IJNRD.ORG ISSN : 2456-4184
International Research Journal
IJNRD
Research Through Innovation
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

NETWATCH: EMPOWERING NETWORK VISIBILITY AND BLACK HOLES DETECTION & INDENTIFICATION IN SEGMENT ROUTING

[1]Sanjay H S, [2]Kumarswamy S
[1]PG Student, [2]Assistant Professor
[1,2]Department of Computer Science and engineering,
[1,2]University visvesvaraya College of Engineering, Bangalore, India

Abstract : Detecting network black holes (BHs) proves to be a formidable undertaking, given their propensity to disrupt services for particular traffic flows due to device misconfigurations. The intricacies escalate within the framework of Segment Routing (SR) due to the necessity of an encapsulation mechanism for enforcing segment lists on packets. While existing active probing tools targeting IPv6 BH detection fall short in the context of SR, this project aims to confront the challenge of identifying SR Black Holes within SR domains. In this endeavor, we draw upon an experimental showcase illustrating the creation of an SR Black Hole to underscore the limitations inherent in active probing-based tools. To surmount these constraints, we introduce a passive framework named Segment Routing Black Holes Detection (SR-BHD). SR-BHD capitalizes on specific traffic counters present in SR-capable nodes to validate the conservation of flow principle across network elements. Functioning by continuously monitoring traffic flow within SR domains, SR-BHD offers an alternative to active probing. During the experimental evaluation, various scenarios are simulated, and SR environments are emulated to assess the efficacy of SR-BHD. The outcomes reveal that SR-BHD adeptly discerns SR BHs by identifying any deviation from the flow conservation principle. This capability enables network administrators to promptly recognize and rectify BHs, mitigating the impact on traffic flows and ensuring a dependable network performance. In accordance with proper citations, the SR-BHD framework stands as an innovative approach for passive SR Black Hole detection, providing an efficient alternative to active probing. Its adaptability within SR domains and adept handling of challenges posed by the encapsulation mechanism solidify its status as a valuable tool for network monitoring and failure detection within SR architectures. Overall, this project makes a substantial contribution to enhancing the reliability and performance of SR networks by addressing the specific issue of detecting SR Black Holes. The SR-BHD framework emerges as a pragmatic and effective solution, empowering network administrators to proactively identify and address BHs, thereby guaranteeing uninterrupted and efficient traffic flow within SR domains.

Index Terms: Segment routing, network black hole, failure detection, network monitoring, flow conservation principles.

Research Through Innovation

1.INTRODUCTION

The emergence of the Network Function Virtualization (NFV) paradigm, coupled with the necessity to configure intricate services through Service Function Chains (SFCs), has driven advancements in routing technology. The Software Defined Networking (SDN) has played a pivotal role in enhancing control plane functionalities, enabling the efficient implementation of flexible routing algorithms. Complementing this, the Segment Routing (SR) architecture addresses data plane requirements, efficiently leveraging the concept of source routing, particularly within the context of SRv6 [1]. SR introduces a robust network programming model [2] that offers unparalleled expressiveness for defining network programs applied to traffic flows. To alleviate the load on network nodes and ensure scalability, network programs are embedded directly into the packet header. Specifically, SR uses the IPv6 data plane (SRv6, [3]), incorporating the segment list into a new extension header named Segment Routing Header (SRH). However, the flexibility of SR comes with a trade-off, introducing overhead. While this cost is manageable given the increasing capacities of backbone links, longer packets in an IPv6 data plane may lead to anomalies in packet forwarding, given IPv6's known challenges with Maximum Transmission Unit (MTU) handling [4]. These challenges may result in black holes [5], where communication failures occur due to the silent discard of oversized packets.

The operational principle of SRv6 further exacerbates MTU constraint issues in IPv6, suggesting the use of a greater MTU value within an SR domain [3]. Efforts to reduce the overhead for packet program enforcement have introduced the concept of microSIDs [6]. Despite these efforts, the risk of network black hole events persists. This framework expands on the initial concept to address realistic scenarios with multiple sources of packet loss. Additionally, a procedure utilizing information from SR traffic counters is defined to enhance the performance of Segment Routing Black Holes Detection (SR-BHD). Through extensive simulations and sensitivity analysis, the efficacy of SR-BHD in detecting SR Black Holes is evaluated. Furthermore, a prototype implementation validates the passive monitoring approach.

Several factors contribute to black hole creation, with network programs potentially enforced at every network node based on logical conditions. The recommendation in [3] aims to increase Δ, the difference between the bottleneck link's MTU and the entering packet's length, while microSIDs aim to reduce overhead (O). However, the fate sharing paradigm assumes that probe and data packets share the same network "fate," which may not hold true in SR architecture due to its policy routing approach. As a result, active detection tools are unsuitable for SR Black Holes.This paper tackles the detection of MTU-related black holes in an SRv6 network, proposing a passive monitoring framework, SR-BHD. This framework accurately detects SR Black Holes, providing a concise list of suspected link/flow pairs impacted by black holes. Building upon the seminal idea presented in [9], this paper demonstrates the existence of SR Black Holes through an experimental demonstration. It also highlights the inadequacy of active probing-based tools for reliable detection and extends the framework to handle realistic scenarios with multiple sources of packet loss. Furthermore, a procedure based on additional information from SR traffic counters is introduced to enhance SR-BHD performance. The paper includes an extensive evaluation, including a sensitivity analysis, and presents a prototype implementation to validate the efficacy of the passive detection approach.

2.NEED OF THE STUDY.

In this paper we address the problem of detecting MTU related black holes in an SRv6 network. In particular, we propose a passive monitoring framework that is able to accurately detect the presence of *SR Black Holes*, providing as output a short list of suspected link/flow pairs, i.e., the list of flows impacted by black holes and the links causing such black holes. The proposed framework, named *Segment Routing Black Holes Detection* (*SR-BHD*) uses a passive approach based on the observation of traffic counters available in SR capable nodes [10]. The present paper extends the seminal idea presented in [9], by achieving the following improvements:

- we prove, through an experimental demonstration, the existence of *SR Black Holes*, that was conjectured in [9];
- we show that *SR Black Holes* cannot be trustworthy detected by means of an active probing based tool;
- we extend the framework presented in [9] to make it work also in realistic scenarios where multiple sources of packet loss exist;
- we define a procedure based on the availability of extra information provided by specific SR traffic counters, to improve the performance of *SR-BHD*;
- we run an extensive evaluation, including a sensitivity analysis, to assess the performance of the proposed framework in terms of *SR Black Holes* detection;
- we present a prototype implementation to validate the effectiveness of the detection through the proposed passive approach.

3.1Literature Survey

In this section we provide an overview of the research activities related to network black holes. In particular, we divide the literature in two categories: i) known types of network black holes and existing frameworks for their detection, and ii) performance measurements tools in the context of Segment Routing architecture.

3.1.1 Network Black Holes and Detection Frameworks

s As specified in [11], network black holes are silent logical failures that often result from events like misconfiguration or software bugs. The utilization of overlay architectures appears to be a common catalyst for black hole creation, as highlighted in [12], where diverse failure modes leading to black hole occurrences are discussed in the context of an IP over MPLS infrastructure. Failures in the Label Distribution Protocol (LDP) execution can create a black hole in this scenario, wherein the underlying IGP domain functions correctly while end-to-end reachability is compromised. This study focuses on black holes emerging in an SRv6 network due to the violation of the MTU constraint caused by Path MTU Discovery (PMTUD) failure, as outlined in [5]. Various failure modes for the PMTUD procedure, including unresponsive routers configured to withhold ICMP Packet Too Big (PTB) messages for oversized packets, are described in detail.

Numerous detection systems have been proposed across different contexts to identify network black holes, all relying on active testing through probe transmissions. In [11], an active probing detection mechanism is defined for detecting network black holes in an IP/MPLS backbone. The method employs a full mesh of probes exchanged periodically among edge routers, utilizing the concept of a failure signature to identify suspicious links. A reliable tool for discovering PMTUD failures is Scamper [5], employing a two-step procedure to determine the largest usable MTU on an end-to-end path and identify routers not participating in PMTUD. Netalyzr, presented in [13], is a network measurement and debugging tool that determines the path MTU to a destination server by emitting UDP probes. Ripe Atlas [14], a global monitoring infrastructure, has been utilized to discover path MTU black holes in the Internet, identifying causes and affected data plane protocols. This paper extends the concept presented in [9] – a passive approach based on processing traffic-related data in SRv6 network devices. This passive approach has been successfully employed in the past to identify and detect network anomalies and failures, as exemplified in [15] for network tomography and [16] for statistical analysis using Signal Processing techniques on SNMP MIB data. However, no passive monitoring tool has previously targeted the detection of SR Black Holes.

3.1.2 SR Performance Measurement Tools

The SR architecture offers a suite of Operation and Maintenance (OAM) tools empowering Network Operators with capabilities for infrastructure performance measurement, troubleshooting, and more. Various tools and methodologies are discussed in the literature. In [17], a scalable and topology-aware data-plane monitoring system for SR-MPLS is presented. Ping and Traceroute functionalities tailored for SR networks are outlined in [18]. Bidirectional Forwarding Detection (BFD) for assessing the vitality of Segment Routing Policies in Traffic Engineering is detailed in [19].

Nodes with SR capabilities can utilize SR Routing Traffic Counters (SRTCs) to gather traffic statistics at different granularities. SRTCs, namely SRINT, PSID, and POL, are integral to the proposed framework. SR-INTs, or link counts, quantify SR traffic on specific links. PSID counters account for received traffic directed to a specific node, while POL counters track traffic steered through a specific SR policy. The comprehensive *Segment Routing Background*. Segment Routing (SR) [23] introduces a innovative network paradigm grounded in source routing, wherein the source node determines the path for each packet. The end-to-end paths are encoded as an ordered list of instructions known as segments, collectively forming the Segment List (SL). These segments are represented as labels, termed Segment IDentifiers or SIDs. In the context of SRv6, where the underlying data plane utilizes IPv6, a SID is expressed as an IPv6 address descriptions of these counters and their interrelations are elucidated in [20], where a mathematical model demonstrates their effectiveness in enhancing various networking algorithms, such as Traffic Matrix Assessment and Traffic Anomaly Detection. This paper draws extensively from the insights presented in [20]. In [21], an SRv6 Performance Monitoring (SRv6-PM) framework is introduced, enabling in-depth performance monitoring in an SRv6 infrastructure. The framework comprises data plane tools for line-rate traffic measurements, a control plane logic for node-specific measurements, and a Cloud Native Big Data Management system for storage, processing, and visualization. A use case involving fine-grained measurement of packet loss in a single SR flow is validated through SRv6-PM, leveraging SR traffic counters at ingress and egress nodes.

The packet forwarding mechanism in SRv6 operates as follows: upon receiving a packet, the border router must direct it along a specified SL determined by a designated SR Policy. After processing the packet in accordance with the matched SR Policy, the outermost IPv6 header is extended with a Segment Routing Header (SRH). This SRH contains the SL and a pointer indicating the active segment, i.e., the current SID for packet forwarding. Specifically, the active segment is replicated in the destination address field of the outer IPv6 header. Transit routers forward incoming packets by examining the IPv6 .

For this study secondary data has been collected. From the website of KSE the monthly stock prices for the sample firms are obtained from Jan 2010 to Dec 2014. And from the website of SBP the data for the macroeconomic variables are collected for the period of five years. The time series monthly data is collected on stock prices for sample firmsand relative macroeconomic variables for the period of 5 years. The data collection period is ranging from January 2010 to Dec 2014. Monthly prices of KSE -100 Index is taken from yahoo finance. destination address of the outer header. Upon reaching a node with the same SID as the active segment, the SID-related function is applied, with various functions possible. A common function is the END, signifying that the active segment needs updating, shifting the pointer to the next SID in the SL. Another action is the enforcement of an SL by introducing a new policy. Finally, before exiting the SR domain, the SRH must be removed.

International Research Journal

4.SEGMENT ROUTING BLACK HOLES DETECTION ALGORITHM (*SR-BHD*)

The SR-BHD algorithm represents a passive monitoring solution designed for the detection of SR Black Holes within SR networks. This section outlines the system model and notation, followed by an explanation of the SR-BHD working principle in two phases: an initial discussion of the ideal model and subsequent modifications to enhance SR-BHD's robustness against "noise signals," such as packet loss due to congestion. Lastly, an improved version, SR-BHD+, is introduced, leveraging advanced traffic counters to enhance the precision of SR-BHD. Let G(N,L) denote the graph illustrating the topology of the SR domain under consideration, where N and L represent the sets of nodes and links, respectively. For a given link l, the head node is denoted as l.h, and the tail node is referred to as l.t. The node-SID for node $i \in N$ is symbolized by the same notation. Tab. II provides a summary of the notation. In this context, three types of traffic counters are employed: link count (yL(l)), Prefix Segment IDentifiers (PSID) counters (yB(i , a)), and POLicy (POL) counters (yP(i , e, c)). The link count, yL(l), measures the traffic (in bytes) transmitted over link . PSID counters, yB(i , a), at node i track the packets received with a as the active segment. POL counters, yP(i , e, c), associated with SR policies, record the traffic steered through them. Practical examples of these counters are illustrated in Tab. I.

The set of application flows injected into the SR domain is denoted as F. Each application flow f traverses the SR domain by being steered through an SR policy. Considering the policy < i, e, c >, the set of application flows handled through it is Πi ,e,c. An SR flow, xi ,e,c, represents the aggregated traffic steered through the same policy. The vector x collects all SR flows. In the IPv6 underlay layer, the shortest path policy computes the path between each pair of nodes in the SR domain. The underlay path from node i to node a is gi ,a, represented by a vector of length L, indicating the percentage of traffic over link l if node i sends one unit of traffic to node a. In the overlay, routing is determined by the configured segment lists. The segment list for SR policy < i, e, c > is σi ,e,c, an ordered sequence of node-SIDs. The vector ri ,e,c, representing the overlay path of SR flow xi ,e,c, is calculated using Eq. (1).

$$\mathbf{r}_{i,e,c} = \sum_{k=1}^{|\sigma_{i,e,c}|-1} \mathbf{g}_{\sigma_{i,e,c}[k],\sigma_{i,e,c}[k+1]} \qquad (1)$$

Equation (1) illustrates that the overlay routing of an SR flow is determined by a linear combination of vectors representing paths in the underlay between nodes specified in the segment list. The routing matrix R encompasses vectors ri ,e,c for all existing policies, featuring L rows and as many columns as the number of flows. Concerning the PSID counter instantiated at node n for the active segment a, the variable bn,a i ,e,c signifies the percentage of SR flow xi ,e,c it represents. The matrix B is a compilation of all bn,a i ,e,c variables for PSID counters and SR flows. Therefore, the relationship between SR flows and the value assumed by each PSID counter can be expressed using Eq.

4.1 SR-BHD Principle

SR-BHD draws its conceptual foundation from the flow conservation principle [27], a fundamental tenet in network theory. This principle dictates that the disparity between incoming and outgoing flows at a network node should equate to the local demand. In instances of an SR Black Hole emerging on a specific link, this principle encounters disruption. Specifically, a segment of the flow intended to traverse the node through the affected link is lost due to MTU constraints, resulting in an imbalance between incoming and outgoing traffic. It is crucial to note that the presence of a black hole is not the exclusive factor causing a violation of the flow conservation principle. Various factors, including congestion, binary errors, and the absence of a route (e.g., destination unknown), can introduce packet loss, disrupting the equilibrium between incoming and outgoing traffic. We will initially present the equations governing the flow conservation principle and subsequently outline an approach to integrate different sources of packet loss into the mathematical model.

The methodology section outline the plan and method that how the study is conducted. This includes Universe of the study, sample of the study,Data and Sources of Data, study’s variables and analytical framework. The detailsare as follows;

$$\sum_{a\in\mathcal{N}} g_{i,a}(l)\cdot y_B(i,a) = y_L(l)\forall l \in \mathcal{L} \qquad (3)$$

$$y_B(i,a) = \sum_{l\in\delta_i^+} g_{l.t,a}(l)\cdot y_B(l.t,a)\forall i,a \in \mathcal{N} \qquad (4)$$

SR-BHD monitors network traffic statistics periodically and assesses the validity of flow conservation principles to detect potential SR Black Holes. The verification involves two distinct equations within SR-BHD. Equation (3) specifically enforces the flow conservation principle at the link level. It ensures that the total traffic received at node i and destined for forwarding over output link l equals the actual traffic sent over link l. Each PSID counter keeps track of traffic received at node i with an active segment 'a'. This traffic follows the underlay path between nodes i and a, represented by the vector gi, a.

The l-th component of this vector, multiplied by the PSID counter value, determines the portion of traffic routed over link l. By summing contributions from all possible active segments, the amount of received traffic forwarded by node i over output link l can be calculated. In an ideal scenario without losses, this quantity aligns with the link count yL(l). For example, applying Eq. (3) to the link between nodes C and D in a toy case scenario (as illustrated in Fig. 5 with blue counters), traffic forwarded by node C towards the next hop D is determined by the sum of PSID counters yB(C,D) and yB(C,E) (with values 110 and 160, respectively), totaling 270 units of traffic.

4.2 Framework Overview

In Fig. 1, we present the key functional components of the proposed SRBHD framework, designed for integration into a centralized monitoring system. The central monitoring system interfaces with network devices via a southbound connection, extracting traffic statistics that are subsequently stored in distinct databases. Upon the availability of new traffic measurements, the monitoring system activates the SR-BHD block. This block takes input from link and PSID traffic counters, network topology, and the existing routing configuration encompassing IPv6 paths and SRv6 segment lists. Additionally, the SR-BHD block receives input from the margin value, encapsulated in a vector, providing estimates of packet loss attributed to various factors such as congestion and transmission errors on each network link. The margin estimation block, dependent on the current link utilization, supplies this information.

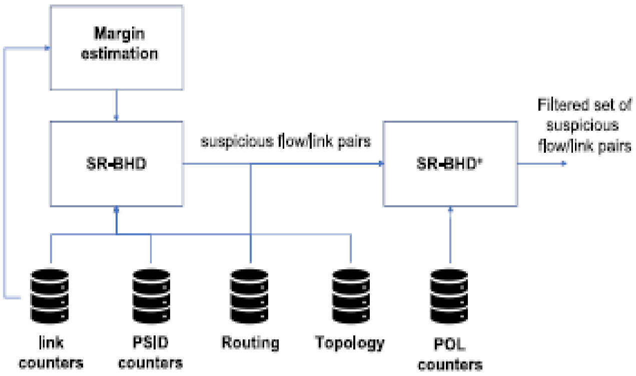
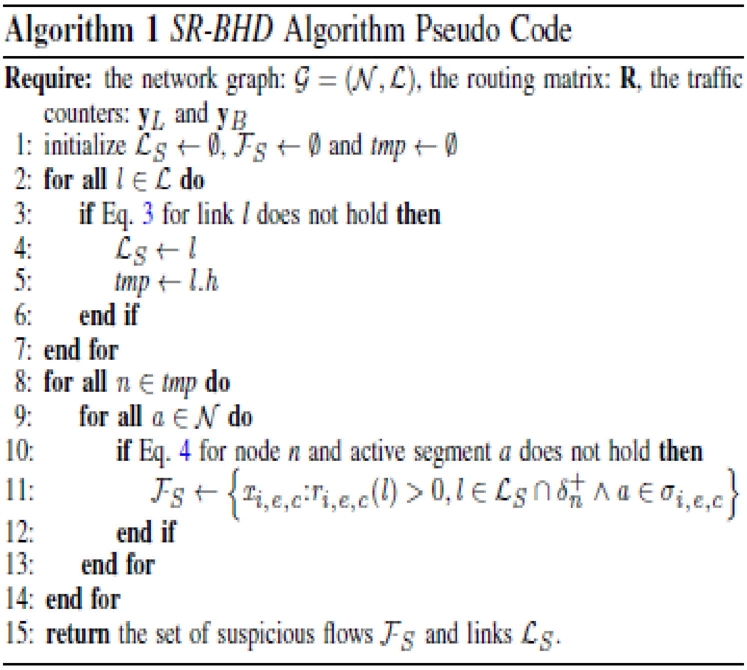


Fig. 1. Scheme of the *SR-BHD* monitoring framework.

The SR-BHD block outputs a list of link/flow pairs suspected of being affected by an SR Black Hole. To enhance precision, a refinement step can be executed using the SR-BHD+ block. Implementation of SR-BHD+ necessitates an additional set of traffic counters, specifically the POL ones. Notably, these counters may not always be accessible in SR-capable nodes, rendering the refinement conducted by SR-BHD+ an optional process.

4.3 Algorithm Explanation



Algorithm 1 *SR-BHD* Algorithm Pseudo Code

```
Require: the network graph: $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, the routing matrix: $\mathbf{R}$, the traffic
    counters: $\mathbf{y}_L$ and $\mathbf{y}_B$
 1: initialize $\mathcal{L}_S \leftarrow \emptyset$, $\mathcal{F}_S \leftarrow \emptyset$ and $tmp \leftarrow \emptyset$
 2: for all $l \in \mathcal{L}$ do
 3:     if Eq. 3 for link $l$ does not hold then
 4:         $\mathcal{L}_S \leftarrow l$
 5:         $tmp \leftarrow l.h$
 6:     end if
 7: end for
 8: for all $n \in tmp$ do
 9:     for all $a \in \mathcal{N}$ do
10:         if Eq. 4 for node $n$ and active segment $a$ does not hold then
11:             $\mathcal{F}_S \leftarrow \left\{x_{i,e,c}{:}r_{i,e,c}(l) > 0, l \in \mathcal{L}_S \cap \delta_n^+ \wedge a \in \sigma_{i,e,c}\right\}$
12:         end if
13:     end for
14: end for
15: return the set of suspicious flows $\mathcal{F}_S$ and links $\mathcal{L}_S$.
```

The SR-BHD block generates a list of link/flow pairs suspected to be influenced by an SR Black Hole. To enhance precision, a refinement step can be taken through the SR-BHD+ block. The execution of SR-BHD+ relies on an additional set of traffic counters known as POL counters. However, since these counters may not always be accessible in SR-capable nodes, the application of SR-BHD+ for refinement is optional. The algorithm involves two primary sets: i) L_S and F_S, representing suspicious links and flows potentially associated with a black hole, and ii) the set tmp storing the head nodes of suspicious links. The validity of Eq. (3) is examined for each link (lines 2–7). If Eq. (3) is not satisfied, the current link is marked as suspicious, and its head node is added to the tmp set (lines 4–5). After identifying the potential SR Black Hole location, SR-BHD detects affected SR flows (lines 8–14). For each node in tmp, the flow conservation law for all active segments is verified (line 10). If the condition is not met for active segment 'a,' the set of suspicious flows (F_S) is updated (line 11), encompassing all SR flows traversing the suspicious link 'l' with 'a' as the active segment. The output of SR-BHD is the list of suspicious flows and links (line 15), mitigating false positive alarms.

The margin's role is to bolster SR-BHD against packet loss sources beyond the targeted SR Black Hole. Considering link 'l,' the neural network's layer comprises two nodes: one for the link count value and the other for its capacity. The output layer features a single node reporting the targeted amount of traffic loss, serving as the margin value. It's essential to note that the presented NN structure is illustrative, and the actual structure may vary considerably in a practical setting, as detailed in Section VI.

5. EXPERIMENTAL DEMONSTRATION OF POSSIBLE EXISTENCE OF SR BLACK HOLES

The goal of this section is twofold: i) to experimentally demonstrate the existence of possible *SR Black Holes* in an SRv6 networks, and ii) to show that active probing based tools are not trustworthy for detecting such a type of failures.

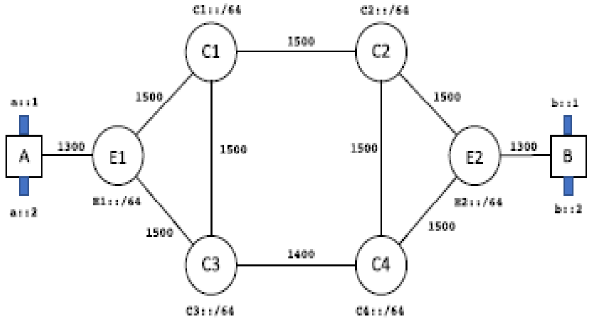


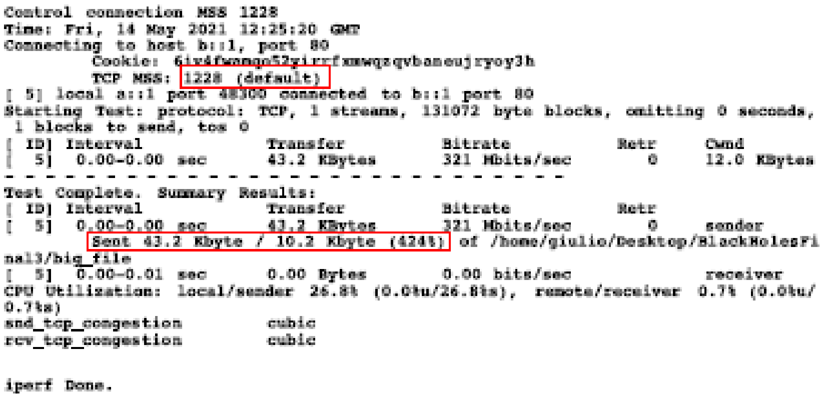
Fig. 2. Reference scenario.

TABLE I
MAIN FEATURES OF THE POLICIES CONFIGURED IN THE EMULATED NETWORK SCENARIO

Policy Name	Scope	Head end node	Segment List
pol_1	high reliability	E1	C1,C2,E2
pol_2	best effort	E1	C3,C4,E2
pol_3	link bypass	C1	C3,C4

The experiment is carried out on an emulated network utilizing virtual routers that support SRv6 as the data plane technology. The Vector Packet Processor (VPP) [24] with SRv6 plugin is employed for this purpose. The experimental topology, depicted in Fig. 2, comprises a SRv6 domain with 6 nodes, identified by locators as shown in the figure. Hosts A and B are connected through this SRv6 domain. Access link MTUs are set at 1300 Bytes, while internal link MTUs are 1500 Bytes, except for the link between nodes C3 and C4, which has a lower MTU of 1400 Bytes. It's essential to note that the chosen MTU configuration aligns with the recommendation in [25], advising higher MTUs for internal links within the SR domain compared to external links

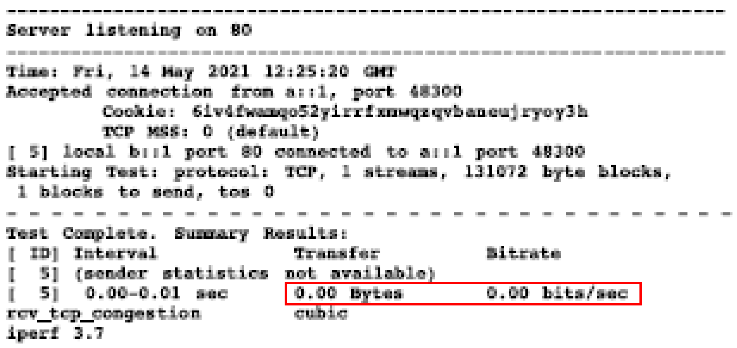
Two SR policies, pol_1 and pol_2, are configured in the head-end node E1, detailed in Tab. I. In this scenario, these policies establish varying levels of reliability for network paths, with pol_1 designed for high-reliability traffic and pol_2 for regular traffic. Pol_3, detailed in Tab. I and configured in node C1, exemplifies re-routing in case of a link failure (C1-C2), providing an alternative path (C1-C3-C4).



```
Control connection MSS 1228
Time: Fri, 14 May 2021 12:25:20 GMT
Connecting to host b::1, port 80
      Cookie: 6ix4fwanqo52yirrfxmwqzqvbaneujryoy3h
      TCP MSS: 1228 (default)
[  5] local a::1 port 48300 connected to b::1 port 80
Starting Test: protocol: TCP, 1 streams, 131072 byte blocks, omitting 0 seconds,
 1 blocks to send, tos 0
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-0.00   sec  43.2 KBytes  321 Mbits/sec    0    12.0 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
Test Complete. Summary Results:
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-0.00   sec  43.2 KBytes  321 Mbits/sec    0       sender
      Sent 43.2 Kbyte / 10.2 Kbyte (424%) of /home/giulio/Desktop/BlackHolesFi
nal3/big_file
[  5]   0.00-0.01   sec  0.00 Bytes   0.00 bits/sec            receiver
CPU Utilization: local/sender 26.8% (0.0%u/26.8%s), remote/receiver 0.7% (0.0%u/
0.7%s)
snd_tcp_congestion       cubic
rcv_tcp_congestion       cubic


iperf Done.
```

Fig 3. Snapshot of the Iperf Window on Client node.



```
-----------------------------------------------------------
Server listening on 80
-----------------------------------------------------------
Time: Fri, 14 May 2021 12:25:20 GMT
Accepted connection from a::1, port 48300
        Cookie: 6iv4fwamgo52yirrfxmwqzqvbaneujryoy3h
        TCP MSS: 0 (default)
[ 5] local b::1 port 80 connected to a::1 port 48300
Starting Test: protocol: TCP, 1 streams, 131072 byte blocks,
 1 blocks to send, tos 0
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Test Complete. Summary Results:
[ ID] Interval           Transfer         Bitrate
[  5] (sender statistics not available)
[  5]  0.00-0.01 sec     0.00 Bytes       0.00 bits/sec
rcv_tcp_congestion       cubic
iperf 3.7
```

To investigate the presence of an SR Black Hole, two traffic flows are initiated between hosts A and B. The first is a TCP connection requiring reliable transfer, steered by a traffic classifier in node E1 following the path specified by pol_1. The second consists of ICMP Echo Requests sent along the regular path using pol_2. The experiment is conducted twice: the first run without link failure events, and the second with the link between nodes C1 and C2 intentionally failed. As anticipated, no black hole is observed in the first case. The MTU of the links along the TCP connection path is 1300 Bytes, while the SRv6 overhead is 96 Bytes. Since the MTU of internal links crossed by the TCP connection is 1500 Bytes, no MTU violation occurs, and the flow reaches the destination without issues.

A distinct scenario unfolds when policy pol_3 is utilized to reroute traffic due to the link failure between nodes C1 and C2. In this case, the SRv6 processing overhead is 176 Bytes. Although the smallest MTU among links along the TCP connection path remains at 1300 Bytes, the bottleneck MTU inside the SRv6 domain is now 1400 Bytes (due to the detour via link C3-C4). Consequently, packets sent through link C3-C4 have an overall length of 1476 Bytes, exceeding the MTU and leading to silent packet drops. Fig.3 illustrates the Iperf window at the TCP connection's client side, displaying the traffic volume and the considered MSS. Fig. 4 demonstrates that the server does not receive the traffic due to the SR Black Hole. It's noteworthy that the TCP connection is successfully established, given the small size of messages during the three-way handshake. Additionally, ICMP traffic crossing the link C3-C4 is correctly delivered to the destination.

5.1 Applying Active Probing Tools to Detect the SR Black Hole

The experimental validation of the existence of the SR Black Hole was conducted within a non-ollaborative network environment, where nodes do not transmit ICMP PTB messages. In this subsection, we assess the efficacy of the Scamper tool [5] in accurately detecting the path MTU. Scamper has been specifically designed for detecting MTU-related black holes in non-collaborative networks. It employs a traceroute-like mechanism to achieve two primary objectives: i) identify the location of the black hole occurrence, and ii) determine the bottleneck MTU. The mechanism involves sending UDP probes along the path between the source and destination nodes. Initially, small UDP probes assess the reachability of the destination. Subsequently, a PMTUD process is initiated by sending probes with increasing sizes. To handle unresponsive nodes, Scamper utilizes a timeout mechanism, assuming packet drop due to MTU constraint violation if no response is received within the timer expiration. After two consecutive timeout events, Scamper determines the maximum supported packet size, leveraging a table of well-known MTU values to expedite the process. Once the path MTU is identified, a traceroute-like procedure is employed to locate the hop containing the bottleneck link. Probes of a size larger than the path MTU are sent, and the Time To Live (or Hop Limit in IPv6) is incrementally increased. The absence of ICMP Time Exceeded messages indicates the identification of the bottleneck link connecting the last responding node with its next hop.

Research Through Innovation

An initial observation from using Scamper in our scenario reveals that the Hop Limit-based mechanism for bottleneck link detection becomes ineffective when the ingress node of the SRv6 domain performs IP in IP encapsulation. Following encapsulation, the Hop Limit of the outer header is set to the default value, which is larger than that in the inner header. The obtained results using Scamper in the network scenario illustrated in Fig. 1 are detailed next. The tested MTU values at each iteration of Scamper's execution are presented in Fig. 5, with the last iteration value being reported as the output to the source host, which generates packets accordingly.

Two experiments are conducted. In the first, the network configuration remains unchanged from the previous experiment where the link between nodes C1 and C2 fails. As previously discussed, the TCP traffic in this scenario undergoes double encapsulation with an overall SRv6 processing overhead of 176 Bytes. Despite the link between the source node and E1 having the smallest MTU in the overall TCP traffic path, the SRv6 overhead imposes a limit on the packet size determined by the link between nodes C3 and C4. Consequently, the maximum allowed size for packets injected into the SRv6 domain is 1224 Bytes. Unfortunately, due to the policy-based routing in SR, Scamper probes follow a different path than the data traffic, handled through policy pol_2.

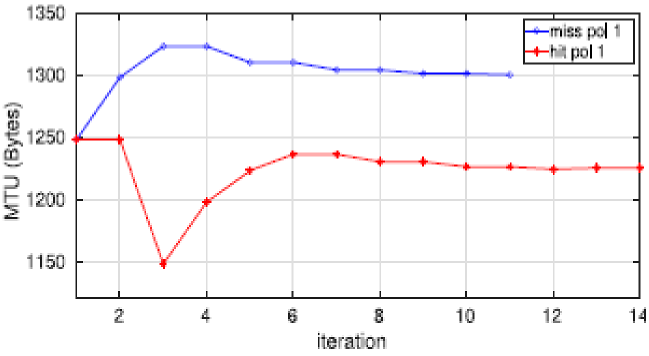


Fig. 5. MTU assessment procedure followed by Scamper

The resulting MTU size obtained by Scamper in this case, represented by the blue curve in Fig. 5, is 1300 Bytes. As a consequence, the TCP source generates packets larger than the supported size, leading to the creation of a black hole. This test confirms that active tools fail in detecting the SR Black Hole due to policy-based routing in SR. To further underscore this point, a second experiment is performed by incorporating a classification rule in node E1, directing Scamper probes to follow the same path as TCP traffic. The outcome of this Scamper execution is represented by the red line in Fig. 5. As expected, under this configuration, Scamper correctly determines that the maximum supported MTU is 1224 Bytes. Injecting packets of such length into the SR domain, enforcing pol_1 and pol_3 (resulting in an SR-related overhead of 176 Bytes), leads to a maximum size of 1400 Bytes, matching the MTU of the bottleneck link. In summary, these experiments demonstrate that an active probing mechanism can fail to detect an SR Black Hole, resulting in false negatives, which are unacceptable for a detection tool.

6. PERFORMANCE EVALUATION

Herein, we present a simulation-based analysis assessing the performance of SR-BHD. We consider three real networks, namely Abilene (N = 12, L = 30), Geant (N = 22, L = 72), and Germany (N = 50, L = 176), sourced from [28]. Real traffic matrices for these networks are utilized. Traffic demand routing adheres to the shortest path policy, where segment lists contain a single SID linked to the destination node. Each traffic flow from the traffic matrix is directed by an SR policy installed at the source node. Capacity planning entails routing the peak traffic matrix according to the shortest path policy, and each link is assigned a capacity randomly selected (uniform distribution) to achieve utilization between 50% and 99%. Simulated packet loss due to congestion on link l, denoted as Ql, is determined by dropping a portion of the traffic flowing over the link. Ql follows a Normal distribution, calculated using Eq. (9), where αC is the congestion amplification factor, Cl is the link capacity, and yIL(l) is the ideal traffic flow over link l without packet loss.

$$\text{loss\_AE} = \frac{1}{N}\sum_{i=0}^{N}(X_i - Y_i)^2 \qquad (2)$$

To establish the margin, a fully connected feed-forward neural network (NN) is trained for each network and each αC value. The NN includes an input layer with two neurons representing link traffic and capacity, a hidden layer with 5 neurons using ReLu activation, and an output layer with a single neuron representing the margin's assessed value. The Mean Squared Error minimization policy guides the learning process. For training sets, 24 simulations (one Traffic Matrix per hour from [28]) are run without considering SR Black Holes, collecting tuples <yL(l), Cl, Ql> for each link.

6.1 Precision and Recall parameters are defined as follows:

We assess the precision and recall of the algorithms in detecting SR Black Holes, assuming the presence of a single black hole in all possible events for each link and flow. A congestion amplification factor of αC = 10^-3 is used. SR-BHD achieves 100% recall, but SR-BHD+ exhibits occasional detection failures for Geant and Germany networks due to estimation errors in traffic loss evaluation (resulting in 0 recall). Precision improvements with SR-BHD+ are evident, particularly in larger networks. For Abilene, SR-BHD+ achieves a mean precision of 97.22 compared to SR-BHD's 7.71. In larger networks, SR-BHD+ significantly outperforms SR-BHD, detecting at most 3 flows in more than 90% of cases (33% precision), making it a valuable troubleshooting tool. We also evaluate the impact of multiple SR Black Holes on SR-BHD+ performance for the Germany network, varying from 1 to 5, randomly selected in terms of location (link/flow pair). Additionally, the Average Run Length (ARL) for different flows considering various αC values is depicted for the Abilene network in Fig. 7(a). Figs. 7(b) and 7(c) illustrate the average recall and precision achieved by SR-BHD+ as αC varies, demonstrating that performance decreases with increasing αC, and tuning the smin tolerance parameter enhances robustness against regular packet loss events, achieving 100% recall even for low ARL values

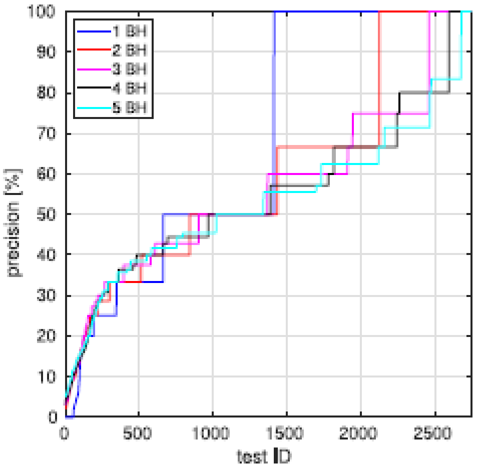


Fig. 6. CDF of the precision of *SR-BHD+* in presence of multiple *SR Black Holes.*

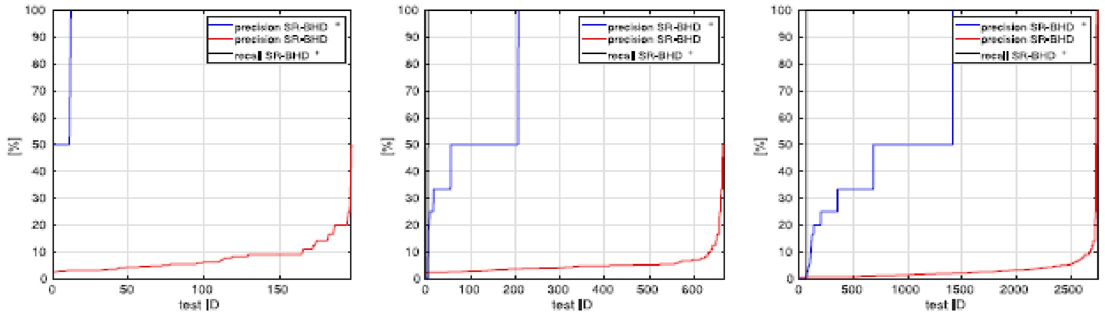
7.SR-BHD PROTOTYPE

In this section a description of the experimental prototype of *SR-BHD* that we have realized is provided. The goal of the prototype is to prove the effectiveness of the proposed approach. Three different aspects are discussed: i) the description of the design implementation of *SR-BHD* prototype, ii) the methodology adopted to run the tests, iii) the results obtained in the different types of performed tests.

7.1Prototype Description

The proposed SR-BHD algorithm is intended for integration into a centralized monitoring system within the SDN paradigm. In the current prototype, this integration is achieved through a set of scripts activated by a timer. Conceptually, the prototype comprises two primary components: i) the Stats Collector module, responsible for gathering SRTCs at each node, and ii) the Black Hole Detection module, which implements the SR-BHD logic on each link to identify potential black holes.

International Research Journal



(a) Abilene.

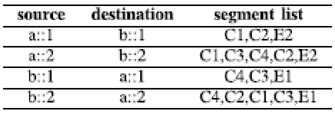
(b) Geant.

(c) Germany.

Fig. 7. Precision and Recall analysis of the *SR-BHD* and *SR-BHD+* in different networks.

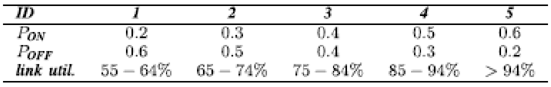
The limitation associated with lPSID is a crucial distinction impacting the implementation of the SR-BHD framework. It relies on statistics derived from PSID counters collected at each node. Conversely, in VPP, these SRTC counters are maintained only at nodes executing SR-related operations (e.g., a router tracking packets subjected to the END operation). To complete our prototype, we have implemented a strict source routing policy, meaning that the segment list for each flow contains an explicit set of intermediate nodes to traverse. As a future endeavor, we aim to extend the application of PSID counters to every node, regardless of whether they perform SR functions. attributes, such as source, destination, and path details, of the four traffic flows in the scenario are outlined in Table III. These flows are generated using the VPP traffic generator, allowing the creation of constant bit rate UDP flows with configurable packet size and data rate, all set at 100 Kbps. Each test selects a target flow and the link where the black hole is induced. Creating the black hole and affecting the target flow involve two actions: generating larger-sized packets for the target flow compared to others and reducing the MTU of the designated link. Subsequently, all traffic flows commence simultaneously and conclude 120 seconds later. The Stats Collector module triggers every 60 seconds, leading to two executions of the Black Hole Detection module in each run.

TABLE III
MAIN FEATURES OF THE TRAFFIC FLOWS INCLUDED IN THE EMULATED ENVIRONMENT



source	destination	segment list
a::1	b::1	C1,C2,E2
a::2	b::2	C1,C3,C4,C2,E2
b::1	a::1	C4,C3,E1
b::2	a::2	C4,C2,C1,C3,E1

TABLE IV
DIFFERENT CONFIGURATIONS OF THE DTMP SOURCE



ID	*1*	*2*	*3*	*4*	*5*
P_{ON}	0.2	0.3	0.4	0.5	0.6
P_{OFF}	0.6	0.5	0.4	0.3	0.2
link util.	$55-64\%$	$65-74\%$	$75-84\%$	$85-94\%$	$>94\%$

To introduce controlled congestion events, varying levels of background traffic are generated. Specifically, a source-destination traffic flow is initiated at the end nodes of each oriented link. The source node is modeled as a Discrete Time Markov Process (DTMP) with ON and OFF states, characterized by transition probabilities pON and pOFF. In the ON state, it generates 1000 Bytes packets at a constant bit rate of Ra. In the OFF state, it remains inactive with probability pOFF or starts sending traffic with probability 1−pOFF in the subsequent time slot. The duration of a time slot is set to 100 milliseconds, Ra is 15 Mbps (exceeding link capacity for congestion creation), and five combinations of transition probabilities (detailed in Table IV) are considered across different tests.

7.2 Experimental Evaluation

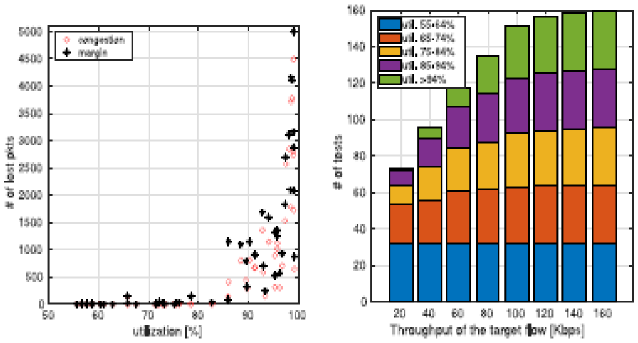


Fig. 8. Performance evaluation of the developed prototype carried out over an emulated environment.

In the following, we outline the Neural Network (NN) structure used in the experimental evaluation of the prototype and the corresponding training procedure. Our choice is a feed-forward NN with two hidden layers featuring 32 and 16 neurons, respectively, employing ReLu as the activation function and trained to minimize the Mean Squared Error (MSE). Constructing the training set involved several steps: i) selection of a set of transition probabilities for the DTMP sources, ii) execution of a 60-second experiment, collecting link utilization every 6 seconds, and iii) determination of the final amount of lost packets. This process was repeated 100 times for each of the 5 combinations of transition probabilities, yielding a training set of 100 × 5 × 16 = 8000 observations (16 being the number of links). Each observation consists of 10 measurements of link utilization and a label indicating the number of lost packets. Therefore, the NN features an input layer with 10 neurons and an output layer with 1 neuron.

The final assessment, depicted in Figure 7(b), pertains to the sensitivity of SR-BHD in detecting the SR flow affected by a black hole (the target flow) as a function of flow throughput. For each test, a combination of transition probabilities for the DTMPs (indicating background traffic level) is selected, and the throughput of the target flow is set. The target flow and the link experiencing the black hole are then chosen. In total, 16 combinations are tested, with SR-BHD applied twice in each test (once every 60 seconds), resulting in 32 observations (referred to as tests in the y-axis of Figure 7(b)) for each pair of background traffic level/target flow value. Figure 7(b) illustrates, for different levels of background traffic, the number of tests where the black hole is correctly detected as a function of the size of the target flow. The analysis reveals that for low background traffic values, SR-BHD can successfully detect even very small flows (20 Kbps over a 10 Mbps link) lost in the black hole. In these cases, the margin approximates the amount of packet loss due to congestion effectively. However, as the background traffic level increases, small flows are not consistently detected. Intriguingly, when the rate of the target flow equals 160 Kbps (approximately 1.6% of the link bandwidth), the black hole is consistently identified. This underscores the notable capability of SR-BHD to detect black holes even when the target flow is significantly lower than the link bandwidth.

CONCLUSION

In this paper, we have tackled the issue of logical failures in Segment Routing networks stemming from the violation of the MTU constraint, referred to as SR Black Holes. Our experimental findings confirm two crucial points: i) SR Black Holes can manifest in misconfigured SR domains, and ii) traditional detection methods relying on active probing prove ineffective in identifying these failures. To address this, we introduced a passive monitoring framework called SR-BHD, leveraging specific SR Traffic Counters. This framework adeptly detects black holes, even in scenarios with multiple sources of packet loss like congestion and transmission errors. SR-BHD enforces the flow conservation principle at various network levels, allowing tolerance for additional packet loss sources through the incorporation of a safety margin. To fine-tune the tolerance level, we presented an estimation framework employing Neural Networks to determine the appropriate margin value. Our performance evaluation illustrates that by appropriately adjusting the tolerance interval of SR-BHD+, a favorable balance can be achieved between algorithm precision and resilience in critical situations. Subsequently, a prototype of SR-BHD was implemented and tested within an emulated environment. The experimental results confirm the efficacy of the proposed framework in successfully detecting the presence of SR Black Holes.

REFERENCES

[1] P. L. Ventre *et al.*, "Segment routing: A comprehensive survey of research activities, standardization efforts and implementation results," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 182–221, 1st Quart., 2022.
[2] C. Filsfils, P. Camarillo, J. Leddy, D. Voyer, S. Matsushima, and Z. Li, "Segment routing over IPv6 (SRv6) network programming," Internet Eng. Task Force, RFC 8986, Feb. 2021. [Online]. Available: https://rfceditor. org/rfc/rfc8986.txt
[3] C. Filsfils, D. Dukes, S. Previdi, J. Leddy, S. Matsushima, and D. Voyer, "IPv6 segment routing header (SRH)," Internet Eng. Task Force, RFC 8754, Mar. 2020. [Online]. Available: https://rfceditor. org/rfc/rfc8754.txt
[4] M. de Boer, J. Bosma, and W. Toorop, *Discovering Path MTU Black Holes in the Internet Using RIPE Atlas*, Univ. Amsterdam, Amsterdam, The Netherlands, 2012.
[5] M. Luckie, K. Cho, and B. Owens, "Inferring and debugging path MTU discovery failures," in *Proc. 5th ACM SIGCOMM Conf. Internet Meas.*, 2005, pp. 1–6.
[6] A. Tulumello *et al.*, "Micro SIDs: A solution for efficient representation of segment IDs in SRv6 networks," in *Proc. 16th Int. Conf. Netw. Service Manage. (CNSM)*, 2020, pp. 1–10.
[7] S. Litkowski, A. Bashandy, C. Filsfils, P. Francois, B. Decraene, and D. Voyer, "Topology independent fast reroute using segment routing," Internet-Draft draft-ietf-rtgwg-segment routing-ti-lfa-07, Internet Eng. Task Force, Fremont, CA, USA, Jun. 2021. [Online].
Available: https://datatracker.ietf.org/doc/html/draft-ietf-rtgwg-segmentrouting- ti-lfa-07
[8] Y. Desmouceaux, M. Townsley, and T. H. Clausen, "Zero-loss virtual machine migration with IPv6 segment routing," in *Proc. 14th Int. Conf. Netw. Service Manage. (CNSM)*, 2018, pp. 420–425.
[9] M. Polverini, A. Cianfrani, and M. Listanti, "Snoop through traffic counters to detect black holes in segment routing networks," in *Proc. Int. Conf. Commun. Netw.*, 2020, pp. 337–350.
[10] C. Filsfils, Z. Ali, M. Horneffer, D. Voyer, M. Durrani, and R. Raszuk, "Segment routing traffic accounting counters," Internet-Draft draftfilsfils- spring-sr-traffic-counters-01, Internet Eng. Task Force, Fremont, CA, USA, Apr. 2021. [Online]. Available: https://datatracker.ietf.org/ doc/html/draft-filsfils-spring-sr-traffic-counters-01
[11] R. R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren, "Detection and localization of network black holes," in *Proc. 26th IEEE Int. Conf. Comput. Commun.*, 2007, pp. 2180–2188.
[12] L. Fang, A. Atlas, F. Chiussi, K. Kompella, and G. Swallow, "LDP failure detection and recovery," *IEEE Commun. Mag.*, vol. 42, no. 10, pp. 117–123, Oct. 2004.
[13] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson, "Netalyzr: Illuminating the edge network," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas.*, 2010, pp. 246–259.
[14] R. Staff, "Ripe Atlas: A global Internet measurement network," *Internet Protocol J.*, vol. 18, no. 3, pp. 1–31, 2015.
[15] M. Mardani and G. B. Giannakis, "Estimating traffic and anomaly maps via network tomography," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1533–1547, Jun. 2016.
[16] M. Thottan and C. Ji, "Anomaly detection in IP networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2191–2204, Aug. 2003.
[17] R. Geib, C. Filsfils, C. Pignataro, and N. K. Nainar, "A scalable and topology-aware MPLS data-plane monitoring system," Internet Eng. Task Force, RFC 8403, Jul. 2018. [Online]. Available: https://rfceditor. org/rfc/rfc8403.txt

[18] N. Kumar, C. Pignataro, G. Swallow, N. Akiya, S. Kini, and M. Chen, "Label switched path (LSP) ping/traceroute for segment routing (SR) IGP-prefix and IGP-adjacency segment identifiers (SIDs) with MPLS data planes," Internet Eng. Task Force, RFC 8287, Dec. 2017.

[19] Z. Ali, K. Talaulikar, C. Filsfils, N. K. Nainar, and C. Pignataro, "Bidirectional forwarding detection (BFD) for segment routing policies for traffic engineering," Internet-Draft draft-ali-spring-bfd-sr-policy- 07, Internet Eng. Task Force, Fremont, CA, USA, May 2021.
[Online]. Available: https://datatracker.ietf.org/doc/html/draft-ali-springbfd- sr-policy-07

[20] M. Polverini, A. Cianfrani, and M. Listanti, "A theoretical framework for network monitoring exploiting segment routing counters," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 3, pp. 1924–1940, Sep. 2020.

[21] P. Loreti *et al.*, "SRv6-PM: A cloud-native architecture for performance monitoring of SRv6 networks," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 611–626, Mar. 2021.

[22] F. Aubry, D. Lebrun, S. Vissicchio, M. T. Khong, Y. Deville, and O. Bonaventure, "SCMon: Leveraging segment routing to improve network monitoring," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.

[23] C. Filsfils, S. Previdi, L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir, "Segment routing architecture," Internet Eng. Task Force, RFC 8402, Jul. 2018.

[24] "CSIT REPORT: The Fast Data I/O Project (FD.io) Continuous System Integration and Testing (CSIT) Project Report for CSIT Master System Testing of VPP-18.04 Release." Aug. 2022. [Online]. Available: https://docs.fd.io/csit/master/report/_static/archive/csit_master.pdf

International Research Journal
IJNRD
Research Through Innovation