Leveraging the IPv4/IPv6 Identity Duality by using Multi-Path Transport

Ioana Livadariu*, Simone Ferlin*, Özgü Alay*, Thomas Dreibholz*, Amogh Dhamdhere† and Ahmed Elmokashfi*

*Simula Research Laboratory, 1364 Fornebu, Norway
Email: {ioana,ferlin,ozgu,dreibh,ahmed}@simula.no
† CAIDA/UCSD, La Jolla, CA 92093, U.S.A
Email: amogh@caida.org

***Abstract*—With the 20th anniversary of IPv6 nearing quickly, a growing number of Internet service providers (ISPs) now offer their customers both IPv6 and IPv4 connectivity. This makes multi-homing with IPv4 and IPv6 increasingly common even with just a single ISP connection. Furthermore, the growing popularity of multi-path transport, especially Multi-Path TCP (MPTCP) that is the extension of the well-known Transmission Control Protocol (TCP), leads to the question of whether this identity duality can be utilized for improving application performance in addition to providing resilience. In this paper, we first investigate the AS-level congruency of IPv4 and IPv6 paths in the Internet. We find that more than 60% of the current IPv4 and IPv6 AS-paths are non-congruent at the AS-level, which motivates us to explore how MPTCP can utilize the IPv4/IPv6 identity duality to improve data transfer performance. Our results show that MPTCP, even with a single dual-stack Internet connection, can significantly improve the end-to-end performance when the underlying paths are non-congruent. The extent of the improvement can reach up to the aggregate of the IPv4 and IPv6 bandwidths.**

I. INTRODUCTION

The free pool of IPv4 addresses is rapidly shrinking. In February 2011 the Internet Assigned Number Authority (IANA) assigned its last set of available addresses to the Regional Internet Registries (RIRs). In the following three years, three RIRs namely APNIC, RIPE, and LACNIC reached their last /8 of free addresses [1]–[3]. ARIN and AFRINIC are projected to reach their last /8 in February 2015 and August 2019 respectively [4]. This impending runout has encouraged many networks to adopt IPv6. The adoption, though dismal, is accelerating [5], [6]. The number of dual-stacked hosts that access popular Internet services over IPv6 is growing at a fast pace. For instance, the number of users that access Google over IPv6 has increased by five-folds in the past two years [7]. Many hosts in the Internet today exhibit an unprecedented identity duality that is likely to remain until IPv4 is phased out completely. This duality presents new opportunities, as dual-stacked hosts can choose either IPv4 or IPv6 to communicate with each other. The task of picking the right address, however, can be challenging. RFC 3484 recommends first trying IPv6 and switching to IPv4 if the IPv6 connection times out [8], which proved to be suboptimal for hosts with poor IPv6 connectivity [9].

Currently, most popular operating systems and web browsers implement the *Happy Eyeballs* algorithm to help pick the best address [10]. For instance, Google Chrome on a dual-stacked host tries IPv4 and IPv6 simultaneously and selects the one with best latency. However, simultaneous use of IPv4 and IPv6 paths is not part of the algorithm.

Multi-path transport allows the simultaneous use of multiple paths in the network to increase end-host robustness and to provide bandwidth aggregation [11]–[13]. Today, many devices could in fact support multi-path transport (e.g. smart phones with both 3G/4G and WLAN interface). However, the most prominent transport protocol, i.e. TCP, is still single path. Multi-Path TCP (MPTCP) [12] is a major extension of TCP that supports multi-path transmission. Its design is motivated by the need to be compatible with network middleboxes and hence, it is backward compatible with TCP. By pooling resources, MPTCP effectively increases end-to-end goodput especially when the paths go over distinct bottlenecks [12], [14]. The performance gains come from aggregating the bandwidth of the paths when the paths are completely incongruent (i.e. two distinct paths), or when the bottleneck is different even if the paths share some hops.

This paper is the first to explore whether the current identity duality can be leveraged to improve performance. We first investigate the congruency of the IPv4 and IPv6 paths by comparing paths between dual-stacked ASes using BGP data. We find that a large fraction of paths are non-congruent at the AS level. We then explore how to leverage this address duality to maximize resource usage with multi-path transport. Particularly, we experiment with a new use case of MPTCP that considers a dual-stacked interface as two separate interfaces. This use case differs completely from the conventional uses of MPTCP. The common use case for MPTCP requires having more than one network interface. Alternatively, if there is only one network interface, MPTCP relies on load balancers in the network to provide performance gains. However, the use case we consider requires only one dual-stack network interface and takes advantage of the non-congruency of the paths. To the best of our knowledge, this use case has not been considered in the literature before, although MPTCP supports both IPv4 and IPv6. We measure a marked increase in MPTCP goodput for IPv4 and IPv6 paths that are not sharing a bottleneck; the MPTCP goodput can be the aggregate of IPv4 and IPv6 goodputs. Our results show that the current identity duality and path incongruence can be leveraged to improve end-to-end performance for large bulk transfers.

II. BACKGROUND AND RELATED WORK

This section briefly describes the main building blocks of MPTCP and reviews the most related work.

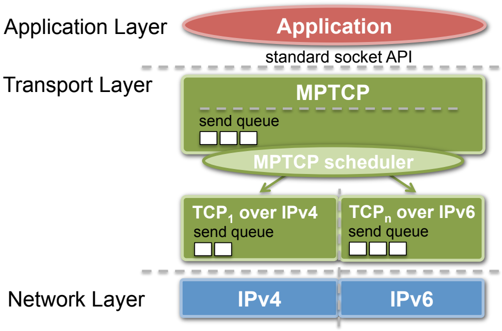


Figure 1: Multi-Path TCP over IPv4 and IPv6: Overview

A. Multi-Path TCP Background

MPTCP is a major extension to TCP, allowing the use of multiple paths simultaneously between two end-hosts for the transmission of a single data stream [15]. By pooling available resources, MPTCP effectively increases the goodput as the paths may go over different interfaces with distinct bottlenecks [12], [14]. The resource pooling is achieved by presenting a regular TCP socket to the application. However, below the socket, TCP connections are created for each path as illustrated in Figure 1. Together, these subflows form a MPTCP connection that uses TCP options to signal the necessary control information between the end-hosts. Thus, these TCP subflows let MPTCP look like regular TCP to firewalls/middleboxes and make it deployable on today's Internet [12].

In addition to resource pooling, another goal of MPTCP is to increase application resiliency. By design, the use of multiple paths implicitly increases application resiliency; data transmission continues as long as one of the paths is still usable. To further improve application resiliency, MPTCP allows senders to retransmit lost segments on different subflows. This retransmission strategy enables moving data from a path that fails during transmission [16].

MPTCP also extends the standard TCP congestion control, since running existing TCP congestion control algorithms independently would give MPTCP connections more than their fair share of the capacity in case of a shared bottleneck by two or more of its subflows. MPTCP proposes the use of coupled congestion control [17]–[19] that couples each subflows' congestion control and dynamically controls the overall aggressiveness of the MPTCP connection. The coupled congestion control also makes resource usage more efficient, as it steers traffic away from more to less congested paths.

Recently, a measurement-based performance evaluation of MPTCP has been carried out in real operational networks where the authors investigated the smartphone use-case with one WIFI and one LTE interface [20]. The authors evaluated the performance for different type of applications (bulk transfer and short flows) and argued that MPTCP offers no appreciable benefit over TCP for shorter flows, while improving the performance for longer flows.

The current MPTCP Linux implementation[1] (v0.89) supports both IPv4 and IPv6, however, no previous work considered the use of IPv4 and IPv6 simultaneously to take advantage of non-congruency of these paths. By transparently supporting multiple subflows from different address families, MPTCP can benefit from path non-congruency to both increase resilience and throughput.

B. IPv4 and IPv6

The impending runout of IPv4 addresses motivated several studies to measure and investigate different aspects of IPv6 adoption, e.g., end-user IPv6 capabilities, AS-level topology and end-user performance. Huston [21] updates IPv6 statistics of the IPv6 topology and routing evolution. In [5], the authors presented an in-depth analysis of the evolution of the IPv6 deployment in terms of AS-level topology, routing, and performance. Recently, Czyz et al. [6] presented a complete analysis of the IPv6 adoption process using multiple sources of data, i.e. routing, traffic, naming and content data, and various metrics to characterize this process. Both studies showed an increase in IPv6 uptake, most of which takes place in the core of the Internet. The former measured IPv4 and IPv6 AS-level congruency and concluded that there is a lack of congruency, but the gap is steadily shrinking. In this paper, we measure and analyze IPv4 and IPv6 congruency in more detail. We focus on assessing the AS-level congruency of the paths using BGP data from a large set of vantage points. Further, we assess the router-level congruency for a selected set of paths. To this end, we employ existing path discovery and bottleneck bandwidth estimation tools.

III. ARE IPV4 AND IPV6 PATHS CONGRUENT?

This section uses BGP routing data to investigate IPv4 and IPv6 AS-level paths similarity from the perspective of a set of dual-stacked vantage points (VPs).

A. Data Set

We use Border Gateway Protocol (BGP) paths data derived from BGP routing table snapshots collected by the RouteViews project [22] and RIPE's Routing Information Service (RIS) [23]. The data consists of monthly snapshots of paths seen at each VP in the first five days of the month. A subset of VPs are default-free, meaning that they have a route to almost all destination ASes. As of June 2014, 119 of the VPs at Route Views and RIPE are dual-stacked i.e. they provide both IPv4 and IPv6 BGP views. To identify the dual-stacked VPs that are default-free, we computed the fraction of origin ASes to which each VP had a route. Out of the total number of VPs, 72 VPs had a route to more that 90% of the ASes in both the IPv4 and IPv6 topology. In the following, we compare paths from these 72 ASes to all dual-stacked ASes.

B. Comparing IPv4 and IPv6 AS-PATHs

For each pair of a VP and a dual-stacked destination AS, we compare the IPv4 and the IPv6 AS-level paths. To compare IPv4 and IPv6 paths from the same source to a given destination, we first remove AS-PATH prepending, then compare the resulting path pair hop-by-hop. The pair is congruent if these paths are identical. Investigating the latest topology snapshot from June 2014, we find that 62% of all dual-stacked paths seen by the 72 VPs are *non-congruent*.

The congruent paths are relatively short; 50% of all congruent paths are 1-hop long, and 38% are 2-hops long. We further analyze the non-congruent paths to understand the extent of the incongruity. 61% of non-congruent paths have no hops in

---

[1] Linux MPTCP: http://www.multipath-tcp.org.

common, and only 8% share more than one hop. For non-congruent paths that share at least one hop, we find that in 91% of the cases the shared hops are adjacent to the source or destination AS on the path. Having analyzed the congruence of all paths, we next break it down per VP. Figure 2(a) shows the CDF of the fraction of non-congruent paths per VP. We note that there is a high variability across VPs; the percentage of non-congruent paths varies between 27% and 91%, depending on the VP. We find that all VPs for which the percentage of non-congruency is high, i.e. more than 85% are edge networks, whereas VPs for which the percentage of non-congruency is less than 80% are both edge networks and transit providers. Moreover, VPs for which the non-congruency is low, i.e. less than 35%, are transit providers.

Congruency over time. We further investigate how path congruency is changing with time. Figure 2(b) shows the time evolution of the distribution of the fraction of non-congruent paths seen by a VP. The plot presents the median, quartiles, maximum, and minimum fraction of non-congruent paths across VPs. The median fraction of non-congruent paths has decreased from 85% in June 2009 to 77% in June 2014. However, in the same period, both the number of IPv6 prefixes and ASes have increased by nine-folds [21]. Accordingly, the path congruency is increasing at a much slower pace compared to the growth of the IPv6 Internet. We investigate the non-congruent paths to build an idea about the causes behind this significant lack of congruency. Back in 2009, about 70% of non-congruent paths include at least one AS that did not adopt IPv6 at the time. Surprisingly, however, this percentage follows a monotonically decreasing trend over time and drops to below 40% in 2014. In other words, almost two thirds of the non-congruent paths are incongruent because ASes are preferring different peers in the IPv6 Internet. This shows that the observed incongruence can not be explained by the lack of IPv6 adoption alone and that *incongruency is likely to remain a feature of the IPv6 Internet*. Further, we compare the paths that become congruent to understand why congruency is increasing over time. After controlling for new VPs, we split the paths that become congruent into paths to new IPv6 adopters and existing paths. We observe that the contribution of the latter is increasing over time. Half of the paths that became congruent in 2014 were non-congruent before that.

C. Summary

Our results show a prevalent lack of congruency that is not a mere transient behavior. We note that the measured percentages should be taken as lower estimates. End-to-end paths could possibly be non-congruent even if they are identical at the AS-level, since they may differ at the router-level. This prevalence in lack of congruency bodes well if we want to leverage the duality in identity. Further, the slow rate at which path congruency is increasing suggests that a solution exploiting the lack of congruency will stay relevant in the medium term and maybe longer. In the next section, we run a set of controlled experiments to check whether MPTCP can capitalize on this lack of congruency to improve end-to-end data transport performance.

IV. EXPERIMENT SETUP AND PERFORMANCE ANALYSIS

A. Experiment Setup

In order to evaluate the benefits of MPTCP using both IPv4 and IPv6, we run experiments using dual-stack VMs hosted in different countries: Germany (DE), Japan (JP), U.S.A. (US) and Norway (NO). We use 4 VMs, three of the VMs are single-homed to a single ISP, while the VM in Oslo is multi-homed to three different ISPs (Uninett, Kvantel and PowerTech). Considering all VMs, we obtain a total of 12 VM-pairs that is 6 IPv4 pairs and 6 IPv6 pairs. We run our experiments just in one direction, hence we measure and experiment on 12 different combinations of paths as shown in Table I. The transfer direction is indicated by → in the table. We would like to note that while trying to find dual-stacked hosts for our measurements, we faced several challenges. First of all, several existing wide-area testbeds (e.g. Planet-lab [24], GENI [25]) do not support IPv6. Furthermore, it is very difficult to find IPv6-ready infrastructure hosting providers outside Europe and North America, which could provide different paths to our analysis. We observed unstable IPv6 performance for the VMs we tested in Asia, South America and Oceania. The instabilities were outside the service provider network and could not be circumvented. The instability made a qualitative comparison between IPv4 and IPv6 impossible. We also note that the main goal of our measurement study is to investigate whether using a dual-stacked interface as two distinct interfaces can actually provide benefits in reality and compare the performance when IPv4 and IPv6 paths are congruent and when they are not. To this end, it is sufficient to experiment with a small number of hosts to quantify and compare the two cases above. A large measurement campaign would be beneficial in revealing the parts of the Internet where MPTCP does not work due to middleboxes and this is part of our future work.

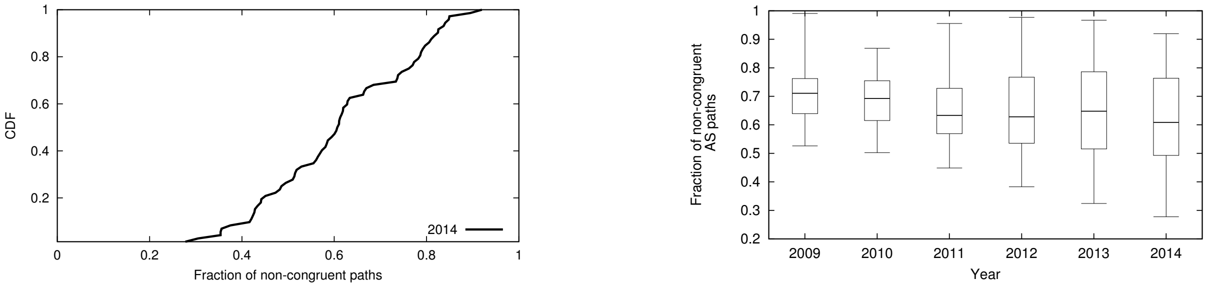
B. Methodology

The path analysis for all pairs in Table I is performed in three main steps. First, we analyze the path stability and congruency by running `traceroute` with ICMP, TCP and UDP every 10 minutes for a period of 22 days. We then identify the most stable path for each pair, which we define as the path that appears at least in 70% of the `traceroute` runs, and map each IP hop to its corresponding AS. Second, we check whether IPv4 and IPv6 paths connecting the same pair are congruent by comparing the identified IPv4 and IPv6 AS paths as well as using DNS names to identify common routers. Paths that share same AS-hops, however, are not necessarily congruent since they may cross different routers. Hence, we finally use `STAB` [26] to estimate the available bandwidth along the IPv4 and IPv6 paths. Next, we use the estimated available bandwidth as a suggestive measure as to whether the hops are shared or not. `STAB` is an end-to-end tool that actively probes all intermediate hops by sending packets of different sizes and rates called *chirps*. We validate `STAB` results using our knowledge about the topology of the VM-sites.

Before experimenting with MPTCP, we confirm that MPTCP transparently supports dual-stacked interfaces, i.e., TCP subflows are established on both IPv4 and IPv6 addresses. When we started, the current implementation of MPTCP in Linux only supported transfers that begin by negotiating over the IPv6 address. A recent modification to the implementation, however, now allows for the first subflow to be opened on either IPv4 or IPv6 address[2]. We perform bulk-transfers using IPv4, IPv6, and MPTCP on a dual-stacked interface between all 15 pairs. We transfer a large file of size 128 MiB for 70 times for each of the 15 paths between July and October

---

[2] Patch: http://github.com/multipath-tcp/mptcp/commit/597173ef46a36521f1dd0d8ff1e00c728aa90130.



(a) Fraction of non-congruent AS paths per VP in June 2014

(b) Fraction of non-congruent path (June snapshots)

Figure 2: Analysis of non-congruent AS paths

2014. TCP buffer sizes are set to the file size (128 MiB) and auto-tuning is enabled [27]. In addition, Linux MPTCP version 0.88.11 is used with its default coupled congestion control, i.e. OLIA [28].

C. Analysis of the Paths used in Experiments

We start by analyzing each of the network paths discussed in Section IV-A. Our goal is to identify whether the IPv4 and IPv6 paths are non-congruent and do not share a bottleneck. To this end, we first compare the IPv4 and IPv6 paths for each pair to determine whether they are AS-level congruent. Second, for the non-congruent paths, we determine whether they share any common hops. Finally, for the paths that share common hops, we determine whether these hops are the bottleneck. In Table I, we present the results of our analysis for each path where we classify the paths as shared or distinct bottlenecks.

We observe that the majority of the measured paths are AS-level non-congruent (7 out of 12). Four of these paths do not share any hops, while three of them share one hop. The measured AS-level paths are between three and four hops long, whereas the router level paths are between 8 and 25 hops. As mentioned earlier, congruent paths at the AS-level are not necessarily congruent at the router level. Further, AS-level non-congruent paths that share at least one hop may still provide great benefits especially if the bottleneck does not lie on the common hop. That is, the performance of end-to-end transport over both IPv4 and IPv6 is mainly decided by the bottleneck.

Therefore, we further investigate all router level paths of AS-level paths and check whether the bottleneck lies on the shared segments by using `STAB`. Note that for non-congruent paths that share at least a single inter-AS link, a similar bottleneck bandwidth suggests that the bottleneck lies on the shared segment(s). Our results indicate that only the first path, HR-U, share a bottleneck link in the first hop. We compare the DNS names of the IP hops that lie on the shared part of the two paths, which confirm that the two paths indeed share common routers. In Table I, we group the pairs in different categories, based on whether the paths have distinct or shared bottlenecks. Next, we analyze the interplay between these paths and MPTCP performance.

D. RTT and Loss Comparison of IPv4 and IPv6 Paths

We start by investigating the delay and loss characteristics of TCP connections over IPv4 and IPv6 paths to check for qualitative differences. This helps to understand the impact on MPTCP's scheduler and congestion control mechanisms.

MPTCP's coupled congestion control shifts traffic away from the lossiest path [28], when there are significant differences in loss between paths. Also, differences in RTT will make MPTCP's default lowest-RTT scheduler to prefer one of the paths, thus affecting the multipath load distribution.

The RTT measurements were taken from the trace files, while losses over time were taken from the congestion window sampled on sub-RTT intervals for each path of each measurement. The RTT gives an indication of differences in the path length and loss distribution over time gives an indication of the path congruency. However, very important to note, loss correlation over time is very noisy, therefore, error-prone to be correlated. In our attempts, we isolated slow-start losses due to their higher rate and bursty nature. We could observe that these losses were clustered on congruent paths compared to non-congruent paths. The average RTT and loss values for each of the paths is shown in Figure 3.

All measurements over IPv4 and IPv6 paths were conducted sequentially with a randomized interval in the range of a few minutes. While the measurement intervals should not be too short, correlating subsequent instances, they should not be too long to capture a very different behaviour of the path.

Figures 3 show the mean (boxes) and the standard deviation (errorbars) of delay and loss for different VM-combinations. We observe for the selected VM combinations that IPv4 and IPv6 RTTs are mostly similar. Exceptions to this are HS-K and RL where the IPv6 RTT is up to 40% higher than it is for IPv4. MPTCP's default lowest-RTT scheduler can equally fill the congestion window of both paths, and since the delays are very similar, the head-of-line blocking effect is low. The loss characteristics on IPv4 and IPv6 paths exhibit slight differences. For example, RL has no loss on the IPv4 path, and HS-K shows almost twice as much loss on IPv6 compared to IPv4. Accordingly, we do not record striking differences in TCP performance between IPv4 and IPv6 paths that connect the same pair. Next, we investigate, in the light of the path characteristics in Table I, whether running MPTCP on IPv4 and IPv6 on the same interface improves the end-to-end performance.

E. MPTCP Performance Analysis

1) Distinct Bottleneck: MPTCP benefits in terms of throughput and resilience when IPv4 and IPv6 paths are disjoint and have distinct bottlenecks as illustrated in the box-plots in Figures 4(a) and 5[3]. The box-plots show the median, quartiles, minimum, maximum of the measured goodput values for each VMs-combination. Below, we summarize our results and findings.

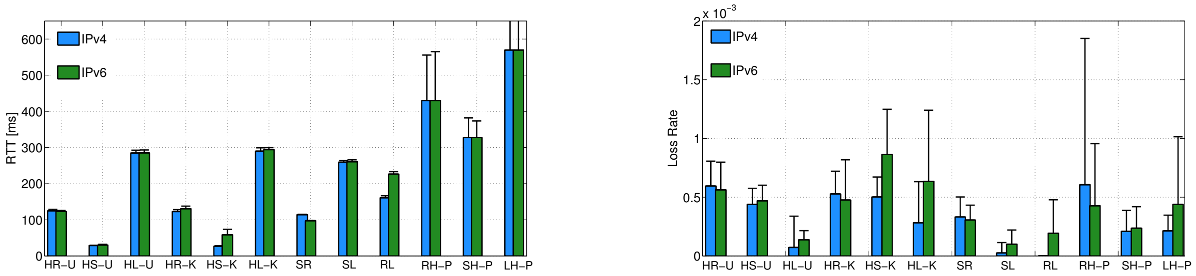
HS-U and HS-K: IPv4 and IPv6 paths for these pairs share a common AS-hop. The bottleneck links, however, do not lie on

---

[3] We present the PowerTech results in a separate figure since the throughput of a DSL link is much lower compared to a high speed link

Table I: Comparing the paths between VMs

Src (→)	**Dst**	**Abbr**	**AS congruency**	**Shared hop**	**Bottleneck**
UNINETT (NO)	Rackspace (US)	HR-U	Non-congruent	1-hop	Shared
UNINETT (NO)	ServerBiz (DE)	HS-U	Non-congruent	1-hop	Distinct
UNINETT (NO)	Linode (JP)	HL-U	Congruent	NA	Shared
Kvantel (NO)	Rackspace (US)	HR-K	Congruent	NA	Shared
Kvantel (NO)	ServerBiz (DE)	HS-K	Non-congruent	1-hop	Distinct
Kvantel (NO)	Linode (JP)	HL-K	Congruent	NA	Shared
ServerBiz (DE)	Rackspace (US)	SR	Non-Congruent	0	Distinct
ServerBiz (DE)	Linode (JP)	SL	Non-Congruent	0	Distinct
Rackspace (US)	Linode (JP)	RL-P	Non-Congruent	0	Distinct
Rackspace (US)	PowerTech (NO)	RH-P	Congruent	NA	Shared
ServerBiz (DE)	PowerTech (NO)	SH-P	Congruent	NA	Shared
Linode (JP)	PowerTech (NO)	LH-P	Non-Congruent	0	Distinct



(a) RTT

(b) Loss

Figure 3: Average RTT and Loss over IPv4 and IPv6

the shared path segment. We observe that MPTCP increases goodput by up to 15% when combining IPv4 and IPv6.

SR and SL: IPv4 and IPv6 paths connecting these pairs are completely non-congruent. In this case, MPTCP perfectly aggregates the bandwidth of IPv4 and IPv6 paths.

RL: In this particular path, TCP performs significantly better than MPTCP. Looking at our dataset, we observe that MPTCP always experiences loss early during the transmission which limits its inflight data and adversely affects its performance.We carry out further analysis to investigate the reason behind this behavior. We first run two competing TCP connections in parallel. Then, we run MPTCP with a single subflow. In both cases, we observe no early loss, so the performance is similar to the TCP performance. Note that when MPTCP is set to open a single subflow, the TCP packet header does not contain the MPTCP option. We do not observe other TCP header fields being changed (window size, port number, etc.). Hence, these losses only happen when using MPTCP with multiple flows, hinting that their cause might be a negative differentiated treatment by a middlebox against MPTCP.

LH-P: Although our analysis indicates that the paths are AS-level non-congruent, the access link (PowerTech DSL link) is the bottleneck. Therefore, MPTCP performs similar to TCP. Note that the access link bandwidth is ≈ 700 Kbit/s, which is much lower than the bandwidth of all other hops on the path.

To sum up, running MPTCP on a single dual-stacked interface indeed improves the overall throughput when paths do not share the tightest link. The benefit is more evident when paths are completely non-congruent. The presence of middle boxes that mishandle MPTCP packets can undermine this gain. Also, path non-congruency is of little utility if the access link is the bottleneck.

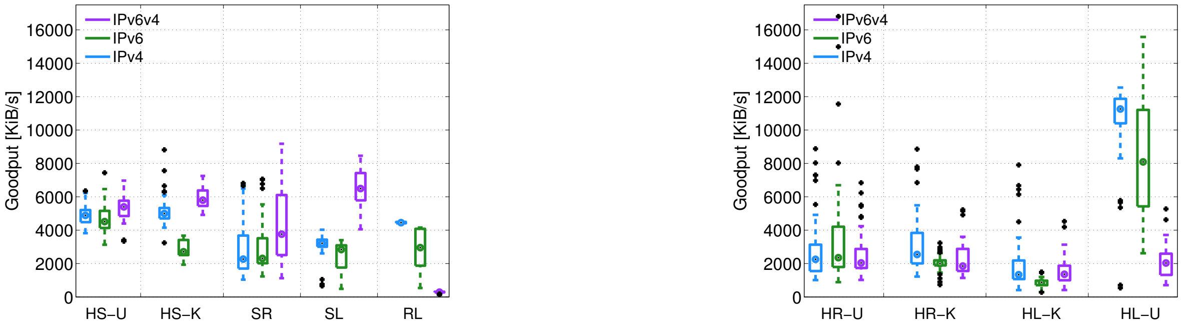
2) Shared Bottleneck: Subsubsection IV-E1 shows that MPTCP increases throughput when IPv4 and IPv6 paths have distinct bottlenecks. However, in the presence of a shared bottleneck, MPTCP does not take more than its fair share of the link, i.e., it behaves like a single TCP connection. This is one of the three main design goals of MPTCP [15], [18]. Hence, in such scenarios with shared bottlenecks, opening multiple MPTCP subflows is just as good as opening a single TCP connection. Figures 4(b) and 5 show the results for paths where IPv4 and IPv6 paths are congruent and/or share a bottleneck.

HR-U: For this pair, the IPv4 and IPv6 paths are non-congruent but they share a single AS-hop. The tightest link on the path is located on the shared hop, meaning that IPv4 and IPv6 paths cross the same bottleneck link. Hence, MPTCP does not provide any performance gains.

HR-K and **HL-K**: The IPv4 and IPv6 paths share all hops, which also includes the bottleneck.

HL-U: The IPv4 and IPv6 paths share all hops, which also includes the bottleneck. However, we again observe what we suspect to be traffic differentiation between TCP and MPTCP as in RL.

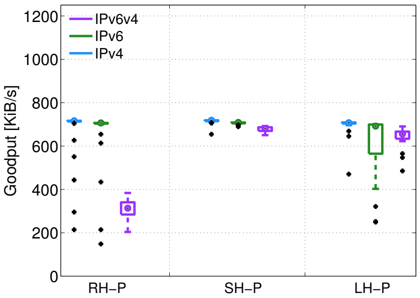
RH-P and **SH-P**: Here, the IPv4 and IPv6 paths share all hops and access link (PowerTech DSL) is the bottleneck. We measure lower MPTCP performance compared to TCP in RH-P. We observe more losses with MPTCP, we further measure with two subflows over IPv4 and two over IPv6 and observed similar results. Since we cannot locate where losses actually happen on the path, neither identify changes in the packet headers, we leave this scenario for future investigation. To summarize, running MPTCP on a single dual-stacked interface does not provide any benefit when the two paths share the tightest link. This is also valid when the access link is the bottleneck. In the presence of a shared bottleneck, MPTCP does not take more than its fair share of the link, i.e., it behaves like a single TCP connection which is one of MPTCP's three main design goals [15], [18].



(a) Distinct Bottleneck

(b) Shared Bottleneck

Figure 4: TCP and MPTCP over IPv4 and IPv6 Paths with Distinct and Shared Bottlenecks



(a) Goodput: Downlink

Figure 5: TCP and MPTCP over IPv4 and IPv6 over DSL

V. DISCUSSION AND CONCLUSIONS

With IANA having reached the IPv4 exhaustion phase, the number of dual-stacked hosts is increasing as more networks are adopting IPv6. During this transition, end-hosts have dual identities and significant performance gains can be achieved if this dual identity is leveraged using multipath transport. In this paper, we analyzed the IPv4 and IPv6 path congruency and investigated how MPTCP can exploit the underlying path diversity to improve the performance. We first evaluated the AS-path congruency in the Internet and showed that more than 60% of the current IPv4 and IPv6 AS paths are non-congruent. Motivated by this high non-congruency, we evaluated the congruency of IPv4 and IPv6 in real networks using a set of VMs and analyzed MPTCP performance. For the non-congruent paths, we first investigated whether there are any common hops and whether these hops share a bottleneck. We observed significant MPTCP performance gains up to the aggregate of the individual IPv4 and IPv6 paths in non-congruent links where the paths are not sharing any bottleneck. However, when the paths are congruent or when IPv4 and IPv6 share a bottleneck, MPTCP is just as good as TCP in accordance with MPTCP's main design goals. Based on our analysis, we believe that there is a great potential in using the current IPv4 and IPv6 identity duality, together with multipath transport to provide performance improvements as well as resilience. The proposed MPTCP use case is easily implementable and can improve the performance for MPTCP-capable dual-stacked users trying to access dual-stacked content.

REFERENCES

[1] APNIC, "APNIC IPv4 Address Pool Reaches Final /8," Apr. 2011.

[2] RIPE NCC, "Allocations from the Last /8," Aug. 2010.

[3] LACNIC, "No More IPv4 Addresses in Latin America and the Caribbean," 2014.

[4] G. Huston, "IPv4 Address Report," Sep. 2014.

[5] A. Dhamdhere, M. Luckie, B. Huffaker, K. C. Claffy, A. Elmokashfi, and E. Aben, "Measuring the Deployment of IPv6: Topology, Routing and Performance," in *IMC*, 2012.

[6] J. Czyz, M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil, and M. Bailey, "Measuring IPv6 Adoption," in *SIGCOMM*, 2014.

[7] Google, "IPv6 Adoption," Sep. 2014.

[8] R. Draves, "Default Address Selection for Internet Protocol version 6 (IPv6)," IETF, RFC, Feb. 2003.

[9] D. Wing and A. Yourtchenko, "Happy Eyeballs: Success with Dual-Stack Hosts," IETF, Standards Track RFC, Apr. 2012.

[10] E. Aben, "Hampering Eyeballs – Observations on Two "Happy Eyeballs" Implementations," Nov. 2011.

[11] C. Raiciu, S. Barré, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving Datacenter Performance and Robustness with Multipath TCP," in *SIGCOMM*, 2011.

[12] C. Raiciu, C. Paasch, S. Barré, A. Ford, M. Honda, F. Duchêne, O. Bonaventure, and M. Handley, "How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP," in *NSDI*, 2012.

[13] T. Dreibholz, "Evaluation and Optimisation of Multi-Path Transport using the Stream Control Transmission Protocol," Habilitation Treatise, 2012.

[14] Y.-C. Chen, Y. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, "A Measurement-Based Study of MultiPath TCP Performance over Wireless Networks," in *IMC*, 2013.

[15] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," IETF, RFC, 2013.

[16] C. Paasch, S. Ferlin, Özgü Alay, and O. Bonaventure, "Experimental Evaluation of Multipath TCP Schedulers," in *SIGCOMM CSWS*, 2014.

[17] M. Becke, T. Dreibholz, H. Adhari, and E. P. Rathgeb, "On the Fairness of Transport Protocols in a Multi-Path Environment," in *ICC*, 2012.

[18] D. Wischik, M. Handley, and M. B. Braun, "The Resource Pooling Principle," *ACM CCR*, 2008.

[19] C. Raiciu, M. Handley, and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols," IETF, RFC, 2011.

[20] S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan, "Wifi, lte, or both?: Measuring multi-homed wireless internet performance," in *IMC*, 2014.

[21] Geoff Huston, "IPv6 Reports," http://www.potaroo.net/index-v6.html.

[22] David Meyer, "Route Views Project," http://www.routeviews.org/.

[23] RIPE, "Routing Information Service (RIS)," http://www.ripe.net/ris/.

[24] PlanetLab, http://www.planet-lab.org.

[25] GENI, https://www.geni.net.

[26] V. J. Ribeiro, R. H. Riedi, and R. G. Baraniuk, "Spatio-temporal Available Bandwidth Estimation with STAB," *SIGMETRICS*, 2004.

[27] C. Paasch, R. Khalili, and O. Bonaventure, "On the Benefits of Applying Experimental Design to Improve Multipath TCP," in *CoNEXT*, 2013.

[28] R. Khalili, N. Gast, M. Popović, and J.-Y. L. Boudec, "MPTCP is not Pareto-Optimal: Performance Issues and a Possible Solution," *IEEE/ACM Transactions on Networking*, 2013.