Unveiling the MPLS Structure on Internet Topology

Gabriel Davila Revelo†, Mauricio Anderson Ricci† , Benoit Donnet*, José Ignacio Alvarez-Hamelin†‡
† INTECIN, Facultad de Ingeniería, Universidad de Buenos Aires – Argentina
{gdavila, anderson, ihameli}@cnet.fi.uba.ar
‡ CONICET – Argentina
∗ Université de Liège, Liège – Belgium, benoit.donnet@ulg.ac.be

***Abstract*— Recently, researches have been conducted to discover and assess the usage of MPLS tunnels. Indeed, recent developments in the ICMP protocol make certain categories of MPLS tunnels transparent to `traceroute` probing. Additional techniques have been proposed to reveal the presence of MPLS tunnels when they do not explicitly appear in `traceroute`. It has been shown that MPLS is a very well deployed technology whose usage (i.e., Traffic Engineering, load balancing, etc.) varies in time and according to ASes. However, the MPLS structure on the Internet architecture has not been studied yet. In this paper, we follow this path by providing two contributions to the state of the art: (*i*) we evaluate the biases involved on MPLS tunnel detection when they are not directly revealed through `traceroute`. (*ii*), we provide some properties and architectural details related with MPLS deployment on router topology based on a k-core decomposition.**

I. Introduction

Internet topology refers to the study of the various types of connectivity structures and representations between directly connected nodes on the Internetq architecture. This representation aims at modeling the Internet with the greatest possible accuracy in order to test new communications protocols, algorithms, QoS policies, traffic engineering, etc.

The Internet topology can be seen at several abstraction levels i.e., IP interface, router, subnetwork, PoP, and Autonomous System (AS) levels. All these models have been widely studied in the past [1]. However, the current state of the art of Internet deployment involves a great number of technologies impacting the Internet Topology. And those technologies deserve a deep study in order to include them in the current Internet models. For instance, *Multiprotocol Label Switching* (MPLS) [2] has been recently the focus of several studies [3], [4], [5]. It has been demonstrated that MPLS is a mature technology widely deployed for (mainly) load balancing reasons or traffic engineering purposes. A few studies have partially questioned its impact on Internet topology [6], [7]. However, the MPLS structure on the Internet architecture has not been studied yet. The importance to study the architectural details of MPLS usage would help to know the way in which the Internet Service Providers (ISPs) use their networks or apply their policies for traffic engineering as well as to better understand the Internet architecture more accurately.

This work mainly provides a study around the structure of MPLS usage over Internet Topology. Firstly, we focus on evaluating the accuracy of MPLS tunnel detection methods. Particularly, we provide a quantification of the biases related with MPLS tunnels that are not revealed explicitly by `traceroute`. Secondly, we study the MPLS structure based on the way MPLS routers and networks interact with non-MPLS capable routers. In order to do it, we define a new abstraction level on Internet graph, distinguishing router-level and MPLS-level links. We also identified non-MPLS capable routers and MPLS clusters. In this way, we contribute to the traditional Internet topology with new details related to MPLS usage. We mainly use k-core decomposition [8] based tools to reveal the fingerprints closely related to MPLS presence. It has been shown previously that k-core decomposition is a relevant tool to describe Internet Topology [9], [10], [11]. Our main findings reveal that the MPLS structure varies depending on the type of MPLS tunnels that prevails for a given AS and that MPLS deployment plays an important role on the Internet Backbone. Specifically, we find that local robustness of Internet topology increase due to MPLS presence.

The remainder of this paper is organized as follows: Sec. II provides the state of the art and the background related to MPLS tunnels discovery. In particular, it describes how MPLS tunnels can be revealed through active measurements; Sec. III explains how we collected data for this work; Sec. IV presents our results related to *mpls signatures* accuracy; Sec. V presents the main contributions of this paper with a detailed study around the behavior of MPLS networks on the Internet Topology and architectural details of some ASes with most MPLS usage; Finally, Sec. VI concludes this paper by summarizing its main achievements.

II. Related Work

In this section, we first provide an overview of MPLS (Sec. II-A) before explaining how MPLS tunnels can be revealed through active measurements (Sec. II-B). We also position this work regarding the state of the art.

A. MPLS Overview

The *Multiprotocol Label Switching* (MPLS) [2] was originally designed to speed up the forwarding process. In practice, this was done with one or more 32 bits *label stack entries* (LSE) inserted between the frame header (Data-link layer) and the IP packet (Network layer). A given packet can manage several LSEs at the same time. In this case, the packet carries a *stack of labels*. Each LSE is made of four fields: a 20-bit label value used for forwarding the packet to the next router, a 3-bit Traffic Class field for quality of service (QoS), priority, and Explicit Congestion Notification (ECN) [12], a 1-bit bottom

of stack flag (when set the current label is the last in the stack [13]), and an 8-bit time-to-live (LSE-TTL) field having the same purpose as the IP-TTL field [14].

MPLS routers, called *Label Switching Routers* (LSRs), exchange labelled packets over *Label Switched Paths* (LSPs). The first MPLS router (*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. In some cases, for performance reasons, the LSE stack may be removed by the penultimate MPLS router (*penultimate hop popping*, PHP). The Egress LER then performs a classic IP lookup and forwards the traffic, reducing so the load on the Egress LER (specially if the Egress LER is shared among several LSPs). This means that, when using PHP, the tunnel exit is one hop before the Egress LER.

B. Revealing MPLS Tunnels

MPLS routers may send ICMP `time-exceeded` messages when the LSE-TTL expires. In order to debug networks where MPLS is deployed, routers may also implement RFC4950 [15], an extension to ICMP allowing a router to embed an MPLS LSE in an ICMP `time-exceeded` message. In that case, the router simply quotes the MPLS LSE (or the LSE stack) of the received packet in the ICMP `time-exceeded` message. RFC4950 is particularly useful for operators as it allows them to verify the correctness of their MPLS tunnels and traffic engineering policy.

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 when using `traceroute` via ICMP messages even if they do not implement RFC4950. Operators can configure this action using the `ttl-propagate` option provided by the router manufacturer [14] (while, to the best of our knowledge, the RFC4950 is just a matter of implementation and cannot be deactivated on recent routers supporting it).

Using those two features, Sommers et al. [3] provide an extensive study of MPLS tunnels as observed in CAIDA's topology data. In this data, they find tunnels in 7% of ASes, and the fraction is constant over the years of data considered. Recently, Vanaubel et al. [16] focus on MPLS deployment and usage under IPv6. Vanaubel et al. [5] propose a classification of path diversity according to MPLS deployment. Their classification reveals the actual usage of MPLS (e.g., load balancing, traffic engineering) according to the inferred label distribution protocol. Finally, it has also been demonstrated that MPLS tunnels may have an impact on Internet topology discovery tools. For instance, the presence of MPLS tunnels may interfere with load balancing detection [6] or violate the destination-based forwarding [7].

Donnet et al. [4] propose a taxonomy of MPLS tunnels based on how they react to `traceroute` probes according to their compliance (or not) to RFC4950 for MPLS and the `ttl-propagate` option. The classes proposed are: *explicit tunnels* (i.e., `ttl-propagate` and RFC4950 are enabled), *implicit tunnels* (i.e., the router that pushes the MPLS label enables the `ttl-propagate` option but LSRs do not implement RFC4950), *opaque tunnels* (i.e., the LH implements RFC4950 but the ingress LER does not enable the `ttl-propagate` option), and, finally, *invisible tunnels* (i.e., the ingress LER does not enable the `ttl-propagate` option and RFC4950 is not implemented by the LH router). Implicit and opaque tunnels can be revealed as follows:

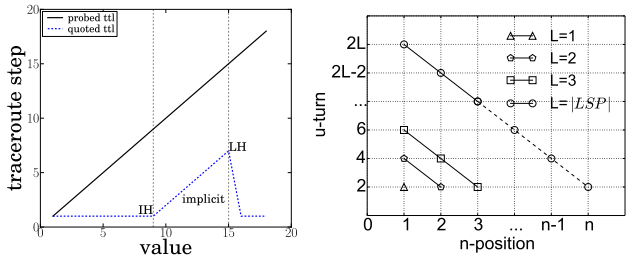
1) a quoted IP-TTL (*qTTL*) in ICMP `time-exceeded` messages > 1 will likely reveal the `ttl-propagate` option at the ingress LER of an LSP. For each subsequent `traceroute` probe within an LSP, the qTTL will be one greater resulting in an increasing sequence of qTTL values in `traceroute`. This is illustrated in Fig. 1(a);
2) #hops differences with the IP-TTL in `echo-reply` messages (*u-turn*). It relies on the fact that LSRs along an LSP present an *original label stack* default routing behavior: when the LSE-TTL expires, an LSR first sends the `time-exceeded` reply to the Egress LER which then forwards the reply on its own to the probing source, while an LSR replies to other probes using its own IP routing table if available. Thereby, u-turn is the signature related with the difference in these values. Summarizing, $\textit{u-turn} = \text{TTL}_{\text{echo-reply}} - \text{TTL}_{\text{time-exceeded}}$. The expected u-turn value is in the form $[2L, 2L-2, 2L-4, ..., 2]$ where L is the LSP length and the array position corresponds to the LSR position within the LSP.
3) opaque tunnels are revealed through the quoted LSE-TTL returned by the LH in the `time-exceeded` reply. By comparing this quoted LSE-TTL and 255 (which is the standard initial LSE-TTL value), one can reveal the presence of an opaque tunnel and its length [4].

Additional study by Vanaubel et al. [17] shows that the probing heuristic to detect implicit tunnels seems quite reliable. However, u-turn signatures are by definition more subject to false positives than qTTL ones. This is exactly what we tackle in this paper (and, consequently, our work is complementary to Vanaubel et al. [17]): we want to test u-turn signature accuracy.

III. Data Collection

In order to collect MPLS data, we develop a tool called Magallanes [18] allowing us to easily run and manage `scamper` [19] based probes through the PlanetLab (PL) infrastructure. Magallanes starts by randomly allocating several vantage points (VP) within the available set of PL nodes. It next distributes, among those VPs, a given number of destinations (or "probe targets"). To achieve some geographical uniformity in target selection, Magallanes select randomly targets from data provided by IP geolocation database maxmind.[1] Additionally, Magallanes allows one to store an experiment results on a centralized database and to perform alias resolution using MIDAR [20].

[1] See `www.maxmind.com`.



(a) qTTL signature. (b) u-turn signature.

Fig. 1. Signatures behavior for implicit MPLS tunnels.

We ran MAGALLANES on October 31st, 2015. We chose 100 VPs and selected 10,000 targets per VP [2]. Each VP managed its own set of targets, meaning that probes targets are disjoint sets between VPs. `scamper` was configured to run ICMP Paris traceroute [21]. To get the u-turn signature, we sent a `ping` to each hop revealed by Paris traceroute. We sent six ICMP `echo-request` packets from the same VP. Six ICMP `echo-reply` allow us to infer with 95% confidence if there is a single return path and, therefore, reduce measurement errors caused by a reverse path containing load-balanced segments of different lengths [6].

As a result we discovered around 270,000 IP interfaces, 520,000 links, 42% of which were available to run MIDAR and we found aliases successfully on 19% of them. To match IP interfaces to ASes, we used the CAIDA dataset [22] derived from Routeviews[3] and collected the same day as the exploration. Additionally, we found that 44% of traces collected traverse at least one MPLS tunnel. The amount of explicit tunnels is highly superior to implicit ones. We discovered explicit tunnels on 34% of traceroutes and at least one implicit tunnel on 16%. Surprisingly, we found more implicit tunnels revealed through u-turn signature (12%) rather than qTTL signature (4%). However, the qTTL signature matched with at least 63% of the explicit tunnels. We discuss these results in the next sections. Finally, we did not found opaque tunnels, confirming so their rarity [17].

IV. MPLS SIGNATURES VALIDATION

In this section, we expose our methodology for validating the MPLS signatures used to reveal implicit MPLS tunnels (see Sec. II-B). Basically, we compare the LSR position within an MPLS tunnel, called the *MPLS position*, with the different signatures values. Our main goal here is to assess the u-turn accuracy.

The MPLS position of an LSR is obtained based on its appearance order in an LSR, as revealed by `traceroute`. The appearance order is called *n-position*, i.e., the first LSR revealed by a `traceroute` probe should be the first LSR within the LSP (1-position) , the LSR revealed by the next consecutive `traceroute` probe should be the second LSR within the LSP (2-position), etc. Given that MPLS tunnels can be configured to perform load balancing (this is quite common, as shown by Vanaubel et al. [5]), the n-position revealed by `traceroute` might lead to a bias with respect to the actual MPLS position. This is illustrated in Fig. 2.

[2]The collected dataset is available at `http://cnet.fi.uba.ar/Sup_Mat_TMA_2016/`.

[3]See `www.routeviews.org`

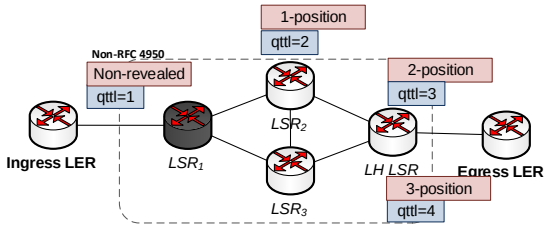
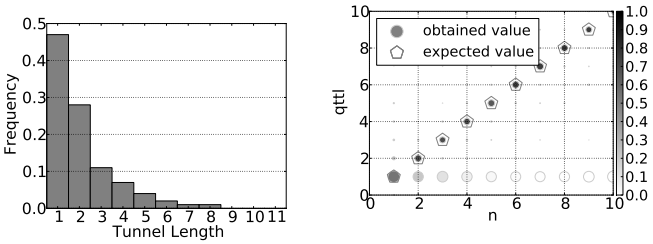


Fig. 2. Example of biased n-position. It shows a scenario where, due to per-packet load balancing issues and the absence of RFC4950 on LSR_1, the n-position could be erroneously inferred: First, `traceroute` reveals LSR_2 (and associates it the 1-position), the next `traceroute` probe reveals the LH LSR (and associates it the 2-position). Finally, due to load balancing issues on LSR_2 the LH LSR is revealed twice (with different qTTL value).



(a) Tunnel length distribution (b) qTTL and n-position comparison

Fig. 3. Comparison between obtained and expected values for qTTL and u-turn.

Implicit tunnels are based either on qTTL or u-turn signatures. Both of them are directly related with MPLS position. Indeed, first, the qTTL value refers to the IP-TTL of the `echo-request` packet when it enters the MPLS tunnel. Therefore, a qTTL of n in the resulting ICMP `time-exceeded` means that the sent probe expired n hops later than the Ingress LER of the LSP, i.e., an LSR reply with qTTL$= n$ means that the LSR appears in the *n-position* in the LSP. This is illustrated in Fig. 1(a). From the Ingress LER, the qTTL starts to grow linearly with the LSP length. We therefore expect observing a qTTL=1 on the first LSR in the LSP, a qTTL$= 2$ on the second LSR in the LSP, etc. Second, a u-turn value is related to the tunnel length, L, and the n-position of the LSR within the tunnel (see Sec. II-B) as is shown on Fig. 1(b).

A. qTTL Signature

Our signature validation relies on the hypothesis that the actual MPLS position matches with the n-position, i.e., the n-position of the LSR within the LSP corresponds to the qTTL value generated by that LSR. Said differently qTTL = n.

In order to validate this assumption, we use the dataset described in Sec. III and compare the qTTL with n-position for explicit and implicit tunnels. The results are shown in Fig. 3. In particular, Fig. 3(a) provides the MPLS tunnel length distribution computed as the number of LSRs in the tunnel.

We observe, corroborating so previous studies [3], [5], [4], that most of tunnels are rather short (length < 3 in more than 80% of the cases).

Fig. 3(a) also provides, by extension, possible values for qTTL (X-axis). This suggests thus that qTTL values should oscillates between 1 and 8, with a strong predominance for short values (i.e., between 1 and 3).

Fig. 3(b) represents a scatter plot showing the relationship between the qTTL (Y-axis) and the n-position (X-axis). The circle size in the scatter plot is related with the occurrence frequency of Y-axis values regarding each n-position. The transparency of the circle is related with occurrence frequency of the n-position regarding each Y-axis value. For instance, on Fig. 3(b) for values where $n > 1$, the biggest circles are mainly located on qTTL$= 1$ and qTTL$= n$. So, this suggests that, for a given n-position, the qTTL value usually takes either the value 1 or n.

However, we notice, on Fig. 3(b) that the qTTL signature highly matches with n. The bias $qTTL = n \pm \epsilon$ could occur due to two causes: one is the limitation in our method to reveal the first LSR in the LSP when RFC4950 is not implemented (by definition of implicit tunnel); and the second cause could occur due to load balancers (using Paris traceroute should avoid load balancing issues, except for "per packet" load balancers), as suggest Fig. 2 suggests. Fig. 3(b) also shows that qTTL frequently takes the value of 1, even for $n > 1$, which means that the LSR implements the RFC4950 but do not match with the qTTL signature.

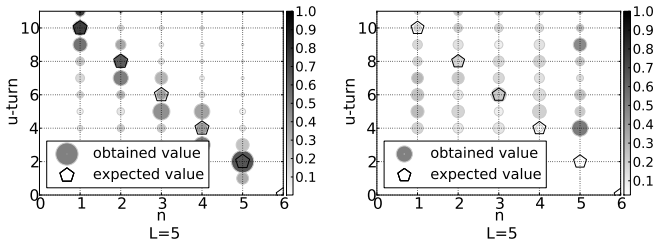
We also find that around 2% of LSRs do not react to qTTL signature, even if their neighbors does, i.e., some LSRs interfaces located at $i_{n\pm1}$ tunnel positions react properly to qTTL signatures but the LSR interface located at i_n position does not.

Nevertheless, the n-position is highly reliable and therefore, the potential load balancer presence on LSPs is not a common issue. Indeed, we find that in 58% of the cases the n-position matches with the qTTL value while in $36,3\%$ of cases the qTTL signature is not present on explicit tunnels and takes the value of 1, and just $6,7\%$ of the cases presented have some bias around the expected value n. Those results support our hypothesis: the MPLS tunnel position highly matches with the n-position. Thereby, we use n-position as a reference value to validate the u-turn signatures.

B. u-turn Signature

As explained in Sec. II-B, the expected u-turn value is of the form $[2L, 2L-2, 2L-4, ..., 2]$ where L is the tunnel length and the array position corresponds to the LSR position within the LSP, i.e., n (see Fig. 1(b)). The relationship between L, n, and the expected value can thus be written $u - turn = 2 \times (L - (n-1))$.

Because u-turn is commonly present in almost all LSRs, first, we compare n with u-turn on LSRs revealed either explicitly or qTTL-based using the dataset presented in Sec. III (Fig. 4(a)). We also study n value on LSRs where u-turn was the only detected signature (Fig. 4(b)). We use the filter u-turn > 3 (i.e., avoiding short tunnels where biases are more likely to appear) to avoid false positives.



(a) u-turn on LSRs revealed through RFC4950 and qTTL

(b) u-turn on LSRs where no other signature was found

Fig. 4. Comparison between obtained and expected values for u-turn signature.

The results for a given tunnel length $L = 5$ are shown on Fig. 4, a scatter plot that must be read the same way as Fig. 3(b). Quickly said, Fig. 4 suggests that u-turn is usually overestimated. Similar results were observed for other tunnel lengths.

We notice that obtained u-turn values are close to expected ones when the LSR was either explicitly revealed or when qTTL is present (Fig. 4(a)). However, for LSRs revealed only by u-turn signatures (Fig. 4(b)) the obtained and expected u-turn values commonly do not match. If we accept a bias of ± 2 around the expected u-turn value over our whole dataset, we notice that, on LSRs explicitly revealed or qTTL based, the 60% of obtained u-turn signatures match with the expected values. However, for LSRs revealed only through u-turn signature, the obtained u-turn signature just match in less than 25% with the expected values.

Therefore, LSRs revealed only through u-turn are highly inaccurate, mainly, because MPLS tunnels are not the only responsible for u-turn signature. Indeed, it is also related with load balancing on the return path, where ICMP `echo-reply` and ICMP `time-exceeded` at different hops may be load balanced [21].

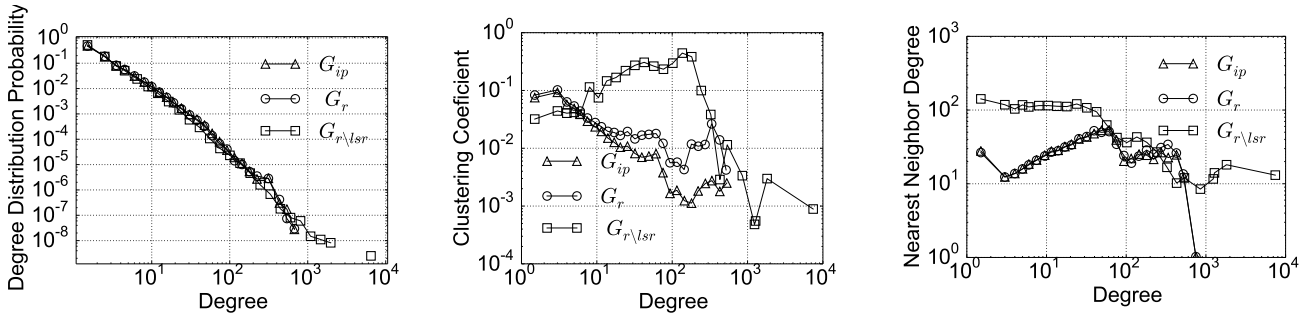
Up to now, we showed u-turn signature's inaccuracy. We also believe it is important to know whether u-turn signature presence has an impact on the Internet architecture. We tackle this (and other interesting questions) in Sec. V.

V. LSRs and MPLS clusters

This section aims at better understanding the impact of MPLS deployment over Internet, specifically over router level topology. To achieve our purpose, we study how LSRs and *MPLS clusters* interacts with non MPLS routers. Generally speaking, we compare the fingerprints on k-core decomposition that MPLS presence causes over the Internet Topology. We propose two main studies: the first one aims at locating the LSRs and MPLS clusters over the entire router topology; and a second one aims at better understanding the MPLS structure for a given AS.

A. Definitions and Background

We define several graphs at different abstraction levels as follows: first, the *IP level graph* G_{ip} is built with the IP



(a) Degree Distribution Probability (b) Clustering Coefficient (c) Neighbor Degree Distribution

Fig. 5. Metrics for IP, router and MPLS cluster interconnection topologies.

addresses and links found through `traceroute`. Second, the *router level graph* G_r is obtained after solving alias resolution process through MIDAR [20]). Third, the *MPLS router level graph* G_r^{mpls} is formed by MPLS links and routers in which at least one IP interface belongs to an LSP.

The *ASes induced graph* $G_r(as)$ is a subgraph of G_r where each vertex has an interface belonging to the same Autonomous System, namely as. In particular, the induced graph of G_r^{mpls} is $G_r^{mpls}(as)$. A connected component C_i^{mpls} in a $G_r^{mpls}(as)$ is called *MPLS cluster*. Finally, the *MPLS cluster interconnection graph* is a hybrid router level graph, $G_{r\setminus lsr}$, where all the MPLS clusters C_i^{mpls} are gathered together in a single node, while non-MPLS capable routers remain unchanged. Broadly speaking, an MPLS cluster interconnection graph refers to a router level graph where all MPLS clusters are treated as a single node. Additionally, we call $G_{r\setminus lsr}(as)$ the subgraphs of $G_{r\setminus lsr}$ induced by routers having at least one interface in the Autonomous System as.

This section mainly focuses on MPLS clusters interconnection graph $G_{r\setminus lsr}$ and their respective ASes induced graphs $G_{r\setminus lsr}(as)$. In this way, we study how MPLS clusters are connected to non-MPLS capable routers. Particularly, as MPLS clusters interconnection graph analysis is mainly based on k-core decomposition, we present the following definitions:

- *k-core:* Given a graph $G = (V, E)$, then the subgraph $H = (C, E|C)$ induced by the set $C \subseteq V$ is a *k*-core of order k iff $\forall v \in C : degree_H(v) \geq k$ and H is the maximum subgraph with this property.
- *Shell index*. A vertex i has a shell index c if it belongs to the c-core but not to $(c+1)$-core. We denote by C_i the shell index of vertex i. A shell C_c consists of all the vertices whose shell index is c. The maximum value c such that C_c is not empty is denoted by $C_{\max}$. Therefore, the k-core is thus the union of all shells C_c with $c \geq k$.
- *Core-connectivity* [11]. Let a core-connected graph, then vertices (i, j) having shell-index a and b respectively, has at least $k = \min(a, b)$ different paths to join i to j.

To retrieve the k-core decomposition of a graph G, we use LANET-VI [9], [11]. This tool returns a two dimensional plot, where the position of each vertex is arranged into a circle depending on its shell index and its neighbors' index. A color code allows for the identification of shell indices, and diameter

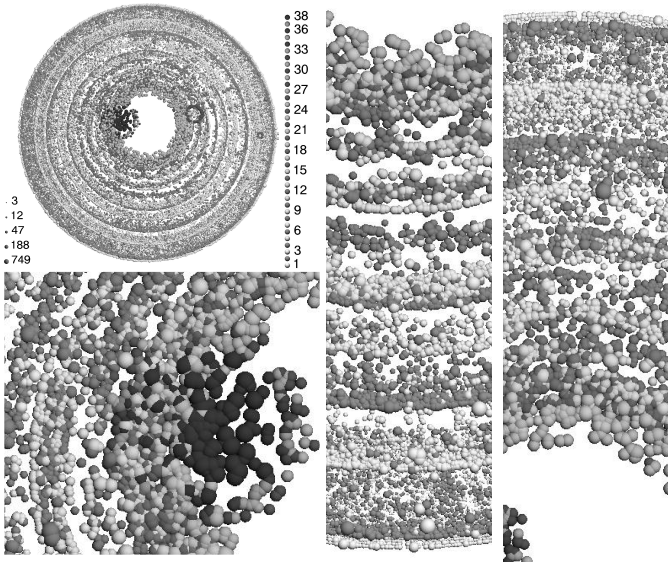


Fig. 6. k-core visualization of router level topology G_r.

of the spheres represent vertex's degree in a logarithmic scale. The k-core decomposition can break the original network into various connected components which are displayed as independent circles.

B. MPLS on Internet Topology

In this section, we study the LSRs and MPLS cluster structure over the Internet topology. We focus on the router level topology for our analysis because it is closer to a realistic Internet one and because we do not notice any strong difference between IP and router level topology as Fig. 5 suggests. Indeed, we only remark that router level topology has a slightly stronger clustering coefficient (see Fig. 5(b)), due to alias resolution process. In this way, Fig. 6 [4] shows the k-core visualization of G_r. The figure is divided in four parts, the main part being in the upper left while the three others are a zoom on the main one. The main part is composed of two scales, the one on the left is the node degree in a logarithmic scale, while the one on the right is a gray scale

[4] Higher resolution figures available at `http://cnet.fi.uba.ar/Sup_Mat_TMA_2016/`.

with each shell index C_i. Between the two scales, we see the shell index with $C_{\max}$ in the center, the other shells being located concentrically around it. Note that $C_{\max}$-core is made of several components with one having the most significant part, and it is shown at the left of the center (black nodes). We also see that all the shell indexes are highly populated and that the node degree is not related with the shell index, i.e., there are many routers with high degree in the outer (lower) shells. Another typical feature of router level topology is that the links between routers mainly occur between routers belonging to neighbors shells (links have been omitted for the sake of visualization), e.g., the routers on the outer shells are not usually connected to the routers located on the $C_{\max}$-core, as it is in the Autonomous Systems maps [9].

In order to locate LSRs-routers with MPLS capabilities into the shell indexes over k-core decomposition, we paint in black the non-MPLS routers and in gray the LSRs. The results are shown in Fig. 7(a).[4] We notice that the LSRs are commonly distributed around the different shells of Internet, with slightly low density in the lower ones. Additionally, we apply the same methodology for the MPLS interconnection cluster level graph $G_{r\setminus lsr}$ (Fig. 7(b)): MPLS clusters (gray nodes) are distinguished from the non-MPLS capable routers (black nodes). In this case, MPLS clusters degree is correlated with shell index: higher the shell cluster, higher its degree. This behavior is observed in hierarchical networks, e.g., the Autonomous System network.

Finally, we evaluate $G_{r\setminus lsr}$ using metrics such as degree distribution, local clustering coefficient, and nearest neighbor degree. These metrics are widely used by the research community to describe network properties [23, p. 61-62]. Local clustering coefficient is a measure of how connected are the vertex's neighbors between them. Nearest neighbor degree is the average of the neighbor's degree for a given vertex. The results are shown on Fig. 5. The x-axis values have been logarithmically binned. We notice that MPLS clusters C_i^{mpls} highly impact the router level topology (notice that C_i^{mpls} is represented by a single node). On one hand, the nearest neighbor degree is higher for lower degree nodes on $G_{r\setminus lsr}$, suggesting that routers with low degree are highly connected to MPLS clusters (the highest degree nodes as Fig. 7(b) shows) and thereby to LSRs. On the other hand, the clustering coefficient of $G_{r\setminus lsr}$ is highly increased for vertices having degree 50 to 200. It means that their neighbors are highly connected between them due to the MPLS clusters aparition. It also implies that MPLS clusters plays an important role on Internet robustness: the more connected the vertex's neighbors are, the more disjoint paths exist between them. Moreover, this network verify the core-connectivity property [11] for the highest cores.

C. MPLS clusters on Autonomous Systems

Although, the previous results give us a general overview about MPLS deployment, we believe that the study of MPLS structure requires going deeper into the individual AS topology. Indeed, we found that around 89.9% of MPLS links are intra-domain. Thereby, we focus on the top ASes in terms of total number of discovered links. On this set of ASes, we discard those having less than 500 MPLS links. The summary of this top ASes is shown in Fig. 8. Additionally, we identify the amount of discovered MPLS links by AS, distinguishing the type of MPLS tunnel as follows: given a link between two MPLS interfaces i_{n-1} and i_n discovered by `traceroute` at $n-1$ and n position, we define :

- *explicit MPLS link:* as links where i_n belongs to an explicit MPLS tunnel;
- *qTTL MPLS link:* as links where i_n belongs to an implicit MPLS tunnel qTTL based;
- *u-turn MPLS link:* as links where i_n belongs to an implicit MPLS tunnel u-turn based.

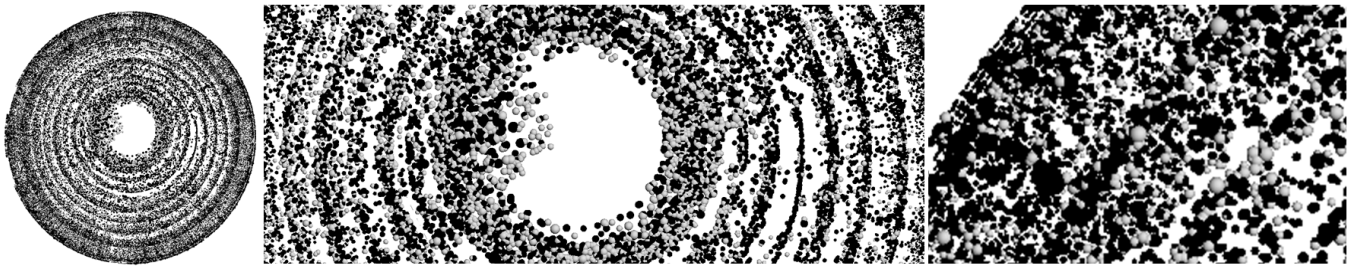
Regarding Fig. 8, we notice that the ratio $r_{mpls} = |E_r^{mpls}(as)|/|E_r(as)|$ is greater when more explicit MPLS links have been discovered. Interestingly, we also see that the ASes with more IP links discovered have the lowest ratio r_{mpls}.

For our purposes, we select the most representatives ASes from those in Fig. 8. In this way, we analyze the graphs $G_r(as)$ and $G_{r\setminus lsr}(as)$ for AS1299, AS174, AS6762, AS2914, AS7018, and AS1273 using k-core decomposition (see Fig. 9). We observe that k-core decomposition structure varies according the type of MPLS tunnels that prevails in the AS. Particularly, for AS1299 (Teliasonera AB), AS174 (Cogent Communication), and AS6762 (Telecom Italia) where prevails u-turn MPLS links, we show that MPLS clusters (represented as gray nodes) are spread out over different shells. These k-core structures are similar in our top five ASes (first five ASes with most links discovered on Fig. 8) where u-turn signature was mostly discovered, i.e., between 30% and 80% over the total amount of MPLS links.

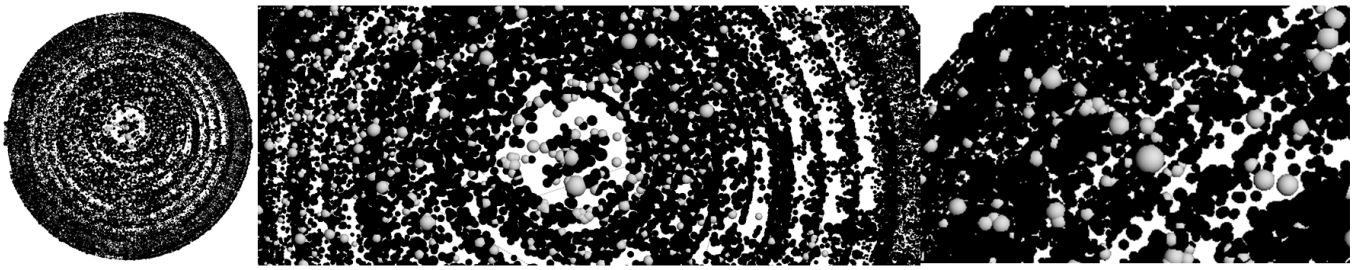
However, for AS2914 (NTT America Inc.), AS7018 (AT&T), and AS1273 (Cable and Wireless Worldwide plc) where explicit MPLS links prevail, we find a highly different k-core structure, i.e., most vertices are directly connected to a predominant MPLS cluster located at $C_{\max}$-core. The remaining ASes in Fig. 8 with high percentage of explicit tunnels have the same structure.

In summary, we notice that ASes where explicit MPLS links prevail have a predominant MPLS cluster while ASes where u-turn MPLS links prevail have several MPLS cluster spread out over the shells. Additionally, We believe that k-core decomposition on the top five ASes could have different structure due to either the u-turn signature inaccuracy or due to some particular MPLS deployment. Indeed, we remark only on these ASes a low ratio r_{mpls} and an unusually high u-turn links presence.

Another remarkable observation relies on the fact that the maximum degree reached by MPLS clusters is considerably high with respect to the network size. Indeed, except for AS174, the rest of ASes suggest that more than 50% of non-MPLS routers are connected to at least one LSR. Actually, even the outer shells of the k-core decomposition are linked directly with the MPLS clusters located in the $C_{\max}$-core. This



(a) The k-core visualization of router level topology G_r.



(b) The k-core visualization of MPLS cluster level topology $G_{r\setminus lsr}$.

Fig. 7. k-core visualization of G_r and $G_{r\setminus lsr}$. On Fig. 7(a), black nodes refer to non MPLS capable routers and gray nodes refer to LSRs. On Fig. 7(b), black nodes refer to non MPLS capable routers and gray nodes refer to MPLS clusters.

behavior matches with our observation of nearest neighbor degree and clustering coefficient discussed in Sec. V-B. Additionally, because MPLS clusters are mainly located on $C_{\max}$-core (even on ASes with high percentage of u-turn MPLS links), we believe that MPLS plays an important role in ISPs' backbone.

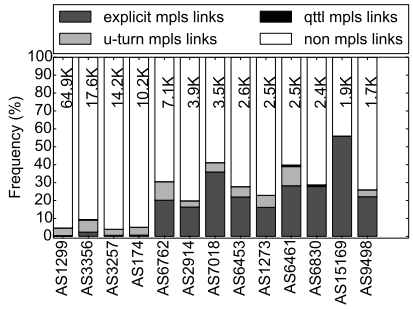


Fig. 8. Top of ASes with most links discovered

VI. Conclusion

MPLS usage has an important role on Internet. Indeed, we found that MPLS tunnels are traversed by around 34% (explicit tunnels) and 42% (explicit and implicit tunnels) of `traceroute` probes in our dataset. In this paper, we tested the detection methods of implicit MPLS tunnels. Our results suggested that although Paris traceroute works properly (avoiding per-packet load balancers) for MPLS tunnels discovery, u-turn signatures are commonly biased due to per-flow load balancing issues in the return path. We also provided in this work a first and novel overview about MPLS structure. On one hand, our findings highlight the importance of MPLS in the Internet robustness; on the other hands we showed that MPLS deployment plays an important role in the ISP's backbone. Additionally, our methodology based on k-core decomposition allowed us to reveal the fingerprints related with the type of MPLS tunnels that prevails for a given AS. However, we believe that it is necessary to continue the current studies by adding new methods and mechanisms to reveal MPLS presence. Mainly, because it is necessary to correct the u-turn biases and infer tunnels not revealed by `traceroute` (invisible tunnels). Additionally, this work did not study the traffic behavior and patterns related with MPLS presence, this information could allow one to infer LSR not revealed by the current methods. Finally, we plan to perform long time measurement campaigns in order to explore the impact of time on MPLS infrastructures.

Acknowledgments

This work is partially funded by the European Commission funded mPlane ICT-318627 project, and also by UBACyT 2014 (20020130200122BA).

References

[1] B. Donnet, "Internet topology discovery," in *Data Traffic Monitoring and Analysis: From Measurement, Classification and Anomaly Detection to Quality of Experience*, M. M. E. Biersack, C. Callegari, Ed. Springer, 2013, pp. 44–81.

[2] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," Internet Engineering Task Force, RFC 3031, January 2001.

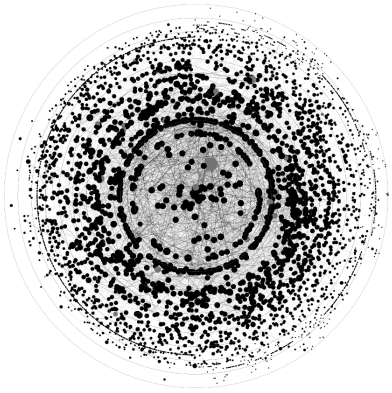
[3] J. Sommers, B. Eriksson, and P. Barford, "On the prevalence and characteristics of MPLS deployments in the open Internet," in *Proc. ACM Internet Measurement Conference (IMC)*, November 2011.

[4] B. Donnet, M. Luckie, P. Mérindol, and J.-J. Pansiot, "Revealing MPLS tunnels obscured by traceroute," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 2, pp. 87–93, April 2012.

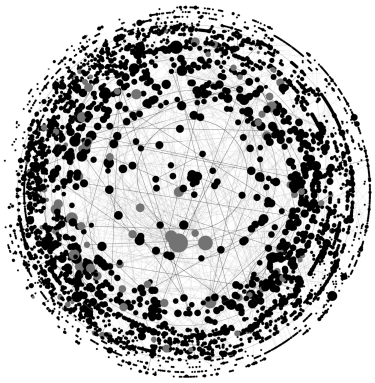
[5] Y. Vanaubel, P. Mérindol, J.-J. Pansiot, and B. Donnet, "MPLS under the microscope: Revealing actual transit path diversity," in *Proc. ACM Internet Measurement Conference (IMC)*, October 2015.

[6] B. Augustin, R. Teixeira, and T. Friedman, "Measuring load-balanced paths in the internet," in *Proc. ACM Internet Measurement Conference (IMC)*, November 2007.

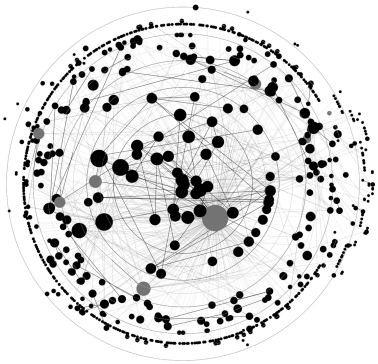
[7] T. Flach, E. Katz-Bassett, and R. Govindan, "Quantifying violations of destination-based forwarding on the Internet," in *Proc. ACM Internet Measurement Conference (IMC)*, November 2012.



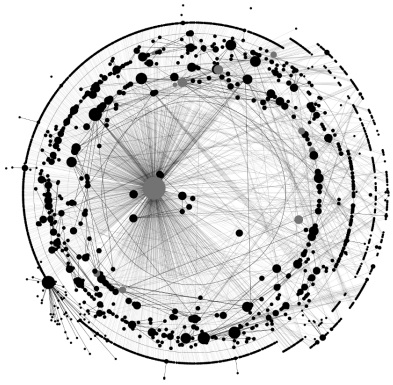
(a) AS1299 Teliasonera AB , $C_{\max} = 21$, $\text{Degree}_{\max} = 2781$, $|V_{r\setminus lsr}| = 4128$, $|E_{r\setminus lsr}| = 24865$



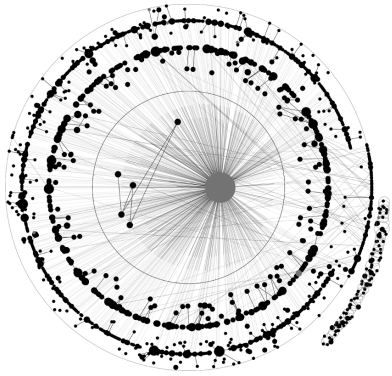
(b) AS174 Cogent Communication, $C_{\max} = 8$, $\text{Degree}_{\max} = 751$, $|V_{r\setminus lsr}| = 4421$, $|E_{r\setminus lsr}| = 8611$



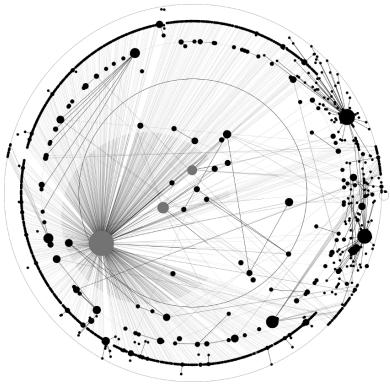
(c) AS6762 Telecom Italia, $C_{\max} = 10$, $\text{Degree}_{\max} = 564$, $|V_{r\setminus lsr}| = 750$, $|E_{r\setminus lsr}| = 1504$



(d) AS2914 NTT America Inc., $C_{\max} = 4$, $\text{Degree}_{\max} = 1019$, $|V_{r\setminus lsr}| = 1807$, $|E_{r\setminus lsr}| = 2360$



(e) AS7018 AT&T, $C_{\max} = 3$, $\text{Degree}_{\max} = 745$, $|V_{r\setminus lsr}| = 1306$, $|E_{r\setminus lsr}| = 1441$



(f) AS1273 Cable and Wireless Worldwide plc, $C_{\max} = 3$, $\text{Degree}_{\max} = 806$, $|V_{r\setminus lsr}| = 1127$, $|E_{r\setminus lsr} = 1215$

Fig. 9. k-core visualization of MPLS cluster interconnection Graph $G_{r\setminus lsr}(as)$. On the top, the ASes show several MPLS clusters spread out around the shells. On the bottom, the ASes show a predominant MPLS cluster located on $C_{\max}$.

[8] V. Batagelj and M. Zaveršnik, "Fast algorithms for determining (generalized) core groups in social networks," *Advances in Data Analysis and Classification*, vol. 5, no. 2, pp. 129–145, July 2011.

[9] J. I. Alvarez-Hamelin, A. Barrat, and A. Vespignani, "Large-scale networks fingerprinting and visualization using the k-core decomposition," in *Proc. Advances in Neural Information Processing Systems*, December 2006.

[10] M. Ángeles Serrano, M. Boguná, and A. Díaz-Guilera, "Modeling the Internet," *Eur. Phys. J. B.*, vol. 50, no. 1–2, pp. 249–254, February 2006.

[11] M. G. Beiró, J. I. Alvarez-Hamelin, and J. R. Busch, "A low complexity visualization tool that helps to perform complex systems analysis," *New J. Phys*, vol. 10, no. 12, p. 125003, 2008.

[12] L. Andersson and R. Asati, "Multiprocotol label switching (MPLS) label stack entry: EXP field renamed to traffic class field," Internet Engineering Task Force, RFC 5462, February 2009.

[13] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, and A. Conta, "MPLS label stack encoding," Internet Engineering Task Force, RFC 3032, January 2001.

[14] P. Agarwal and B. Akyol, "Time-to-live (TTL) processing in multiprotocol label switching (MPLS) networks," Internet Engineering Task Force, RFC 3443, January 2003.

[15] R. Bonica, D. Gan, D. Tappan, and C. Pignataro, "ICMP extensions for multiprotocol label switching," Internet Engineering Task Force, RFC 4950, August 2007.

[16] Y. Vanaubel, P. Mérindol, J.-J. Pansiot, and B. Donnet, "A brief history of MPLS usage in IPv6," in *Proc. Passive and Activement Measurement Conference (PAM)*, March/April 2016.

[17] Y. Vanaubel, J.-J. Pansiot, P. Mérindol, and B. Donnet, "Network fingerprinting: TTL-based router signature," in *Proc. ACM Internet Measurement Conference (IMC)*, October 2013.

[18] CoNexDat-UBA, "Magallanes: Large scale internet explorer tool," December 2015, https://github.com/ihameli/magallanes.

[19] M. Luckie, "Scamper: a scalable and extensible packet prober for active measurement of the Internet," in *Proc. ACM Internet Measurement Conference*, November 2010.

[20] K. Keys, Y. Hyun, M. Luckie, and k. claffy, "Internet-scale IPv4 alias resolution with MIDAR," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 383–399, April 2011.

[21] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding traceroute anomalies with Paris traceroute," in *Proc. ACM Internet Measurement Conference (IMC)*, October 2006.

[22] T. C. UCSD, "Routeviews Prefix-to-AS mappings (pfx2as) for IPv4 and IPv6," October 2015, see http://data.caida.org/datasets/routing/routeviews-prefix2as/2015/10/routeviews-rv2-20151030-1400.pfx2as.gz.

[23] R. Pastor-Satorras and A. Vespignani, *Evolution and structure of the Internet: A statistical physics approach*. Cambridge University Press, 2007.