nodes on max-ILC. From the figure, we can see that the number of SDN links in hybrid SDN decreases gradually with the increase of the number of N-SDN nodes (the decrease of the number of SDN nodes), resulting in the gradual decline of the network intercept link coverage. And max-ILC = 0.00% denotes that all links in the whole network are N-SDN links that cannot be intercepted, namely, all nodes in the network are N-SDN nodes. Additionally, we can see that the intercept link coverage of the whole hybrid SDN can reach 80.53~100% when the number of N-SDN nodes is between 0 and 57 (i.e., the number of SDN nodes is between 99 and 156), namely, only when the number of SDN nodes in hybrid SDN is more than 99 can SDN nodes intercept more than 80% of the links of the whole network.

Figure 7 shows the influence of different numbers of N-SDN nodes on numP. From the figure, we can see that when the number of N-SDN is 0 (i.e., the number of SDN nodes is 156), 79 SDN nodes are required to achieve the maximum intercept link coverage; the number of SDN nodes required to intercept the whole network gradually increases first and then decreases gradually. This is because that when the number of N-SDN nodes is between 0 and 37 (i.e., the number of SDN nodes is between 119 and 156), though the increase of N-SDN links results in the decrease of the degree of some SDN nodes, the total number of SDN links does not decrease significantly. Thus, more SDN nodes are needed to intercept the same number of links. Accordingly, the number of SDN nodes required to intercept the whole network increases. While when the number of N-SDN is between 38 and 156 (i.e., the number of SDN is between 0 and 118), the number of SDN links greatly decreases with the increasing number of N-SDN nodes, so the number of SDN nodes needed to achieve maximum intercept link coverage also decreases gradually. Moreover, when all nodes in the network are N-SDN nodes, all links are N-SDN links, and thus the minimum vertex cover set P is empty (i.e., numP = 0). To sum up, according to Figures 4 and 5, we only need 69~95 SDN nodes to achieve 80.53~100% intercept link coverage of the whole interception system when the number of SDN nodes in the whole hybrid SDN is between 99~156.

Next, we will study and analyze the influence of three different approaches and three SDN interception models on intercept link coverage (ILC) as shown in Figure 8. From the figure, we can see that ILC of PA and HA with RMVCA is higher than that of RA without RMVCA in general, and ILC of PA and HA are relatively close, whether using T model, ECMP-T model or Fermat-point interception model. Additionally, compared with RA, PA and HA can significantly improve the intercept link coverage when the number of N-SDN nodes is between 0 and 60 (i.e., the number of SDN nodes is between 96 and 156). And this improvement decreases with the decrease of SDN nodes.

Figure 8. The impact of three approaches in three SDN interception models on ILC under CRN topologies. ILC. The higher, the better.

Meanwhile, another conclusion we can make is that the three SDN interception models have nearly the same intercept link coverage. In other words, the intercept link coverage (ILC) has no relationship with SDN interception models and the SDN interception models have little impact on ILC.

5.3.2. TTC

Using RMVCA, we will analyze the impact of RMVCA on the efficiency of deploying intercept access points in whole hybrid SDN. In many experiments, we run three SDN interception models to deploy IAPs in three approaches during which the shortest paths need to be calculated, and thus the total times of calculating the shortest path (TTC) is different. In order to evaluate the performance of RMVCA, we employ TTC as its most important performance metric. We predict that RMVCA can improve the efficiency of deploying IAPs (i.e., reduce the total deployment time). The experimental results are shown in Figure 9.

Figure 9. The impact of three approaches in three SDN interception models on TTC under CRN topologies. TTC. The lower, the better.

From Figure 9, we can see that TTC of Fermat-point interception model is the highest whether in PA, RA or HA, namely, running Fermat-point model may take the longest time to deploy IAPs. In addition, TTC of T model and ECMP-T model is similar and is far lower than that of Fermat-point model. Therefore, in terms of the efficiency of deploying IAPs, T model and ECMP-T model are better than Fermat-point model.

Also, Figure 9 show the impact of three approaches in three SDN interception models on TTC under CRN topologies. From the figure, we can see that compared TTC in PA and RA, TTC in HA is the lowest, whether running T model, ECMP-T model or Fermat-point model in hybrid SDN. Namely, HA is the best approach in terms of the efficiency of deploying IAPs based on thorough analysis and comparison.

Meanwhile, we also can see that TTC in PA is the highest and thus PA is the most undesirable approach. Considering that TTC is the most important performance metric of RMVCA, we can abandon PA. According to Figure 9, we can conclude by calculating that with the increasing number of N-SDN nodes (i.e., with the decreasing number of SDN nodes) in hybrid SDN, HA can significantly improve the deployment efficiency of intercept access points for the reason that compared with RA, HA can decrease TTC on average by 41.14%, 44.07%, 53.32% respectively in T model, ECMP-T model and Fermat-point model. In conclusion, PA is the most undesirable approach that should be abandoned. While HA is the best approach in terms of the deployment efficiency of IAPs.

5.3.3. MILC, TOC and UQoS

After deploying the best intercept access point (IAP; I), the interception center (the law enforcement agencies; L) hopes to receive the data intercepted by the intercept access point with the minimum cost (i.e., the minimum cost or hop-count from the intercept access point (I) to the interception center (L); MILC). Therefore, MILC is one of the most important performance metrics of lawful interception system. In addition, the network operators are most concerned about the total cost of running intercept operation in global network (i.e., TOC) which is the prominent performance metrics of lawful interception system. Meanwhile, running different SDN interception models to deploy intercept access point may lead to the different selection of the best intercept access point (namely the placement location of IAP differs) and the different amount of calculation, thus affecting the quality of service of normal user's data stream (UQoS). Thus, UQoS is also one of the important performance metrics of lawful interception system. In a word, MILC, TOC and UQoS are of great significance for the Law Enforcement Agencies, the network operators and the users, respectively. Focused on three hybrid SDN topologies CRN, NSFNet and COST 239, we study and analyze the impact of running three different SDN interception models to deploy the best IAPs on MILC, TOC, and UQoS of whole lawful interception system under different number of SDN nodes. The experimental results of the three topologies are shown in Figure 10a–c.

Figure 10. The impact of three SDN interception models on MILC, TOC, UQoS under three topologies. Hop-count. The lower, the better.

From the figures, we can see that MILC, TOC in T model are relatively close to the ones in ECMP-T model. And MILC and TOC consumed by ECMP-T model are lower than that of T model, so ECMP-T model is better than T model. More importantly, compared with MILC and TOC in T model and ECMP-T model, MILC and TOC in Fermat-point model are the lowest in all number of SDN nodes. In other words, Fermat-point model can decrease MILC and TOC compared with T model and ECMP-T model. More specifically, compared with T model and ECMP-T model, Fermat-point model can decrease MILC on average by 13.41%, 11.11% in CRN topology, 14.91%, 8.73% in COST 239, and 19.72%, 16.04% in NSFNet, and TOC on average by 1.91%, 0.99% in CRN topology, 2.82%, 0.46% in COST 239, and 2.65%, 1.03% in NSFNet. These simulation results verify that the performance of Fermat-point model outperforms T model and ECMP-T model and thus Fermat-point model is the best SDN interception model in terms of MILC and TOC.

Meanwhile, from the figures, we can see that no matter in CRN, COST 239 or NSFNet, ECMP-T model and T model have the same UQoS. In other words, T model and ECMP-T model have little impact on the transmission quality of traffic normally accepted by users and on deployment efficiency of IAP. According to the principle of T model and ECMP-T model, we know the simulation results in three hybrid SDN topologies are consistent with the theory, so these results are true and reliable. In addition, we can also clearly observe from the figures that UQoS in Fermat-point model is higher than the one in T model and ECMP-T model, which means that Fermat-point model slightly affect UQoS. Thus, Fermat-point model has poor performance in terms of UQoS.

5.3.4. NFD

Due to the hybrid SDN topologies where N-SDN nodes cannot be selected as IAP, not every combination of S, D and L can successfully deploy intercept access point. We count the total number of failures to deploy IEP (i.e., NFD), to evaluate the performance of SDN interception models. The statistical results are shown in Figure 11a–c. We can clearly observe from the figures that in the three hybrid SDN topologies, the total number of failures to deploy IAPs (NFD) in Fermat-point model is the least compared with NFD in T model and ECMP-T model, which means that Fermat-point model has a high success rate to deploy intercept access point. More specifically, compared with T model and ECMP-T model, Fermat-point model decreases NFD on average by 88.21%, 86.87% in CRN topology, 76.9%, 74.68% in COST 239, and 67.53%, 66.26% in NSFNet. To sum up, the performance of Fermat-point model outperforms T model and ECMP-T model and thus Fermat-point model is the best interception model in terms of NFD.

Figure 11. The impact of SDN interception models on NFD under three topologies. NFD. The lower, the better.

6. Conclusions

In this paper, we proposed an improved equal-cost multi-path shortest path algorithm (ECMP-Dijkstra) and accordingly proposed three SDN interception models T model, ECMP-T model, and Fermat-point model, to deploy the best intercept access point reasonably in three real-world hybrid SDN topologies. Subsequently, we proposed a restrictive minimum

vertex coverage algorithm (RMVCA) to intercept the whole interception system with the least SDN nodes, and to optimize the deployment efficiency of intercept access points and improve the intercept link coverage, so as to optimize the performance of the whole intercepting system. According to RMVCA, we analyze the effect of different SDN node ratios on the intercept link coverage and the minimum vertex coverage set. Considering the intercept link coverage and the minimum vertex coverage set, we found a suitable SDN node ratio for deploying intercept access points reasonably, namely, to intercept the whole hybrid SDN with the least SDN nodes.

Based RMVCA, we put forward three approaches PA, RA, and HA for experiments, and compared the three experimental approaches. The experimental results show that HA is the best approach, which can significantly optimize the efficiency of deploying intercept access points (i.e., optimize TTC) and improve the intercept link coverage of the whole hybrid SDN.

By the way, we analyzed the influence of three SDN interception models on various performance metrics of lawful interception system using three real-world topologies. The simulation results reveal that the three SDN interception models have little effect on the intercept link coverage, and T model and ECMP-T model have no effect on user's traffic transmission quality. Compared with T model and ECMP-T model, Fermat-point model is the best interception model for the reason that Fermat-point model can make MILC, TOC, NFD the lowest by sacrificing a small part of user's traffic transmission quality (UQoS) and deployment time (TTC), intercepting the whole hybrid SDN at dramatically lower costs.

This paper has not considered the traffic bottleneck (link capacity) problem but has proposed the deployment and optimization strategy of intercept access points that pave the way for the future work that joint deployment of IAPs and LEAs in H-SDNs based on the consideration of the traffic bottleneck problem.

Author Contributions: X.X. and W.-K.J. conceived and designed the study. X.X. performed the simulations. X.X. wrote the paper. Y.W. and X.W. reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by the National Natural Science Foundation of China No. U1805262, No. 61871131, and No. 61701118 in part by the Natural Science Foundation of Fujian Province, China No.2018J05101, 2018H6007, and Special Fund for Marine Economic Development of Fujian Province (ZHHY-2020-3).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data are not publicly available due to their containing information that could compromise the privacy of research participants.

Acknowledgments: The authors would like to thank the anonymous reviewers for their useful comments and careful reading of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Based on Proactive Approach (PA) or Hybrid Approach (HA), the details of pseudo code of T, ECMP-T or Fermat-point model can be presented respectively in Algorithm A1 or Algorithm A2.

Algorithm A1 Proactive Approach—T or ECMP-T or Fermat-point Model

Input: $P; S; D; L$
Output: I
1: **if** $P \neq \emptyset$ **then**
2: $I = \emptyset$
3: **for** node i in P **do**
: ...
4: **return** I

Algorithm A2 Hybrid Approach—T or ECMP-T or Fermat-point Model

Input: $N_{SDN}; P; S; D; L$
Output: I
1: $I = \emptyset$
: ...
2: $SP \leftarrow \text{Select}(N_{S,D}, P)$ or $\text{Select}(N_{S,D,L}, P)$
3: **if** the SDN node-set $SP \neq \emptyset$ **then**
4: **for** node i in SP **do**
: ...
5: **else**
6: $\text{Algo3}(N_{SDN}, S, D, L)$ or $\text{Algo4}(N_{SDN}, S, D, L)$
7: **end if**
8: **return** I

References

1. Foundation, O.N. Software-Defined Networking: The New Norm for Networks. *ONF White Paper* **2012**, *2*, 11.
2. Kadhimi, A.J.; Seno, S.A.H. Maximizing the Utilization of Fog Computing in Internet of Vehicle Using SDN. *IEEE Commun. Lett.* **2019**, *23*, 140–143. [\[CrossRef\]](#)
3. Gökemli, B.; Tatlıcioğlu, S.; Tekalp, A.M.; Civanlar, S.; Lokman, E. Dynamic Control Plane for SDN at Scale. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2688–2701. [\[CrossRef\]](#)
4. McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Turner, J. OpenFlow: Enabling innovation in campus networks. *ACM Sigcomm. Comput. Commun. Rev.* **2008**, *38*, 69–74. [\[CrossRef\]](#)
5. Cheng, T.Y.; Jia, X. Compressive Traffic Monitoring in Hybrid SDN. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2731–2743. [\[CrossRef\]](#)
6. Dijkstra, E.W. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1959**, *1*, 269–271. [\[CrossRef\]](#)
7. Li, G. *An Improvement of Dijkstra Algorithm*; Computer Development & Applications: New York, NY, USA, 2009.
8. Wang, S.; Zhao, X. The improved Dijkstra's shortest path algorithm. In Proceedings of the 2011 Seventh International Conference on Natural Computation, Shanghai, China, 26–28 July 2011; pp. 2313–2316. [\[CrossRef\]](#)
9. Kadry, S.; Abdallah, A.; Joumaa, C. On The Optimization of Dijkstras Algorithm. In *Informatics in Control, Automation and Robotics*; Springer: Berlin/Heidelberg, Germany, 2012.
10. Zhang, X.; Chen, Y.; Li, T. Optimization of logistics route based on Dijkstra. In Proceedings of the 2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 23–25 September 2015; pp. 313–316. [\[CrossRef\]](#)
11. Bozyiğit, A.; Alankuş, G.; Nasiboğlu, E. Public transport route planning: Modified dijkstra's algorithm. In Proceedings of the 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey, 5–8 October 2017; pp. 502–505. [\[CrossRef\]](#)
12. Makariye, N. Towards shortest path computation using Dijkstra algorithm. In Proceedings of the 2017 International Conference on IoT and Application (ICIOT), Nagapattinam, India, 19–20 May 2017; pp. 1–3. [\[CrossRef\]](#)
13. Wei, K.; Gao, Y.; Zhang, W.; Lin, S. A Modified Dijkstra's Algorithm for Solving the Problem of Finding the Maximum Load Path. In Proceedings of the 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT), Kahului, HI, USA, 14–17 March 2019; pp. 10–13. [\[CrossRef\]](#)
14. Wenzheng, L.; Junjun, L.; Shunli, Y. An Improved Dijkstra's Algorithm for Shortest Path Planning on 2D Grid Maps. In Proceedings of the 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 12–14 July 2019; pp. 438–441. [\[CrossRef\]](#)
15. Levin, D.; Canini, M.; Schmid, S.; Schaffert, F.; Feldmann, A. Panopticon: Reaping the benefits of incremental SDN deployment in enterprise networks. In Proceedings of the 2014 {USENIX} Annual Technical Conference, Philadelphia, PA, USA, 19–20 June 2014; pp. 333–345.
16. Hong, D.K.; Ma, Y.; Banerjee, S.; Mao, Z.M. Incremental deployment of SDN in hybrid enterprise and ISP networks. In Proceedings of the Symposium on SDN Research, Santa Clara, CA, USA, 14–15 March 2016.

17. Poularakis, K.; Iosifidis, G.; Smaragdakis, G.; Tassioulas, L. One step at a time: Optimizing SDN upgrades in ISP networks. In Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017.
18. Zhou, H.; Zhang, D.; Xie, K.; Wang, X. Data reconstruction in internet traffic matrix. *China Commun.* **2014**, *11*, 1–12. [\[CrossRef\]](#)
19. Chen, J.; Kanj, I.A. Constrained minimum vertex cover in bipartite graphs: Complexity and parameterized algorithms. *J. Comput. Syst. Sci.* **2003**, *67*, 833–847. [\[CrossRef\]](#)
20. Behsaz, B.; Hatami, P.; Mahmoodian, E.S. On minimum vertex cover of generalized Petersen graphs. *Australas. J. Comb.* **2010**, *40*, 253–264.
21. Madhavi, L.; Maheswari, B. Edge Cover Domination in Mangoldt Graph. *Momona Ethiop. J. Sci.* **2011**, *3*, 37–51. [\[CrossRef\]](#)
22. Wu, Y.; Yu, Q. A Characterization of Graphs with Equal Domination Number and Vertex Cover Number. *Bull. Malays. Math. Soc. Ser.* **2012**, *3*, 803–806.
23. Volkmann, L. On graphs with equal domination and covering numbers. *Discret. Appl. Math.* **1994**, *51*, 211–217. [\[CrossRef\]](#)
24. Grandoni, F.; Könnemann, J.; Panconesi, A.; Sozio, M. A Primal-dual Bicriteria Distributed Algorithm for Capacitated Vertex Cover. *Siam J. Comput.* **2008**, *38*, 825–840. [\[CrossRef\]](#)
25. Zhang, N.; Sheng, Z. *Improved Greedy Algorithm for the Minimum Vertex Cover Problem*; Mongolia Normal University: Hohhot, China, 2012.
26. Tomar, D. An Improved Greedy Heuristic for Unweighted Minimum Vertex Cover. In Proceedings of the 2014 International Conference on Computational Intelligence and Communication Networks, Bhopal, India, 14–16 November 2014; pp. 618–622. [\[CrossRef\]](#)
27. Angel, D. A breadth first search approach for minimum vertex cover of grid graphs. In Proceedings of the 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 9–10 January 2015; pp. 1–4. [\[CrossRef\]](#)
28. Fayaz, M.; Arshad, S.; Zaman, U.; Ahmad, A. A Simple, Fast and Near Optimal Approximation Algorithm for Optimization of Un-Weighted Minimum Vertex Cover. In Proceedings of the International Conference on Frontiers of Information Technology IEEE, Islamabad, Pakistan, 19–21 December 2017.
29. Çınaroğlu, S.; Bodur, S. A new hybrid approach based on genetic algorithm for minimum vertex cover. In Proceedings of the 2018 Innovations in Intelligent Systems and Applications (INISTA), Thessaloniki, Greece, 3–5 July 2018; pp. 1–5. [\[CrossRef\]](#)
30. Zhou, Y.; Qiu, C.; Wang, Y.; Fan, M.; Yin, M. An Improved Memetic Algorithm for the Partial Vertex Cover Problem. *IEEE Access* **2019**, *7*, 17389–17402. [\[CrossRef\]](#)
31. Branch, P.A. Lawful Interception of the Internet. *Int. J. Emerg. Technol. Soc.* **2003**, *1*, 38–51.
32. Csikor, L.; Szalay, M.; Rétvári, G.; Pongrácz, G.; Pezaros, D.P.; Toka, L. Transition to SDN is HARMLESS: Hybrid Architecture for Migrating Legacy Ethernet Switches to SDN. *IEEE/ACM Trans. Netw.* **2020**, *28*, 275–288. [\[CrossRef\]](#)
33. Bshouty, N.H.; Burroughs, L. Messaging a linear programming solution to give a 2-approximation for a generalization of the vertex cover problem. In *Symposium on Theoretical Aspects of Computer Science*; Springer: Berlin/Heidelberg, Germany, 1998. [\[CrossRef\]](#)
34. Cao, W.; Feng, X.; Jia, J.; Zhang, H. Characterizing the Structure of the Railway Network in China: A Complex Weighted Network Approach. *J. Adv. Transp.* **2019**, *2019*, 3928260. [\[CrossRef\]](#)
35. De Maesschalck, S.; Colle, D.; Lievens, I.; Pickavet, M.; Demeester, P.; Mauz, C.; Jaeger, M.; Inkret, R.; Mikac, B.; Derkacz, J. Pan-European optical transport networks. *Photonic Netw. Commun.* **2003**, *5*, 203–225. [\[CrossRef\]](#)
36. Claffy, K.C.; Polyzos, G.C.; Braun, H. Traffic characteristics of the T1 NSFNET backbone. In Proceedings of the IEEE INFOCOM '93 The Conference on Computer Communications, San Francisco, CA, USA, 28 March–1 April 1993; Volume 2, pp. 885–892. [\[CrossRef\]](#)