Through the Wormhole: Tracking Invisible MPLS Tunnels

Yves Vanaubel*, Pascal Mérindol‡, Jean-Jacques Pansiot‡, Benoit Donnet*
∗ Montefiore Institute, Université de Liège – Belgium
‡ Icube, Université de Strasbourg – France

ABSTRACT

For years, Internet topology research has been conducted through active measurement. For instance, Caida builds router level topologies on top of IP level traces obtained with `traceroute`. The resulting graphs contain a significant amount of nodes with a very large degree, often exceeding the actual number of interfaces of a router. Although this property may result from inaccurate alias resolution, we believe that opaque MPLS clouds made of invisible tunnels are the main cause. Using Layer-2 technologies such as MPLS, routers can be configured to hide internal IP hops from `traceroute`. Consequently, an entry point of an MPLS network appears as the neighbor of all exit points and the whole Layer-3 network turns into a dense mesh of high degree nodes.

This paper tackles three problems: the revelation of IP hops hidden by MPLS tunnels, the MPLS deployment underestimation, and the overestimation of high degree nodes. We develop new measurement techniques able to reveal the presence and content of invisible MPLS tunnels. We assess them through emulation and cross-validation and perform a large-scale measurement campaign targeting suspicious networks on which we apply statistical analysis. Finally, based on our dataset, we look at basic graph properties impacted by invisible tunnels.

CCS CONCEPTS

• **Networks → Network measurement**; **Topology analysis and generation**;

KEYWORDS

network discovery; MPLS; traceroute; fingerprinting; Internet modeling

ACM Reference Format:
Yves Vanaubel*, Pascal Mérindol‡, Jean-Jacques Pansiot‡, Benoit Donnet* ∗ Montefiore Institute, Université de Liège – Belgium ‡ Icube, Université de Strasbourg – France . 2017. Through the Wormhole: Tracking Invisible MPLS Tunnels. In *Proceedings of IMC '17, London, United Kingdom, November 1–3, 2017,* 14 pages.
https://doi.org/10.1145/3131365.3131378

x

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
IMC '17, November 1–3, 2017, London, United Kingdom
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5118-8/17/11…$15.00
https://doi.org/10.1145/3131365.3131378

1 INTRODUCTION

Since the end of the nineties, the Internet topology discovery has been extensively investigated. Indeed, numerous analyses [18, 26] have been proposed to describe various types of connectivity structures and representations of the Internet architecture. In particular, inferring the router level topology of IP networks is an important concern, notably to study routing characteristics. These router level maps are obtained by grouping together IP addresses collected with `traceroute`: this process is called *alias resolution*. Inferring the architecture of an Autonomous System (AS) is also crucial for analyzing the performance of routing protocols. Using random graph models rather than realistic networking topologies may result in biased or even wrong conclusions. For example, the performance of fast-rerouting schemes or multipath transport protocols strongly depend on the underlying topology.

Typically, router level topologies are undirected graphs built upon IP level traces obtained from `traceroute`; then, they can be statistically analyzed [33]. In particular, the node degree distribution fascinates the research community, specially since the Faloutsos et al. [21] seminal paper highlighting the power-law shape of this distribution. However, one may observe a significant amount of nodes with a very large degree, often exceeding the actual number of interfaces of a router. For instance, Fig. 1 illustrates the degree distribution of nodes in the Caida ITDK dataset [10] where we observe a large amount of nodes having a very large degree.

This large amount of high degree nodes might be explained by several factors. A `traceroute` campaign conducted from a limited number of vantage points can tend to induce a subgraph in which the inferred node degree distribution does follow a power law even if this is not the actual distribution [28]. Clauset and Moore [16] have since demonstrated analytically that such a phenomenon is to be expected for the specific case of the Erdös-Rényi random graphs [20]. Second, others [31] have stated that high degree nodes can emerge from Layer-2 (L2) clouds (such as Ethernet switches). L2 devices interconnect a large number of Layer-3 (L3) routers, themselves being also involved in multiple L2 interconnections. Such a situation induces nodes with very high degrees when analyzing the L3 graph with `traceroute` probing.

In this paper, we investigate another reason for HDNs in the Internet graph: opaque MPLS clouds hiding their content to `traceroute` probing [19]. *MultiProtocol Label Switching* (MPLS) [35] is a technology that has been designed to speedup forwarding decisions (through exact label matching instead of longest prefix matching on IP addresses) but is nowadays mainly deployed for providing IGP/BGP scalability, virtual private network (VPN) services [32], and traffic engineering capability [37, 41]. It has been shown that MPLS is largely deployed by operators [19, 36, 38, 39] thanks to dedicated measures making use of MPLS transparency features:

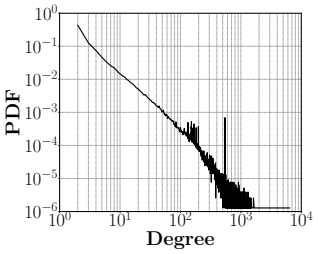


Figure 1: Node degree distribution in Caida ITDK dataset.

(*i*) the ability of MPLS routers to generate ICMP `time-exceeded` packets with MPLS label information [9] and, (*ii*), the ability of the TTL to be decremented (and thus making probing packets expiring) within the MPLS tunnel [1]. This MPLS popularity is also confirmed by a survey we made between August 28th, 2017 and September 12th, 2017. In the data collected (50 answers, from Stub ISPs to Tier-1) through direct contacts with operators or the Nanog community, it shows up that 87% of the surveyed operators deploy MPLS. It has also been demonstrated that MPLS tunnels may have an impact on Internet topology discovery tools [2, 6, 22].

Unfortunately, ISPs may want to hide, or, at least, not provide in details the structure and the configuration of their internal MPLS networks. For VPN services, provider networks generally prefer to simplify the routing view of their customers: they just see provider equipment directly connecting their different customer sites instead of viewing all the provider internal architecture details. To achieve this, they may restrict the deployment of MPLS transparency features, leading so to *invisible MPLS tunnels* [19] (i.e., the content of the tunnel is hidden to `traceroute` probes). Doing so, they can avoid competitor networks to imitate their finely tuned calibration and also avoid attackers to get knowledge of their internal organization [24].

Consequently, the data obtained by researchers from `traceroute` measurements is incomplete and the resulting Internet maps are potentially biased. Indeed, as the content of the tunnel is hidden, a direct, but false, link between the entry and exit points of the tunnel is inferred. Further, an entry point of a MPLS network appears as the direct neighbor of all exit points. The whole L3 network turns, then, into a dense mesh of high degree nodes [42]. It means that (*i*) current basic `traceroute` campaigns cause false router-level links to be inferred (between two edge routers separated by an invisible tunnel), that (*ii*) MPLS deployment in the Internet may be underestimated (missing internal IP links), and that (*iii*) node degree distribution, and other graph properties such as density or clustering coefficient, may be shifted to higher values. Another reason for identifying invisible MPLS tunnel is to better capture network delay anomalies [23]. Indeed, as the content of the tunnel is hidden to `traceroute`, the delay between the entry and exit point of the tunnel might appear as being artificially high, possibly leading to wrong conclusion when tracking connectivity issues.

In this paper, our aim is to fix those issues by proposing new probing mechanisms and analyses when exploiting IP level traces. In particular, our contributions are threefold. First, we develop and validate new active measurement techniques based on `traceroute` and TTL estimation that are able to, at worst (and as long as the UHP feature[1] is not enabled), reveal the presence of invisible tunnels and to, at best, expose their content in standard configuration cases. All proposed techniques have been assessed through emulation testbeds with GNS3, an emulator running actual IOS in a virtualized router[2] and through cross-validation. In particular, we show that our algorithms are efficient in roughly 86% of the cases. Second, with our specific dataset collected with our measurement mechanisms and its analysis, we are therefore able to improve the MPLS knowledge of the research community and provide an insight on ISPs standard and common practices. Finally, and as an illustration of our contribution purpose, we show how to improve classical Internet topology models by correcting the biases in terms of node degree, route length distributions, and graph density. Our dataset and GNS3 configuration scripts are freely available.[3]

The remainder of the paper is organized as follows: Sec. 2 provides the required background for this paper. Sec. 3 is the heart of the paper as it presents and validates our measurement techniques to reveal the content of invisible MPLS tunnels. Sec. 4 explains how we deploy our measurement techniques in the wild, while Sec. 5 presents the results. Sec. 6 discusses, based on the data we collected, the ISPs standard practices in deploying MPLS tunnels. Sec. 7 reviews a few basic Internet modeling features based on revealed invisible MPLS tunnels. Finally, Sec. 8 concludes this paper by summarizing its main achievements.

2 BACKGROUND

2.1 MPLS

MPLS routers, i.e., *Label Switching Routers* (LSRs), exchange labeled packets over *Label Switched Paths* (LSPs). In practice, those packets are tagged with one or more *label stack entries* (LSE) inserted between the frame header (data-link layer) and the IP packet (network layer). Each LSE is made of four fields: an MPLS label used for forwarding the packet to the next router, a Traffic Class field for quality of service, priority, and Explicit Congestion Notification [3], a bottom of stack flag bit (to indicate whether the current LSE is the last in the stack [34]), and a time-to-live (LSE-TTL) field having the same purpose as the IP-TTL field [1] (i.e., avoiding routing loops).

The first MPLS router (the *Ingress Label Edge Router*, or Ingress LER, i.e., the tunnel entry point) adds the label stack, while the last MPLS router (*Egress Label Edge Router*, or Egress LER, i.e., the tunnel exit point) removes the label stack (*Ultimate Hop Popping*, UHP, where the Egress LER advertises an explicit null label – label value of 0 [34]). In practice, and in most cases (at least this is the default configuration), the top LSE is removed by the penultimate LSR, that we call the *Last Hop* (LH). This operation is called *Penultimate Hop Popping* (PHP) and is activated by the Egress LER when it advertises an implicit null label (label value of 3 [34]). Since the top LSE has been removed by the Last Hop, the Egress LER performs then only a classic IP lookup to forward the traffic. It allows

[1]See Sec.2 for MPLS technical details about Penultimate Hop Popping (PHP) and Ultimate Hop Popping (UHP).
[2]See https://gns3.com/
[3]See http://www.montefiore.ulg.ac.be/~bdonnet/mpls

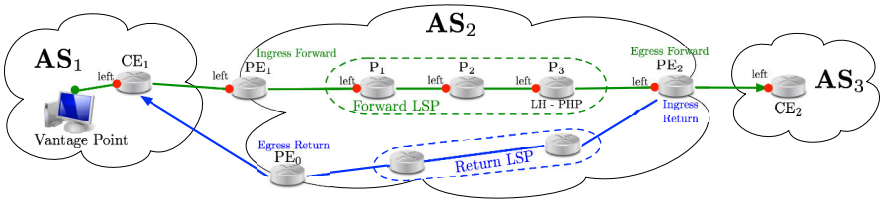


Figure 2: GNS3 topology. AS_2 is a transit AS with MPLS enabled (labels distributed with LDP). PE_1 acts as Ingress LER, PE_2 as Egress LER, and P_i are LSRs. CE_2 is the `traceroute` destination. Notation P_i`.left` refers to left interface of router P_i (the same applies for CE_j and PE_k). PHP is applied by P_3.

thus to reduce the load on the Egress LER, especially if it is the root of a large LSP-tree. This means that, when using PHP, the last MPLS operation (i.e., popping) is performed one hop before the Egress LER, on the Last Hop. On the contrary, UHP is generally used only when the operator implements sophisticated traffic engineering operations. This is confirmed by our survey as UHP is only deployed by 10% of the operators. Fig. 2 illustrates, among others, the main vocabulary associated to MPLS tunnels.

Labels may be allocated through the *Label Distribution Protocol* (LDP) [4]. Each LSR announces to its neighbors the association between a prefix in its routing table and a label it has chosen. Therefore, labels are allocated from downstream and, for a given prefix, a router advertises the same label to all its neighbors. Depending on the implementation, LDP may advertise a label for all prefixes in its IGP routing table (default case for Cisco routers [17, Chap. 4], [8]) or only for loopback addresses (default case for Juniper routers [8]). LDP is mainly used for scalability reasons (e.g., to limit BGP-IGP interactions to edge routers) as deployed tunnels are congruent with the IGP. Labels can also be distributed through RSVP-TE [7], when MPLS is used for Traffic Engineering (TE) purposes. In practice, most operators consider the use of RSVP-TE in addition to the use of LDP. This is confirmed by our survey. While LDP-only is used by 50% of the operators, RSVP-TE is used alone by only 8% of the operators. RSVP-TE and LDP are used in conjunction by 42% of the operators. Note that only a single operator considers another labeling protocol.[4]

2.2 Measuring MPLS Tunnels

LSRs may send ICMP `time-exceeded` messages when the LSE-TTL expires. If the LSR implements RFC 4950 [9] (as it should be the case for all recent OSes), it simply quotes the MPLS LSE stack of the received packet in the ICMP `time-exceeded` message.

If the Ingress LER copies the IP-TTL value to the LSE-TTL field rather than setting the LSE-TTL to an arbitrary value such as 255, LSRs along the LSP will reveal themselves via ICMP messages, even if they do not implement RFC4950 (in such a case they do not quote the LSE but just reveal their incoming IP address). Operators can configure this transparency operation using the `ttl-propagate` option provided by the router manufacturer [1] (while, to the best of our knowledge, RFC4950 compliance is just a matter of implementation, and cannot be deactivated on recent OSes supporting it). Donnet et al. [19] have discussed in detail the impact of those two features (i.e., RFC4950 and `ttl-propagate`) on MPLS tunnel discovery based on `traceroute`.

[4]Probably for Segment Routing as LDP or RSVP-TE are not required to distribute labels in this case [14, Chap. 1, pg. 2].

Router Signature	Router Brand and OS
$< 255, 255 >$	Cisco (IOS, IOS XR)
$< 255, 64 >$	Juniper (Junos)
$< 128, 128 >$	Juniper (JunosE)
$< 64, 64 >$	Brocade, Alcatel, Linux

Table 1: Summary of main router signature, the first initial TTL of the pair corresponds to ICMP `time-exceeded`, while the second is for ICMP `echo-reply`.

In this paper, we focus on *invisible MPLS tunnels*, i.e., tunnels that are completely obscured from `traceroute`: the Ingress LER does not enable the `ttl-propagate` option, and the last hop does not send back an ICMP `time-exceeded` message (that may embed a MPLS LSE). Due to the PHP feature (with UHP the tunnel is even more invisible than with PHP), the last hop is the LSR in charge of converting the MPLS data packet into a standard IP one. It does not send back neither a RFC4950 nor a standard ICMP error message, because it does not decrement the IP-TTL. As a matter of fact, the last hop considers now this transit packet as an IP one, and simply pushes it to the Egress LER that will decrement the IP-TTL. Hence, all IP hops inside the tunnel are hidden, and the topology information is missing from `traceroute` exploration, providing so a biased view of the network. This is illustrated in Fig. 2 and Fig. 4d (for the Paris traceroute [5] output) when performing a `traceroute` from the Vantage Point towards the target, CE_2 in AS_3. In our survey, a surprising large share of 48% of the operators make use of the `no-ttl-propagate` option.

2.3 Network Fingerprinting

Vanaubel et al. [40] have presented a router fingerprinting technique that classifies networking devices based on their hardware and OS. This method infers initial TTL values used by a router when generating its different kinds of reply packets. It then builds the router *signature*, i.e., the n-tuple of n initial TTLs. A basic pair-signature (with $n = 2$) simply uses the initial TTL of two different messages: an ICMP `time-exceeded` message elicited by a `traceroute` probe, and an ICMP `echo-reply` message obtained from an

echo-request probe. Table 1 summarizes the main router signatures, with associated router brands and router OSes. This feature is really interesting since the two most deployed router brands, Cisco and Juniper, have, in theory, distinct MPLS behaviors.

In our survey, if a large proportion (i.e., 25%) of operators used a mix of router technologies, Cisco routers are the most prominent (58%), followed by Juniper (28%).

3 DISCOVERING INVISIBLE MPLS TUNNELS

This section describes our techniques for revealing the content, or at least identifying the presence, of invisible MPLS tunnels. We propose four complementary mechanisms simply based on traceroute or ping and falling into two categories. First, *Forward/Return Path Length Analysis* (**FRPLA**) and *Return Tunnel Length Analysis* (**RTLA**) are only able to provide more or less high-level information about invisible MPLS tunnels[5]: an estimation (FRPLA) or the exact number (RTLA) of hops hidden by the return MPLS tunnel[6] between the Ingress and Egress LERs in the return LSP. Second, *Direct Path Revelation* (**DPR**) and *Backward Recursive Path Revelation* (**BRPR**) are able to explicitly reveal the content of the obfuscated tunnel (the hidden LSR hops in the LSP), either in a single probe or hop by hop with a recursive probing process.

Combining those four techniques allows us to capture a majority of MPLS use cases: Juniper and Cisco standard behaviors and typical network MPLS/IGP/BGP configurations (per default in particular). Table 2 summarizes the scope of the four measurement techniques for different MPLS configurations. While the two following sections deepens our measurement techniques (Sec. 3.1 is dedicated to FRPLA and RTLA and Sec. 3.2 to DPR and BRPR), Sec. 3.3 validates them using several studies: (*i*), experimentally with GNS3, an emulator running the actual IOS of real routers in a virtualized environment[8], and (*ii*), with a dedicated cross-validation campaign on explicit tunnels. Finally, Sec. 3.4 discusses the inherent limitations of our techniques.

3.1 Inferring the Length of Tunnels

The two first techniques (FRPLA and RTLA) are based on the same principle. When entering an invisible tunnel in the forward path (i.e., from source to destination), the IP-TTL is not copied in the MPLS LSE (the Ingress LER does not enable the ttl-propagate option, as explained in Sec. 2.2), making the tunnel appear to be a single hop route. In Fig. 2, P_1, P_2, and P_3 in AS_2 are not revealed by the traceroute run from the Vantage Point to the target CE_2, located in AS_3. Instead, the link $PE_1 \rightarrow PE_2$ appears as a single hop, as illustrated by the first Paris traceroute [5] output on Fig. 4c.

Hopefully, when performing the traceroute from the Vantage Point to the target, when the TTL expires at the Egress LER, it generates an ICMP time-exceeded message. If this reply also goes back to the Vantage Point through an MPLS tunnel, when leaving this return tunnel at its last hop, the LSE-TTL is copied in the IP-TTL only if it is lower than the IP-TTL in order to avoid routing loops (this *min* behavior is implemented by Cisco [17]). More formally, if we denote $TTL_{IP}(X)$ (resp., $TTL_{LSE}(X)$) the IP-TTL (resp., LSE-TTL) of the reply in transit at a node X on the return path, and $h(X, Y)$ the number of hops (of the return path) from nodes X towards Y, it comes:

$$TTL_{IP}(VP) = min(TTL_{IP}(E), TTL_{LSE}(E)) - h(E, VP). \quad (1)$$

where VP is the vantage point that receives replies coming back via E, the Egress LER of the return path.[9] Thus, we have $TTL_{IP}(VP) = TTL_{LSE}(E) - h(E, VP)$ when there is a tunnel on the return path. Indeed, it is very likely the LSE-TTL and the IP-TTL have been initialized to the same value (e.g., 255 in the vast majority as described in [40]) at the router that originates the ICMP reply, e.g., the Egress LER of the forward path. Thus, the LSE-TTL, at E, is always lower than the IP-TTL as it has been decremented along the return LSP. While the forward tunnel is totally invisible, one can infer the length of the return tunnel ($LSE_{TTL}(E)$ is visible). For instance, imagine on Fig. 2 that the forward tunnel from the Vantage Point to CE_2 is the same as the return tunnel when the IP packet expires at PE_2. In that case, when generating the ICMP time-exceeded, PE_2 sets the IP and LSE-TTL to 255 [40] but only the LSE-TTL is decremented in the tunnel. When arriving at the Egress LER of the return path (i.e., PE_1), the *min* scheme is applied, resulting in the value 252 being copied in the IP-TTL (i.e., $min(255, 252)$). Thus, the IP-TTL observed by the Vantage Point when receiving the ICMP time-exceeded message would be 252 - 2 = 250.

In practice, this *min* scheme allows Egress routers to behave the same (i.e., applying a minimum function on both TTLs before leaving the MPLS tunnel) whether the ttl-propagate option is used or not at the Ingress LER. Therefore, avoiding routing loop occurrences is performed in a stateless manner without any signalization. With this standard behavior, the number of hops of the tunnel is included in the return path length. However, the return tunnel length is still not clearly retrievable. Indeed, the Egress of the return path is not necessarily the Ingress of the forward path, and the forward/return paths are not the same in general as routing, and BGP in particular, may introduce path asymmetry.

With FRPLA, we compare, at the AS granularity, the length distribution of forward and return paths. Then, we can statistically analyze whether we observe a significant differential (the so-called "shift" in Table 2) as return paths are expected to be longer than forward ones. Tunnel hops being not counted in the forward paths while they are taken into account in the return paths. We expect that, when no IP hops are hidden, the resulting distribution will look like a normal distribution centered in 0 (i.e., forward and return paths have, on average, a similar length). If we observe, rather, a shift towards positive values, it is then likely that the AS makes use of the no-ttl-propagate option. This difference may provide the average tunnel length of the AS.

As an example, on Fig. 2, the Egress LER, PE_2, is located six hops from the Vantage Point while only the two LERs (PE_1 and PE_2) are exposed when the LSP is turned invisible. On the Paris traceroute output, when the tunnel is made visible on Fig. 4a, we can observe each internal hop of the forward LSP. One can conclude that if the

[5] Those methods are not able to reveal the internal IP hops hidden by the tunnels.
[6] A *return tunnel* refers to the MPLS tunnel taken by the ICMP time-exceeded or ICMP echo-reply packet.
[7] On Juniper routers, the TTL of the ping reply differs from the traceroute one [40]. We may exploit this singularity and analysis the potential gap between the two.
[8] See https://gns3.com/
[9] In practice, this min operation is rather implemented at the penultimate hop of the LSP, the LH LSR, when PHP is the rule. However, for readability reason, we keep the notation as simple as possible.

LDP advertising policy	`traceroute` target	TTL propagation policy (assuming the same configuration on both LERs)		
		`ttl-propagate`	`no-ttl-propagate`	
			$< 255, 255 >$	$< 255, 64 >$ [7]
All internal prefixes	external	explicit LSP no shift & no gap	invisible LSP shift (**FRPLA**) & no gap	invisible LSP shift (**FRPLA**) & gap (**RTLA**)
	internal	explicit LSP no shift & no gap	Last Hop without label via **BRPR** shift (**FRPLA**) & no gap	Last Hop without label via **BRPR** shift (**FRPLA**) & gap (**RTLA**)
Loopback address only	external	explicit LSP no shift & no gap	invisible LSP shift (**FRPLA**) & no gap	invisible LSP shift (**FRPLA**) & gap (**RTLA**)
	internal	explicit IP route no shift & no gap	route without labels via **DPR** shift (**FRPLA**) & no gap	route without labels via **DPR** shift (**FRPLA**) & gap (**RTLA**)

Table 2: Visibility effects of basic MPLS configurations according to the label advertisement policy (see Sec. 2.1 for details), `traceroute` target (i.e., "external" refers to a `traceroute` target outside the AS having the invisible MPLS tunnel, while, with "internal", the `traceroute` destination is inside the AS), and TTL policy (i.e., does the ISP hides the tunnel using a no propagation feature or not?) and the signature of the LER. We assume PHP is applied.

$$h(I,E) = \overbrace{(255 - TTL_{IP}(VP))}^{\texttt{time_exceeded}} - \overbrace{(64 - TTL_{IP}(VP))}^{\texttt{echo reply}}$$

$$(255 - min(TTL_{IP}(E), TTL_{LSE}(E)) - h(E,VP))$$

$$(64 - min(TTL_{IP}(E), TTL_{LSE}(E)) - h(E,VP))$$

$$\begin{aligned} h(I,E) &= TTL_{IP}(E) - TTL_{LSE}(E) + 191 \\ &= 64 - TTL_{LSE}(E) + 191 \\ &= 255 - TTL_{LSE}(E) \end{aligned}$$

Figure 3: Return Tunnel Length Analysis (RTLA). I **and** E **are respectively the Ingress and the Egress LERs of the return LSP.** $h(X,Y)$ **is the number of hops from node X to node Y.**

forward and return paths are the same, the LSP is made of three LSRs (a difference of $6-3 = 3$ hops between the return and forward path lengths).

FRPLA is our most generic method (see Table 2) as it should work at least for all Cisco LSRs using PHP as default configuration.[10] On the contrary, RTLA only works for networks deploying Juniper LERs on the edges. It produces similar results to FRPLA, but with more accuracy (because Juniper TTL signatures provide more information – see Table 1): while FRPLA only provides the return path length until the vantage point and is, thus, sensitive to route asymmetry (due to BGP in particular), RTLA provides exactly the return tunnel length (at least in the absence of ECMP routes that have distinct hop number) by exploiting the TTL gap between two kinds of probes.[7] Hence, when both RTLA and FRPLA apply, we prefer to exploit the result given with RTLA.

With Juniper Egress LERs or other routers with signature <255, 64> (see Table 1), and when the *min* behavior is enabled on the return LSP, two kinds of probes can be used for revealing the actual return path length with RTLA. Indeed, while the TTL of the return path is initialized at 255 for ICMP `time-exceeded` replies, it is initialized at 64 for ICMP `echo-reply` [40]. In this latter case, the IP-TTL of the answer is always lower than the LSE-TTL (because this one is always set to 255 and tunnels are short enough). Consequently, the last hop of the return tunnel, when applying the *min* function, does not copy the LSE-TTL inside the IP-TTL packet, but rather keeps the IP-TTL, which is is still at 64 for the reply that is originated from the Egress LER of the forward path. That is what we call a "gap" in Table 2. More formally, we can deduce the length of the return tunnel, $h(I,E)$, as illustrated in Fig. 3. We observe that the gap between the two return path lengths, given in the first line, is the number of hops of the return tunnel, $h(I,E)$.

For example, applying this computation to the example of Fig. 2 and assuming that the return path is the same as the forward path, and that PE_2 has a $< 255, 64 >$ signature, we would have for the `time-exceeded` packet $TTL_{IP}(VP) = 250$, while for the `echo-reply` packet $TTL_{IP}(VP) = 62$. It comes $(255-250)-(64-62) = 3$, which is the length of the (return) tunnel.

3.2 Revealing the Hidden Hops

The basic idea of these methods is that inside an MPLS network, not all packets are forwarded through LSPs. This is because LSPs may be constructed towards only a subset of internal prefixes (for example loopback addresses for Juniper routers, while Cisco routers create LSPs for all internal prefixes) or only packets destined to a BGP next-hop may be switched through MPLS (also a default behavior of Juniper routers [13, 27]). If one is able to `traceroute` one of the router's internal IGP IP addresses (belonging to the prefixes related to internal traffic), e.g., the incoming interface of the Egress LER (revealed with PHP), one can see explicit IGP routes without labels, and so infers the hidden LDP tunnel.

Besides, Cisco routers can also be configured that way when the network is partitioned into core and edge routers regarding the IGP/BGP structuration (e.g., to avoid external routes redistribution to IGP-only LSR). One can easily configure LDP prefixes filters in order to limit LDP signalling for external BGP transit traffic. In both cases (Juniper per default or basic Cisco configuration), and when using the `BGP next hop feature` on LERs, all the external BGP transit traffic goes through MPLS tunnels while the traffic

[10] Our survey highlights that 58% of operators deploy Cisco hardware.

targeting internal IP prefixes is routed via IGP explicit routes. On Fig. 4c, we illustrate the use of the command `mpls ldp label allocate global host-routes` that mimics a Juniper behavior on Cisco routers: we reveal the explicit IGP route in a single probing shot if we target the incoming interface of the Egress PE_2.left (this interface shares a prefix with the last hop P_3 on Fig. 4c). This is the principle of the Direct Path Revelation (DPR).

Our last method, Backward Recursive Path Revelation (BRPR), is based on the PHP feature when the network enables LDP everywhere (the standard and per default behavior of Cisco LSRs). Since `traceroute` naturally reveals the incoming IP interface of each Egress LER, we can apply a recursive `traceroute` approach that targets this last internal prefix to reveal each intermediate hop in a backward fashion from the Egress LER until the Ingress LER. This approach works well when the BGP routes remain similar for all internal prefixes of the targeted AS, i.e., they enter via the same Ingress LER and follow the same shortest IGP path inside (this is the default LDP behavior). It is worth mentioning that the incoming IP interface of each Egress LSR appears thanks to both PHP and the fact that the IP prefix belongs to the last hop and the Egress LSR. On Fig. 4b, we show that four steps are necessary to stop the recursion and reveal all internal LSRs in a backward fashion.

3.3 Validating our Measurement Techniques

In order to validate our measurement mechanisms, we conducted experimentations using GNS3. It allows us to run the actual Cisco IOS system, in our case IOS 15.2(4), over an emulated platform. We also analyzed a similar Juniper testbed, except for the UHP case which is not available for LDP on Junos. We do not show the Juniper emulation here due to space limitations. For our experimentations, we setup a simple configuration (see Fig. 2) with three ASes: AS_1 is the client AS, with router CE_1 (the `traceroute` source is connected to CE_1), a transit AS (AS_2) running MPLS and LDP as the labeling protocol for the LSP setup between the five routers PE_1, P_1, P_2, P_3, and PE_2, and finally, another client AS, AS_3, with router CE_2 connected to PE_2. The initial `traceroute` target is an internal prefix of AS_3 (i.e., a loopback of CE_2). Routing between ASes is handled with BGP, while internal routing is managed through OSPF.

We tested several MPLS feature combinations on the network given in Fig. 2. All of them are simple to enable (a few basic commands per LSR) and close to the Cisco MPLS default configuration.

The first scenario is the so-called **Default** configuration. PHP (with implicit null label – label value of 3) and TTL propagation are enabled by default, and all internal IP prefixes are announced through LDP. In this case, `traceroute` explicitly shows LSPs, with MPLS labels, as shown by the simulation output in Fig. 4a. Note that the return TTL shown between brackets for nodes P_1 and P_2 are 247 and 248, because `time-exceeded` messages generated inside a tunnel are first forwarded to the end of the tunnel [19].

The second scenario is the **Backward Recursive** configuration. It is the same as Default except that the TTL propagation is disabled (command `no mpls ip propagate-ttl` applied on all LERs). `traceroute` does not show MPLS tunnels anymore, as illustrated in Fig. 4b. However, as mentioned in Sec. 3.2, retracing the previous trace recursively backward starting initially from the Egress LER PE_2 until the Ingress LER PE_1 allows us to reveal the entire tunnel, but without any MPLS flag, one LSR at a time: tracing towards PE_2 reveals the incoming address of P_3 (thanks to PHP), and tracing towards P_3 reveals the incoming address of P_2, and so on. This corresponds to the expected output with the Backward Recursive Path Revelation technique (BRPR).

BRPR or DPR fail	8%
DPR successful	57%
BRPR successful	3%
hybrid DPR/BRPR	5%
BRPR or DPR	26%

Table 3: Cross-validation results on 5,364 Ingress-Egress LERs pairs, scattered in 271 different ASes.

The third scenario is the **Explicit Route** configuration. It is similar to Backward Recursive, but only loopback addresses (i.e., "host addresses" instead of all prefixes) are announced into LDP (command `mpls ldp label allocate global host-routes` applied on all LERs). As said in the introduction of Sec. 3, this is also the default Juniper configuration. In this case, a trace towards CE_2 reveals PE_2 incoming address (which is not a loopback address, hence it is not announced through LDP), and then a trace towards this address reveals the full LSP $PE_1 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow PE_2$ (Fig. 4c) but without MPLS flags since it is not switched through MPLS space. It corresponds thus to the output expected with the Direct Path Revelation (DPR).

Finally, the last scenario is the **Totally Invisible** configuration. In this case UHP is enabled on all LERs (the command `mpls ldp explicit-null` triggers the use of explicit null labels). The (forward) TTL propagation is also disabled. A trace towards CE_2 does not reveal PE_2 (CE_2 seems directly connected to PE_1), as shown on Fig. 4d. Assuming that an address of PE_2 has been discovered by another mean and can be used as a new target, none of the techniques given in this paper reveal anything.

For FRPLA and RTLA, we analyzed the TTL of the ICMP `time-exceeded` messages received when tracing. For example, while PE_2 appears to be at six hops (see Fig. 2 and Fig. 4c) or three hops (see Fig. 4d), the return TTL (provided between brackets on Fig. 4) always indicates six hops (when tracing towards CE_2), except in the UHP case, providing so the shift and the gap that might be exploited by FRPLA and RTLA.

To further validate our measurement techniques, we also performed a cross-validation of hidden hops revelation techniques (i.e., DPR and BRPR) on explicit tunnels. To do so, we collected data from PlanetLab. We considered 23 PlanetLab vantage points spread into five teams. Each team is responsible to probe towards 10,000 destinations (no overlapping between the teams) obtained from the Archipelago dataset [15]. This leads to a total of 269,096 traces collected. We extracted 14,771 distinct Ingress-Egress LERs pairs, with LSRs being explicitly revealed between the LERs. Note that both LERs should be in the same AS and the content of the LSR be fully revealed (i.e., no anonymous hops).

On those pairs, we rerun DPR and BRPR. On the one hand, DPR is considered as successful if, when targeting the Egress LER, we obtain the exact same number of hops between the Ingress and

```
$pt CE2.left
1 CE1.left [255]
2 PE1.left [254]
3 P1.left  [247]
  MPLS Label 19 TTL=1
4 P2.left !T2 [248]
  MPLS Label 20 TTL=1
5 P3.left  [251]
  MPLS Label 21 TTL=1
6 PE2.left [250]
7 CE2.left [249]
```

(a) Default Configuration: explicit tunnel.

```
$pt CE2.left
1 CE1.left [255]
2 PE1.left [254]
3 PE2.left [250]
4 CE2.left [250]

$pt PE2.left
1 CE1.left [255]
2 PE1.left [254]
3 P3.left  [251]
4 PE2.left [250]

$pt P3.left
1 CE1.left [255]
2 PE1.left [254]
3 P2.left  [252]
4 P3.left  [251]

$pt P2.left
1 CE1.left [255]
2 PE1.left [254]
3 P1.left  [253]
4 P2.left  [252]

$pt P1.left
1 CE1.left [255]
2 PE1.left [254]
3 P1.left  [253]
```

(b) Last Hops without labels discovered with a recursive process (BRPR).

```
$pt CE2.left
1 CE1.left [255]
2 PE1.left [254]
3 PE2.left [250]
4 CE2.left [250]

$pt PE2.left
1 CE1.left [255]
2 PE1.left [254]
3 P1.left  [253]
4 P2.left  [252]
5 P3.left  [251]
6 PE2.left [250]
```

(c) Route without labels in a single probe (DPR).

```
$pt CE2.left
1 CE1.left [255]
2 PE1.left [254]
3 CE2.left [252]

$pt PE2.left
1 CE1.left [255]
2 PE1.left [254]
3 PE2.left [253]
```

(d) Invisible UHP tunnel.

Figure 4: Emulation results for each basic configuration (`pt` stands for the `paris-traceroute` command [5]). The TTL of each ICMP `time-exceeded` reply received at the vantage point on the return path is provided between brackets for each hop. This is the return IP-TTL used for FRPLA and RTLA analyses (the latter using two distinct return IP-TTL).

Egress LER[11] and all MPLS labels have disappeared from the `traceroute` output. On the other hand, BRPR is considered as successful if, at each step of the recursion, the last hop does not exhibit any label.

On the 14,771 distinct Ingress-Egress LERs, we obtained 9,407 pairs for which the re-run failed, either because the Ingress or the Egress was not re-discovered. Table 3 summarizes the cross-validation on the 5,364 remaining pairs.

We see that in 8% of the cases, DPR or BRPR fail. However, in 60% of the cases, we have got a success with DPR and BRPR. Note that, in a few particular cases (5%), tunnels were revealed partially by DPR and partially by BRPR. Finally, in 26% of the cases, we successfully retrieve the tunnel but we cannot discriminate between DPR and BRPR as the LSP counts only one LSR.[12]

Finally, Table 3 shows a very low success rate for BRPR (3%). This suggests that, given the large proportion of Cisco routers deployed by operators, Cisco devices are configured to inject loopback addresses into LDP instead of all prefixes. This is particularly true when the operator deploys hybrid hardware (Juniper and Cisco – this hybrid situation is confirmed by our survey, see Sec. 2.3). In that case, Juniper devices systematically filters any piece of information not associated to loopback addresses. Table 5 also experimentally confirms this interesting result.

3.4 Discussion

Our techniques cover all basic MPLS configurations, except the totally invisible one with UHP enabled (not the standard configuration as it is useless for basic LDP tunneling – our survey highlights the fact that only 10% of the operators deploy UHP). The main configuration of Cisco (PHP and all prefixes enabled) is seen with FRPLA and BRPR. The basic Juniper configuration is seen with DPR and also causes a shift visible with FRPLA. The mix with Juniper LER and Cisco LSR is specifically covered with RTLA. It is likely that all other mixes with PHP as the default mode also triggers one signal, i.e., with Cisco LER, at least FRPLA should work (and so BRPR) and with Juniper LER, RTLA and DPR (and/or BRPR) should work. BRPR and DPR can be combined in several ways for tracking more advanced configurations with LDP filters or heterogeneous OS, as when some internal prefixes are not announced in LDP.

On the contrary to other techniques, FRPLA should not be used in the wild at the tunnel scale, otherwise it faces the risk of producing false positives (i.e., a tunnel length of X hops is inferred because the return path has X more hops than the forward one due to routing asymmetry) and false negatives (i.e., the return pat is shorter of X hops compared to the forward one due to routing asymmetry). Instead, FRPLA should be used, using multiple independent vantage points, as a statistical method in order to correctly infer the existence, and possibly the average length, of invisible tunnels at the AS scale. Indeed, if the `traceroute` campaign produces traces entering through a sufficient number of Ingress LER in the target AS, it is very likely the routing asymmetry will follow a normal law centered on 0 (as already confirmed by our experimental results, see Fig. 7 in particular). In other words, FRPLA actually measures the sum of the length of the return tunnel and the IP routing asymmetry of the trace. This asymmetry being the length difference between forward and return paths (a positive or a negative shift).

Generally speaking, if MPLS is enabled in a given network with PHP (it is by default for basic LDP tunneling, whatever the configurations and the OSes), we should see it with at least one of our techniques. On the contrary, UHP, mainly designed for traffic engineering oriented tunnels, turns RSVP-TE tunnels really invisible. Since network operators have no reasons to deploy RSVP-TE without also enabling LDP, we argue our set of techniques provides at least one MPLS signal in the vast majority of cases, provided some transit traces traversing LDP signaled LSPs.

[11] In practice, most of the time, both paths (i.e., the explicit tunnel and the one revealed by DPR), are exactly the same. However, in some cases, load balancing with ECMP may exhibit a similar path but with distinct IP addresses.

[12] This last statement is aligned with the very short tunnel length distribution [19, 36].

4 DATA COLLECTION

Prior to deploying in the wild our measurement techniques, we looked for HDNs in the CAIDA ITDK dataset [10] (it provides router level topologies). We first cleaned up the dataset by removing non publicly-routable IP addresses and pseudo-addresses allocated to non-responsive routers. After this pruning, 45,021,817 IP addresses, 44,700,863 nodes, 2,705,780 links, and 43,178 ASes remained.

We set the threshold for differentiating low degree nodes from high degree nodes (HDNs) to 128 (i.e., any node with a degree greater or equal to 128 is tagged as an HDN) and select areas of interest on where to send our probes. The 128 degree HDN threshold is selected to be a lower bound relative to well-known physical hardware, in particular PE routers which we expect to terminate invisible tunnels. For instance, the ASR9000 series is one of the best selling Cisco PE routers. This router can be equipped with up to 20 linecards, each containing up to 16 interfaces. Thus, a threshold of 128 is a reasonable balance between the volume of probes sent (we do not want to burden the network) and the amount of interesting data collected. Obviously, invisible tunnels do not only occur between HDNs, but we expect that many HDN pairs hide invisible tunnels (proportionally much more than non HDN pairs). Considering a threshold of 128 let us with 17,944 HDNs.

To efficiently guide our measurements, we retrieved, from the CAIDA ITDK dataset [10], the neighbors of the HDNs (set A – 599,467 unique nodes) and, next, the neighbors of neighbors of HDNs (set B – 983,793 unique nodes). This latter set is able to provide us IP addresses that do not belong to the same AS as addresses in the former set (our basic hypothesis is that MPLS LERs also define borders of domains[13]). Our aim is to simulate transit traffic, i.e., entirely traversing the suspicious AS made of many HDNs and ending up to one of its neighbors. Since we are looking for targets around HDNs in general, the destination set of our measurement campaign is the union of both sets (set $A \bigcup B$). We obtained a total of 1,306,545 destination IP addresses.

Our measurement scripts used `scamper` [29], and its Paris traceroute [5] implementation with ICMP `echo-request` packets, starting at TTL equals 2. In addition, `echo-request` probes are also sent to all IP addresses appearing in the traces for router signature inference [40]. For each of these traces, we look at the last three hops, say `X`, `Y`, `D` (where `D` $\in A \bigcup B$). `X` and `Y` are candidate endpoints of an invisible tunnel. A second trace with `Y` as target is then launched. If this trace ends with `X`, `H`, `Y`, we infer that one hop, `H`, has been revealed from an invisible tunnel. A new trace is then launched towards `H` (recursive processing with BRPR) as an attempt to reveal more hops. If a new hop, say `H'`, is discovered in the trace towards `H` (this trace should end with `X`, `H'`, `H`), the recursion continues with `H'` as the target and so on. This recursive process stops either if no new address is revealed (so we cannot distinguish DPR from BRPR if the recursion stops at the second trace – the revealed tunnel has only one hop), or if the new trace does not go through X. Note that multiple IP addresses may have been revealed in a single shot, i.e., the trace towards Y ends with `X`, $H_1, H_2, \ldots, H_{n-1}, H_n$, `Y`, with $n > 1$ being the number of hops discovered with DPR.

[13]This has been verified with the bdrmap [30] dataset, but partially (as both measurement campaigns are strongly different, intersection between datasets is sometimes weak).

Since we are looking for invisible MPLS tunnels spanning a single AS, and with HDNs as a trigger, our post-processing methods select only traces ending by `I`, `E`, `D` where `I` and `E` are HDNs located in the same AS. For AS resolution, we referred to CAIDA node-to-AS mapping when available, otherwise IP-to-AS mapping from Team Cymru.[14] Our findings for these `I`, `E` couples are discussed in the next section.

Our tool was deployed on the PlanetLab testbed on 91 Vantage Points (VP) distributed all around the world (USA, Canada, Europe, Japan, Russia, Brazil, China, Australia, New Zealand). The VPs were distributed equally in five groups, paying attention to the geographical locations. The destination set ($A \bigcup B$) was distributed amongst the different groups of VPs as follows: (i) the HDN neighbors (set A) were randomly spread over five subsets, (ii) the neighbors of each neighbors (set B) were added in the corresponding VPs subsets. It is worth noticing that the different destinations subsets were thus consistent, i.e., if neighbor N is in VP set 1, then all neighbors of N are also in VP set 1. The sizes of the destination subsets are similar amongst the VP sets: 579,012 (VP set 1), 583,173 (VP set 2), 586,363 (VP set 3), 588,771 (VP set 4), and 586,229 (VP set 5). All measurements on each VP set were launched simultaneously on November 18th, 2016 with `scamper` probing at a rate of 25 packets/second. The fastest VP set finished the measurements on November 29th, 2016, while the slowest finished on December 6th, 2016.

5 MEASUREMENT RESULTS

This section provides three kinds of analysis to demonstrate the efficiency of our contributions. First, we start by studying and comparing the efficiency of our two most powerful techniques, DPR and BRPR, for revealing IP internal hops (Sec. 5.1). Second, we analyze the *Return & Forward Asymmetry* (i.e., the difference between the return and forward path in term of number of hops), with FRPLA in particular, and we cross-validate it when it intersects with DPR and BRPR (Sec. 5.2). Finally, we study the distribution of return tunnel length with RTLA, its incidence on path asymmetry, and, again, we cross-validate it using DPR and BRPR with a return and forward path asymmetry perspective (Sec. 5.3). We demonstrate thus that most of the invisible tunnels can be identified in some way, either explicitly (DPR or BRPR) or implicitly (FRPLA or RTLA)

Table 4 provides many pieces of information for ASes presenting the largest number of HDNs for which we were able to reveal the content of invisible tunnels (with the exception of AS2856). The two columns labeled "HDNs" refer to the number of HDNs found in the CAIDA dataset ("ITDK") but also to those ("Candidate"), encountered in our measurement campaign, that can potentially act as Ingress or Egress LER. The "I – E pairs" columns refer to IP address pairs, belonging to candidate HDNs, that potentially act as Ingress or Egress LER ("Candidate"). The next column, "%Rev.", provides the proportion of Ingress – Egress pairs for which we were able to reveal the content of hidden tunnels. The next three columns provide raw statistics about revealed MPLS LSPs for those candidates. The column labeled "Raw LSPs" gives the number of unique LSP (as a sequence of IP addresses) we identified, while column "#IPs

[14]See http://www.team-cymru.org/IP-ASN-mapping.html

ISP (ASN)	HDNs		I – E Pairs (IP)		Revealed LSPs			Graph Density	
	ITDK	Candidate	Candidate	%Rev.	Raw LSPs	#IPs LSRs	%IPs LERs	Before	After
Telia (1299)	1,819	1,317	58,548	0.2	102	59	42.4	0.024	0.019
China Telecom (4134)	1,212	1,078	31,728	2.8	1,016	281	61.6	0.008	0.007
Tinet Spa (3257)	1,032	654	12,411	55.1	12,577	1,092	44.2	0.033	0.009
Level 3 (3549)	708	425	9,028	65.6	8,675	757	32.6	0.065	0.007
Deutsche Telekom (3320)	497	364	21,189	68.2	29,395	1,385	40.0	0.108	0.013
Telecom Italia (6762)	346	129	6,235	73.6	7,548	214	83.6	0.236	0.094
Qwest (209)	271	110	1609	28	552	65	0	0.151	0.056
Bharti Airtel (9498)	159	150	11,909	12.5	4,199	493	44.8	0.138	0.041
PCCW Global (3491)	92	57	3,512	52.6	3,704	264	5.3	0.300	0.045
British Telecom (2856)	1,944	148	5656	0.1	3	0	0	0.2	0.2

Table 4: Invisible MPLS tunnels discovery for ASes of interest (I – E stands for Ingress – Egress). Most ASes are Tier-1 or Transit (Tier-2, etc.) ISPs having large inter-connections, possibly resulting in dense HDN graphs.

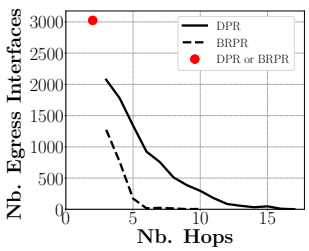


Figure 5: Forward Tunnel Length (FTL).

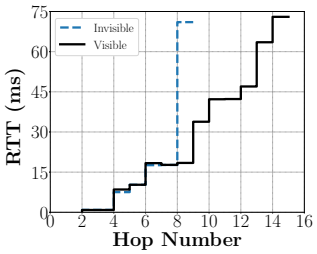
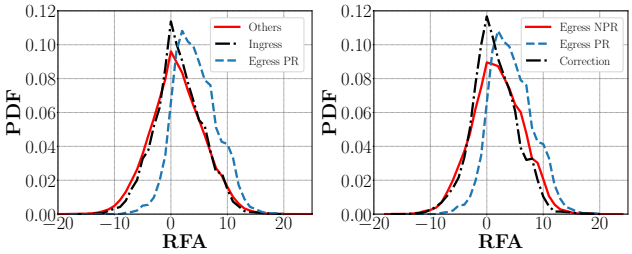


Figure 6: RTT correction with hop revelation (AS3549).



(a) Non HDN and Ingress LER vs. Egress LER.

(b) Corrected distribution of FRPLA with Path Revelation.

Figure 7: Return vs. Forward path Asymmetry (RFA).

LSRs" gives the number of unique IP addresses revealed. The last column is the proportion of those revealed IP addresses also identified as Ingress or Egress LER. Finally, the column "Graph Density" indicates how the density[15] of those ISP graphs is corrected when revealing invisible MPLS tunnels. It is worth noticing that the density is, here, computed only based on Ingress – Egress pairs (and not on the whole ISP graph).

5.1 Path Revelation with DPR and BRPR

In our measurement campaign, a total of 13,771 invisible tunnels were revealed. Among this number, 8,477 were elicited by DPR, 2,270 by BRPR and, finally, 3,024 were too short (i.e., LSP made of a single hop) to determine which measurement technique applies. The additional probing induced by BRPR (i.e., the recursion to reveal the tunnel content) was 8,180 probes.

Fig. 5 illustrates the revealed tunnels length as the number of hops (X-axis) required to reach the tunnel exit point (i.e., Egress LER). A value of 2 means thus a tunnel made of a single LSR. Note that a tunnel of length 1 cannot hide an LSR. The Y-axis provides the raw number of IP addresses acting as Egress LER. The red dot refers to very short tunnels, i.e., a single LSR. In that case, DPR and BRPR are indistinguishable. The distribution does not really look like a power-law with a strong shape and heavy tail. But still, this is a strongly decreasing function bounded with relatively short tunnels, i.e, very few of them exceed 12 hops. This tunnel length distribution is aligned with previous results on visible tunnels [19, 36]. We can also observe on Fig. 5 that the distributions for DPR and BRPR behave differently. This is because DPR discovers the whole tunnel with only one additional trace while BRPR needs one trace for each IP address. A significant share of its attempts may fail before discovering the whole tunnel, resulting in shorter average tunnels. Table 4 shows the number of newly discovered IP addresses for revealed LSPs (column "#IPs LSRs").

Fig. 6 shows the RTT evolution for each hop of a trace traversing an invisible tunnel in AS3549. When the tunnel is invisible (blue dashed line), we observe a jump of about 50 ms in the RTT values between hops 8 (Ingress LER) and 9 (Egress LER). However, once the tunnel has been revealed (black curve), this large delay is actually decomposed between the seven hops of the tunnel.

5.2 Return vs. Forward Asymmetry

Fig. 7 provides the Return & Forward Asymmetry (RFA) distribution. This distribution is based on FRPLA, i.e., it reveals the actual return path length (in terms of IP hops) while the forward path length is underestimated due to the invisibility of the forward tunnel. With RFA, a value of 0 in the distribution is inconclusive for us as it means that return and forward paths have the same length. Similarly, a negative value in the distribution (the return

[15] The density of a graph with E edges and V vertices is $\frac{2\times E}{V\times(V-1)}$.

path is shorter than the forward one) does not bring any information about a potential invisible tunnel. Finally, a positive value is the ideal case for us, as the return path is longer than the forward one. We can therefore assume the presence of an invisible tunnel.

Fig. 7a provides the RFA distribution in several cases. First, we have a look at paths not involved in MPLS tunnels or HDNs, i.e., the red curve ("Others" means any IP address except those tagged as HDN) and the black curve (IP address identified as HDN Ingress LER). In both cases, the path asymmetry follows a normal law centered in 0, with a median value of 1, the symmetry being not perfect. In general, paths between two nodes in the Internet are not the same in both directions. It is due to, among others, BGP hot-potato routing. However, on average over a large number of pairs, the distribution should be (almost) symmetrical.

However, the story is different when HDNs are classified as Egress LER in our campaign (blue curve on Fig. 7a: they are Egress LERs for which we revealed nodes on the forward path – "PR" means "Path Revelation"): the normal law is now significantly shifted towards positive values (median of 4) and a bit flattened out as three values show almost the same level of density (difference of 0, 1, and 2 hops). We also considered all Egress LERs, even those where we do not find any forward paths with a Path Revelation technique (red curve on Fig. 7b – NPR means "No Path Revelation"). Since it underestimates the difference when comparing it to Egress LERs for which we reveal a forward path (Egress PR), we can conclude that FRPLA and RTLA are not really efficient when path revelation does not work either. Generally speaking, the difference observed between the Egress curve and the other curves is due to the return tunnel path length, following a kind of power-law, that is taken into account with FRPLA. This significant shifting of the median is a direct consequence of the forward tunnel invisibility. Indeed, the forward path length does not include the hidden hops, while the complete return path length is obtained based on the TTL in the ICMP replies sent by the Egress LER.

Fig. 7b tries to fix this shift by using the actual lengths of forward paths revealed by DPR or BRPR. This cross validation is performed on the intersection of the path revelation methods and FRPLA. For each revealed tunnel towards an Egress LER, we add the number of revealed hops (either with DPR or BRPR) to the forward path length, and then, we re-analyze the Return and Forward Asymmetry (RFA). We notice that it works very well for most networks. In particular, for all Egress LERs considered in Fig. 7b, we see that the corrected Egress curve (black curve on Fig. 7b) is almost centered at 0 compared to the curves for the non corrected distribution. Again, we observe that the shift in asymmetry is much more remarkable when looking only at forward tunnels revealed with path revelation mechanisms: it means that FRPLA is much more coherent for Egress for which we revealed a tunnel than for those we were unable. These results are aligned with our discussion about the standard configurations when we state that both techniques apply (revelation and length analysis) to the same configurations (see Sec. 3.4).

5.3 Return Tunnel Length

It is worth reminding that RTLA is more accurate than FRPLA, but specific to LERs with $< 255, 64 >$ signature (instead of all LERs

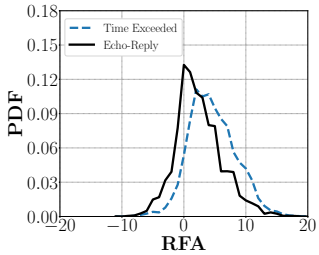
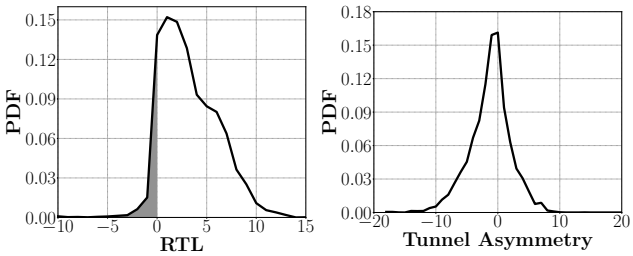


Figure 8: RFA `time-exceeded` and `echo-reply` messages.



(a) Return tunnel length distrib. **(b) Tunnel Asymmetry.**

Figure 9: RTLA with Juniper Egress LER (at the IP level).

for FRPLA). This higher accuracy comes from the fact that it provides exactly the return tunnel length instead of the total return path length (as with FRPLA). This analysis, based on RTLA results, works thanks to our particular campaign design: we specifically target HDN Egress LERs (and their neighbors, again Ingress/Egress LERs or borders of other domains) that are likely to be themselves entry points for tunnels being the first segment of the return path. When building ICMP messages, the Egress LER sets the IP-TTL to its maximum value (255 – `time-exceeded`– or 64 – `echo-reply`– for Juniper, always 255 for Cisco). If 255 is used, the LSE-TTL will be lower or equal to the IP-TTL when exiting the return tunnel on the return path. The *min* behavior will then replace the IP-TTL by the lower LSE-TTL value. However, when using 64 as initial TTL, as it is the case for `echo-reply` on Juniper routers, the LSE-TTL will always be greater that the IP-TTL, and the *min* behavior will let the IP-TTL unchanged. Hence the gap between the path lengths obtained with `time-exceeded` and `echo-reply` messages directly provides the return tunnel length.

Fig. 8 shows the gap between the lengths of return/forward paths (RFA) considering both IP-TTLs of Juniper LERs. If we compare the two curves, we observe a shift towards the positive values for the `time-exceeded` messages. Indeed, the asymmetry of paths does not follow a normal law centered in 0 for this type of message (blue curve), as the median equals 4. However, if we consider `echo-reply` messages (black curve), the distribution is almost centered in 0 (the highest peak is at 0 while the median is 2). The reason is that with an IP-TTL value of 64, the return path does not exhibit a really significant signal of any return tunnel. The asymmetrical shape of the black curve may be due to some TTL operation variation at one hop (the last hop of the return tunnel in particular).

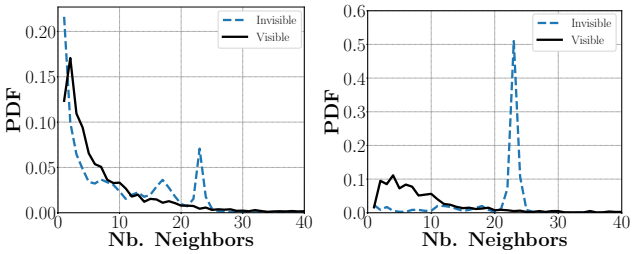
Fig. 9a shows the tunnel length distribution for the return LSP as revealed with RTLA. We can compare it to the one of forward LSP given in Fig. 5. Both distributions look very similar. On Fig. 9a, the low amount of negative values (the shaded area) probably comes from ECMP variations, or other specific return path noise for some of our Vantage Points. On Fig. 9b, we try to assess the accuracy of RTLA. In the fashion of what we have done for Fig. 7b, we subtract the return tunnel length (as depicted in the scheme given in Fig. 3) with the actual forward tunnel length (FTL) obtained either with DPR or BRPR. It seems to work well at this global scale: the distribution almost follows a normal law centered in 0 as expected.

6 MPLS ANALYSIS

One lesson of our measurements is that MPLS deployment, hence its behavior, is greatly variable from one ISP to another, as can be seen from Table 5. In this table, we detail MPLS deployment characteristics for the same networks given in Table 4. First, we sort them considering their signatures (see Table 1): first ranked ASes belong mostly to Cisco while last ranked ones mainly include Juniper devices. Second, we provide the scores of our two active revelation techniques (their relative efficiency) and their possible combination ("Others"). Finally, we show how FRPLA and RTLA perform compared to them for estimating the average tunnel length ("FTL" gives the Forward Tunnel Length in term of number of hops, as revealed with our path revelation technique).

In Table 5, when looking at hardware deployed (TTL signature columns) and hidden hop discovery techniques, we observe two tendencies. First, we have several ASes that show a consistent behavior. For instance, AS3257 (and, to a lower extent, AS9498) is built around Juniper hardware. As expected, the vast majority of hidden IP interfaces is revealed with DPR. On the contrary, while AS3491 deploys mostly Cisco hardware, BRPR succeeds in general to reveal hidden hops. Second, other ASes appear to deploy a mix of router vendors and, and thus, as mentioned in Sec. 3.3, DPR provides better results. It is worth noticing that AS3549 is the only one with a high prevalence of the TTL signature <64,64>, and the most efficient discovery method is DPR. So, the behavior associated to this signature looks similar to the Juniper routers behavior. Another finding for AS3549 (not shown on the Table) is that Juniper seems prevalent at the edge (Ingress and Egress) while the <64,64> signature is prevalent in the core (revealed IP addresses).

The last group of columns in the table looks at the return tunnel length estimation (with FRPLA) and inference (RTLA), and compares them to the forward tunnel length revealed by path revelation techniques. AS2856 is not significant in this case since almost no tunnels were revealed (as stated in Table 4). We see that, for FRPLA, the median is not far from the actual median tunnel length, considering that this method is sensitive to asymmetric routing. RTLA, when feasible (i.e., Juniper Egress routers), provides a value consistent with the tunnel length (see "FTL" in Table 5). This indicates that the TTL behavior at the Egress router on the return tunnel is often the *min* one. This was expected for AS6762 or AS3320 as they mix Cisco and Juniper routers. This is more surprising for AS3257 as it seems fully Juniper. This probably suggests that AS3257 deploys unusual MPLS configuration. Finally, as AS1299 seems to mainly contain really short tunnels (77% of revealed tunnels are classified as "DPR or BRPR", meaning that only one LSR is retrieved). It explains, at least partially, why FRPLA and RTLA do not provide significant information for it.



(a) all ASes. (b) AS3320.

Figure 10: Effects of invisible MPLS tunnels on degree distribution. Peak values disappear.

7 INTERNET MODEL UPDATE

One of the key metrics in Internet modeling comes from the seminal paper by Faloutsos et al. [21]: the *node degree* distribution. This metric gives the proportion of nodes with k adjacencies (for all integer k, see Fig. 1). This distribution may be an indicator of the network resilience to failures and attacks [25]. Faloutsos et al. found that the node degree distribution follows a power-law shape. If this has been heavily questioned in the past [16, 28, 31], we advocate in this paper that invisible MPLS tunnels might artificially increase node degrees, since each Ingress LER appears as the neighbor of all exit points in a given AS. This assumption has been, in this paper, the starting point of measurement techniques for revealing hidden tunnels. Fig. 10 shows the effects of hidden tunnels on the degree distribution, and how this distribution is corrected once the tunnels content is taken into account.

We achieve this as follows: we mapped each Ingress - Egress pair to a router identifier using the Caida ITDK dataset [10]. The obtained graph is used to compute the degree distribution in the invisible case (blue dashed line on Fig. 10). Then, we also mapped all the IP addresses revealed by our techniques in MPLS tunnels. The updated graph is used to compute the degree distribution in the visible case (black line on Fig. 10). If we were unable to perform the mapping with the Caida dataset, we assigned a new identifier to the IP address. However, note that we were able to map 97% of the revealed IP addresses in the Caida dataset. After the mapping, we counted the number of neighbors for the obtained routers.[16] On Fig. 10, the Y-axis provides the PDF, while the X-axis gives the number of neighbors.

Fig. 10a illustrates the degree distribution for all ASes. Two main results arise: as expected, (*i*), when tunnels are hidden (blue dashed line), the proportion of HDNs is larger than when the revealed content is taken into account and, (*ii*), two peaks are observed (for a number of neighbors of 17 and 23). Those two peaks are due to

[16]Note that, for the comparison, we consider the intersection between our dataset and the ITDK one. It justifies that the degrees observed in Fig. 10 are much lower than the ones in Fig. 1 as our dataset is limited to a relatively small sample.

ASN	TTL signature (%)			Hidden Hop Discovery (%)				#Hidden Hops (median)		
	<255,255>	<255,64>	<64,64>	DPR	BRPR	DPR or BRPR	Others	FRPLA	RTLA	FTL
3491	93	0	0	2	74	20	4	4	-	2
4134	73	0	0	13	3	83	1	1	-	1
2856	67	30	1	33	0	67	0	-3	-	1
3320	53	41	0	50	9	2	40	4	2	2
6762	37	53	0	6	69	17	7	4	3	2
209	27	37	0	98	0	2	0	3	2	4
1299	25	74	0	19	3	77	0	0	0	1
3549	11	45	38	73	3	1	24	5	4	5
9498	7	72	0	99	0	1	0	4	4	4
3257	0	96	0	99	0	1	0	4	2	4

Table 5: MPLS deployment per AS. The percentage for TTL signatures is rounded (the total may exceed 100%). "DPR or BRPR" refers to hops revealed by either DPR or BRPR (when only one IP address is discovered in a tunnel, there is no difference between the two methods). "Others" refers to a mix of discovery techniques (a tunnel might be, in some case, revealed by DPR and, in another trace, by BRPR).

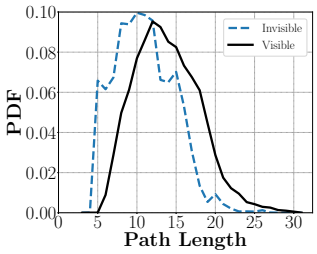


Figure 11: Effects of invisible MPLS tunnels on path length distribution for all ASes.

two ASes in particular: the peak at 23 is caused by invisible tunnels in AS3320 (Deutsche Telekom – see Fig. 10b), while the other one is due to AS3549 (Level3 – not shown here due to space limitations). Focusing on a given AS (as in Fig. 10b) provides very insightful results: for AS3320, we identify a kind of full-mesh made of 23 routers (a representative sample of the real network) that we are able to turn into a more general graph where the shape of the degree distribution becomes standard. This is confirmed by the graph density analysis for AS3320, as provided by Table 4. Indeed, its density is divided by a factor of ten once invisible tunnels are revealed.

The *path length* (i.e., the number of hops between two devices in the network) is an important metric for modeling the Internet topology as it takes part into the *shortest path* (i.e., the path offering the minimum distance between a given pair of nodes), the *average path length* (i.e., the average length of shortest paths for all pairs of networking devices), or the *graph diameter* (i.e., the longest shortest path) [33]. Obviously, if many nodes are hidden by long invisible MPLS tunnels, the path length distribution will be biased and the resulting inferred model (such as small world) biased.

Fig. 11 shows the effects of invisible MPLS tunnels on path length distribution (blue dashed line) and the shift when hidden routers are revealed by our methods (black plain line). This has been computed on the data we collected (see Sec. 4). If both distributions (invisible and visible) more or less display the familiar bell-shaped curve typical of Internet distance distributions, it is clear that, by revealing hidden hops, we can observe a shift towards longer routes. In particular, the mean is at 10 with invisible tunnels, while it is at 12 when one lifts the curtain on MPLS tunnels. In addition, it is worth noticing that it is still an underestimation because, when a trace goes through several invisible tunnels, our current set of techniques only reveal the last one. Thus, as a significant share of routing traces is likely to traverse up to two invisible MPLS networks, one may conclude that the actual length shift may be higher. Finally, as most of our targets are Egress LERs belonging to Tier-1 networks, one may multiply by almost a factor two our results about path length to infer the typical length of routes in the Internet (we did not take into account the descending route from the Tier-1 to another Stub network).

Obviously the results presented in this section are only illustrations of the effect of invisible tunnels on basic graph characteristics as seen with our dataset. Much more extensive measurement campaigns and analyses are required as far as the whole Internet is concerned.

8 CONCLUSION

In this paper, we presented and evaluated several kinds of techniques for revealing IP level information hidden by invisible MPLS tunnels. Our set of active and analytical mechanisms allowed us to provide insights about standard MPLS practices of ISPs. Besides, we revisited some basic Internet graph characteristics that are biased by invisible MPLS tunnels.

We validated our set of active techniques through emulation, cross-validation and a survey of operators and, then, implemented them on PlanetLab. We also propose and validate two analytical techniques. To summarize, we distinguished between a set of techniques (FRPLA and RTLA) able to provide the length distribution of invisible tunnels, and others (DPR and BRPR) that indeed reveal the IP hops hidden by invisible tunnels. In particular, FRPLA has the advantage of being scalable (as it is a pure analytical technique) and to work with any IP level dataset, as it only relies on

Brand	MPLS		Trigger		Revelation	
	LDP	Popping	FRPLA	RTLA	DPR	BRPR
Cisco	all prefixes	PHP	✓	–	–	✓
Juniper	loopback	PHP	(✓)	✓	✓	(✓)

Table 6: Measurement techniques applicability.

standard `traceroute` campaigns. RTLA is based on a similar analysis as FRPLA that studies the return path of replies, but it requires an additional `echo-request` per IP address. In practice, for tunnels endpoints being Juniper routers, it provides a more accurate estimation of the invisible tunnels length than FRPLA. Those two techniques are enough to determine whether an AS hides an invisible MPLS cloud. They are also sufficient to evaluate the stretch in terms of Internet path length caused by invisible MPLS tunnels.

The DPR and BRPR techniques imply a more specific and complex measurement campaign since route tracing is aimed at dynamically revealing IP addresses originally hidden by MPLS tunnels. This additional IP level information allows us to gain knowledge on the internal architecture of opaque MPLS ASes. More generally the Internet graph and its node degree distribution in particular can be corrected. Finally, we identified a few ASes where our techniques did not succeed, while they claim to deploy MPLS features (according to their websites). This is probably because they use MPLS only with UHP, for VPN and/or traffic engineering, leaving tunnels truly invisible for the time being.

In this work, the measurement campaign has been driven by the presence of abnormal high degree nodes in the router level topology. Those nodes were a trigger for performing dedicated invisible MPLS tunnel discovery. However, in this paper, we have shown that FRPLA and RTLA techniques are able to infer the presence of invisible tunnels. We could then envision a modification of `traceroute`, using FRPLA and RTLA as triggers for the presence of invisible tunnels, and BRPR and DPR to reveal the internal nodes on the fly, as suggested by Table 6.

ACKNOWLEDGMENTS

We would like to thank operators for taking time to answer to their survey. We are also grateful to the IMC reviewers and our shepherd, Rob Beverly, for their useful feedback on the paper.

REFERENCES

[1] P. Agarwal and B. Akyol. 2003. *Time-to-Live (TTL) Processing in Multiprotocol Label Switching (MPLS) Networks.* RFC 3443. Internet Engineering Task Force.
[2] Z. Al-Qudah, M. Alsarayreh, I. Jomhawy, and M. Rabinovich. 2016. Internet Path Stability: Exploring the Impact of MPLS Deployment. In *Proc. IEEE Global Communication Conference (GLOBECOM).*
[3] L. Andersson and R. Asati. 2009. *Multiprotocol Label Switching (MPLS) Label Stack Entry: EXP Field Renamed to Traffic Class Field.* RFC 5462. Internet Engineering Task Force.
[4] L. Andersson, I. Minei, and T. Thomas. 2007. *LDP Specification.* RFC 5036. Internet Engineering Task Force.
[5] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. 2006. Avoiding Traceroute Anomalies with Paris Traceroute. In *Proc. ACM Internet Measurement Conference (IMC).*
[6] B. Augustin, R. Teixeira, and T. Friedman. 2007. Measuring Load-Balanced Paths in the Internet. In *Proc. ACM Internet Measurement Conference (IMC).*
[7] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. 2001. *RSVP-TE: Extensions to RSVP for LSP Tunnels.* RFC 3209. Internet Engineering Task Force.
[8] D. Aydin. 2014. CISCO vs. Juniper MPLS. (June 2014). See http://monsterdark.com/cisco-vs-juniper-mpls/.
[9] R. Bonica, D. Gan, D. Tappan, and C. Pignataro. 2007. *ICMP Extensions for Multiprotocol Label Switching.* RFC 4950. Internet Engineering Task Force.
[10] Center for Applied Data Analysis. 2016. The CAIDA UCSD Internet Topology Data Kit. (March 2016). See http://www.caida.org/data/internet-topology-data-kit.
[11] CISCO. [n. d.]. *CISCO ASR9922 Router.* see https://goo.gl/KYyfbr.
[12] CISCO. [n. d.]. *CISCO Line Cards.* see https://goo.gl/XqUN3q.
[13] Cisco. 2013. *MPLS Label Distribution Protocol Configuration Guide, Cisco IOS Release 15S.* Cisco, Chapter MPLS LDP Local Label Allocation Filtering. See https://goo.gl/rF975K.
[14] Cisco. 2017. *Segment Routing Configuration Guide, Cisco IOS XE Release 3S.* Cisco Press.
[15] kc claffy, Y. Hyun, K. Keys, M. Fomenkov, and D. Krioukov. 2009. Internet Mapping: from Art to Science. In *Proc. IEEE Cybersecurity Applications and Technologies Conference for Homeland Security (CATCH).*
[16] A. Clauset and C. Moore. 2004. *Traceroute Sampling Makes Random Graphs Appear to Have Power Law Degree Distributions.* cond-mat 0312674. arXiv.
[17] L. De Ghein. 2006. *MPLS Fundamental: A Comprehensive Introduction to MPLS (Theory and Practice).* CISCO Press.
[18] B. Donnet and T. Friedman. 2007. Internet Topology Discovery: a Survey. *IEEE Communications Surveys and Tutorials* 9, 4 (December 2007), 2–15.
[19] B. Donnet, M. Luckie, P. Mérindol, and J.-J. Pansiot. 2012. Revealing MPLS Tunnels Obscured from Traceroute. *ACM SIGCOMM Computer Communication Review* 42, 2 (April 2012), 87–93.
[20] P. Erdös and A. Rényi. 1960. On the Evolution of Random Graphs. *Publ. Math. Inst. Hung. Acad. Sci.* 5 (1960), 17–61.
[21] M. Faloutsos, P. Faloutsos, and C. Faloutsos. 1999. On Power-Law Relationships of the Internet Topology. In *Proc. ACM SIGCOMM.*
[22] T. Flach, E. Katz-Bassett, and R. Govindan. 2012. Quantifying Violations of Destination-Based Forwarding on the Internet. In *Proc. ACM Internet Measurement Conference (IMC).*
[23] R. Fontugne, E. Aben, C. Pelsser, and R. Bush. 2017. Pinpointing Delay and Forwarding Anomalies Using Large-Scale Traceroute Measurements. In *Proc. ACM Internet Measurement Conference (IMC).*
[24] G. Geshev. 2015. Warranty Void if Label Removed: Attacking MPLS Networks. In *Proc. Zero Nights.* see http://2015.zeronights.org/assets/files/02-Geshev.pdf.
[25] J.-L. Guillaume, M. Latapy, and C. Magnien. 2004. Comparison of Failures and Attacks on Random and Scale-Free Networks. In *Proc. 8th International Conference on Principles of Distributed Systems (OPODIS).*
[26] H. Haddadi, G. Iannaccone, A. Moore, R. Mortier, and M. Rio. 2008. Network Topologies: Inference, Modeling and Generation. *IEEE Communications Surveys and Tutorials* 10, 2 (April 2008), 48–69.
[27] Juniper. 2014. Configuring the Prefixes Advertised into LDP from the Routing Table. (December 2014). See https://goo.gl/jwdr4Q.
[28] A. Lakhina, J. Byers, M. Crovella, and P. Xie. 2003. Sampling Biases in IP Topology Measurements. In *Proc. IEEE INFOCOM.*
[29] M. Luckie. 2010. Scamper: a Scalable and Extensible Packet Prober for Active Measurement of the Internet. In *Proc. ACM Internet Measurement Conference (IMC).*
[30] M. Luckie, A. Dhamdhere, B. Huffaker, D. Clark, and k claffy. 2016. bdrmap: Inference of Borders Between IP Networks. In *Proc. ACM Internet Measurement Conference (IMC).*
[31] P. Mérindol, B. Donnet, O. Bonaventure, and J.-J. Pansiot. 2010. On the Impact of Layer-2 on Node Degree Distribution. In *Proc. ACM Internet Measurement Conference (IMC).*
[32] K. Muthukrishnan and A. Malis. 2000. *A Core MPLS IP VPN Architecture.* RFC 2917. Internet Engineering Task Force.
[33] R. Pastor-Satorras and A. Vespignani. 2004. *Evolution and Structure of the Internet: A Statistical Physics Approach.* Cambridge University Press.
[34] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, and A. Conta. 2001. *MPLS Label Stack Encoding.* RFC 3032. Internet Engineering Task Force.
[35] E. Rosen, A. Viswanathan, and R. Callon. 2001. *Multiprotocol Label Switching Architecture.* RFC 3031. Internet Engineering Task Force.
[36] J. Sommers, B. Eriksson, and P. Barford. 2011. On the Prevalence and Characteristics of MPLS Deployments in the Open Internet. In *Proc. ACM Internet Measurement Conference (IMC).*
[37] C. Srinivasa, L. P. Bloomberg, A. Viswanathan, and T. Nadeau. 2004. *Multiprocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB).* RFC 3812. Internet Engineering Task Force.
[38] Y. Vanaubel, P. Mérindol, J.-J. Pansiot, and B. Donnet. 2015. MPLS Under the Microscope: Revealing Actual Transit Path Diversity. In *Proc. ACM Internet Measurement Conference (IMC).*
[39] Y. Vanaubel, P. Mérindol, J.-J. Pansiot, and B. Donnet. 2016. A Brief History of MPLS Usage in IPv6. In *Proc. Passive and Activement Measurement Conference (PAM).*
[40] Y. Vanaubel, J.-J. Pansiot, P. Mérindol, and B. Donnet. 2013. Network Fingerprinting: TTL-Based Router Signature. In *Proc. ACM Internet Measurement Conference (IMC).*

[41] N. Wang, K. Ho, G. Pavlou, and M. Howarth. 2008. An Overview of Routing Optimization for Internet Traffic Engineering. *IEEE Communications and Surveys Tutorials* 10, 1 (April 2008), 36–56.

[42] W. Willinger, D. Alderson, and J. C. Doyle. 2009. Mathematics and the Internet: a Source of Enormous Confusion and Great Potential. *Notices of the American Mathematical Society* 56, 5 (May 2009), 586–599.