LRVP: Lightweight Real-Time Verification of Intradomain Forwarding Paths

Wenlong Chen, Xiaolin Wang, Xiaoliang Wang, Ke Xu, and Sushu Guo

***Abstract*—The correctness of user traffic forwarding paths is an important goal of trusted transmission. Many network security issues are related to it, i.e., denial-of-service attacks, route hijacking, etc. The current path-aware network architecture can effectively overcome this issue through path verification. At present, the main problems of path verification are high communication and high computation overhead. To this aim, this article proposes a lightweight real-time verification mechanism of intradomain forwarding paths in autonomous systems to achieve a path verification architecture with no communication overhead and low computing overhead. The problem situation is that a packet finally reaches the destination, but its forwarding path is inconsistent with the expected path. The expected path refers to the packet forwarding path determined by the interior gateway protocols. If the actual forwarding path is different from the expected one, it is regarded as an incorrect forwarding path. This article focuses on the most typical intradomain routing environment. A few routers are set as the verification routers to block the traffic with incorrect forwarding paths and raise alerts. Experiments prove that this article effectively solves the problem of path verification and the problem of high communication and computing overhead.**

***Index Terms*—Autonomous systems, cyberspace, forward error correction, security, routing.**

I. Introduction

As INTERNET users or Internet service providers only care about whether packets or traffic is successfully transmitted to the destinations, its specific transmission path receives less attention. However, inconsistent forwarding paths make some malicious Internet behaviors difficult to trace, such as distributed denial of service [1] and Internet protocol (IP) source address spoofing. Moreover, the attacker may change a certain part of the traffic forwarding path to make it pass through the nodes or links that are not supposed to be passed and implement malicious behaviors in these places, such as illegal traffic monitoring and modification. Fortunately, after more than ten years of network study, the path perception network architecture today can effectively solve many Internet security problems. By encrypting the authentication information of packets, many mainstream routing verification protocols [2]–[5] are used to strengthen the network transmission path verification on the data plane, which can effectively solve most network security problems. The existing mainstream schemes are to add an additional protocol header between the TCP header and the IP header, which will increase the communication overhead . In addition, the existing schemes need to rely on cryptographic techniques such as key management and hashing to ensure data security, which will increase the computation overhead of the router. This article aims to implement a path verification architecture with no communication overhead and low computational overhead, and to improve the possibility of actual deployment. Although the size of the PPV header [4] is constant, it is not validated in real time, resulting in the bandwidth usage of flows that have already had errors. Although unicast reverse path forwarding (uRPF) [7] performs real-time detection of forwarding paths, it requires an additional forward information database (FIB) lookup with the source IP of the packet, which increases the processing load of the forwarding engine. Moreover, uRPF does not support network environments with asymmetric paths.

This article proposes a lightweight real-time verification mechanism of intradomain forwarding paths (LRVPs) in autonomous systems (ASes). The LRVP focuses on the most typical intradomain routing environment: Link-state routing protocols (LSRP). Internal gateway protocol (IGP) is usually used for decision routing within an AS. Currently the two standardized protocols are routing information protocol (RIP) and open shortest path first (OSPF). OSPF is a link-state routing protocols. It uses the Dijkstra algorithm to calculate the shortest path tree and make routing decisions. The LRVP is based on the OSPF protocol. The OSPF protocol running in an AS is not changed, and the OSPF complexity is not increased. Each router in an AS is assigned a different ID. IP packets entering the AS carry the IDs of the corresponding router in the IP header to identify the source router node of the packet. In the AS, the forwarding path of each pair of source and destination nodes is assumed to be deterministic. The LRVP selects some routers in the network to act as verification routers (VRs), blocking the traffic with incorrect forwarding paths and raising alerts. The existing routing protocol does not require any modification for the LRVP. The verification rules are integrated into the existing FIB table. Only one FIB lookup (longest prefix matching) is required to obtain the outgoing interface as well

Manuscript received 30 June 2021; revised 24 August 2021, 7 October 2021, 22 December 2021, and 16 February 2022; accepted 28 March 2022. Date of publication 19 May 2022; date of current version 9 December 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61872252, in part by the National Key Research and Development Program of China under Grant 2018YFB1800403, and in part by Beijing Natural Science Foundation under Grant 4202012. *(Corresponding author: Xiaoliang Wang.)*

Wenlong Chen, Xiaolin Wang, Xiaoliang Wang, and Sushu Guo are with the Information Engineering College, Capital Normal University, Beijing 100048, China (e-mail: chenwenlong@cnu.edu.cn; 2191002036@cnu.edu.cn; wangxiaoliang@cnu.edu.cn; 2191002042@cnu.edu.com).

Ke Xu is with the Department of Computer Science and Technology, Tsinghua University, Beijing 100190, China (e-mail: xuke@mail.tsinghua.edu.cn).

Digital Object Identifier 10.1109/JSYST.2022.3165826

TABLE I
COMPARISON OF SOME RELEVANT PARAMETERS

	ICING	OPT	PPV	Atomos[6]	LRVP
Real-time	✓	✓	✗	✓	✓
New protocol header	✓	✓	✓	✓	✗
New verification table[1]	✗	✗	✗	✗	✓
Negotiation[2]	✓	✓	✓	✓	✗
Verification device	all	all	Destination	all	VRs
Source initialization complexity	O(n)	O(n)	O(1)	O(1)	-
Packet verification complexity	O(n)	O(1)	O(1)	O(1)	O(1)
Packet update complexity	O(n)	O(1)	O(1)	O(1)	-

1.The verification table (VIT) is only stored in VRs.
2.Negotiate information such as encryption keys before communication

TABLE II
SUMMARY OF NOTATION

Notation	Meaning
ID	It uniquely identifies the router
FR	The faulty router
CFP	The correct forwarding path
BP	The wrong forwarding branch path
VR	The verification router
S_N(r)	The set of neighbors of r
S_VR	The set of VR
S_BP	The set of BP
S_BR	The set of candidate VRs for BP
S_ID	The set of router's ID in VIT
I	The number of VIT-index of VR
L	The number of interface that the VR connects to other routers
e	The number of forwarded items of VIT

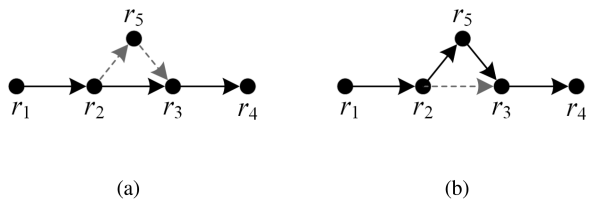


Fig. 1. Two typical cases of wrong forwarding. Solid lines indicate expected forwarding. Dotted lines indicate incorrect forwarding.

as the next-hop of the packet and check the forwarding path at the same time. A minimization deployment scheme of VRs is designed accordingly to reduce the deployment cost and increase the possibility of the actual deployment. The deployment scope of the forwarding path verification is definable, which can be deployed on one, several or all the paths on the forwarding path in the network. Moreover, a simple VR deployment scheme based on the minimum loopback topology (LT) is proposed. Table II lists the main notations and their meanings in this article.

There are two types of routers in LRVP, namely VR and non-VR. To improve the efficiency, the LRVP only needs to deploy a few key crossover nodes instead of all nodes as VRs. VR also participates in normal forwarding. The difference between VR and non-VR is that VR adds a verification module, which verifies packets before forwarding. In addition, the LRVP is deployed in an AS that uses OSPF and has no special topological requirement.

The LRVP is different from the existing schemes that require information such as encryption keys to be negotiated before communication begins. For some schemes, packets are marked by intermediate routers and verified by the destination, which leads to delayed blocking and wasted bandwidth (bandwidth occupied by the transmission of incorrect traffic). In the LRVP, VRs directly verify the packets so that the wrong traffic is blocked in real time and the bandwidth waste is avoided. In addition, the verification process is different. While other schemes are where the router verifies the proof carried in the packet header, our scheme verifies the packet by querying the verification table (VIT) stored in the router. For some of the differences, please refer to Table I. Source initialization complexity refers to the computation complexity of hashing (or encryption) when initializing packets at the source. Packet update complexity refers to the computation complexity of intermediate router updating, i.e., hashing or encrypting the information in the header. The LRVP does not use cryptography algorithms or have additional protocol headers, so they are not considered in this article.

The advantages of the LRVP are as follows.

1) The LRVP is suitable for networks with asymmetric round-trip paths.
2) The implementation cost of the LRVP is small. The existing routing protocol does not require modification. We have designed a minimization deployment scheme of VRs with the fusion storage of VITs and an existing forwarding table, which reduces the deployment cost and increases the possibility of the actual deployment.
3) The deployment scope of path verification is definable.

Regarding to achieving the desired effect (no communication overhead and low computing overhead), this article has the following limitations.

1) The LRVP is only be applied to the AS running OSPF, and the number of edge routers in the AS is less than 256 (the reasons will be explained in Section III).
2) The LRVP increases the storage overhead of the router. In the extreme condition (full connection), VIT occupies about 8 kB of storage space, which is acceptable.
3) The deployment and operation of VR inevitably incur some additional overhead. Please refer to Section V for detailed analysis.

In the experimental section, the LRVP is compared with PPV [4] in terms of throughput and packet processing delay, etc. The result shows that the LRVP increases the throughput by 10%, and packet processing delay is reduced by half. Moreover, it only requires a small VIT storage cost without bringing any additional transmission load.

The remainder of this article is organized as follows: Section II analyzes the problem of path validation. Section III introduces the design of the LRVP in detail. Section IV introduces the deployment scheme of VR. Section V gives the experimental performance and evaluation. Section VI covers the related previous work in the field of source authentication and path validation. Finally, Section VII concludes this article.

II. Problem

A. Problem Scenario

The forwarding path verification is implemented for packet transmission in the AS. We only study the scenario where a packet is finally transmitted to the destination. Packets are usually discarded when there is no matching forwarding item, or the routing loop causes the "Time To Live" (TTL) to be 0. So, the scenario that the packet cannot reach the destination will be directly perceived by the terminal users. To sum up, after the LRVP is enabled, two situations may occur when packets are incorrectly forwarded: one is that the loop is formed (the loop causes the TTL to be exhausted and the packet is discarded), and the other is that packets are discovered and blocked by VRs.

The path verification process needs to know which edge router a packet comes from. Although the source IP in the packet is the basis for analysis, the source IP or its corresponding IP prefix has a large number, and the cost of information storage and analysis is high. Considering the aforementioned issue, the idea that an IP header carries the ID of the source router is proposed, which simplifies the matching information and the processing logic. As shown in Fig. 1, suppose that R_1 has ten network segments. If their IP prefixes are used as the verification information, ten verification items need to be set for the packets with R_1 as the source router. In the LRVP, we uniformly identify the packets of the ten subnets with the ID of R_1 so that only one verification item needs to be set.

In most of the existing path verification schemes, IP packet size changes with its forwarding path length. To avoid fragmentation or transmission failure caused by IP packets whose lengths exceed the maximum transmission unit (MTU), a packet needs to reserve enough space for the ever-lengthening IP header when it is sent, which will lead to additional bandwidth overhead. For example, data can be transmitted by only one packet. But due to the length of the header, two packets have to be used to transmit the data, which increases the communication overhead (one more packet means one more extension IP header). It is worth mentioning that PPV [4] solves this problem well, and its header is not affected by path length. However, the existing schemes still have additional communication overhead (i.e., the extension headers). Therefore, this article attempts to design a scheme that requires as few packet extension headers as possible.

B. Types of the Incorrect Forwarding Path

In an AS, the path of packet transmission is usually specific (one or more paths). PPV [4] classifies wrong path types in great detail and lists five of them. In this article, both source and destination are routers, so the wrong path types can be summarized into the following two types. As shown in Fig. 1, for the packet p sent by r_1 to r_4, there are two possible error conditions in the forwarding process.

1) New Nodes Are Added to the Path: As shown in Fig. 1(a), r_1 sends a packet p to r_4, and the path is $\{r_1, r_2, r_3, r_4\}$. If r_2 forwards p to r_5 by mistake and r_2 is the next-hop of the path from r_5 to r_4, a forwarding loop is formed between r_2 and r_5. The scenario where p does not reach the destination is not considered here. If the next-hop of the path from r_5 to r_4 is r_3, p is successfully sent to the destination. But the path of p is wrong, which is a scenario that should be focused on.

2) Some Nodes Are Missing From the Path: p passes through neither certain routers that it should have passed through, nor some routers in the correct order. As shown in Fig. 1(b), r_2 directly forwards p to r_3, ignoring r_5. Although p is successfully sent to the destination, the path is wrong.

If r_5 in Fig. 1(a) or r_3 in Fig. 1(b) uses the LRVP to know whether p is a packet that is not to be forwarded. Take a flow(s, d) as an example. There is an expected correct forwarding path from s to d on which VR is not required. While there may be other paths (unexpected paths) from s to d, VRs are deployed on these unexpected paths. Ensure that if the packet of flow(s, d) deviates from the expected path, it will be captured by a VR. If the verification object are flows of all the ODs (origin-destinations), then each BP (Branch Path) of each OD requires a VR. If the object are flows of an OD, one VR is required for each BP of that OD. BP is defined in Section III.A. It seems to be a complicated problem, but a solution is designed that is easy to implement, which is introduced in Section III.

The reasons for the path error are as follows.

a) *Source forgery:* Assume that the forwarding paths of packets p_1 and p_2 are p_1:$\{s_1, r_1, r_2, ..., d\}$ and p_2:$\{s_2, r_1, r_2, ..., d\}$, respectively. If p_2 is forged p_1, then the path of p_2 becomes $p1(p2)$:$\{s_2, r_1, r_2, ..., d\}$.

b) *Intermediate router failure or malicious behavior:* Assume that $p_1 : \{r_1, ..., r_i, r_{i+1}, ..., r_n\}$, r_i has malicious forwarding or failure, resulting in p_1:$\{r_1, ..., r_i, r'_{i+1}, ..., r_n\}$. This type of situation may be caused by security issues such as route hijacking, router failure, etc.

C. Disadvantages of Identifying Sources With IP

The source IP address in the IP header corresponds to the source router. Moreover, using the source IP to represent the packet source is a common packet processing scheme. However, a VR performs forwarding verification based on the source IP, which has the following disadvantages.

1) The terminal systems of the edge network are messy and complex, and there are fake source IP addresses. So, the source IP fails to provide a credible basis for path verification.
2) If we pay attention to the source IP, the verification rules include two dimensions of information (source and destination IPs), resulting in a larger storage of verification rules in VRs. It should be noted that the number of IP prefixes in the external networks of an AS is huge.

3) VRs need to obtain the correspondence between the source IP and the source router, increasing the processing complexity.

III. Detailed Design

A. Wrong Forwarding Branch Path

Definition 1: The correct forwarding path (CFP). For example, CFP(s, d) is a legal path from s to d, which is an ordered node set:$\{a_1, a_2, ..., a_n\}$. Among them, $a_1 = s$, $a_n = d$.

Definition 2: The wrong forwarding branch path (BP). For a CFP(s, d):$\{a_1, a_2, ..., a_n\}$, BP(s, d) is a path whose start node and end node are both in CFP(s, d), which is expressed as an ordered set of nodes $\{b_1, b_2, ..., b_m\}$ that satisfy (1). Loops do not constitute BPs. And it should be emphasized that BP(s, d) is not a path from s to d.

$$\begin{cases} b_1 = a_i, b_m = a_j, i < j \\ \{b_1, ..., b_m\} \neq \{a_i, ..., a_j\} \\ \forall b_x \in \{b_2, ..., b_{m-1}\}, b_x \notin \{a_i, ..., a_j\} \end{cases} \tag{1}$$

Node b_1 in the aforementioned description is the origin of the BP, and it is defined as a faulty router (FR). One CFP(s, d) may have many BPs. Each BP has a corresponding FR, and some BPs may share the same FR.

As shown in Fig. 1(a), the correct path from r_1 to r_4 is expressed as CFP(r_1, r_4):$\{r_1, r_2, r_3, r_4\}$. There is a wrong path $\{r_1, r_2, r_5, r_3, r_4\}$. For CFP($r_1$, r_4), there is a BP(r_1, r_4): $\{r_2, r_5, r_3\}$ and r_2 is the FR. Fig. 1(b) shows another scenario, i.e., the correct path from r_1 to r_4 is expressed as CFP(r_1, r_4):$\{r_1, r_2, r_5, r_3, r_4\}$. There is a wrong path $\{r_1, r_2, r_3, r_4\}$. For CFP(r_1, r_4), there is a BP(r_1, r_4): $\{r_2, r_3\}$ and r_2 is the FR.

Given a CFP(s, d), if there are two BPs: BP$_i(s, d)$: $\{b_1^i, b_2^i, ..., b_k^i\}$ and BP$_j(s, d)$:$\{b_1^j, b_2^j, ..., b_l^j\}$, they are regarded as different BPs as long as the two paths are not exactly the same. In fact, we can judge this based on the first two hops of the path.

Deduction 1: If $b_1^i = b_1^j$ and $b_2^i = b_2^j$, then BP$_i(s, d)$= BP$_j(s, d)$.

Proof: If there are two BPs for a CFP(s, d), they are expressed as follows:

BP $_i(s, d)$:$\{b_1^i$, CFP(b_2^i, d)$\}$,CFP(b_2^i, d) is unique.

BP $_j(s, d)$:$\{b_1^j$, CFP(b_2^j, d)$\}$,CFP(b_2^j, d) is unique.

If $b_2^i = b_2^j$, then CFP (b_2^i, d) = CFP(b_2^j, d), and if $b_1^i = b_1^j$, then BP $_i(s, d)$ = BP$_j(s, d)$.

This ends our proof.

One CFP(s, d) may have many BPs. Let the set of BPs of CFP(s, d) be S_BP(s, d). The algorithm for calculating S_BP(s, d) is designed as shown in Algorithm 1. φ is an ordered set of nodes of BPs.

B. Verification Router (VR)

Definition 3: The VR: A VR is appointed by the management server according to the requirements, for the aim of verifying the path of the packets forwarded by itself. It blocks the packets with a wrong path and sends them to the management server. Obviously, the blocked packets should not pass through the VR.

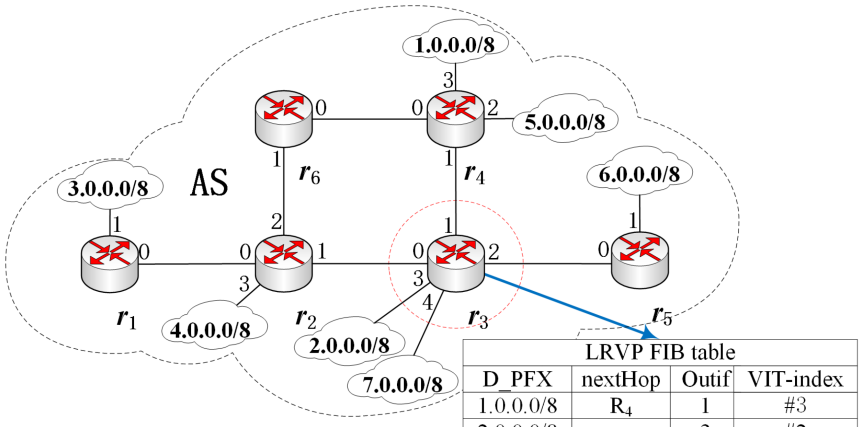
1) *Which node is selected as VR?* In LRVP, not all the nodes need to be deployed as VRs. Every BP$_k(s$, $d)$ of CFP(s, d) needs one VR. For BP$_k(s, d)$:$\{b_1, b_2, ..., b_m\}$, b_1 is the origin of BP$_k(s$, $d)$ and a VR is selected in the set: $\{b_2, b_3, ..., b_m\}$ to verify flow (s, d). S_BR$_k(s$, $d)$:$\{b_2, ..., b_m\}$ is the set of candidate VRs for BP$_k(s$, $d)$. The number of VRs are as few as possible. Suppose there is a CFP(s, d) with n BPs, i.e., BP$_1(s, d)$, BP$_2(s, d)$,..., BP$_n(s, d)$. For BP$_k(s, d)$, $1 \leq k \leq n$, a VR needs to be deployed. One VR may exist in the S_BR of many BPs, so the number of VRs may be less than n, and there is the minimization goal of (2). For the selection method of the VR, please refer to Section IV.

$$\begin{aligned} &\text{o.f. Minmum}(|S\_VR(s,d)|) \\ &\text{s.t.}\forall \\ &(BP_k(s,d), \exists r_i, r_i \in S\_VR(s,d), r_i \in S\_BR_k(s,d). \end{aligned} \tag{2}$$

2) *How is VR designed?* The traditional forwarding engine forwards packets based on the destination IP only. While in the LRVP, the router analyzes the incoming interface of the packet and the corresponding source router to determine whether an error occurred on the previous forwarding path. It integrates the path verification processing and the existing FIB lookups by using IDs.
   a) *ID:* Each router in an AS is assigned an ID, which is stored in the type-of-service (ToS) field of the IP header. Therefore, the LRVP supports up to 256 routers in an AS, which is sufficient in a real network. The packets entering the AS through a router are marked with the router's ID so that a VR can know the router corresponding to the received IP packet. It should be emphasized that a packet does not have to be marked

Algorithm 1: Calculation of S_BP(s, d)

```
Require s, d;
Ensure S_BP(s, d);
calculate the CFP(s, d);
φ = {};
for each node in CFP(s, d) do
  CFP(s, d)=CFP(s, d) − node;
  for each node′ ∈ node.neighbor & node′ ∉
  CFP(s, d) do
    φ.first = node;
    φ.add(node′);
    while node′.nexthop ∉ φ do
      if node′.nexthop ∈ CFP(s, d) then
        φ.add(node′.nexthop);
        S_BP(s, d).add(φ);
        φ.clear;
        BREAK;
      else
        φ.add(node′.nexthop);
        node′ = node′.nexthop
      end if
    end while
    φ.clear;
  end for
end for
```



ER and ID correspondence table

Router	R_1	R_2	R_3	R_4	R_5	R_6
ID	1	2	3	4	5	6

Verification Information Table(VIT)

	Inif	S_ID
#1	0	00000000
	1	00000000
	2	00001000
	default	00100000
#2	0	11000000
	1	00010000
	2	00001000
	default	00100000
#3	0	00000000
	1	00000000
	2	00001000
	default	00100000
#4	0	11000000
	1	00010000
	2	00000000
	default	00100000

LRVP FIB table

D_PFX	nextHop	Outif	VIT-index
1.0.0.0/8	R_4	1	#3
2.0.0.0/8	-	3	#2
3.0.0.0/8	R_2	0	#1
4.0.0.0/8	R_2	0	#1
5.0.0.0/8	R_4	1	#3
6.0.0.0/8	R_5	2	#4
7.0.0.0/8	-	4	#2

Fig. 2. Example of the AS with LRVP. The CFP(r_1, r_4) is r_1, r_2, r_6, r_4, the $BP_1(r_1, r_4)$ is r_2, r_3, r_4, and r_3 is the VR for BP_1.

as the ID of the source router, and any verifiable ID can be used. For example, the packet from 3.0.0.0/8 in Fig. 2 can be marked as either 1 or 2 without being misjudged. In addition, for an unmarked packet, the router R determines whether to mark (R is the source router) or drop (R is not the source router) according to the interface on which the packet enters R.

b) *FIB:* The main fields of the existing FIB entries of the router includes:"*D_PFX*, *nextHop*, *Outif*." The LRVP extends this structure:"*D_PFX*, *nextHop*, *Outif*, *VIT-index*." In addition, a VIT is added in the VR, which includes several groups of VIT entries. If the VR has x interfaces, then each group has $x + 1$ entries, describing the valid source ID for each interface. The VIT in Fig. 2 includes four groups of entries. The "VIT-index" field of FIB is used to point to the group of the VIT.

Given CFP(s, d), v is deployed as a VR for a BP(s, d). According to the link state information of the network, v knows all the legal source routers passing through v to d, and the set of IDs of these routers is defined as S_ID. Obviously, $s \in$S_ID. Therefore, when v receives the packet destined for d, it obtains its ID from the IP header. If S_ID contains the ID and the incoming interface of the packet is correct, the packet is successfully forwarded. Otherwise, it means that the previous forwarding of the packet is wrong. The bit-compressed structure is used to describe the legal S_ID, ensuring that it takes up as little storage space as possible. Each bit in S_ID represents a router. The bit width of S_ID is determined according to the number of routers in an actual AS network. For example, an AS with no more than 16 routers requires 2 bytes to store S_ID.

For the destination d of CFP(s, d), the LRVP inherits the destination IP prefix in the existing forwarding entry and maps it to the destination router. One router may correspond to many IP prefixes. Therefore, a VR is deployed for many destination IP prefixes for a $BP_i(s, d)$. As shown in Fig. 2, for CFP(r_1, r_4):$\{r_1, r_2, r_6, r_4\}$, there is a $BP_1(r_1, r_4)$:$\{r_2, r_3, r_4\}$. r_3 is selected as the VR for BP_1, so r_3 configures an S_ID for all the IP prefixes belonging to r_4, including 1.0.0.0/8 and 5.0.0.0/8.

3) *How does a VR work?* A VIT entry includes the incoming interface of a packet and the legal source nodes: Inif and S_ID. Only when these two fields are matched successfully, the forwarding path of the packet is judged as correct. As shown in Fig. 2, when r_3 receives a packet with "*incoming interface* $= 0$, *destination IP* $=$ 6.0.0.1, *ID* $=$ 4," r_3 lookups the FIB table based on the destination IP 6.0.0.1 and learns that VIT-index $= 4$. Then, according to the incoming interface, the S_ID directly found in VIT is "11000000." However, "ID $= 4$" in the packet is represented as "00010000," which fails to match. So, it is determined that the packet forwarding path is wrong and r_3 drops the packet and sends an alert message to the management server. When r_3 receives a packet with "*incoming interface* $= 0$, *destination IP* $=$ 6.0.0.1, *ID* $= 2$" and "*ID* $= 2$" in the packet is represented as "01000000," it passes the verification. VIT is stored as an array. S_ID is accessed directly based on the pointer and subscript (incoming interface), so the lookup table complexity is O(1).

C. Analysis of the VIT size

In the LRVP, a VIT has a small number of entries. If an AS contains n routers, there are at most n groups of entries. The number of entries in each group depends on the number of interfaces of a VR. A router connects many user IP prefixes, which correspond to the same VIT index in the FIB of the VR. In Fig. 2, both "1.0.0.0/8" and "5.0.0.0/8" belong to r_4, and their VIT indexes in the FIB of r_3 are all #3.

Let e denote the number of VIT items, let I denote the number of VIT indexes, and let L denote the number of interfaces that a VR connects to the other routers. Each VIT-index points to L+1 forwarding items and the VIT of a VR contains at most n VIT indexes. The total number of VIT items is shown as follows:

$$e = I(L + 1). \tag{3}$$

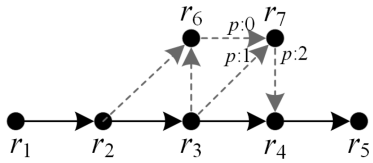


Fig. 3. Example of fault location.

A router with only one interface connected with other routers does not need to be deployed as a VR, e.g., for r_1 in Fig. 3, the packet with r_1 as the destination router can be checked by r_2. According to (3), the size of e depends on the parameters I and L. Assuming $I = n$ (I is the maximum), in the best case, a VR contains two interfaces connected to the other routers, and $L = 2$. While in the worst case, when a VR is connected to all the routers, $L = n$. The value range of e is shown as

$$3n \leq e \leq n(n+1). \tag{4}$$

According to (4), the storage space complexity of a VIT is O(n) in the best case and O(n^2) in the worst case. Since the actual situation depends on the network topology, i.e., in most real networks, routers connecting all or most of the others are not common, for most VRs, L in (3) is small.

If the ToS field of the IP header is used to store ID, in an AS, the maximum storage space occupied by a VIT entry for each lookup is 256 × (256 + 1) ≈ 8K.

Moreover, for a VR, if the incoming interface and source routers to several destination routers are all the same, they also share a VIT index. As shown in Fig. 2, for the FIB of r_3, the VIT index of the destination IP prefixes to r_1 and r_2 ("3.0.0.0/8," "4.0.0.0/8") are both #1. Therefore, in the actual network, I in (3) is smaller than the theoretical maximum value n. In summary, the actual cost of a VIT is far less than the theoretical maximum. A specific experimental analysis on the value of e is given in Section V.

In addition, since a VIT uses bit compression to store the source router ID, it is better adapted to multipath transmission such as equal-cost nultipath (ECMP) [8], which does not have additional storage overhead.

D. Fault Location

Under the premise of minimizing the number of VRs, it is difficult to accurately locate the router where the forwarding errors occur, because one VR may check many CFPs. However, we can still locate them in a range as small as possible. The algorithm for locating FRs is shown in Algorithm 2, where v is the VR, inif is the incoming interface for packet into v, and v.inif is the incoming interface of v in the BP.

The fault location process is as follows.

1) A VR gets the incoming interface of a packet, the source router ID from the IP header, and the destination router ID according to the destination IP of the packet, which are "iif, s, d," respectively.
2) The VR determines CFP(s, d) according to s and d before calculating S_BP(s, d) of CFP(s, d).

Algorithm 2: Calculation of S_FR;

Require s, d, v, inif;
Ensure S_FR;
1: S_FR={};
2: S_BP(s, d) ← Algorithm1(s, d);
3: **for** each BP $_i(s, d) \in$ S_BP(s, d) **do**
4: **if** $v \in$BP$_i(s, d)$ **then**
5: **if** (v.inif == inif) in BP$_i(s, d)$ **then**
6: S_FR .add(BP$_i(s, d)$.firstNode);
7: **end if**
8: **end if**
9: **end for**

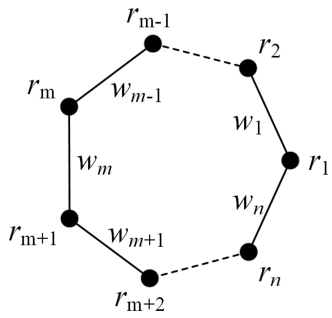


Fig. 4. Example of MLT.

3) The VR finds all the BP$_i(s, d)$ that contains the VR, and the iif' of the VR in BP$_i(s, d)$ is equal to the iif. The first node b_1 of BP$_i(s, d)$ is the possible faulty router.

As shown in Fig. 3, suppose that r_7 is a VR and CFP (r_1, r_5) = $\{r_1, r_2, r_3, r_4, r_5\}$. According to the aforementioned process, if r_7 receives a packet ($iif = 0$; $s = r_1$; $d = r_5$), it calculates all the BP (r_1, r_5): $\{r_2, r_6, r_7, r_4\}$ and $\{r_3, r_6, r_7, r_4\}$, which contain r_7 and the incoming interface of r_7 is 0. So, the set of possible error nodes is: $\{r_2, r_3\}$. Similarly, if r_7 receives a packet ($iif = 1$; $s = r_1$; $d = r_5$), the set of possible FRs is: $\{r_3\}$.

In addition, it is possible to improve the locating accuracy of the faulty router by deploying more VRs. For example, in Fig. 3, if r_6 is also set to a VR, the FR can be accurately located.

IV. VR Deployment Scheme Based on Link-State Routing Protocols

A. Selection Method of VRs

This section describes the algorithm for selecting VRs in an AS.

Definition 4: LT and Minimal LT (MLT). Let R be a set of nodes $\{r_1, r_2, ..., r_n\}$. Let $S\_N(r_i)$ be the set of neighbors of r_i. R is an LT if (5) is established. Then, if $\forall$ S_R' $\subseteq$ S_R, S_R' is not an LT, S_R is an MLT. It should be emphasized that LT or MLT is a network topology and does not refer to the path.

$$r_{i+1} \in S\_N(r_i), r_n \in S\_N(r_1), i \in \{1, 2, ..., n-1\} \tag{5}$$

Deduction 2: For an MLT, only two VRs are needed to verify all the wrong paths.

Proof: In an MLT as shown in Fig. 4, w_i is the weight (cost) of links, satisfying (6). Each link has equal weights in both directions.

$$w_i = \begin{cases} \text{the weight of link } (r_i, r_{i+1}), 1 \le i \le n-1 \\ \text{the weight of link } (r_i, r_1), i = n \end{cases} \tag{6}$$

In the MLT, r_1 and r_m are selected as two VRs, satisfying (7). We assume that there is no ECMP in the network.

$$\begin{aligned} &\sum_{i=1}^{m-1} w_i < \sum_{i=m}^{n} w_i \ \& \\ &\sum_{i=m+1}^{n} w_i < \sum_{i=1}^{m} w_i \ \& \\ &\sum_{i=1}^{m-1} w_i + \sum_{i=m+1}^{n} w_i > w_m \end{aligned} \tag{7}$$

Based on r_1 and r_m, the MLT is divided into two parts: $\{r_1, r_2, ..., r_{m-1}, r_m\}$ and $\{r_1, r_n, ..., r_{m+2}, r_{m+1}\}$. Then, the two VRs can verify all the wrong forwarding paths in the MLT.

According to the positions of s and d in the MLT, CFP(s, d) has the following four situations.

1) If $s = r_k$, $d = r_l$, $k < l$, r_k, and $r_l \in \{r_2, ..., r_{m-1}\}$, then we have

$$\sum_{i=k-1}^{l} w_i < \sum_{i=m}^{n} w_i. \tag{8}$$

Since all the wrong forwarding paths only form a loop, there is no BP(s, d).

2) If $s = r_k$, $d = r_l$, $k < l$, r_k, and $r_l \in \{r_{m+2}, ..., r_{n-1}, r_n\}$, then

$$\sum_{i=k-1}^{l} w_i < \sum_{i=1}^{m} w_i. \tag{9}$$

The conclusion is the same as (1).

3) $s \in \{r_1, r_m, r_{m+1}\}$ or $d \in \{r_1, r_m, r_{m+1}\}$. If there is a BP$(s, d)$, it must go through one of the two VRs: r_1 or r_m.

4) $s = r_k$, $d = r_l$, $k < l$, $r_k \in \{r_2, ..., r_{m-1}\}$, and $r_l \in \{r_{m+2}, ..., r_{n-1}, r_n\}$. Any BP$(s, d)$ must pass through one of the VRs: r_1 or r_m.

This ends our proof.

The calculation of the number of MLTs is equivalent to the calculation of the minimum loop path. The algorithm for calculating VR based on MLTs is designed as Algorithm 3, where S is the set of MLTs, N is the set of the intersection points of any two MLTs, and R is the VR set that meets the conditions. The algorithm takes each $node$ as the starting point to calculate a set R_{node}, where $node$ is the common node of any two MLTs. Finally, the VR set is the R_{node} with the fewest elements.

The VR selection algorithm needs to first traverse all of the overlapping nodes, which are nodes that belong to two or more MLTs, before traversing all the MLTs. The complexity is o(n^2).

Algorithm 3: Calculation of S_VR(s, d);

Require S_BP(s, d);
Ensure S_VR(s, d);
1: $S \leftarrow$ calculate all MLT;
2: $N \leftarrow \text{MLT}_i \cup \text{MLT}_j$;
3: $R = \{R_1, ..., R_n\}$;
4: node $\in N$;$m = 1$
5: **for** node; node $\in N$; node $++$ **do**
6: R_m.add(node)
7: $S' = S$
8: **while** $S' \neq \emptyset$ **do**
9: **if** node $\in \text{MLT}_x$ **then**
10: calculate node$'$, node$' \in \text{MLT}_x$; //node and node$'$ are VRs for MLT_x
11: S'.remove(MLT_x);
12: R_m.add(node$'$);
13: node $=$ node$'$;
14: **end if**
15: **end while**
16: $m++$;
17: **end for**
18: S_VR $(s, d) = \min\{|R_1|, ..., |R_n|\}$;

B. Appointment and Removal of VR

After being selected as VR, the router enables the verification function and plays the VR role. If the VR role of the router is revoked, the router disables the verification function. For VR appointment and discharge, each AS sets up a VR server for the appointment and removal of VRs. The VR server connects with a router in the AS to obtain the network link state of the AS in real time. When the link state changes, the VR server needs to calculate the new VR set. If the VR set changes, it sends the instruction of appointing or removing the VR to the router that needs to change the role. When the VR server finds that a router needs to become a VR, it first checks whether the router is a VR or not. If not, it sends the appointed VR instruction and VIT to the router. Once the link state changes, the VR server calculates the new VIT and sends it to the corresponding VRs, which updates their VITs. The VR life cycle is shown in Fig. 5. The advantage of this strategy is that it will not increase the extra overhead of the router, but the disadvantage is that it needs extra equipment, and the security of the VR server needs extra attention.

C. VIT Renewal

When the link state changes, a VIT needs to be recalculated and updated. Depending on the strategy in Section IV-B, VIT update calculations are done using different devices. The main steps are as follows.

1) The computing device uses the Floyd Algorithm to calculate the shortest distance between any two nodes.
2) For each destination node d, the computing device traverses all the source nodes s. If the VR is on the path (s, d), then (3); otherwise (4).
3) The computing device traverses all the interfaces of the VR. If the interface is consistent with $Interface$ (VR-1,VR), then the corresponding source node is set to 1, where $Interface$ (VR-1,VR) represents the $interface$

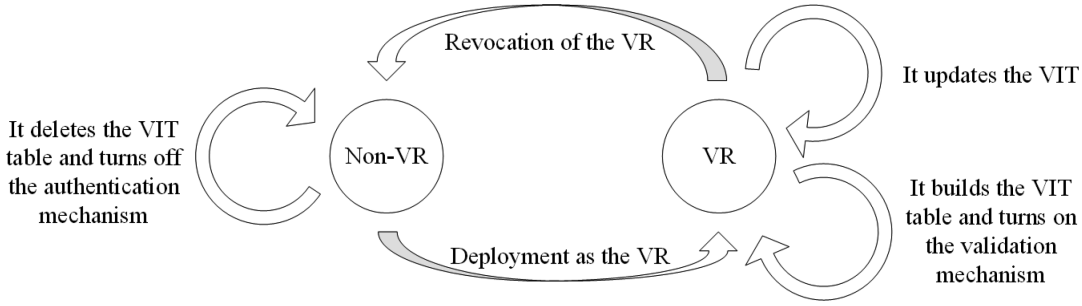


Fig. 5. Life cycle of VR.

TABLE III
COMMUNICATION OVERHEAD OF HEADER

Path Length	LRVP	EPIC(L0)	EPIC(L1)	EPIC(L2-L3)	ICING	OPT	PPV
l	0	3l	5l+8	5l+24	42l+13	19l+52	64
2	0	6	18	34	97	90	64
16	0	48	88	104	685	356	64

through which the message enters the VR from the previous hop.

4) If the VR is not on the path (s, d), the corresponding source node is set to 0.

The VIT renewal needs to traverse the source and destination nodes. The update complexity of the content pointed to by each VIT-index is O(n^2). In this article, the size will not exceed 256, so the actual VIT generation and update time is small . The VIT computing overhead is borne by the VIT server. The VIT is built based on the OSPF routing table. The routing table itself will also be updated with topology changes. VIT update and routing table update are combined without additional update overhead.

V. EXPERIMENT

A. Communication Overhead

The LRVP does not add additional fields to the packet, which is one of the "lightweight" aspects of this scheme. The existing schemes such as EPIC and PPV add protocol headers to the packets, which consume a certain amount of network bandwidth. We estimate the communication overhead with the parameter Goodput Ratio (GR), where GR $= p/(p +$ header), p is the payload (i.e., the size of the packet except for the extension header) and header is the protocol extension header. Let the path length be l (hops). Table III shows the relationship between the header and l of some existing schemes. The header's length of the PPV scheme is fixed, LRVP has no header, and the headers of the other schemes increase with the increasing l.

We assume that the packet payload is 1000B, the headers of the EPIC, ICING, and OPT schemes will increase with the paths. Among them, EPIC can achieve 99.40% GR under the optimal condition, and it still reaches 90.57% [EPIC (L2-L3)], 91.91% [EPIC (L1)], and 95.41% [EPIC (L0)] when $l = 16$. Compared with EPIC, the header's overhead of ICING and OPT is larger than EPIC. On the short paths, GR reaches 91%, but GR will decrease significantly on the long paths. In the PPV scheme, only two routers are marked each time, so its GR is always 93.98%. While the path is short, the communication overhead of PPV is larger than EPIC. But in a long-path transmission, PPV has obvious advantages in terms of communication overhead. The LRVP uses the idle field in the IP header, so the theoretical GR is 100%. The additional communication cost of the LRVP is minimal and it is not limited by the path length, so our scheme is lightweight.

Fig. 6(a) and (b) shows the relationship between GR and the paths for 1000B and 1400B payloads (excluding extended headers), and LRVP and PPV have no relationship with l. The GR of the other schemes will decrease to a certain extent while the path gets longer. The scheme of dynamically increasing the length of the header may lead to transmission failure or sharding. For example, if the MTU is 15000B, when p reaches 1400B, as shown in Fig. 6(b), EPIC(L2-L3), if the path length exceeds 14, the sum of header and p exceeds the maximum load of a packet. For ICING and OPT, one packet cannot be used to transmit p of 1400B with a path length greater than 2. Particularly, if the path length is uncertain before transmission, enough header space must be reserved, which has a potential bandwidth waste and transmission risk.

B. Storage Costs

The storage cost here refers to the extra storage on the router, not the header storage cost. This article requires an additional VIT, which is the primary storage overhead. The number of entries in VIT is calculated by (3), which depends on two parameters I (the number of VIT indexes) and L (the number of interfaces that a VR connects to the other routers). There is a strong correlation between the two parameters and the topological structure. I depends on the actual forwarding path. In the experiments, we still let I get the maximum value, i.e., $I = n$, where n is the number of routers in the AS. L depends on the number of interfaces of the VR. To quantify L, we randomly select 143 topologies [9], which contain 2444 nodes. After counting the number of neighbors (L) of each node and get

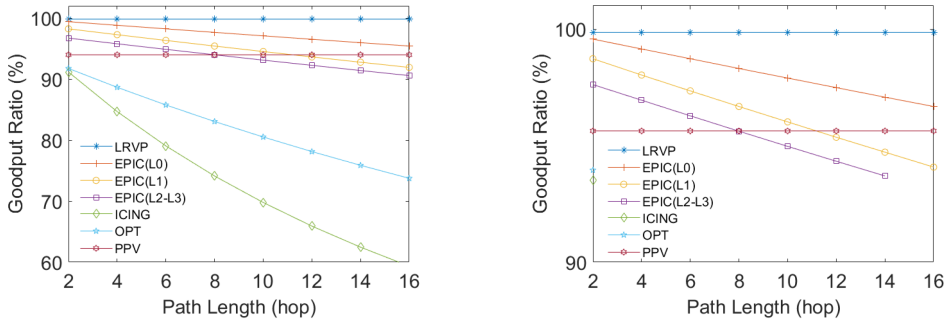


Fig. 6. GR values under different payloads. (a) GR under different path lengths at 1000B payload level. (b) GR under different path lengths at 1400B payload level.

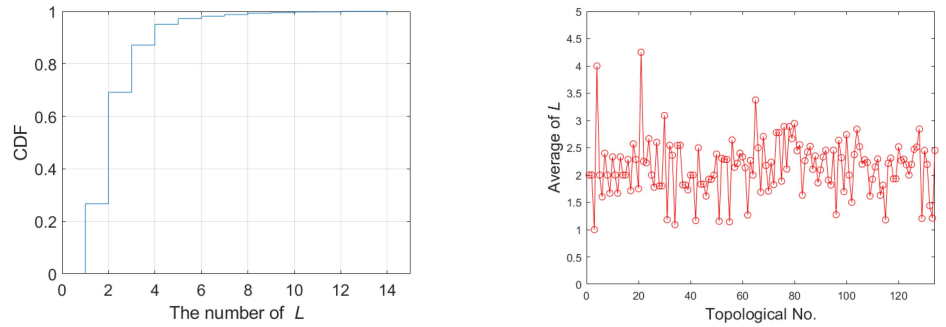


Fig. 7. VIT storage overhead analysis. L is the number of interfaces that a VR connects to the other routers. (a) The CDF of L. (b) The average of L.

the cumulative distribution function figure, as shown in Fig. 7(a), we can see that 95.04% of L is between 1% and 4. 4.96% of L is greater than or equal to 5. Only 0.49% of L is more than 10. In addition, to prove that L of each node is small, we calculated the average value of L. As shown in Fig. 7(b), we also calculated the average value of L for 134 topologies. In most common topologies, the average value of L for each node is mostly concentrated between 1.5 and 3, which is an acceptable value. Suppose that an AS contains n nodes, the number of VIT entries in most VRs are between n and $6n$, and the maximum value is $6 \times 256 = 1536$. The average number of VIT entries of a VR is between n and $4n$, and the maximum number is $4 \times 256 = 1024$.

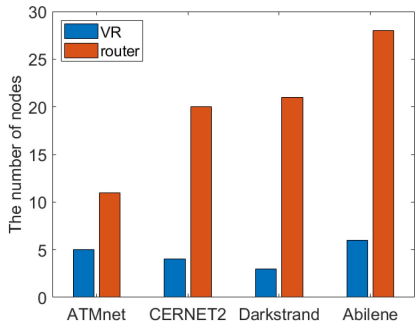


Fig. 8. Number of VRs. ATMnet, CERNET2, Darkstrand, and Abilene [9] are four typical topologies.

C. VR Number

Not all routers need to be deployed as VRs. Whether a router needs to be deployed as VRs depends on the topology. The number of VRs is related to the number of MLTs described in Section IV. We selected four typical topologies: ATMnet(13), CERNET2(20), Darkstrand(21), and Abilene(28), with the number of topology nodes in parentheses. The number of VRs and the total number of nodes in each topology are calculated, as shown in Fig. 8. For most topologies, especially large ones, only about a quarter of the routers in the topology need to be deployed as VRs. This is also one of the "lightweight" embodiment of the LRVP.

D. Throughput Overhead

In this section, we compare the packet processing time of a VR in the LRVP and a tag router in PPV. In LRVP, the VR needs to query the VIT once, and the PPV's tag router has to calculate a HASH value. Compared with the existing schemes, the computing cost of this scheme is mainly the query of the VIT. The LRVP needs to verify the validity of the incoming interface and the source ID during forwarding, which may increase the processing overhead. However, the VIT designed in this article greatly optimizes the lookup performance. Through the previous experiments, it is verified that the maximum storage cost of the VIT is medium. But the actual storage cost in the

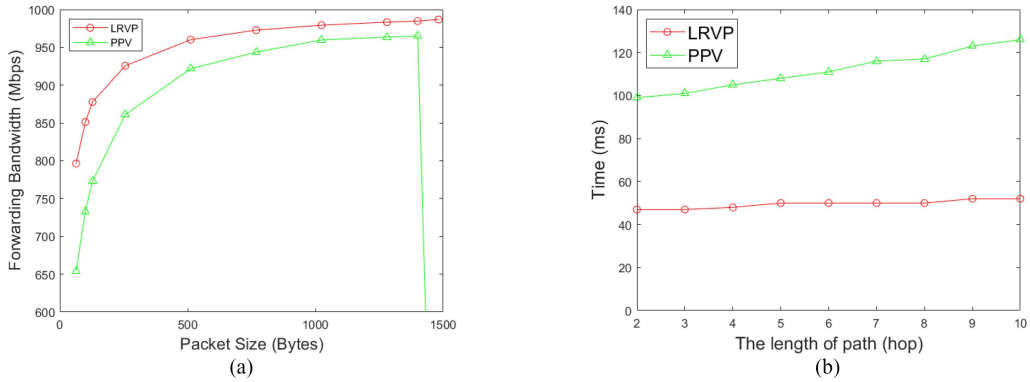


Fig. 9. Throughput overhead and computation overhead. (a) Throughput of PPV and LRVP under different payloads. (b) Packet processing time under different path lengths.

topology is small. So, its computational cost consumed is far less than that of encryption, hash, and other operations. The throughput test is carried out on four real devices (P4 devices), i.e., the path length is four, and the throughput of LRVP and PPV is compared. As shown in Fig. 9(a), we tested throughput under different payloads. The payload is described as an IP packet. The maximum PPV payload is 1436B, because the PPV requires an additional 64B PPV header. When the load is greater than 1436B, the PPV cannot use a single packet for transmission.

In addition, we use BMv2 to simulate the packet processing time. The simulation environment is: i9 CPU, virtual machine ubuntu 16.04, configured with four CPUs, and 4G memory. As show in Fig. 9(b), we tested the time required for each packet processing time under different path lengths. It can be seen that the LRVP takes about half of the time to process a single package as PPV.

E. Balancing VR and Verifiable Paths

Due to the close correlation between the number of VRs and topology, the tradeoff between the number of VRs and effective incorrect forwarding path detection can only be analyzed topologically. The four topologies selected in Section V-C are used for analysis. For each topology, an empty set of paths to verify is set. A path is randomly selected from the topology and placed into the set. Calculate the minimum number of VRs required to validate all paths within this set. Each time a path is added, the VR count is counted until all paths are added to the set. Fig. 10 shows the cumulative status of the number of VRs of four topologies. Experimental results show that the paths have a very high overlap–many paths share the same VR. When the proportion of paths to be verified is within 20%, the number of VRs increases significantly. After more than 20%, the number of VR basically reaches the maximum value.

F. Summary of Other Overhead

VR deployment increases capital expenditure and operating expenditure. It is mainly reflected in the following aspects: Existing routers need minor modifications to support VR functions. On the control plane, after the shortest path is calculated, the information about the incoming interface is added to the route entries. On the data plane, VR checks whether the incoming interface of packets is valid. The ID of the packet is added in edge routers. The VR server is used to complete the calculation of the selected VR and update the VIT.

VI. Related Work

A. Secure Traceroute

Unlike traditional traceroute solutions, Secure traceroute uses encryption to protect the metadata involved in validating packet paths [10], enabling the destination to retrieve a packet path and verify its authenticity. Padmanabhan and Simon[11] propose a probe-based scheme in which each router responds to the mac-protected address for the next-hop so that the source can reconstruct the path that the probe packet has traveled. Wong *et al.* [12] remove the secret channel and the preestablished key of [11], stores the key locally, and uses forwarding matching to correspond to the source. But the scheme based on forwarding matching cannot adapt to the granularity of all packages. AudIt [13] uses service-level agreements (SLAs) instead of encryption to rebuild paths based on aggregating statistics in the response packets of a series of routers. RPVM [14] changes the statistical distribution of the packet technology by dividing the transmission time into multiple time slots. Compared with the probe-based scheme [13], [14], the efficiency is greatly improved. However, due to the lack of encryption, the statistically based scheme is vulnerable to attacks. SPP [15] maintains an encrypted security history for routers to counter packet forgery.

B. Secure Source Routing

Secure source routing focuses on protecting the routing process and prevents packet forgery and tampering by encrypting the path instructions carried by packets [16]. Intermediate routers do not embed proofs in packet tags. Platypus [16] specifies the relay station that must be accessed. Avramopoulos *et al.* [17] improve robustness by adding acceptance of confirmation and fault notification. In addition, the authors in [19]–[22] encrypt the path element and payload in the header considering anonymity and other issues.

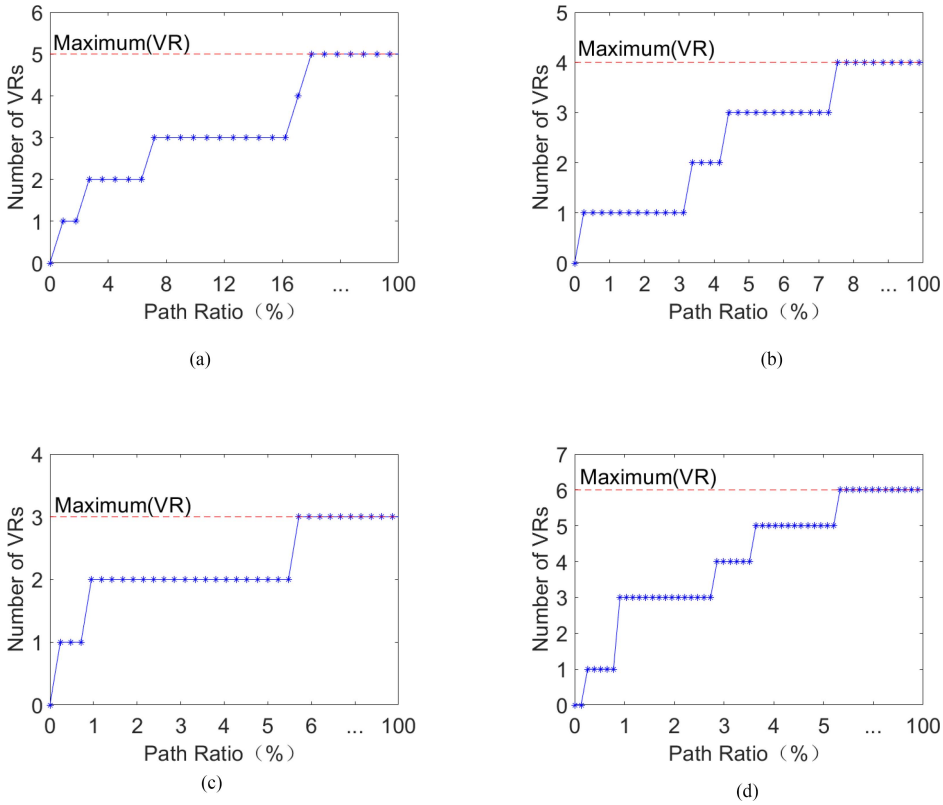


Fig. 10. Cumulative status of the number of VRs of four topologies. Path ratio represents the proportion of paths with verification requirements. (a) ATMnet. (b) CERNET2. (c) Darkstrand. (d) Abilence.

C. Path Validation

This requires that the encrypted states embedded in the packet header execute the specified path and the router updates the status to verify the path [23]. PFRI [24] is the first scheme that integrates the aforementioned two schemes and validates the path traversed by the package through introducing motivation, accountability, Knobs, and Dials. ICING [25] strengthens the requirements of path validation protocol design and selects two key designs: aggregate message authentication code MACs and self-certifying names. Due to more symmetric encryption calculation and key generation, the calculation cost of ICING is high. OPT [2], by sacrificing some intermediate routing security, reduces computing overhead and communication overhead compared with ICING to a certain extent. But the communication overhead is still high. OSV [3], [26] improves the system efficiency by introducing orthogonal sequences. PPV [4] is calculated by two routers at the same time in a probabilistic manner to reduce the computing and communication overhead. However, since packets need to be collected within a certain period of time to "piece together" the path, there are certain hidden dangers. By designing a data plane protocol with an optional security level at the three source terminals and using a small number of efficient symmetric encryption operations, EPIC [5] reduces the computing and communication overhead significantly. Atomos [6] still uses the method of adding a proof to the header that is validated by other routers. Asymmetric cryptography is used to minimize the number of proofs required, and proofs of constant size are designed, which reduces the time to process proofs. Alibi Routing [27] is a novel scheme that can prove that packets cannot travel through certain nodes.

VII. Conclusion

In this article, we design and evaluate an efficient path compliance verification scheme, an LRVP based on the OSPF protocol for ASes to reduce communication overhead. We assign an ID to each router, and the source router marks the packet with its own ID. The VR is added a VIT on the basis of the traditional routing table. The VR records the interfaces of packets entering the router and the marked IDs of the packets, and by querying the VITs, achieves efficient and real-time verification path. The storage overhead of a VIT is small (the table entry size is about 8K) and the communication overhead is low. Compared with the more advanced schemes at present, our scheme has the following advantages: first, it does not introduce additional extended headers, which greatly reduces the communication overhead; and second, it does not use any encryption algorithm, which reduces the packet processing time.

The LRVP needs to modify the forwarding engine of VRs, and the VIT is added, which brings additional storage overhead. It is only deployed for ASes. If the experiment is actually deployed in real time, the challenges include maintaining a VR server to manage VRs, and the VR server needs to calculate which

routers to deploy VR functions. In addition, to further improve the efficiency of path authentication, no encryption algorithm is used, which sacrifices the security of the packet content.

REFERENCES

[1] C. Zhang, F. Luo, and G. Ranzi, "An advanced persistent distributed denial-of-service attack model with reverse-path forwarding-based defending strategy," *IEEE Access*, vol. 7, pp. 185590–185596, 2019.

[2] T. H. J. Kim *et al.*, "Lightweight source authentication and path validation," in *Proc. ACM Conf. SIGCOMM*, 2014, pp. 271–282.

[3] H. Cai and T. Wolf, "Source authentication and path validation with orthogonal network capabilities," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2015, pp. 111–112.

[4] B. Wu *et al.*, "Enabling efficient source and path verification via probabilistic packet marking," in *Proc. IEEE/ACM 26th Int. Symp. Qual. Serv.*, 2018, pp. 1–10.

[5] M. Legner, T. Klenze, M. Wyss, C. Sprenger, and A. Perrig, "EPIC: Every packet is checked in the data plane of a path-aware Internet," in *Proc. 29th USENIX Secur. Symp.*, 2020, pp. 541–558.

[6] A. He, K. Bu, Y. Li, E. Chida, Q. Gu, and K. Ren, "Atomos: Constant-size path validation proof," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3832–3847, Jun. 2020.

[7] F. Baker and P. Savola, "Ingress filtering for multihomed networks," RFC Editor, IETF, 2004.

[8] T. W. Chim, K. L. Yeung, and K. S. Lui, "Traffic distribution over equal-cost-multi-paths," *Comput. Netw.*, vol. 49, no. 4, pp. 465–475, 2005.

[9] The Topology, 2013. [Online]. Available: http://www.topology-zoo.org/dataset.html

[10] K. Bu *et al.*, "What's (Not) validating network paths: A survey," 2018, *arXiv:abs/1804.03385*.

[11] V. N. Padmanabhan and D. R. Simon, "Secure traceroute to detect faulty or malicious routing," *ACM Sigcomm Comput. Commun. Rev.*, vol. 33, no. 1, pp. 77–82, 2003.

[12] E. L. Wong *et al.*, "Truth in advertising: Lightweight verification of route integrity," in *Proc. 26th ACM Symp. Princ. Distrib. Comput.*, 2007, pp. 147–156.

[13] S. Shenker, P. Maniatis, K. Argyraki, O. Irzak, and S. Ashish, "Loss and delay accountability for the Internet," in *Proc. IEEE Int. Conf. Netw. Protoc.*, Beijing, China, 2007 pp. 194–205.

[14] J. Jiang *et al.*, "A network accountability based verification mechanism for detecting inter-domain routing path inconsistency," *J. Netw. Comput. Appl.*, vol. 36, no. 6, pp. 1671–1683, 2013.

[15] A. Chen *et al.*, "One primitive to diagnose them all: Architectural support for Internet diagnostics," in *Proc. 12th Eur. Conf.*, 2017, pp. 374–388.

[16] B. Raghavan and A. C. Snoeren, "A system for authenticated policy-compliant routing," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, 2004, pp. 167–178.

[17] I. Avramopoulos *et al.*, "Highly secure and efficient routing," in *Proc. IEEE INFOCOM 2004*, 2004, p. 208.

[18] M. Casado *et al.*, "ETHANE: Taking control of the enterprise," in *Proc. ACM SIGCOMM Conf. Appl., Technol., Architectures, Protoc. Comput. Commun.*, Kyoto, Japan, 2007, pp. 1–12.

[19] Si. Peter, U. Javed, Q. Zhang, D. Woos, T. Anderson, and A. Krishnamurthy, "One tunnel is (often) enough," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 99–110, Oct. 2014.

[20] M. G. Reed and P. F. Syverson, "Anonymous connections and onion routing," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 4, pp. 482–494, May 1998.

[21] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proc. USENIX Secur. Symp. 2004*, 2004, pp. 303–320.

[22] C. Chen, D. E. Asoni, D. Barrera, G. Danezis, and A. Perrig, "HORNET: High-speed onion routing at the network layer," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1441–1454.

[23] K. Bu *et al.*, "Unveiling the mystery of Internet packet forwarding: A survey of network path validationy [J]," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–34, 2020.

[24] K. L. Calvert, J. Griffioen, and L. Poutievski, "Separating routing and forwarding: A clean-slate network layer design," in *Proc. IEEE Int. Conf. Broadband Commun.*, 2007, pp. 261–270.

[25] J. Naous, M. Walfish, A. Nicolosi, D. Mazières, M. Miller, and A. Seehra, "Verifying and enforcing network paths with ICING," in *Proc. 7th Conf. Emerg. Netw. Exp. Technol.*, 2011, pp. 1–12.

[26] H. Cai and T. Wolf, "Source authentication and path validation in networks using orthogonal sequences," in *Proc. Int. Conf. Comput. Commun. Netw.*, 2016, pp. 1–10.

[27] D. Levin *et al.*, "Alibi routing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 611–624, 2015.

Wenlong Chen received the Ph.D. degree in communication and information system from the University of Science and Technology Beijing, Beijing, China, in 2011.

He is currently a Professor with the College of Information Engineering, Capital Normal University, Beijing. His research interests include network protocol, Internet architecture, high-performance router, and wireless sensor networks.

Xiaolin Wang received the B.S. degree in information security from the College of Computer and Information Technology, Beijing Jiaotong University, Beijing, China, in 2019. He is currently working toward the postgraduate degree in software engineering with the College of Information Engineering, Capital Normal University, Beijing.

His main research interests include computer networking.

Xiaoliang Wang received the Ph.D. degree in computer science and technology from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2017.

He is currently a Lecturer with the Information Engineering College, Capital Normal University, Beijing. His research interests include wireless networks and network security.

Ke Xu was born in 1974. He received the Ph.D. degree in computer network from Tsinghua University, Beijing, China, in 2001.

He is currently a Professor with Tsinghua University. His research interests include architecture of next-generation of Internet, high-performance router, peer-to-peer and overlay network, and Internet of Things.

Sushu Guo received the B.S degree in engineering from the School of Software Engineering, Shanxi Agricultural University, Jinzhong, China, in 2018. She is currently working toward the postgraduate degree in software engineering with the College of Information Engineering, Capital Normal University, Beijing, China.

Her main research interest include IPv6 and wireless sensor network.