A survey of MPLS deployment on internet topology

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

***Abstract*— toto**

I. Introduction

Internet topology refers to the study of the various types of connectivity structures and representations between directly connected nodes on the Internet architecture [1]. This representation aims at obtaining models that represent 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 [2] . However, the current state of the art of Internet deployments involves a great number of technologies impacting the Internet Topology. And those technologies deserve a deep study in order to include them in Internet topology models. For instance, *Multiprotocol Label Switching* (MPLS) [3] has been recently the focus of several studies [4], [2], [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].

Although, the MPLS deployment impact on the Internet architecture has not been studied yet. The importance to study new architectural details and topological features related with MPLS usage and its impact 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 today's Internet architecture more accurately. Our work provides a study around the impact of MPLS deployments over Internet Topology. Principally, we focus on the properties and features that MPLS modifies on traditional networks maps such as router level topology. For our purpose we mainly based on k-core decomposition method [8]. In this way, it has been shown previously that the k-core decomposition is a relevant tool to describe Internet Topology, by being capable of identifying networks sources by means of the visualization [9], may be used to validate models [10], and discover exploration biases on the Internet [11].

In this way, this paper adds complementary and enriched information around MPLS usage on Internet. First, we present a quantification of the biases involved on implicit MPLS tunnel detection. Additionally, we provided some properties and architectural details related with MPLS deployment on router topology. We found that routers with lower degree are frequently connected with MPLS capable routers and routers with high degree are usually connected to common MPLS networks. Additionally, we describes the behaviour of MPLS networks within some ASes with most MPLS usage. We found that given an AS, the MPLS networks tend to form few and well defined clouds. However, this observation could change on regarding with the type of MPLS tunnel that prevails.

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 behaviour 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) [3] 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).[1] A given packet can manage several LSEs at the same time. In this case, the packet is said having 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).

[1] MPLS is IP layer protocol independent.

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 [18], 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 TE 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. [4] 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. In addition, Sommers et al. propose a statistical methodology to infer MPLS tunnels in archived data where ICMP extensions are not recorded. 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. [19] 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 revelead 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`;
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. Summarizing, *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 tunnel length and the array position correspond to the LSR position within the LSP.
3) opaque tunnels are revealed through an *abnormal* LSE-TTL ($1 <$LSE-TTL< 255) returned by the LH in the `time-exceeded` reply.

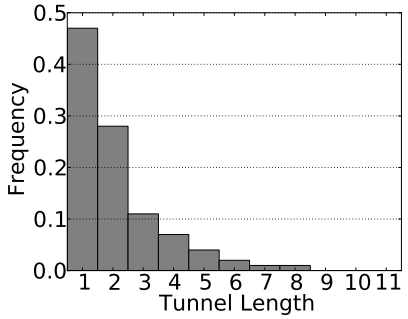
Additional study by Vanaubel et al. [20] shows that the probing heuristic to detect implicit tunnels seems quite reliable. However, u-turn signatures are by definition more subject to false positives than qTTL ones. This is exactly what we tackle in this paper (and, consequently, our work is complementary to Vanaubel et al. [20]): we want to test u-turn signature accuracy.

III. Dataset

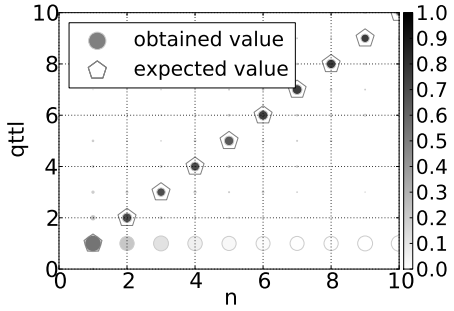
Because part of our works focus on signature validation, we developed our own measurement tool in order to get *qTTL* and *u-turn* signatures. We use *Magallanes* [REF] which is an open source software that allows to run and to manage Scamper [21] based probes through Planet Lab (PL) infrastructure. *Magallanes* allocates randomly several Vantage Points (VP) within the available set of PL nodes and it distributes a given number of destination targets to each VP. To achieve uniformity in target selection, *Magallanes* uses data provided by MaxMind[2]. In this way, it chooses the targets randomly and proportionality distributed according to the number of subnets assigned by the Regional Internet Registry (RIR) to each region. Additionally, it allows to storage the experiments results on a centralized database and to solve alias resolution process using MIDAR [22].

For our experiment, we run the exploration October 31, 2015. We choose 100 VP and we selected 10K of destination targets per VP. Each set of destination targets are disjoint sets. We use ICMP-Paris probes in order to avoid load balancing issues in the forwarding path. To get the *u-turn* signature, we send a *ping* to each hop revealed by traceroute. We sent six

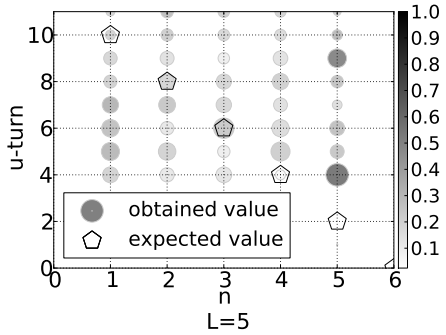
[2] `www.maxmind.com`



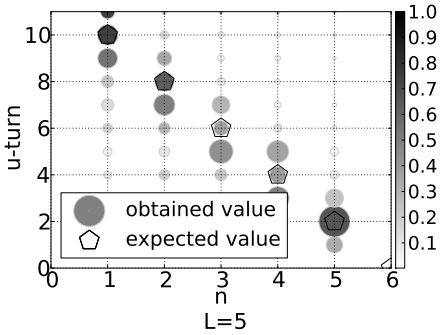
(a) Tunnel Length Distribution



(b) *qTTL* and n-position comparison



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



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

Fig. 1: Comparison between obtained and expected values for *qTTL* and *u-turn* signatures. On figures (b), (c) and (d) 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 n-position regarding each *y-axis* value, e.g., on figure (b) for values where $n > 1$, the biggest circles are mainly located on $qTTL = 1$ and $qTTL = n$ so this suggest that for a given n-position the *qTTL* value usually takes either the value of 1 or n; in the same way, the transparency value suggests that for a given *qTTL* value the n-position usually takes the same *qTTL* value. **Figure (c)** and **Figure (d)** suggest that *u-turn* value is overestimated.

ICMP-echo packets from the same monitor. Six ICMP-echo responses allows us to infer with 95% confidence if there is a single return path length and therefore reduce measurement error caused by a reverse path containing load-balanced segments of different lengths [6]. As a result we discovered around 270K of IP interfaces 520K links, 42% of which were available to run MIDAR and we found aliases successfully on 19% of then. To match IP interfaces to ASes, we use the RouteViews Dataset provided by CAIDA. Additionally we found MPLS tunnels on 44% of the traceroutes. The amount of explicit tunnels is highly superior to implicit tunnels. 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 match with at least 63% of the explicit tunnels. We discuss these results in the next sections. Finally, we did not found opaque tunnels.

IV. MPLS Signatures Validation

In this section we describe the used methodology to validate the MPLS signatures revealed by traceroutes. Basically, we compare the LSR position within an MPLS tunnel with the signatures values.

Implicit tunnels are based either on *qTTL* or *u-turn* signatures. Both of them, directly related with MPLS position:

i *qTTL* value refers to the IP-TTL of ICMP-*echo* message when it enters to the MPLS tunnel. A quoted TTL of n in the incoming ICMP reply means that the sent probe expired n hops later to ingress to the *LSP*, i.e., a LSR reply where $qTTL = n$ means that the LSR appears in the n position of MPLS tunnel.

ii *u-turn* value is related with the tunnel length L and the n-position of the LSR within the tunnel (see section II-B)

Our signature validation relays on the hypothesis that the MPLS position match with the n-position which an LSR is revealed by traceroute. Indeed, we use a paris-traceroute [23] based tool in order to sure that the probes follow the same path. In order to validate our hypotheses, first, we compare a

high reliable signature such as *qTTL* with n-position.

To probe our assumption, the *qTTL* values should to match with n-position, i.e., *qTTL* $= n$. The results are showed in figure 1b. The figure 1a shows the MPLS tunnel length distribution that additionally can help us to understand the amount LSR distributed by position. We principally noticed that *qTTL* signature highly match with n. The bias *qTTL* $= n \pm \triangle$ could occur due to the limitation in our method to reveal the first LSRs in the tunnel if it does not implement the RFC4950. A second deviation could occur due to possible load balancer presence in the follow path. In this case, the traceroute probes follow paths with different lengths before to reach the MPLS tunnel. This issue produce that the same *qTTL* value could be revealed several times in different n-positions. The figure also shows that *qTTL* frequently takes the value of 1. This usually means that `ttl-propagate` is not implemented. However, we found that around 2% of LSRs do not react to *qTTL* signature, even if `ttl-propagate` option is activated, 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. However, n-position are highly reliable. We found that in 58% of the cases the n-position match with the *qTTL* value while in $36,3\%$ of cases the *qTTL* signature is not present and takes the value of 1, and just $6,7\%$ of the cases presented some bias around the expected value. This facts support our hypothesis: the MPLS tunnel position highly match with the n-position which an LSR is revealed by traceroute. Thereby, we use n-position as a reference value to validate the *u-turn* signatures. The expected *u-turn* value is related with n in the way *u-turn* $= 2(L-(n-1))$, where L is the tunnel length. Because *u-turn* is commonly present in almost all LSRs, first, we compare n vs *u-turn* on LSRs revealed either explicitly or *qTTL* based.

Subsequently, we study n value on LSRs where *u-turn* was the only detected signature. We use the filter *u-turn* > 3 to avoid false isolated *u-turn* signatures. The results for a given tunnel length L are showed on figures 1c, 1d. Similar results was noticed for other tunnel length values. Basically, we noticed that the obtained values are close to expected values on the LSRs explicitly revealed and *qTTL* based (figure 1c). However on the set of LSRs revealed just by *u-turn* signatures (figure 1d) the values commonly does not match. If we accept a bias of ± 2 around the expected value, we noticed that on LSRs explicitly revealed and *qTTL* based, the 60% of obtained *u-turn* values match with the expected values. However, on LSRs revealed just trough *u-turn* signature (therefore where it is really useful), the obtained *u-turn* value just match in 25% with the expected value $2(L-(n-1))$. Therefore, LSRs revealed through *u-turn* is highly inaccurate and overestimated. Mainly, because MPLS tunnels are not the only behaviour that could causes *u-turn* signatures. Indeed, has been showed that load balancing is common on traverse paths even between the same pairs source-destination [6]. This issue is called *per-flow* load balancing. Basically, packets that belongs to the same *flow* are treated similarly [23]. A *flow* is identified by the first 32 bits of the IP *payload*. In the case of ICMP messages, this fields refers to *Type*, *Code* and *Checksum*. By definition, *u-turn* signature is based on two kinds of ICMP messages: ICMP `echo-reply` *Code 11* and ICMP `time-exceeded` *Code 0*. Due to the different codes values related with each ICMP message, there is no way to sure that they belong to the same *flow* identifier and thereby to be sure that *u-turn* value is caused just by MPLS tunnels.

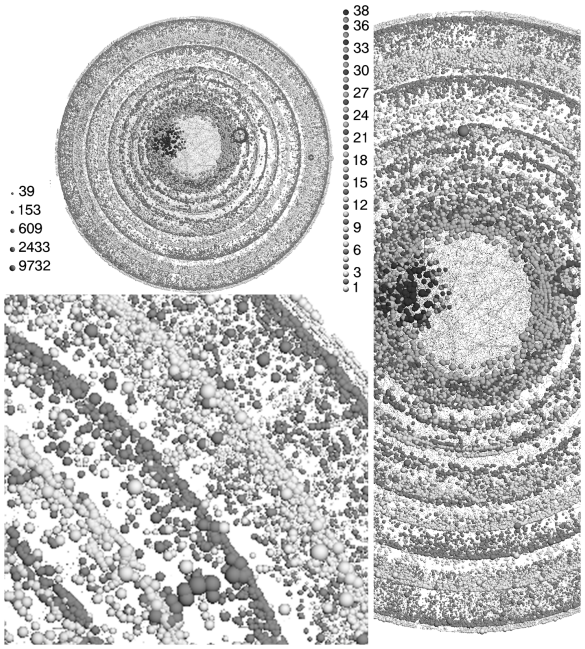


Fig. 2: k-core visualization of router level topology G_r

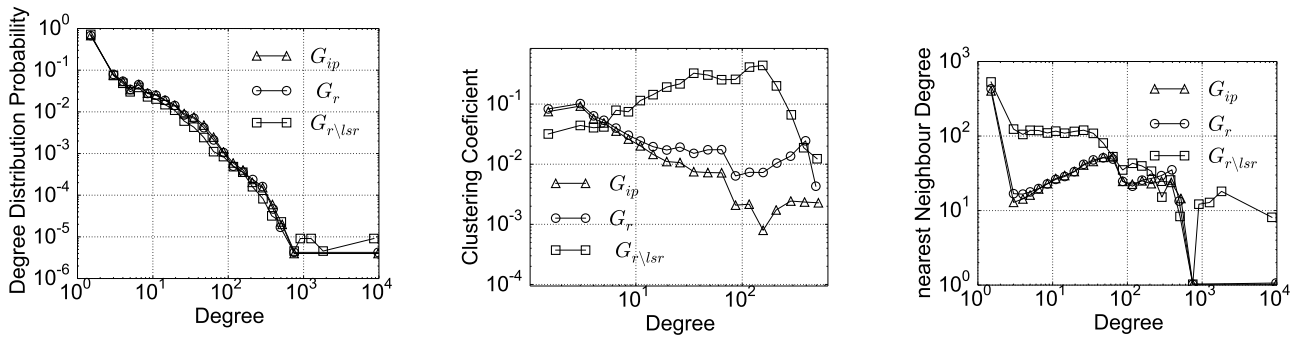
V. LSRs AND *MPLS clusters*

For the best of our acknowledge, MPLS interconnection architecture on Internet Topology has not been yet studied. In this way, in this section we focus our attention on study some properties of LSRs and *MPLS clusters*; and their interaction with non MPLS networks. Our study aims to better understand the impact of MPLS deployments over internet, specifically over router level topology.

A. Methodolody

We defined several graphs at many abstraction levels. First, we make an IP level topology G_{ip} from traceroutes. Secondly, we identified the MPLS tunnels. Later, we solved the alias resolution process using MIDAR and we develop a router level topology G_r (from IP interfaces) and an MPLS router (LSRs) level topology G_r^{mpls} (from IP interfaces belonging to MPLS tunnels). Additionally, we identified the *MPLS clusters* C_i^{mpls} and we treat each cluster as single node in order to study how they interacts with the entire router level topology $G_{r\setminus lsr}$. Finally, in order to better understand the MPLS behaviour we studied this interaction for some ASes $G_{r\setminus lsr}(as)$.

Following, we define with more detail terminology used in this paper.



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

Fig. 3: **Metrics for IP, router and MPLS cluster interconection topologies.** IP, router and MPLS cluster interconection topologies have similar degree distribution. Clusterin Coeficient and Neighbor Degree Distribution do not change significantly between IP level and router level topologies. However the *MPLS clusters* presence highly impact on internet topology. The figure suggest that routers with low degree are highly connected with LSRs (Neighbor Degree Distribution) and routers with high degree ussually have *MPLS clusters* as common neighbors (Clusterin Coeficient).

IP level Graph: $G_{ip} = (V_{ip}, E_{ip})$, where V_{ip} is the IP address discovered in our exploration and E_{ip} is the set of the links found trough traceroute.

Router Level Graph: $G_r = (V_r, E_r)$, where V_r is the set of routers where we found at least one IP address through MIDAR and E_r is the set of all the links found between any pair of routers.

MPLS router level Graph: $G_r^{mpls} = (V_r^{mpls}, E_r^{mpls})$ where V_r^{mpls} is the set of routers where we found at least one IP address belonging to an MPLS tunnel and E_r^{mpls} is the set of all *mpls links* $(v_r^{mpls}(i), v_r^{mpls}(j))$ such as $v_r^{mpls}(i)$ and $v_r^{mpls}(j) \in V_r^{mpls}$.

ASes induced Graphs: $G_r(as) = (V_r(as), E_r(as))$ and $G_r^{mpls}(as) = (V_r^{mpls}(as), E_r^{mpls}(as))$ are the induced graphs of G_r and G_r^{mpls} respectively, such as each vertex has a router's interface belonging to the same Autonomous System as.

MPLS clusters: C_i^{mpls} is a connected component i of $G_r^{mpls}(as)$. One AS could to have several *MPLS clusters*.

MPLS cluster interconnection Graph: $G_{r\setminus lsr} = (V_{r\setminus lsr}, E_{r\setminus lsr}^{mpls})$ is an hybrid router level graph where all the *MPLS clusters* C_i^{mpls} were contracted to a single node while non MPLS capable routers remains unchanged. Broadly speaking, an *MPLS cluster interconnection Graph* refers to a router level graph where all *MPLS clusters* are treated as a single node. In this way, we can study how IP interfaces and non MPLS capable routers interact with *MPLS clusters*. Additionally , we called $G_{r\setminus lsr}(as)$ to the induced subgraphs of $G_{r\setminus lsr}$ such as each vertex has a router's interface belonging to the same Autonomous System as.

Our analysis is mainly based on k-core decomposition. This is a tool where we can easily read basic features of the graph (degree, hierarchical structure, etc.) as well as more entangled features, e.g., the relation between a vertex and the hierarchical position of its neighbours. It is also useful to discriminate between networks with different topological properties and structural arrangement.

Formally, let the graph $G = (V, E)$, k-core decomposition analysis is defined as:

i ***k*-cores**. A 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.
A k-core of G can therefore be obtained by recursively removing all the vertices of degree less than k, until all vertices in the remaining graph have at least degree k.

ii **Shell index**. A vertex i has shell index c if it belongs to the c-core but not to $(c+1)$-core. We denote by C_i to the shell index of vertex i. A shell C_c is composed by all the vertices whose shell index is c. The maximum value c such that C_c is not empty is denoted $C_{\max}$. The k-core is thus the union of all shells C_c with $c \geq k$.

Finally, we use *LaNet-vi* [9] to get the k-core decomposition. *LaNet-vi* returns a two dimensional plot, where the position of each vertex depends on its shell index and on the index of its neighbours. A color code allows for the identification of shell indices, while the vertex's original degree is provided by its size that depends logarithmically on the degree. A central role in the visualization method is played by multi-components representation of k-cores. In the most general situation, indeed, the recursive removal of vertices having degree less than a given k can break the original network into various connected components, each of which might even be once again broken by the subsequent decomposition.

In this paper, we use k-core decomposition focused on properties around *mpls clusters* interconnection. This provide us an idea about the structural arrangement and topological properties caused by MPLS usage. Additionally, the visualization help us to find properties and fingerprints tightly related with MPLS presence.

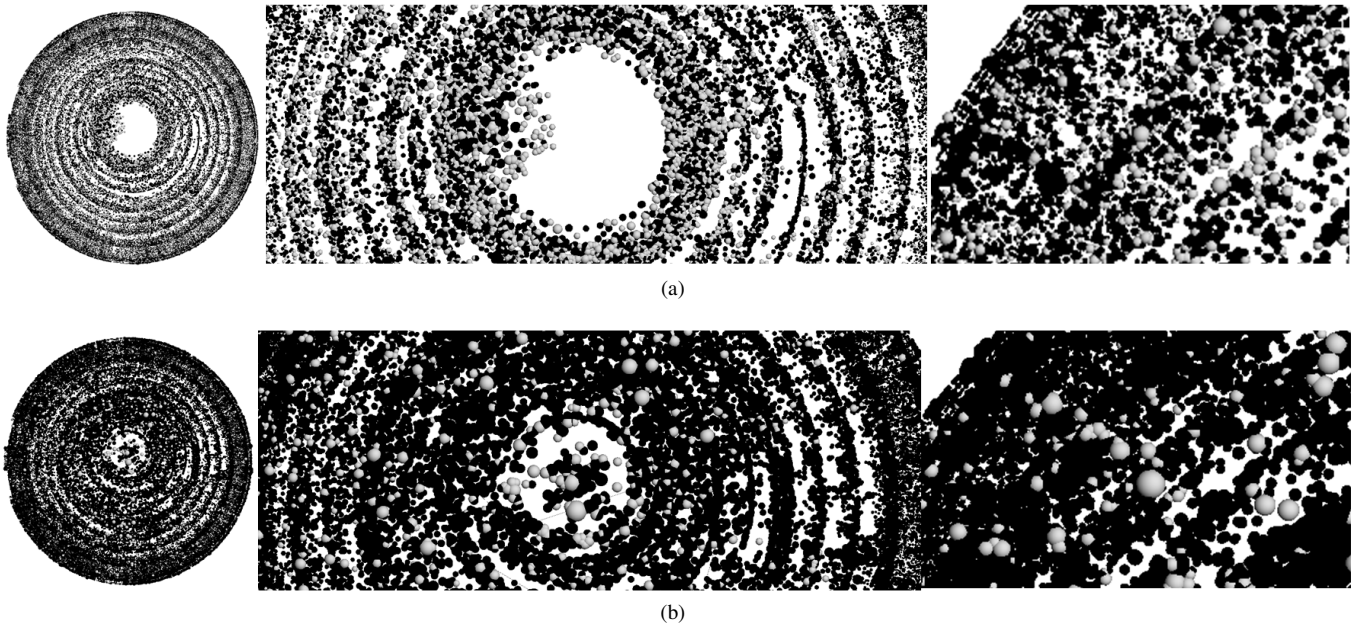


Fig. 4: **(a) k-core visualization of router level topology G_r.** Black nodes refer to non MPLS capable routers and gray nodes refer to LSRs. **(b) k-core visualization of MPLS cluster level topology $G_{r\setminus lsr}$.** Black nodes refer to non MPLS capable routers and gray nodes refer to *MPLS clusters*.

B. MPLS on Internet Topology

First, we analysed the graphs G_{ip} and G_r in order to know if there is some strong difference between their structure and properties. As was expected we noticed that the router level topology has slightly stronger clustering coefficient than IP level topology (see 3b). This fact occurs due to the join of IP interfaces into single routers. However, the main structure of both topologies are similar. Because we did not notice any meaningful difference between this topologies, and because router level topology are more approached to a realistic Internet behaviour, we choose it to the remain of our analysis.

Figure 2 shows the k-core visualization of G_r. In the center is located the shell index with the $C_{\max}$ and the rest of the shells are located concentrically around it. On the right there is a gray scale with each shell index c_i and left is a node degree scale represented by the size of each one. We see that all the shells index are highly populated and that the node degree is not related with the shell index i.e., there is many routers with high degree in the outer shells. Another typical feature of router level topology is that the links between routers mainly occurs between routers belonging to neighbours shells, e.g., the routers on the outer shells are not usually connected to the routers located on the top core.

In order to locate LSRs-routers with MPLS capabilities into the shells index over k-core decomposition, we painted in black the *non MPLS capable routers* and in gray color the LSRs. The results are showed in figure 4a. For the sake of the visualization we do not include neither the shell index, degree scale nor edges between shells. We noticed that the LSRs are commonly distributed around the different shells of Internet but lightly tends to concentrate with more density nearby to the core. Additionally, we apply the same methodology for the MPLS interconnection cluster level graph $G_{r\setminus lsr}$: *MPLS clusters* (gray nodes) are distinguished from the non MPLS capable routers (black nodes). The figure 4b shows the results. In this case, *MPLS clusters* are also spread out on the Internet, indeed we see some well defined gray nodes on the periphery of the decomposition. However *MPLS clusters* shows a stronger tendency to concentrate it near to the core. Finally, we evaluated the impact of *MPLS clusters* on the typical router level topology i.e., *MPLS cluster interconnection graph* $G_{r\setminus lsr}$. We use metrics such as degree distribution, clustering coefficient, and nearest neighbour degree. The results are showed on figure 3. We noticed that *MPLS clusters* highly impact over the router level topology. The nearest neighbour degree highly increments for low degrees nodes on $G_{r\setminus lsr}$. This suggest that routers with low degree are highly connected to *MPLS clusters* and thereby to LSRs. On the other hand, the clustering coefficient of $G_{r\setminus lsr}$ change significantly their slope in regards with IP and router level topology. This suggest that routers with high degree are connected to *common MPLS clusters*.

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 a zoom in on each AS. Indeed, we found that around 89.9% of *mpls links* are intra-AS. Thereby, we choose the top of ASes according to the amount of links discovered. In order to focus our analysis on those ASes were MPLS is relevant, we discard those where we found less than five hundred of *mpls links*. Additionally, we identified the

amount of *mpls links* by AS distinguishing the type of tunnel, e.g., we define as *explicit mpls link* to those link between an interface i_n revealed as explicitly MPLS capable and its previous interface i_{n-1} discovered by traceroute, provided that i_{n-1} was also revealed as MPLS interface. In the same way for *qTTL* and *u-turn mpls links*.

The summary of the top ASes is showed in the figure 5. We noticed 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 links discovered have the lowest ratio r_{mpls}.

For our purposes, we select the most representatives ASes from our top. In this way we analysed the graphs $G_r(as)$ and $G_{r\setminus lsr}(as)$ for AS1299, AS174, AS6762, AS7018, AS1273 and AS2910. The most remarkable observation occurs regarding the graph $G_{r\setminus lsr}(as)$: k-core decomposition highly differs on those ASes where prevails explicit tunnels in regard to those where prevails *u-turn* tunnels. The figure 6 show this observation. We show that *MPLS clusters* (represented as gray nodes) for AS1299 (Teliasonera AB), AS174 (Congent Communication) and AS6762 (Telecom Italia) are spread out over different shells index. This k-core structure are similar in our top four of ASes where we additionally noticed the *u-turn* signature was majority discovered i.e., between 30% and 80% over the total amount of *mpls links*. However, for AS7018 (AT&T), AS1273 (Cable and Wireless Worldwide plc) and AS2910 (Citicorp) where prevail explicit tunnels, we found a k-core structure highly different. In these cases, the ASes have few and well defined *MPLS clusters*, mainly belonging to the top core $C_{\max}$. The same k-core decomposition structure was noticed on the rest of top ASes with high percentage of *qTTL* signature.

Another remarkable observation relays on the fact that the maximum degree reached by *MPLS clusters* is considerably high in regard to the network size. Indeed, with exception of 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 top core. This behaviour match with our observation of nearest neighbour degree and clustering coefficient noticed on section V-B. Additionally, because *MPLS clusters* are mainly located on top core of the k-core decomposition (even on the ASes with high percentage of *u-turn mpls links*), we believe that MPLS plays an important role in the backbone of the ISPs.

Summarizing, we observed that k-core decomposition structure varies according the type of MPLS tunnels that prevails in the AS. This observation could suggest that exists a great number of MPLS links non revealed by traceroute in ye ASes where prevails *u-turn* signatures, therefore, the LSRs discovered can not build large *MPLS clusters* so they are thoroughly spread out. Additionally, because *u-turn* signatures are overestimated, as we showed previously, there could exists several false LSRs spread out over the shell indexes, adding false *MPLS clusters*. Finally, it draws our attention that the ratio of *u-turn links* differs greatly just on ASes with low r_{mpls}. The issues that produces theses *u-turn* bias in some ASes and not in others are not clearly at all.

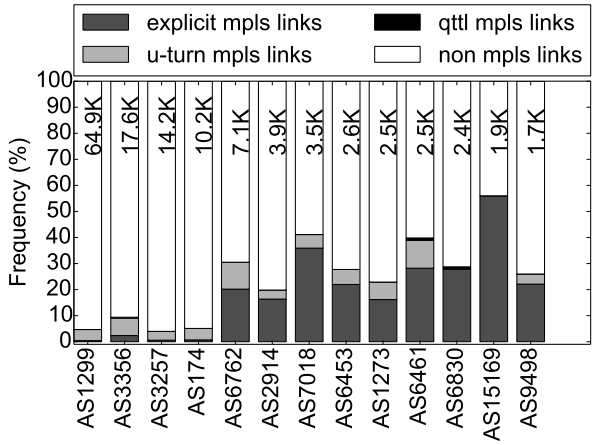


Fig. 5: **Top of ASes with most links discovered** On top four ASes prevails *u-turn mpls links*. ON the rest of ASes prevails *qttl mpls links*.

VI. Conclusion

Acknowledgments

This work is partially funded by the European Commission funded mPlane ICT-318627 project.

References

[1] K. Calvert, M. Doar, and E. Zegura, "Modeling Internet topology," *IEEE Communications Magazine*, vol. 35, no. 6, pp. 160–163, June 1997.

[2] 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.

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

[4] 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.

[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.

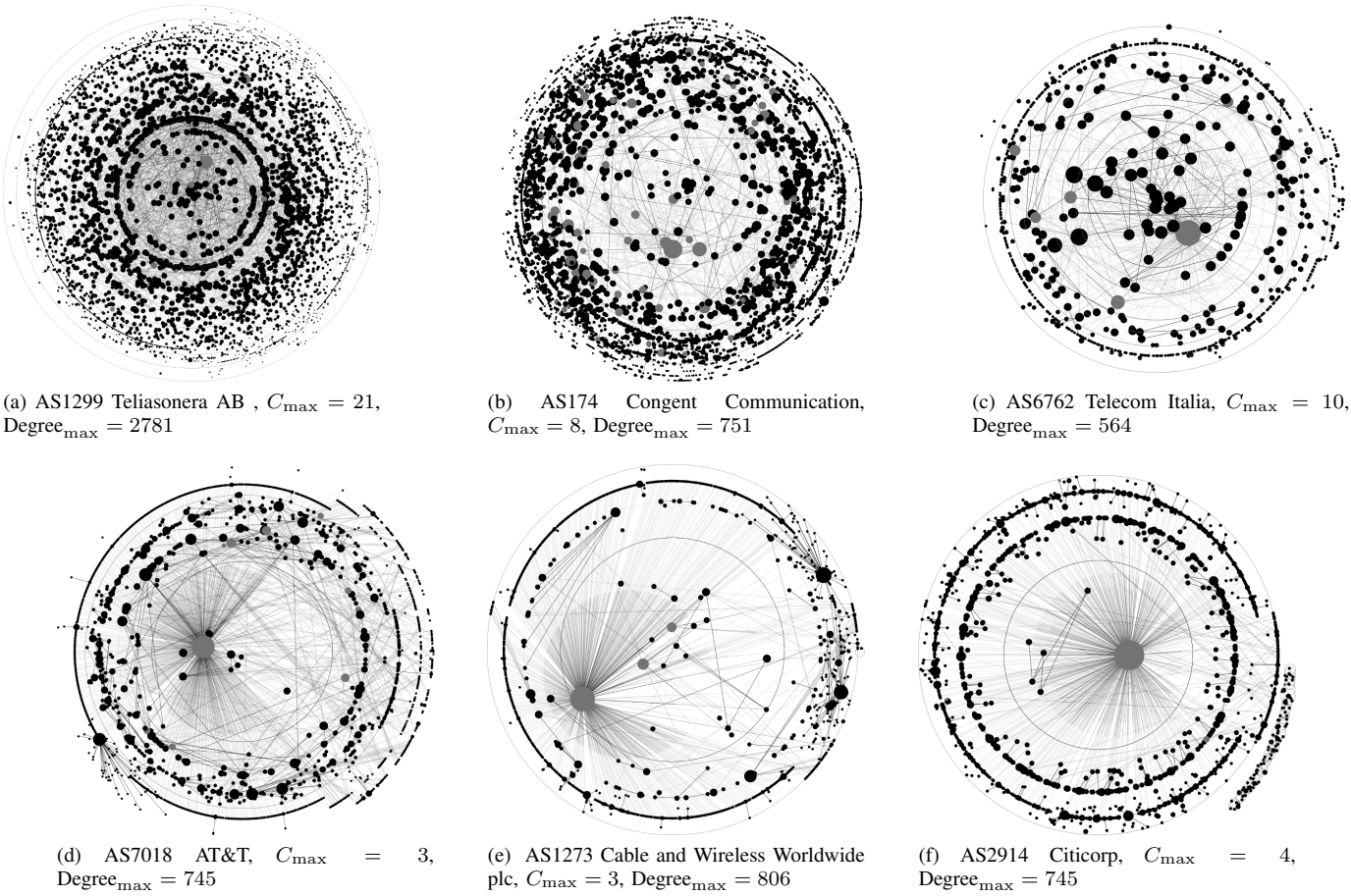
[8] V. Batagelj and M. Zaveršnik, "Generalized cores," *arXiv preprint cs/0202039*, 2002.

[9] J. I. Alvarez-Hamelin, A. Barrat, and A. Vespignani, "Large-scale networks fingerprinting and visualization using the k-code 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] J. I. Alavarez-Hamelin, A. Barrat, and A. Vespignani, "K-coe decomposition of Internet graphs: Hierarchies, self-similarity and measurement biases," *Networks and Heterogeneous Media*, vol. 3, no. 2, pp. 371–393, June 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.



(a) AS1299 Teliasonera AB , $C_{\max} = 21$, $\text{Degree}_{\max} = 2781$

(b) AS174 Congent Communication, $C_{\max} = 8$, $\text{Degree}_{\max} = 751$

(c) AS6762 Telecom Italia, $C_{\max} = 10$, $\text{Degree}_{\max} = 564$

(d) AS7018 AT&T, $C_{\max} = 3$, $\text{Degree}_{\max} = 745$

(e) AS1273 Cable and Wireless Worldwide plc, $C_{\max} = 3$, $\text{Degree}_{\max} = 806$

(f) AS2914 Citicorp, $C_{\max} = 4$, $\text{Degree}_{\max} = 745$

Fig. 6: **k-core visualization of MPLS cluster interconnection Graph** $G_{r\backslash lsr}(as)$. On the top the ASes show *MPLS clusters* spread out around the shell index of the decomposition. ON the bottom the ASes show *MPLS clusters* well defined and located on the top core $C_{\max}$.

[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] L. Andersson, I. Minei, and T. Thomas, "LDP specifications," Internet Engineering Task Force, RFC 5036, October 2007.

[16] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP tunnels," Internet Engineering Task Force, RFC 3209, December 2001.

[17] K. Muthukrishnan and A. Malis, "A core MPLS IP VPN architecture," Internet Engineering Task Force, RFC 2917, September 2000.

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

[19] 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.

[20] 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.

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

[22] 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.

[23] 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.