Snoop Through Traffic Counters to Detect Black Holes in Segment Routing Networks

Marco Polverini(✉), Antonio Cianfrani, and Marco Listanti

University of Roma "Sapienza", Via Eudossiana 18, 00184 Roma, Italy
{marco.polverini,antonio.cianfrani,marco.listanti}@uniroma1.it

Abstract. The new Segment Routing paradigm provides network operator the possibility of highly increasing network performance exploiting advanced Traffic Engineering features and novel network programability functions. Anyway, as any new solutions, SRv6 has a side effect: the introduction of unknown service disruption events. In this work we focus on packet lost events due to the incorrect computation of the Maximum Transmission Unit (MTU) value of an end-to-end path in an SRv6 network. This event, referred to as *MTU dependent SR Black Hole*, cannot be detected by known monitoring solutions based on active probing: the reason is that in SRv6 probe packets and user data can experience different network behaviors. In this work we propose a passive monitoring solution able to exploit the SRv6 Traffic Counters to detect links where packets are lost due to MTU issues. The performance evaluation shows that the algorithm proposed is able to identify the link affected by the blackhole with a precision equal to 100%; moreover, the flow causing the blackhole cannot be detected with the same precision, but it is possible to identify a restricted set of flows, referred to as suspected flows, containing the target one.

Keywords: Segment routing · Network failures · Black holes · Interface counters

1 Introduction

The introduction of a new technology in the networking area leads to new or optimized functions with respect to existing one, and it translates in improved performance for users. As a side effect, it also represents a possible vector for the creation of new flaws and bugs. These kind of effects can be considered as technology dependent failures, which can determine a service disruption. Due to their nature, technology dependent failures are usually transparent to existing monitoring solutions. In literature, they have been identified with the term *network black holes*. As an example of technology dependent black hole, in [9] the service disruption due to LSP failure in MPLS network is reported.

Recently, Segment Routing Architecture [8] has been introduced as a practical realization of the source routing paradigm. SR allows for a better usage

H. Gao et al. (Eds.): ChinaCom 2020, LNICST 352, pp. 337–350, 2021.
https://doi.org/10.1007/978-3-030-67720-6_23

of the bandwidth, increasing the Traffic Engineering capabilities of traditional networks, completely reusing existing data plane functionalities. Currently, two implementations of SR, named SR-MPLS and SRv6 are available [19], using MPLS and IPv6 data planes, respectively. This work is focused on SRv6.

SR comes with a set of tools for Operation and Maintenance (OAM) [11], which allow to troubleshoot the network and to identify possible anomalies (such as failures and misconfigurations) in the current configuration. These tools extend the existing ones, by providing further capabilities and enhancement to the performance. As an example, in [1] SR is exploited to realize a link failure detection infrastructure, named SCMon, able to check the status of a layer 3 link realized as a bundle of layer 2 connections.

One of the main innovation introduced by SR is that the flow states are moved from the routers to the packets headers. This enables SR networks to highly scale, at the cost of increasing the packet length (i.e., overhead); in fact, the length of a packet can increase during the journey from the source up to the destination nodes, due to the application of specific SR information (Segment List, TLV, etc.). Depending on the complexity of the flow states, the packet size can eventually become greater than the minimum MTU, causing drop phenomena, which in turns determine QoS degradation and, in some cases, service disruption. Unfortunately, existing OAM tools are not able to detect this type of failure. We refer to it as *MTU dependent SR Black Hole*.

As we will explained in Sect. 3, classical methods fail in detecting this type of failure since they rely on an active approach, i.e., they are probe based techniques. Probe messages are sent along an end to end path to assess the Path MTU, or to detect a possible black hole in the Path MTU Detection (PMTUD) procedure. Due to the fact that SR forwarding is based on the concept of policy routing, it is not guaranteed that the sent probes follow the same end to end path of the flow hit by the *MTU dependent SR Black Hole*, or that they are processed in the exact same way. On the contrary, in this paper we present a novel algorithm, called *Segment Routing Black Holes Hound* ($SRBH^2$), that uses a passive approach to diagnose the current network status and possibly detect *MTU dependent SR Black Holes*. In particular, $SRBH^2$ exploits the information gathered by the SR Traffic Counters [6], a new OAM tool available in all SR capable nodes. Since SR Traffic Counters collect statistics directly on the data traffic, if a traffic flow falls into a black hole, then this event is somehow coded into the traffic counters. Then $SRBH^2$ represents a method to search and extract this information from the Traffic Counters.

The proposed solution is a dedicated solution to identify *MTU dependent SR Black Holes* and, in a real scenario, will run in parallel with different solutions able to detect different packet losses causes, such as physical failures and traffic congestion. It is important to remark that $SRBH^2$ is based on data traffic information, i.e. SRv6 traffic counters, already available at the OAM module, with no need of generating additional traffic, such as packet probes and traffic mirroring.

To summarize, the main contributions of this work are:

– the identification of a new type of black hole, related to the deployment of SRv6 technology;
– the characterization of classical OAM tools weakness in detecting such types of failures;
– the definition of a low complexity algorithm exploiting SR Taffic Counters to detect *MTU dependent SR Black Holes*.

The rest of the paper is organized as follows: In Sect. 2 we provide a background on SRv6, in Sect. 3 we explain in detail the *MTU dependent SR Black Hole* and show an example of such a failure, in Sect. 4 it is presented the $SRBH^2$ algorithm able to detect them. Sect. 5 provide a performance evaluation, and finally Sect. 6 concludes the paper.

2 Segment Routing Background

Segment Routing (SR) [8] is a novel network paradigm based on source routing, i.e., the source node decides the path that each packet has to go through. The end to end paths are encoded as an ordered list of instructions, also referred to as segments. Thus the full list of segments is named Segment List (SL). Each segment can represent any topological (e.g., send the packet to a given node) or service based (e.g., duplicate the packet) instruction. Segments are expressed as labels, named Segment IDentifiers or SIDs. If the underlay data plane is based on IPv6, a SID has a is represented by an IPv6 address.

In SRv6, SR related information are included in a new defined extension header, named Segment Routing Header (SRH) [7], which is added to each incoming packet in two different ways. The first option is the *insert mode*, i.e., the SRH is included directly on the original IPv6 packet header. The second method is the *encap mode*, i.e., the original packet is encapsulated in an outer IPv6 packet, having the SRH. SL is a special field of the SRH, and is represented by a list of IPv6 addresses, having the meaning of SIDs.

The SRv6 packet forwarding works as follows. When the border router receives a packet, it has to steer it over a specific SL. This last is chosen according to a given *SR Policy*. Specifically, each node of the SR domain maintains an *SR Policy* table, where all the policies available at that node are stored. After a packet has been processed according to a matched *SR Policy*, its most external IPv6 header is extended by the inclusion of an SRH. This last containing the SL and a pointer named *segment left* (sl), which indicates what is the active segment (*actsgm*). It represents the current instruction to be applied on the packet. In particular, the *actsgm* is copied also in the destination address field of the outer IPv6 header, while the SID of the node that has performed the encapsulation is used as source address. As a consequence, the path followed by the packet from a SID to the next one is the path computed by IGP routing protocols (usually the shortest one). Transit routers forward incoming packet by inspecting the IPv6 destination address of the outer header. Once the node having the same SID of the *actsgm* is reached, the so called *MyLocalSID* table is inspected, in order to verify what function has to be applied. A common function is the END one, which implies that the *actsgm* has to be updated, thus the *sl* is decremented and the new *actsgm* is copied into the IPv6 destination field. Finally, before the packet leaves the SR domain, the SRH has to be removed.

3 Introducing the *MTU Dependent SR Black Hole*

In this section we provide a discussion about the main causes of the *MTU dependent SR Black Hole*. Then, we show an example of such a black hole, and we provide a discussion about the weakness of existing monitoring tools when an *MTU dependent SR Black Hole* occurs. Finally, guidelines for the definition of an algorithm able to identify *MTU dependent SR Black Hole* are identified.

3.1 MTU Related Issues in IPv6

We recall that SRv6 reuses the IPv6 dataplane, by defining a new extension header called SRH. As known, IPv6 [4] discourages the use of fragmentation, limiting this operation to be performed only at the source node. Specifically, the source node performs the Path MTU Discovery technique [15] (PMTUD) to acquire the knowledge of the maximum packet size that do not cause violations of the MTU constraint. This procedure is based on the following steps: i) the source node starts emitting packets having a length equal to the MTU of the output link (the one used by the source to reach the next hop); ii) each router along the path checks whether the size of the received packet meets the requirement on the maximum allowed size, with respect to the MTU on the link toward the next hop; if MTU constraints are satisfied, then the packet is normally forwarded, otherwise, an *ICMP Packet Too Big* (PTB) message is sent back to the source node, which notify the maximum allowed size. The source node is now able to fragment the packet taking into account the MTU constraint.

One of the main weakness of this mechanism is the fact that it assumes that all intermediate routers participate to the discovery process. PMTUD failures are a well studied problem in IPv6 networks. Some studies [3,13] have shown how different failure modes are possible. Major causes are due to routers that do not send PTB messages, either due to configuration issues or because of MTU mismatch. In other cases, firewalls might discard fragmented packets, creating a service disruption. In particular [3] has pointed out that a common policy adopted by service provider is represented by TCP Maximum Segment Size (MSS) clamping. This technique forces servers to advertise a MSS smaller than the maximum allowed one, so that to force the clients to generate small packets, eventually avoiding problems related to MTU constraint violation. Clearly, UDP flows cannot exploit MSS clamping, increasing the chances of the creation of a black hole.

To overcome these limitations, the so called Packetization Layer Path MTU Discovery [14] (PLPMTUD) is used in conjunction with the regular PMTUD.

PLPMTUD procedure is the same one of PMTUD, but it is performed at the transport layer; in this way no ICMP messages are required. Clearly this mechanism can be performed only between end systems.

In the context of an SRv6 network, the problems related to the packet fragmentation in IPv6 are further enhanced by the following aspects: i) SRv6 policies require the encapsulation of packets, ii) the size of the packets can change during the journey from source to destination, and iii) due to network programming, end to end paths can change frequently and in an unplanned manner. In the next we detail how, each of these SR features, can lead to the creation of an *MTU dependent SR Black Hole*. Considering the first point, when the encap mode is used to apply the SRH on packets, the source address of the external IPv6 header is set equal to the address of the head end node. This last, as specified in [7], is in charge of performing the PMTUD to discover maximum packet size that can be used to reach the end point of the SR policy. The main consequence of this fact is that PLPMTUD cannot be used, since the MTU discovery is not carried out between end systems, forcing the procedure to rely on the correct delivery of ICMP PTB messages.

Due to the fact that the PMTUD could fail, the following recommendation is specified in [7]: *"deploy a greater MTU value within the SR Domain than at the ingress edges"*. Even though this last expedient can often prevent from critical situations in classical IPv6 networks, it is less likely to work in SRv6 environments. In fact, as previously stressed, the size of the packets can change during the journey from source to destination. There are different factors that can cause an increase of the overall packet size, such as the application of a further level of SR encapsulation and the insertion of TLVs. As a consequence, even though a packet meets the MTU constraint when injected in the SR domain, it can eventually grow up to reach a size that overcomes the allowed threshold. To better explain this problem, let us refer with the term *margin* (M) as the difference between the minimum MTU among the links internal to the SR domain and the one on the links at the edge. If during its journey inside the SR domain, the length of a packet grows more than M, then the MTU constraint is violated and an *MTU dependent SR Black Hole* might happens.

A possible countermeasure to this problem could be to limit the number of extra bytes added to each packet, by properly configuring the SR policy to be applied on each traffic flow: knowing the end to end path followed by a packet, it is possible to configure the SR processing applied on it, so that to limit the maximum amount of extra information to add on it. Unfortunately, the set of functions applied on a traffic flow is not static, due to the network programming paradigm, which can make end to end paths change in an unpredictable manner. As an example, in [5] it is described a procedure to provide zero-loss Virtual Machine migration. This process is based on the definition of a special SID (named *forward to local IF present*), which forces a router to decide the next node to visit based on the verification of a condition. Another example of unpredictable path change is represented by the Topology Independent Loop Free Alternate (TI-LFA), which consists in a backup scheme to provide protection against link failures: when a link fails, the head end node react by re-routing the traffic flows over a backup path. This re-routing is performed by means of an SR encapsulation. Then, it is not possible to plan in advance the set of functions applied on a traffic flow and consequently limit the amount of extra information added to each packet.

3.2 An Example of MTU Dependent SR Black Hole

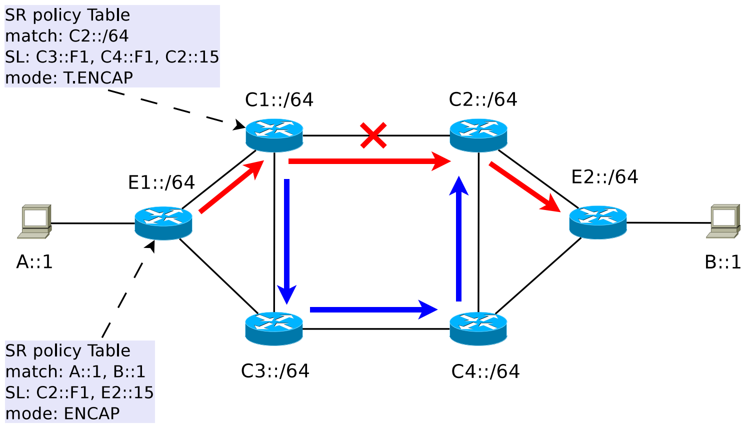


Fig. 1. Example of creation of an *MTU dependent SR Black Hole*. (Color figure online)

In Fig. 1 we present a simple example showing how all the aforementioned events can interact to generate an *MTU dependent SR Black Hole*. Let us assume that, in the scenario reported in Fig. 1 the margin is $M = 100$ Bytes, and that all the links internal to the SR domain have the same MTU. The traffic flow originated by the host A and directed to the host B is steered through an SR policy installed at node $E1$, whose end to end path is shown in red. This SR policy increases the size of each packet of 64 Bytes, thus no fragmentation is needed. In normal conditions, the considered traffic flow is steered over the red path and delivered to the host B. Now, let us assume that the link between nodes $C1$ and $C2$ fails. A backup scheme is implemented at node $C1$ by means of an SR policy, that encapsulates packets that were supposed to be routed over the failed link, with an SRH having a SL of depth 3. The SR policy at node $C1$ adds 80 Bytes of overhead to the incoming packets. Note that the application of this policy is unplanned, since it is used only when a link failure happen. Globally, each packet is increased in size (with respect to the length before being injected in the SR domain) of 144 Bytes, which is a quantity greater than the margin. Consequently the MTU constraint is violated. If ICMP PTM messages are not generated, the head end node of the outer policy ($C1$ in the example) is unaware of this situation and do not fragments the packets. Thus an *MTU dependent SR Black Hole* is created.

3.3 Weakness of Classical Monitoring Tools in Detecting MTU Dependent SR Black Holes

In this subsection we describe known monitoring tools used to identify black holes when the PMTUD procedure fails.

One of the most reliable tools to discover PMTUD failures is Scamper [12]. Scamper in a two steps procedure to determine either the largest MTU that can be used on a end to end path, and to discover (in case of a failure) what is the router that is not participating to the PMTUD. Both the phases of Scamper are based on the enforcement of probes along the end to end path to check. These probes are represented by a set of UDP segments destined to an unused port, when performing the first step, and a set of ICMP Echo messages destined to intermediate routers, in the second phase.

Netalyzr is presented in [10], it is a network measurement and debugging tool to monitor the Internet. The architecture is provided with a set of applets pre-installed, and one of these aims at determining the path MTU toward a destination server. This search is based on a process that require to send a set of UDP probes to the target destination.

Ripe Atlas [18] is a worldwide monitoring infrastructure based on the use hardware probes placed in the so called vantage point. In [2] Ripe Atlas has been used to discover path MTU black holes in the Internet, with the specific focus of assessing the main causes and the most affected data plane protocol. The obtained results show that black holes due to failure in the PMTUD procedure affect both the IPv4 and the IPv6 data planes. Specifically, Ripe Atlas was able to detect the main causes and the location of these failures, such as PTB messages and fragmented packets filtered by firewalls.

Unfortunately, none of the aforementioned tools is able to detect *MTU dependent SR Black Holes*. The reason is that all tools are based on an active approach, i.e. probe packets must be sent through an end-to-end path to be tested. The use of probes is inefficient when dealing with SR for two reasons: i) the presence of routing policies makes possible that the SR path followed by user data could be different than the one followed by the probes, even if the source and the destination of the path are the same; ii) even if the same end to end path is used for data and probes, the network programming feature of SRv6 could results to different SR policies for user data flows and probes, which turns in different SRH applied on probes (and different MTU constraints).

To overcome this limitation, in the next we introduce $SRBH^2$, a passive monitoring system able to detect *MTU dependent SR Black Holes* simply relying on data plane information gathered by the network devices.

4 *Segment Routing Black Holes Hound* Algorithm

The idea behind *Segment Routing Black Holes Hound* ($SRBH^2$) is to use a passive approach: instead of actively probe destination addresses, it searches possible black holes by analyzing passive measurements collected at the data plane. Specifically, the proposed algorithm is based on the Segment Routing

Traffic Counters [6], a new OAM feature available in SR capable routers. SR Traffic Counters, described in [17], allow to collect statistics on the processed SR traffic on SR capable nodes. Among the different types of available counters, we consider only the Base Pcounters ones. A Base Pcounter is able to account the amount of traffic crossing a given node and having a specific SID as active segment. As an example, referring to the scenario reported in Fig. 1, the Base Pcounter at node $C1$ related to the active segment $C2$ is accounting all the packets sent from host A to host B. With the notation $y_B(i, a)$ we refer to the value assumed by the Base Pcounter instantiated at node i and accounting packets having a as active segment.

$SRBH^2$ is a two steps algorithm which is able to detect *MTU dependent SR Black Hole* based on the analysis of the SR Traffic Counters. In the first step, the link where the *MTU dependent SR Black Hole* occurs is detected; in the second step, the affected flow (flows) is (are) found. $SRBH^2$ exploits the relationships existing among different counters to detect a possible *MTU dependent SR Black Hole*.

To better explain the algorithm, let's start describing the effect of an *MTU dependent SR Black Hole* on SR Traffic Counters. Let $\mathcal{G}(\mathcal{N}, \mathcal{L})$ be the direct graph modeling the network, where $\mathcal{N}$ is the set of nodes, and $\mathcal{L}$ is the set of links. The link connecting nodes i and j is referred to as $l_{i,j}$. A traffic counter $y_L(l_{i,j})$ (classical link count) is associated to each link and measures the aggregated amount of data sent through it. In the considered scenario, when the node i receives a packet having the SID a as active segment, the shortest path between nodes i and a is used to forward it. This path is represented to as $P_{i,a}$. A real number ranging between 0 and 1 is used to specify the percentage of the total traffic received at node i and destined at node a that is forwarded over the link $l_{i,j}$. This quantity is referred to as $P_{i,a}(l_{i,j})$, namely routing split coefficient, and can be easily determined by knowing the IGP paths used in the underlay IPv6 network. Using the presented notation, considering the interface counter associated to the link $l_{i,j}$, the following condition holds:

$$y_L(l_{i,j}) = \sum_{a \in \mathcal{N}} P_{i,a}(l_{i,j}) y_B(i, a) \tag{1}$$

Equation (1) shows that the traffic sent over the link $l_{i,j}$ is equal to the summation of the traffic received by the node i and that it is forwarded over the target link. Specifically, each Base Pcounter $y_B(i, a)$ accounts the traffic received by node i having a as active segment. This quantity is then multiplied for the routing coefficient $P_{i,a}(l_{i,j})$. In normal condition, i.e., in absence of *MTU dependent SR Black Hole*, Eq. 1 holds, since all the received traffic is regularly forwarded over the output link. On the contrary, if the MTU constraint over the link $l_{i,j}$ is violated, then a fraction of the received packets is dropped, leading to a discrepancy in the two sides of Eq. 1.

After the detection of the link affected by the *MTU dependent SR Black Hole*, the next step consists in determining the traffic flow involved. It is clear that the involved flow is one of the flows routed over the target link, referred to as suspected flows. SR Traffic Counters can be exploited to reduce the size of the set of suspected flows. In particular, referring to as $\mathcal{F}(l_{i,j})$ to the set of flows crossing link $l_{i,j}$, this can be further divided into disjoint subsets, one for each possible active segment. Then, in order to check whether the affected flow is in the subset related to the active segment a, the validity of the following condition has to be verified:

$$y_B(i,a) = \sum_{n \in \mathcal{N}_i} P_{n,a}(l_{n,i}) y_B(n,a) \tag{2}$$

where i is the ID of the node where the target link is entering on, and $\mathcal{N}_i$ represents the set of neighbors of the node i. Equation 2 shows the relationship arising between Base Pcounters instantiated in neighboring nodes. Specifically, considering the active segment a, the Base Pcounter at node i accounts all the packets forwarded by its neighbors, properly weighed by the routing split coefficients. In presence of an *MTU dependent SR Black Hole* on the link between the node i and one of its neighbors, there will be a mismatch on the value of the SR counters due to the fact that some of the traffic received at the previous hop are dropped. In this way it is possible to detect the SID used as active segment by the target flow; the set of suspected flows can be now reduced to the set of flows having the previous SID as active segment.

To summarize, $SRBH^2$ works as follows: i) for each link, Eq. (1) is checked to detect *MTU dependent SR Black Holes*; ii) for each link violating Eq. (1), the SID used by the flow affected by the *MTU dependent SR Black Hole* is identified using Eq. (2). The final outcome of the procedure will be a set of suspected flows. In the performance evaluation, the algorithm will be characterized by its ability in detecting the link affected by a *MTU dependent SR Black Hole*, and by the size of the set of suspected flows obtained.

Before concluding, it is important to remark that, as far as Eqs. 1 and 2 are concerned, the presented approach is valid under the hypothesis that the only source of packet loss is due to the presence of an *MTU dependent SR Black Hole*. Anyway, the applicability of $SRBH^2$ can be easily extended also to cases where multiple causes of packet loss (such as congestion and other types of failures) exist. This can be done by considering different available statistics (all types of SR Traffic Counters, and packet drop on router interfaces), and using dedicated monitoring system (e.g. the approach proposed in [1]) to detect different causes of packet loss causes, such as physical failures, running in parallel to $SRBH^2$. The analysis of such an architecture is currently under investigation.

5 Performance Evaluation

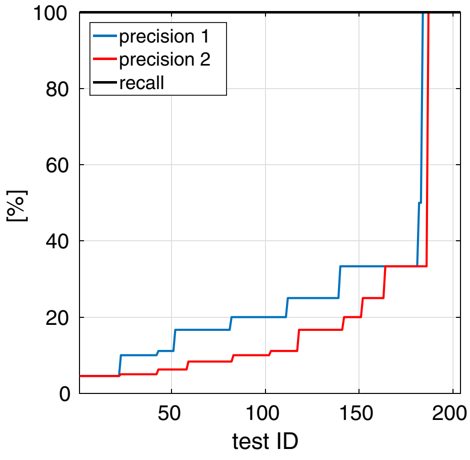
In this section we provide a preliminary performance evaluation of $SRBH^2$ algorithm when used to detect *MTU dependent SR Black Holes* (simply blackholes in the following). The experiments are conducted considering the France Telecom (38 nodes and 144 links) and the Brain (161 nodes and 166 links) networks [16] as the underlay IPv6. For these networks, real traffic matrices are available. Shortest paths are calculated using the Dijkstra algorithm and assuming all the links having the same cost. Segment lists are enforced at the ingress node on the incoming traffic flows according to the egress node and a specific traffic class. A single traffic class is considered, i.e., the best effort one. Best effort traffic is routed over IGP paths, consequently, the resulting SL is composed by a single SID indicating the egress node.

The performance evaluation is carried out in a simulated environment, using Matlab. A blackhole is simulated by dropping all the packet of a target flow when transiting over a target link. For each target flow a set of simulation is run. In each simulation the location of the blackhole is chosen among the links belonging to the path followed by the target flow. In order to get more realistic results, the first link of the path (i.e., the one leaving from the source node) is not considered as a possible location for the black hole, since the source node knows the value of the MTU over this link, then can fragment the packets accordingly. Finally, it is assumed that the only cause of packet loss is due to the presence of a black hole.

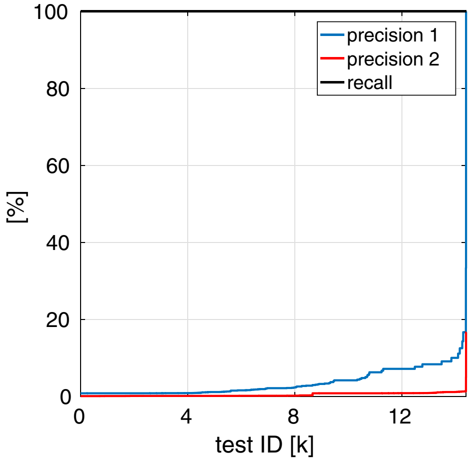
In the next, it is presented a set of experiments aiming at evaluating the performance of the $SRBH^2$ algorithm. The performance are evaluated in terms of *precision* and *recall.* The *precision* represents the fraction of correctly detected blackhole with respect to the total number of black holes (both true and false) computed by $SRBH^2$. The *recall* express the percentage of blackholes that have been correctly detected.

These parameters are used to determine the ability of $SRBH^2$ in assessing the link (first analysis) where the black hole occurs, and the flow (second analysis) causing it. Regarding the assessment of the link where the failure is happening, we have found that $SRBH^2$ always gets *precision* and *recall* equal to 100%. It means that the proposed algorithm is always able to isolate the link affected by the blackhole from the rest of the devices, highly reducing the portion of the network to be checked. This is a valuable result, since it highlights that blackholes can be identified and located with no errors using the proposed passive approach.

The second analysis regards the target flow: it has to be found among the ones routed over the link affected by the black hole. To give an idea of the impact of Eq. 2 on the performance of $SRBH^2$, we have reported in Fig. 2 the *precision* obtained by reporting as output the full set of flows traversing the affected link (referred to as *precision 2*) and in case the suspects are limited only to the subset of flows which violate the condition of Eq. 2 (referred to as *precision 1*). As it can be seen from the results reported in Fig. 2, the *recall* achieved by the proposed approach is always equal to 100%, which means that the affected flow is always among the suspected ones (our algorithm has no false negatives). This finding is encouraging to support the validity of $SRBH^2$ as a method to detect blackholes. Concerning the *precision*, from the analysis reported in Fig. 2 emerges that: i) it is affected by the size of the network, ii) $SRBH^2$ has a poor *precision* in assessing the ID of the flow affected by the black hole, iii) in some cases it reaches 100%, and iv) in general there are some failures that are more difficult to analyze. Furthermore it is clear the improvement on the *precision*



(a) France Telecom.

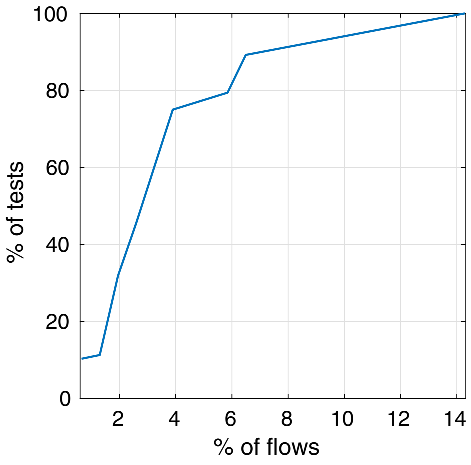


(b) Brain.

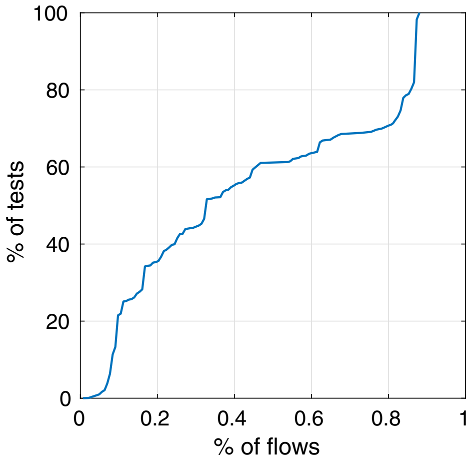
Fig. 2. Precision and Recall of $SRBH^2$ in the detection of the affected flow.

due to the use of Eq. 2 to reduce the size of the set of suspected flows. With reference to point one, the relation between the size of the network and the obtained *precision* is due to the number of traffic flows. When this last grows, the average number of flows routed over a link increases, making harder to detect the target flow by means of Eq. 2.

In order to better investigate the results, in Fig. 3 the CDF of the percentage of suspected flows detected by $SRBH^2$ is reported. In the France Telecom network (Fig. 3(a)), it is possible to see that in the 80% of the considered failures, the percentage of suspected flows is less than 6%. Considering that the total number of flows in this case is equal to 154, then the set of suspected flows (which contain the true positive) has a size of 22 flows at maximum for the 80%



(a) France Telecom.



(b) Brain.

Fig. 3. CDF of the percentage of suspected flows.

of cases. Similarly, in case of the Brain network (Fig. 3(b)), the percentage of suspected flows never exceeds the 1% of the overall number of flows. Since in this case there are more than $14K$ traffic relations, then the number of false positives is equal to 140. Thus, even though the *precision* is poor, actually $SRBH^2$ is greatly reducing the size of the suspected flows. To conclude, the performance evaluation has shown that $SRBH^2$ algorithm is a suitable approach (due to the high recall) for the detection of *MTU dependent SR Black Holes*, even if further work is needed to improve the precision.

6 Conclusion

In this paper we demonstrated that a new type of failure related to MTU can be generated in a SRv6 network. This failure, referred to as *MTU dependent SR Black Hole*, cannot be detect by known monitoring tools since they are based on active probing. We proposed $SRBH^2$, a passive measurement based algorithm able to detect links affected by blackholes only looking at SRv6 Traffic Counters collected by network SR capable routers. The main outcome of performance evaluation is that our solution is always able to detect the link affected by the blackhole; it is still not possible to univocally detect the flow causing the blackhole, but a significant reduction of the number of suspected flows is obtained.

As a future steps we aim at improving the precision and enable the use of $SRBH^2$ when multiple sources of packet loss are present. This imply to consider a wider set of network statistics, such as different SR Traffic Counters and interface counters, and to run in parallel additional monitoring tools.

References

1. Aubry, F., Lebrun, D., Vissicchio, S., Khong, M.T., Deville, Y., Bonaventure, O.: Scmon: leveraging segment routing to improve network monitoring. In: IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. IEEE (2016)
2. de Boer, M., Bosma, J.: Discovering path MTU black holes in the Internet using ripe atlas (2012)
3. Custura, A., Fairhurst, G., Learmonth, I.: Exploring usable path MTU in the internet. In: 2018 Network Traffic Measurement and Analysis Conference (TMA), pp. 1–8. IEEE (2018)
4. Deering, D.S.E., Hinden, B.: Internet protocol, version 6 (IPv6) specification. RFC 8200, July 2017. https://doi.org/10.17487/RFC8200
5. Desmouceaux, Y., Townsley, M., Clausen, T.H.: Zero-loss virtual machine migration with IPv6 segment routing. In: 2018 14th International Conference on Network and Service Management (CNSM), pp. 420–425. IEEE (2018)
6. Filsfils, C., Ali, Z., Horneffer, M., Voyer, D., Durrani, M., Raszuk, R.: Segment routing traffic accounting counters. Internet-Draft draft-filsfils-spring-sr-traffic-counters-00, Internet Engineering Task Force, June 2018. Work in Progress
7. Filsfils, C., Dukes, D., Previdi, S., Leddy, J., Matsushima, S., Voyer, D.: IPv6 segment routing header (SRH). Internet-Draft draft-ietf-6man-segment-routing-header-21, Internet Engineering Task Force, June 2019. Work in Progress
8. Filsfils, C., Previdi, S., Ginsberg, L., Decraene, B., Litkowski, S., Shakir, R.: Segment routing architecture. RFC 8402, July 2018. https://doi.org/10.17487/RFC8402
9. Kompella, R.R., Yates, J., Greenberg, A., Snoeren, A.C.: Detection and localization of network black holes. In: IEEE INFOCOM 2007–26th IEEE International Conference on Computer Communications, pp. 2180–2188. IEEE (2007)
10. Kreibich, C., Weaver, N., Nechaev, B., Paxson, V.: Netalyzr: illuminating the edge network. In: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, pp. 246–259. ACM (2010)

11. Kumar, N., Pignataro, C., Swallow, G., Akiya, N., Kini, S., Chen, M.: Label switched path (LSP) ping/traceroute for segment routing (SR) IGP-prefix and IGP-adjacency segment identifiers (SIDs) with MPLS data planes. RFC 8287, December 2017. https://doi.org/10.17487/RFC8287
12. Luckie, M., Cho, K., Owens, B.: Inferring and debugging path MTU discovery failures. In: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement. USENIX Association (2005)
13. Luckie, M., Stasiewicz, B.: Measuring path MTU discovery behaviour. In: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, pp. 102–108. ACM (2010)
14. Mathis, M., Heffner, J.: Packetization layer path MTU discovery. RFC 4821, March 2007. https://doi.org/10.17487/RFC4821
15. McCann, J., Deering, S.E., Mogul, J., Hinden, B.: Path MTU discovery for IP version 6. RFC 8201, July 2017. https://doi.org/10.17487/RFC8201
16. Orlowski, S., Pióro, M., Tomaszewski, A., Wessäly, R.: SNDlib 1.0-survivable network design library. In: Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium, April 2007
17. Polverini, M., Cianfrani, A., Listanti, M.: Interface counters in segment routing v6: a powerful instrument for traffic matrix assessment. In: 2018 9th International Conference on the Network of the Future (NOF), pp. 76–82, November 2018. https://doi.org/10.1109/NOF.2018.8597768
18. Staff, R.: Ripe atlas: a global internet measurement network. Internet Protocol J. **18**(3), 1–31 (2015)
19. Ventre, P.L., et al.: Segment routing: a comprehensive survey of research activities, standardization efforts and implementation results. CoRR abs/1904.03471 (2019). http://arxiv.org/abs/1904.03471