INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2008; **21**:643–663
Published online 28 December 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/dac.916

IPv6 delay and loss performance evolution

Xiaoming Zhou[3,*,†], Martin Jacobsson[1], Henk Uijterwaal[2] and Piet Van Mieghem[1]

[1]*Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands*
[2]*RIPE NCC TTM, Singel 258, 1016 AB, Amsterdam, The Netherlands*
[3]*Philips Research, High Tech Campus 34 5656 AE, Eindhoven, The Netherlands*

SUMMARY

Although packet delay and loss are two important parameters of the Internet performance, to the best of our knowledge, the evolution of large-scale IPv6 delay and loss performance has previously not been studied. In this paper, we analyze more than 600 end-to-end IPv6 paths between about 26 testboxes of RIPE Network Coordination Centre over two years, and compare the delay and loss performance over time with their IPv4 counterparts. We present and discuss the measurement methodologies and show that IPv6 paths have a higher delay and loss than their IPv4 counterparts. The main reason for the worse performance stems from IPv6-in-IPv4 tunnels rather than from native IPv6 paths and such tunnels are still widely used today. Copyright © 2007 John Wiley & Sons, Ltd.

Received 1 February 2007; Revised 25 September 2007; Accepted 30 October 2007

KEY WORDS: IPv6; Internet measurement; delay; loss

1. INTRODUCTION

Qualifying the IPv6 performance [1] requires measurements collected over time, combined with information about the delay, the path, and the tunnel discovery. Earlier studies have mainly focused on IPv6 transition technologies [2] or on identifying IPv6 network problems in a dual-stack world by using measurements from only a few days [3, 4]. Compared with IPv4, IPv6 is still in its infancy and it is rarely used by real-life applications. Hence, there is a lack of knowledge about network performance of end-to-end IPv6 communication. In general, it can be stated that large-scale deployment of applications will only be successful if the perceived quality of these applications is sufficiently high. Therefore, we argue that studying the evolution of the large-scale IPv6 delay

*Correspondence to: Xiaoming Zhou, Philips Research, High Tech Campus, HTC-34 (5.063), 5656 AE, Eindhoven, The Netherlands.
†E-mail: Xiaoming.Zhou@philips.com

Contract/grant sponsor: NWO SAID

and loss is important in order to understand the performance of the current IPv6 networks, and to provide high-quality services for future Internet applications. Hopcount and delay jitter are two other important parameters, and their primary results of both IPv6 and IPv4 have been compared and analyzed by Zhou and Van Mieghem [5].

We investigate the IPv6 network performance in terms of delay and packet loss measured over two years. The data set for our study was obtained through measurements conducted by the RIPE Network Coordination Centre (NCC)[‡] and their Test Traffic Measurements (TTM) project [6]. RIPE NCC is a regional Internet registry for Europe, Middle East, and Central Asia. The TTM project measures the quality of connectivity of the Internet, including both IPv4 and IPv6. The data set used in this study contains active measurements between a set of about 26 test boxes supporting IPv6.

Our contribution is twofold. First, we present a measurement methodology to evaluate the IPv6 evolution performance by comparing IPv6 and IPv4 performance on a path-by-path basis (Section 3). Second, we investigate the different behaviors of native IPv6 paths and IPv6 tunnel paths over time. We show that the IPv6-in-IPv4 tunnel paths lead to a worse performance of IPv6 compared with IPv4, while most native IPv6 paths show similar performance as their IPv4 counterparts (Section 4). On the other hand, there is little difference in the loss performance between native paths and tunnel paths over time.

2. BACKGROUND

2.1. The transition techniques from IPv4 to IPv6

IPv6 allocates 128 bits to represent an address, while IPv4 only allocates 32 bits. The number of different combinations therefore increases from 2^{32} to 2^{64} networks of 2^{64} addresses, which significantly increases the pool of available addresses. This enables more efficient address allocations and reduces the need for address translation. The use of network address translators (NATs) sustains the explosion of end devices, but also greatly increases the network complexity, which is a barrier to the widespread introduction of point-to-point applications [7]. IPv6 can give every device its own IP address, which alleviates the need for NATs. In addition, IPv6 also offers other advanced capabilities with respect to security, autoconfiguration, and mobility.

Since a world-wide scale migration from IPv4 to IPv6 within a short period is unfeasible, three main transition techniques were invented to make the continuous transition from the current IPv4 Internet to IPv6 possible.

The first technique is *the dual-stack network*. This approach requires hosts and routers to implement both IPv4 and IPv6 using the same link layer. This enables networks to support both IPv4 and IPv6 services and applications at the same time during the transition period. At the present time, the dual-stack approach achieves a relatively good performance [2]. Although the dual-stack approach carries twice the complexity of a single stack, it appears to be the natural choice in the long run.

The second technique relies on *tunneling*. Tunneling enables new IPv6 networking functions while still preserving the underlying IPv4 network as it is. For instance, when an IPv6 packet is leaving an IPv6 domain and entering an IPv4 domain, the packet is encapsulated in an IPv4

[‡]www.ripe.net.

DOI: 10.1002/dac

packet by a border router and transmitted through the network. When the packet reaches the other end of the IPv4 network, it is decapsulated at the border of the receiving IPv6 network. Tunnels can be statically or dynamically configured: 6to4 (RFC 3056 [8]) is a technique to transport IPv6 traffic over IPv4 networks without the need for automatic or configured tunneling. 6over4 (RFC 2529 [9]) is another technique that uses an existing IPv4 domain with multicast support to create a virtual link layer for IPv6 hosts. Tunnels over generic routing encapsulation (GRE) (RFC 2473 [10] and RFC 1701 [11]) have an extra encapsulation header to enable IPv6 traffic forwarding over an existing IPv4 infrastructure, with minimum changes.

The last technique uses *a proxy and translation mechanism*. Translation is necessary in case no other methods like tunneling or native IPv6 are available. For example, when an IPv6-only host wants to communicate with an IPv4-only host. An example of such a technique is NAT–PT (Network Address Translator–Protocol Translator, RFC 2766 [12]), which performs address and protocol translation at the borders between non-homogeneous networks at the IP level. The drawback of using a translation mechanism is that there must be a mapping from each IPv4 address to an IPv6 address. Therefore, NAT-PT needs to use a pool of IPv4 addresses for assignment to IPv6 nodes on a dynamic basis as sessions are initiated across IPv4–IPv6 boundaries.

With different IPv6 transport mechanisms, IPv6 connectivity across the backbone can be set-up with multiple segments managed independently. For example, an enterprise may decide to deploy a dual-stack network to connect to an ISP with a native IPv6, while connecting to another ISP with IPv6 over IPv4 tunnels. Since different organizations are at different stages in their transition to IPv6, we have a mix of native paths and tunnels as well as a mix of single- and dual-stack nodes today.

Nevertheless, it is important to remember that the IPv4 infrastructure still remains the main source of revenue for ISPs and supports the most important services. Introducing IPv6 into these networks should not impact the current IPv4 network negatively.

2.2. Current IPv6 equipment support

Since IPv6 routers with multiple very high-speed interfaces (such as 10 Gigabit Ethernet) are not generally available, a good way to start an in-depth investigation of IPv6 performance is to know how the IPv6 performance differs from that of IPv4 on a low-speed router (e.g. Cisco 3700 series). Depending on the router type and the implementation of the forwarding plane, IPv6 address look-ups are performed by a software forwarding router (i.e. a device using its main central processing unit for basic and enhanced packet forwarding) or a hardware forwarding router (i.e. a device that has hardware assistance for the basic and/or enhanced packet forwarding) [13].

Because the IPv6 look-up is—theoretically four times—more demanding, there is a natural tendency to leverage hardware-based look-up engines (e.g. Cisco 12000 series and the Cisco Carrier Routing System) as much as possible. On the other hand, the IPv6 header is considerably simplified; the classic IPv4 header contains 14 fields, whereas IPv6 only requires eight fields, which results in more efficient processing by routing nodes.

Hardware-based IPv6 look-up generally leads to line-rate forwarding at all interface speeds for most packet sizes. Software processing of the IPv6 look-up (e.g. Cisco 7500 series router) takes more time than for IPv4 because more bits must be processed. For example, when both IPv6 and IPv4 UDP packets with different packet sizes are forwarded through a Cisco-software-based forwarding platform (e.g. Cisco 3700 series), IPv6 and IPv4 forwarding performance of packets with large packet sizes (i.e. $\geqslant$128 bytes) are comparably good. However, when forwarding

DOI: 10.1002/dac

packets with small packet sizes (i.e. $\leqslant$128 bytes), IPv4 outperforms IPv6 by about 28% in terms of throughput [13]. If we repeat the experiment with a Cisco-hardware-based forwarding platform (e.g. Cisco 12000 OC48), the forwarding performance with small packet sizes (i.e. $\leqslant$128 bytes) improves because of the hardware implementation. In this case, the IPv6 and IPv4 forwarding performances are comparable [13]. The other advantage of hardware forwarding is that IPv4 and IPv6 traffic will not compete for processor resources.

2.3. Related work

To the best of our knowledge, hardly any work has quantified the IPv6 performance over time.

Srivastava *et al.* [14] describe the implementation of an IPv6 testbed and the inter-connection between three domains using IPv6-in-IPv4 static tunnels. They investigated performance issues (like throughput, packet loss and delay) of aviation applications (such as Controller to Pilot Data Link Communication) using Diffserv on that IPv6-based backbone network. Their results suggest that Diffserv implementation and support in IPv6 has matured enough to provide stable and reliable quality of service (QoS) for the aviation applications.

Adam *et al.* [2] analyzed the issues of the implementation of the IPv6 service, IPv6 performance (in the context of a high-speed network), the advantages of current transition technologies, and the problems encountered. They also provided a performance comparison between three different transition mechanisms: IPv6 in IPv4 tunneling, 6PE tunneling (IPv6 over an IPv4 MPLS network), and dual stack in a local very high-speed broadband network. Their experiments indicated that the current dual-stack approach already achieves good performance.

Cho *et al.* [3] measured both IPv6 and IPv4 round-trip delays from two locations. Their results show that the majority of IPv6 paths have delay characteristics comparable to those of IPv4.

In [5], we compared and analyzed the hopcount, end-to-end delay, and delay variation (jitter) between IPv6 and IPv4 in a month.

3. METHODOLOGY

First, we review measurement metrics. Second, we gather RIPE IPv6 data, which include traceroute, delay, and loss measurements among a list of IPv6 sites since 2003. We also run a tunnel discovery mechanism to distinguish between native IPv6 paths and tunneled paths and study their differences. Finally, based on our observations, we list some challenges for measurement and analysis in IPv6 networks.

3.1. Definitions of one-way delay and loss

In QoS enabled networks, QoS in one direction may be different than that in the reverse direction. Thus, measuring one-way delay and loss allows us to better understand the asymmetric path routing performance. Understanding one-way packet delay and loss from a source (Src) to a destination (Dst) is motivated by the fact that excessive packet delay or loss (relative to some threshold value) could degrade the perceived quality of certain real-time applications. The larger the delay and loss are, the more difficult it is for transport-layer protocols to sustain high throughput.

The one-way packet delay is the difference between the arrival time at Dst and the departure time at Src [15]. If a packet fails to arrive within a reasonable period of time (such as 10 s), the one-way delay is undefined (RFC 2679 [16]). In the measurements, each measurement packet is

DOI: 10.1002/dac

time stamped just before it is sent on the socket by the application and transmitted to the network interface card (NIC). If the packet arrives within a reasonable period of time, the application at the destination takes the arriving timestamp from the kernel. By subtracting the two timestamps, an estimate of the one-way delay can be computed. The delay between two nodes is the result of many factors, for example, the geographical distance, the number of hops, the load and capacity of the links, the policy routing decisions made along the path and even the way IP packets are transported in the layer 2 architecture. However, the minimum delay mainly reflects the propagation and transmission delay.

The global positioning system (GPS) gives a measurement accuracy of about 10 μs, while the network time protocol only gives a measurement accuracy of several ms. Since delay values often can be as low as 100 μs, it is important for the Src and Dst to synchronize their clocks precisely with GPS. Uncertainty in these values must be taken into account in the error analysis. By observing the TTM delay distribution between a random Src–Dst path over a day, we discovered that few packets suffered from a much longer delay than others. To better understand the sources of uncertainty or how errors affect the sending and receiving sides, we run test measurements (with different packet inter-arrival times, such as 100 packets per second) to verify the delay accuracy. These tests were executed in the lab of RIPE using a simple setup: two test-boxes connected back-to-back with a 1-m-long Ethernet cross-over cable. Our experiments showed that errors are not mainly created on the sending side, because each measurement packet is actually transmitted immediately after it is time stamped and sent by the measurement application. However, measurement errors were created on the receiving side. This is due to the way the kernel handles received packets and the time it takes before the timestamp is actually read. When a packet arrives, the NIC sends an interrupt to the operating system (OS). At that time, the OS may be busy with another process and this introduces a short delay before the packet is handled, which leads to an additional random delay. However, when this is accounted for (by taking the timestamp in the kernel), the accuracy of the delay measurements becomes about 25 μs.

The one-way packet loss is exactly zero when the one-way delay is a finite value, and 1 when the one-way delay is undefined (RFC 2680 [17]). Packet loss occurs where network traffic fails to reach the destination within a reasonable period of time. Losses may be due to congestion of the network, changes in paths between the source and destination, or incorrect routing [15].

3.2. Experimental setup: RIPE TTM

Our data set was provided by RIPE NCC. The RIPE TTM configuration is described by Georgatos *et al.* [6]. When the data were collected, the TTM infrastructure (which is solely in the core network) consisted of approximately 26 IPv6 measurement boxes scattered over Europe, Asia, and the USA as shown in Figure 1(a). As RIPE NCC is connected to the Amsterdam Exchange Point, it maintains many IPv6 peers with other 6net participants, using BGP4+ as Exterior Gateway Protocol, see Figure 1(b). Between each path of measurement boxes, both IPv6 and IPv4 UDP packets with a fixed payload (100 bytes), called probe packets, are continuously transmitted with an average inter-arrival time of about 30 s, resulting in a total of about 2886 probe packets on each path per day. These probe packets perform approximately a Poisson sampling [18] of the traffic along the path between two measurement boxes. The sending measurement box generates an accurate time stamp synchronized via GPS in each probe packet, while the receiving measurement box reads the GPS time of the arrival of the probe packet. The end-to-end delay is the difference between

DOI: 10.1002/dac

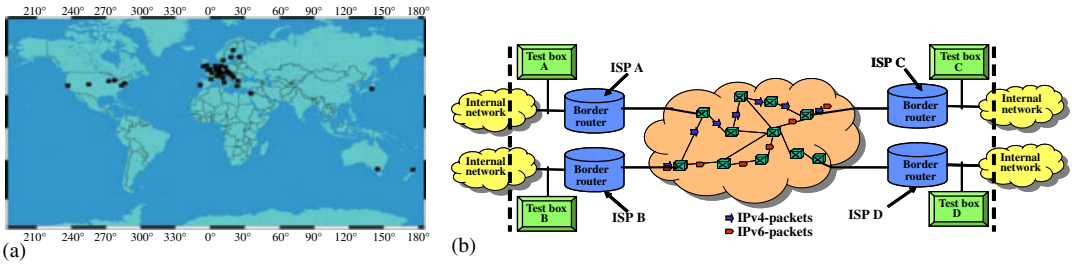

Figure 1. RIPE TTM: (a) location of the RIPE test-boxes and (b) experiment setup.

these two time stamps. The hopcount of a path between two measurement boxes is measured every 6 min using traceroute. IPv6 paths are monitored in a similar way with traceroute6.

A maximum transmission unit (MTU) detection algorithm, written by Colitti *et al.* [19], is run once per hour. In addition to the traceroute measurements, we use a tunnel discovery tool to identify different tunnels on those IPv6 paths. The tunnel discovery tool detects an IPv6 tunnel by measuring the MTU size (normally 1500 bytes) over an entire path. If a path contains a tunnel, the MTU on that path will usually be lower than 1500, since extra headers are added to the packet. However, this method is not perfect as not all links have an MTU of 1500 bytes. Another problem is that the tunnel discovery tool cannot detect more than one tunnel. For instance, if there is an IPv6-in-IPv4 tunnel (MTU 1480) followed by a GRE tunnel (MTU 1472), the tunnel discovery tool is only able to detect the second tunnel (which has the lowest MTU). In any case, a returned value of '1500' indicates a native path, while anything lower most likely means that the path contains at least one tunnel. The MTU value of a tunnel depends on the specific tunnel type: 1480 for IPv6-in-IPv4 tunnels; 1476 and 1472 for GRE tunnels; and 1280 for BSD tunnels.

3.3. Research challenges

Before presenting the analysis, we formulate two research challenges.

The *first* research challenge lies in analyzing the big measurement database. We have analyzed more than 400 GB zipped data collected over 2 years (from October 1, 2003 to October 31, 2005). Figure 2 shows the numbers of active IPv6 and IPv4 testboxes in the TTM infrastructure over these 2 years. Not all boxes are active all the time due to system updates or failures. The number of IPv4 paths is much larger than the number of IPv6 counterparts. Figure 2 also shows that the numbers of active IPv6 testboxes and paths of TTM have been steadily increasing over time. On October 1, 2003, there were 15 active IPv6 testboxes with 210 active IPv6 source–destination paths; by the end of October 2005, these numbers have increased to 29 and 811, respectively. For a fair comparison, only those testboxes supporting both IPv4 and IPv6 traffic were selected in our study.

The *second* challenge is that the evolution of IPv6 tunnels complicates the analysis. Some IPv6 tunnel paths changed to native paths, and *vice versa*. Figure 3 shows the numbers of active native and tunnel paths over the last 2 years. Figure 3 shows that in October 2003 about 61% of the total IPv6 paths were native paths, while by the end of October 2005, this number increased to 86%. We observe that there were 328 IPv6 tunnel paths before August 30, 2005, while about 31% of

DOI: 10.1002/dac

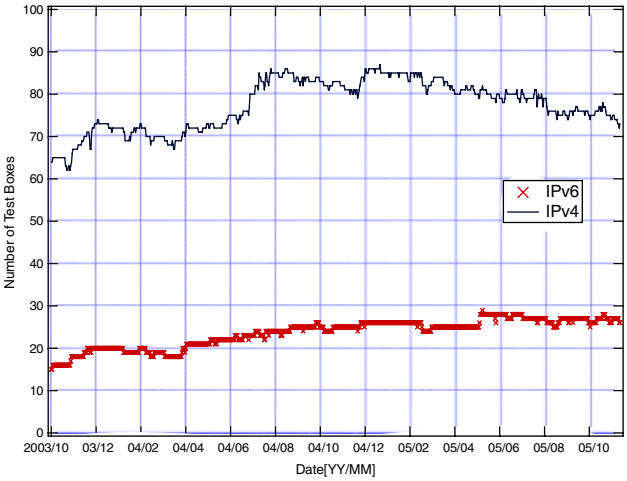

Figure 2. The number of active testboxes over time.

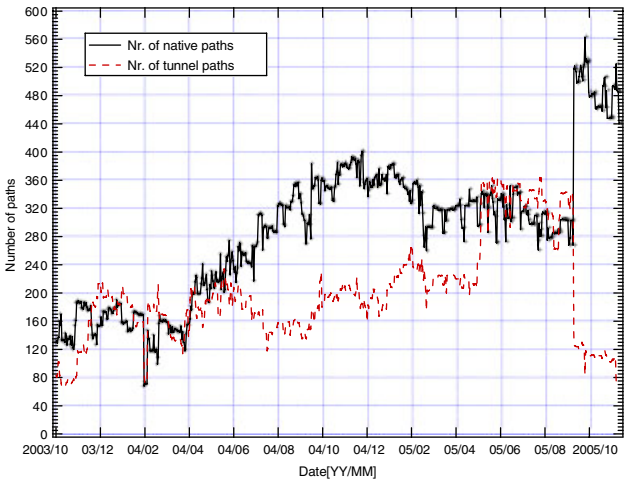

Figure 3. The number of active IPv6 native and tunnel paths over time.

those paths have changed to IPv6 native paths after that, since several ISPs upgraded their routers to dual stack for better performance.

3.4. Presentation of the data

We use the IPv4 performance as a comparison base, and compare the IPv6 delay and loss performance and the corresponding IPv4 performance on a path-to-path basis. Note that the minimum, average, and maximum IPv6 and IPv4 delays are sensitive to clock errors. To exclude the systematic error and the random error from the results, following the idea of RFC 2679 [16], we show the 2.5 percentile, the median, and 97.5 percentile delay for each Src–Dst path. That is, for each

DOI: 10.1002/dac

Src–Dst path i, we first made the delay histogram distribution over a time interval (e.g. one day). Then, we computed the 2.5 percentile $D_{2.5}(i)$, median $D_{50}(i)$, and 97.5 percentile $D_{97.5}(i)$ delay values for that path. We repeated this experiment for all the paths. When all paths were computed, we presented the average values of the 2.5 percentile (such as $(1/n)\sum_{i=1}^{n} D_{2.5}(i)$, where n is the number of paths), median, and 97.5 percentile values of all paths.

4. DELAY AND LOSS PERFORMANCE

4.1. Evolution of delay performance of all TTM paths over 2 years

First, we compare the general IPv6 delay and loss performance with their corresponding IPv4 performance over time, then we study and discuss the native IPv6 and IPv6 tunneled paths performance, separately. Figures 4(a)–(c) show the average 2.5 percentile, median, and 97.5 percentile over 2 years with one day intervals, and Figure 4(d) shows the loss comparison between IPv6 and IPv4.

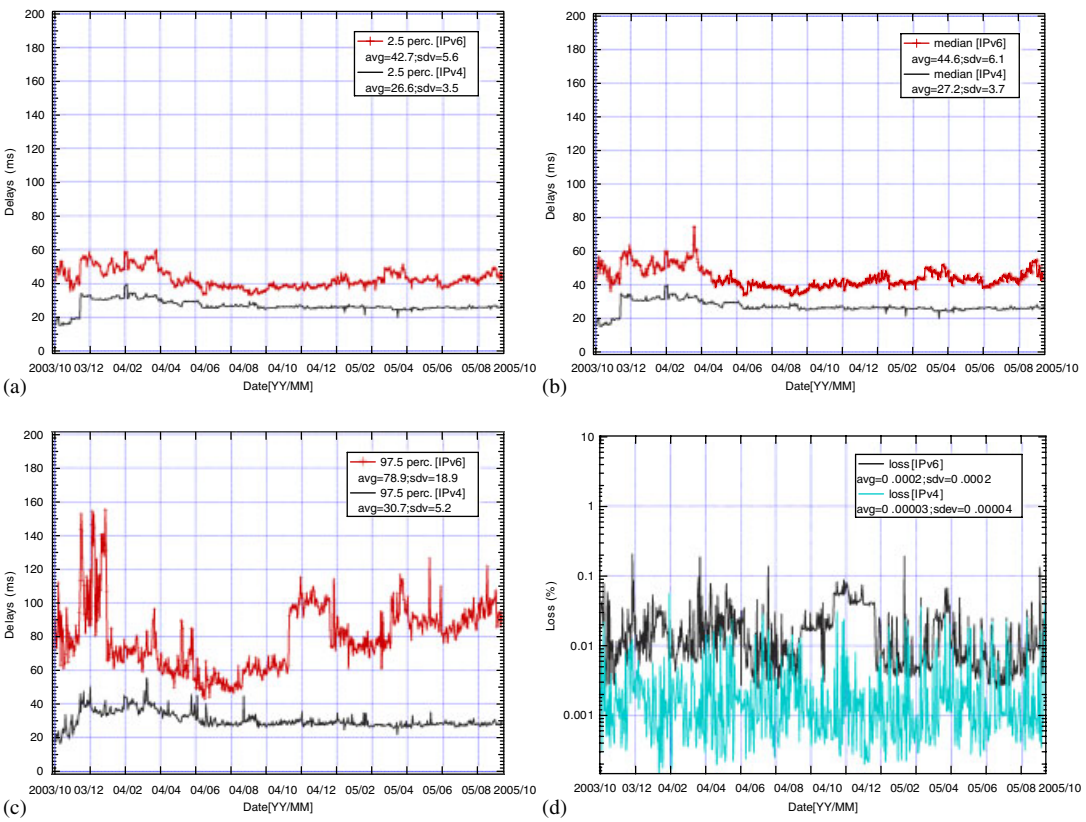

Figure 4. Performance of IPv6 paths and the IPv4 counterparts over time: (a) 2.5 percentile; (b) median; (c) 97.5 percentile; and (d) loss performance.

DOI: 10.1002/dac

Figure 4(a), (b), and (c) illustrate that, on average, IPv6 has an about 61% higher 2.5 percentile delay than the IPv4 counterparts, while about 64 and 157% higher median and 97.5 percentile delays, respectively. Hence, both the average and the variation of the delay are worse for IPv6. Figure 4 shows that since late 2003, IPv6 has a larger average delay than the IPv4 counterpart. Over time, the latter has been steadily and slightly decreasing. Unfortunately, the same cannot be said for IPv6.

Figure 4(d) shows that the packet loss of all paths over the 2 years for both IPv4 and IPv6 are small (less than 0.02%), and do not change much over time. On the other side, our results show that over the years, the IPv6 loss is about one order of magnitude larger than the IPv4 loss. Owing to the development and enormous diversity of the Internet, different average packet loss rates are reported in different studies: Yajnik *et al.* [20] show that the packet loss rates varies between 1.38% (IPv4 Unicast) and 11.03% (IPv4 Multicast) based on the study of speech data transmission in 1999. Wang *et al.* [4] report in 2005 average packet loss rates of 3.09 and 0.76% of the IPv6 and the IPv4 connections, respectively. Our results are much smaller than theirs. The main reason lies in the experimental setup: we send probe packets (100 bytes UDP packets) only in the TTM infrastructure, which is solely in the core network.

On the basis of the above measurements, we conclude that even though new testboxes were added into the measurement testbed in 2 years, the IPv6 and IPv4 delay and loss did not change significantly, and IPv4 outperformed IPv6 in terms of delay and loss during the whole period.

However, given the large percentage of tunneled IPv6 paths in the current Internet (see Figure 3), it is not sufficient to study only aggregated measurements. Therefore, the following three questions are still unanswered by the previous studies:

(1) How much different do IPv6 tunnels paths behave from their IPv4 counterparts over time?
(2) How much different do native IPv6 paths behave from their IPv4 counterparts over time?
(3) Can that difference explain the reason why IPv6 is outperformed?

4.1.1. Delay performance of native IPv6 paths. Figure 5 shows the 2.5 percentile (a), median (b), and 97.5 percentile (c) delay performance of only IPv6 native paths *versus* their corresponding IPv4 counterparts. The results in Figures 5(a) and (b) suggest that the 2.5 percentile and median delay of those native IPv6 paths are only slightly worse (23 and 27% higher) than their corresponding IPv4 paths, which means that native IPv6 can achieve a relatively good performance.

However, the results in Figure 5(c) indicate that the 97.5 percentile delays of those native IPv6 paths are much worse (157% higher) than their corresponding IPv4 paths. One reason is that IPv6 packets have a lower priority than IPv4 and thus have a longer delay in the processing time during peak hours (more details are shown on the RIPE TTM Web site). These results indicate that in most cases, IPv6 native paths have a similar performance as IPv4 paths.

Figure 5 also shows that before August 30, 2005, both IPv4 and IPv6 2.5 percentile and median delays have slightly decreased over time, but increased with about 120 and 73% after that date. This difference was mainly caused by a change in about 30% long-distance IPv6 tunnel paths that were switched to native ones (see jump in Figure 5(a) and (b)) at that date. This increased the number of long native paths and also added extra average delay of these native paths. These observations also hold for the IPv4 counterparts since more long-distance paths were suddenly included.

DOI: 10.1002/dac

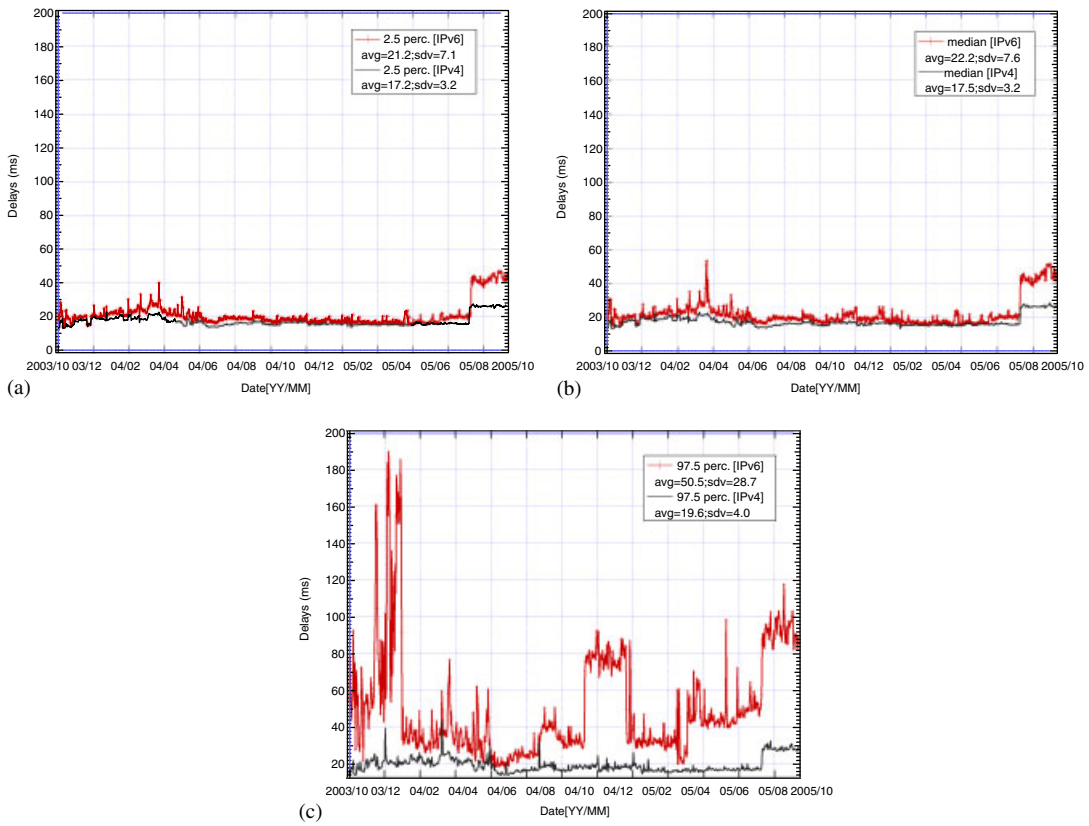

Figure 5. Native IPv6 paths and their IPv4 counterparts over time: (a) 2.5 percentile; (b) median; and (c) 97.5 percentile.

4.1.2. IPv6 tunnels delay performance. Figure 6 shows that IPv6 tunnel paths have a larger average delay than their IPv4 counterparts over time. When considering tunneling transition mechanisms, IPv6 traffic is expected to show a degraded performance, since IPv6 packets are encapsulated in IPv4 packets. Our results do show the evidence of this. On average, IPv6 has about 83% higher 2.5 percentile delay than their IPv4 counterparts (Figure 6(a)), while about 87 and 165% for the cases of median and 97.5 percentile delays (Figure 6(b) and (c)), respectively. Compared with their corresponding IPv4 counterparts, IPv6 native paths perform a little worse, while IPv6 tunnel paths perform much worse. The difference here shows that tunnels degrade the network delay performance and this explains why IPv6 is outperformed by IPv4.

Nevertheless, the delay performance difference compared with IPv4 is larger for IPv6 tunnel paths than for IPv6 native paths. This can partly be explained by the fact that almost all IPv6 tunnels are software based that incur more delay than hardware-based implementations. Another reason can be less optimal routing of IPv6 paths and their tunnels compared with IPv4.

The worse delay performance of IPv6 tunneling is mainly due to the poor management of IPv6-in-IPv4 tunnels; misconfigurations of tunnels were not uncommon. These tunnels have in common that after configuration, they behave like point-to-point links and will only appear as one

DOI: 10.1002/dac

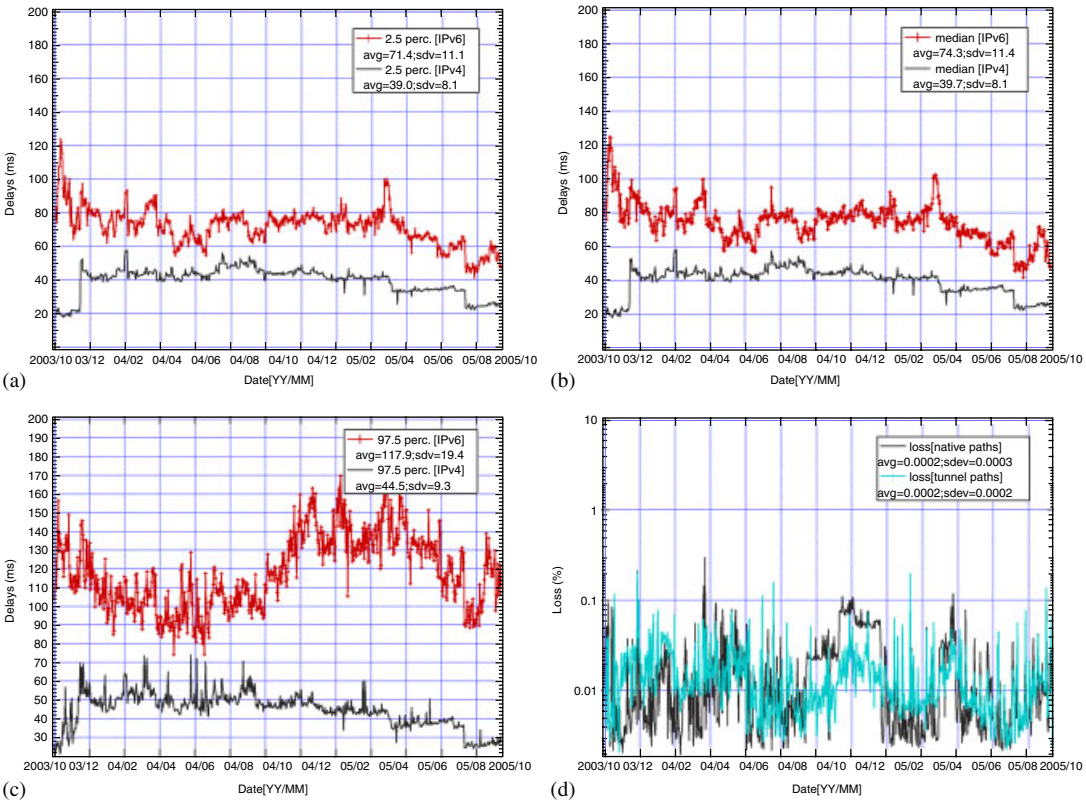

Figure 6. IPv6 tunnel paths and their IPv4 counterparts performance over time: (a) 2.5 percentile; (b) median; (c) 97.5 percentile; and (d) loss performance.

hop in traceroute measurements. We will return to this point in Section 4.4 when we investigate single paths.

Approximately, both IPv6 tunnels and their IPv4 counterparts in Figure 6(a) and (b) have slightly decreased over time. However, the former shows a relatively bigger variation.

4.1.3. IPv6 loss performance. Figure 6(d) shows the loss performance of native IPv6 paths and tunnel paths over time. We observe that the loss in both native IPv6 paths and IPv6 tunnel paths are roughly equal and very small (0.02%) and do not change much over time. Since tunneling degrades the delay performance, one may infer that tunneling will also result in higher packet loss. Our measurement results, however, do not imply any strong correlation between tunneling/native and packet loss rate.

4.2. Delay trends of two source–destination paths over 2 years

Figure 7 shows the one-way delay trend of a typical tunnel path and that of its IPv4 counterpart (from a testbox located in Amsterdam to a testbox located in Dublin) over 2 years. Using measurements during 1 h, we calculated the 2.5 percentile, median, and 97.5 percentile for the path. Each point

DOI: 10.1002/dac

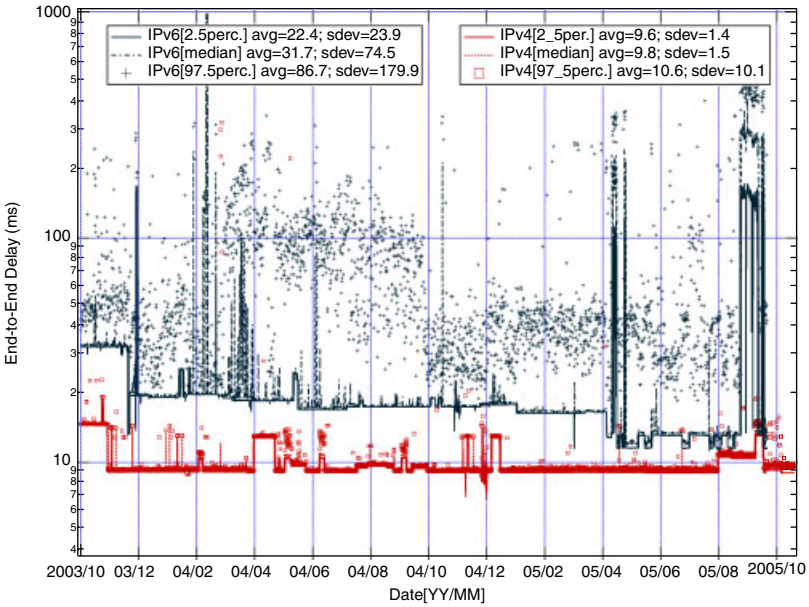

Figure 7. Delay trends of a tunnel pair over 2 years.

in the graph is then the average of this over a 6-h interval, which gives four samplings per day for each value. Figure 7 shows that IPv6 packets via a tunnel have a larger delay (the IPv6 median delay is 31.72 ms, while that of IPv4 delay is 9.76 ms) as well as a much larger variance. Figure 7 also indicates that after 2 years of evolution, the 2.5 percentile and median IPv6 delays of this path are approaching that of the IPv4 delays, which suggests that the IPv6 performance was improving. However, the 97.5 percentile IPv6 delays are much larger than the IPv4 counterparts. Another interesting observation is that there are some high peaks in the IPv6 delay and those peaks suggest very poor delay performance. One such peak lasted from October 4, 2005 to October 21, 2005. An investigation of the traceroutes from those days reveals serious routing and/or link failures (most likely due to misconfigurations or chronic instability in routing tables).

Figure 8 shows a delay trace of a typical native IPv6 path and that of its corresponding IPv4 counterpart (both testboxes are located in Amsterdam) over 2 years. The key observation is that the native IPv6 path behaves similarly as that of an IPv6 tunnel path: the 2.5 percentile and median IPv6 delays of this path are approaching that of IPv4 delays, but it still shows a much higher 97.5 percentile delay. This result suggests that in the worst delay cases (97.5 percentile), both IPv6 tunnel and native paths perform worse than IPv4, and the high peaks (large delay) in the IPv6 performance were not all caused by tunneling.

4.3. Delay and loss performance of all TTM paths over a day

To know how all IPv6 paths perform on a random day, this section shows the delay and loss performance of the IPv6 paths and their IPv4 counterparts on September 19, 2004. Specifically, for each source–destination path, we collected the delay and loss performance over that day. Then

DOI: 10.1002/dac

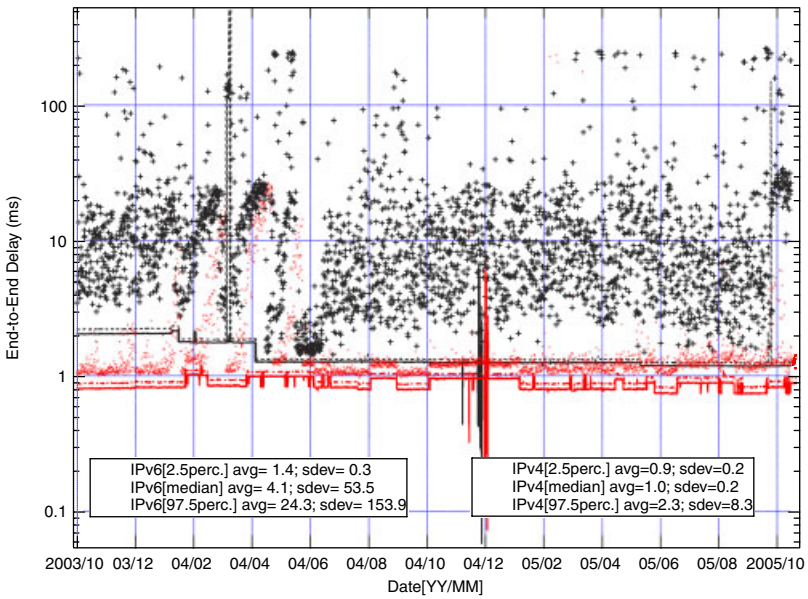

Figure 8. Delay trend of a native IPv6 path over 2 years.

we studied the different behaviors of IPv6 native paths and IPv6 tunnel paths by comparing their delay performance with their IPv4 counterparts on a path-by-path basis.

During that day, there were 359 IPv6 native paths and 181 IPv6 tunnel paths. The native paths were all within the same continent (e.g. Europe, U.S.A.), while about 28% of the tunnel paths were inter-continental (e.g. Europe–Japan). We observe the following performance differences between native IPv6 paths and IPv6 paths with tunnels.

4.3.1. Native paths delay performance

- Figure 9(a) shows the plots of 2.5 percentile and median delay of the IPv6 native paths over that day. For clarity, we have sorted the paths based on the 2.5 percentile values. Figure 9(a) indicates that IPv6 paths with a relatively small end-to-end delay (delay $\leqslant$26.5 ms, shown in part A) have comparable 2.5 percentile and median delay as those of their IPv4 counterparts. While those IPv6 paths (in part B) with a relative large end-to-end delay (delay $\geqslant$26.5 ms) suffer a larger delay than their IPv4 counterparts, for 90% of the IPv6 paths, the 2.5 percentile delay is less than 36.1 ms, and the maximum 2.5 percentile delay of the paths is 59.7 ms, while it is 29.2 and 44.6 ms for the IPv4 2.5 percentile delays, respectively. For 90% of the IPv6 paths, the median delay is less than 37.1 ms, and the maximum median delay of the paths is 60.9 ms, while it is 30.2 and 44.7 ms for the IPv4 counterparts, respectively.
- Figure 9(b) shows that in most cases, IPv6 paths have a larger 97.5 percentile delay. Some IPv6 paths even have a much higher delay. When looking closer at these paths, we found that all those paths contain a site located in Hungary. It was also found that this site-specific effect later changed completely. For 90% of the IPv6 paths, the 97.5 percentile delay is less

DOI: 10.1002/dac

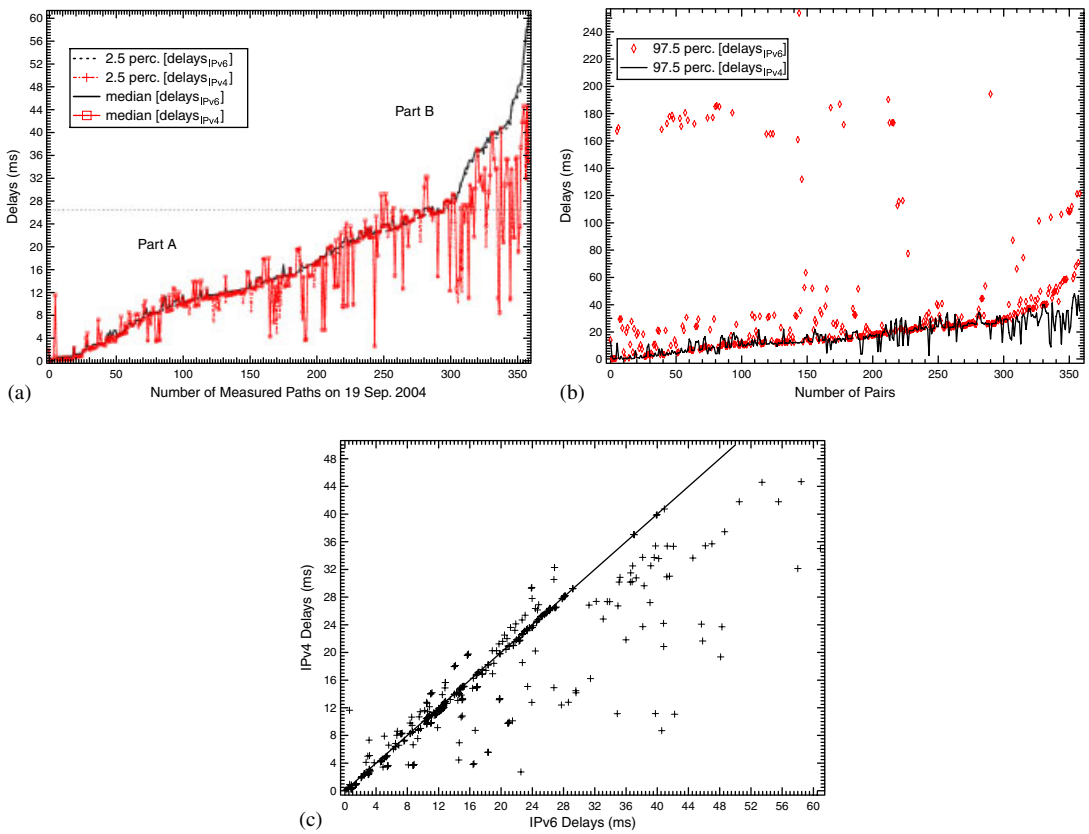

Figure 9. Delay performance of (a) the (sorted) 2.5 percentile and median; (b) 97.5 percentile delays; and (c) scatter plot.

than 94.3 ms, and the maximum 97.5 percentile delay of the paths is 115.6 ms, while they are much less (31.1 and 48 ms, respectively) for the IPv4 counterparts.

- Figure 9(c) shows the scatter plots of the native IPv6 median delays *versus* the IPv4 median delays, where the IPv6 delay is on the X-axis and the IPv4 delay on the Y-axis. Each data point corresponds to a path. We also plot the diagonal line. For points below this line, IPv4 outperforms IPv6. The key observation is that most IPv6 paths perform similar to their IPv4 counterparts. This result complies with that of Figure 9(a).

4.3.2. Tunnel paths delay performance

- Figure 10(a) shows 2.5 percentile and median delays of the IPv6 tunnel paths and the corresponding IPv4 paths. For clarity, we have first classified the paths based on the IPv6 tunnel types (which were deduced from the path MTUs) and then sorted based on the 2.5 percentile values. The results suggest that IPv6-in-IPv4 tunnel paths have a larger delay than their IPv4 counterparts, while packets using other tunnel types only show slightly worse behavior than their IPv4 counterparts. For 90% of the IPv6 tunnel paths, the 2.5 percentile delay is less than

DOI: 10.1002/dac

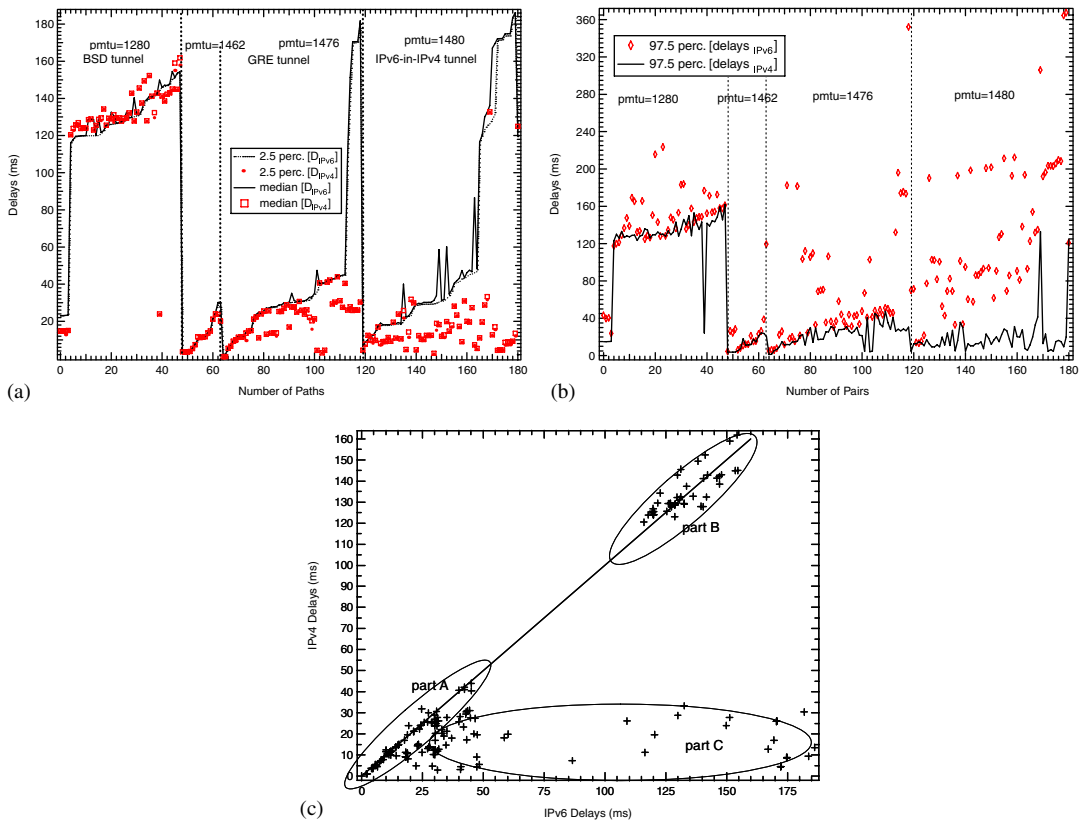

Figure 10. (a) The 2.5 percentile and median; (b) 97.5 percentile delays in the IPv6 tunnel pairs and IPv4 counterparts; and (c) scatter plot of the median delays in IPv6 tunnel paths and IPv4 counterparts.

146.8 ms, and the maximum 2.5 percentile delay of the paths is 184.9 ms, while it is 132.3 and 157.9 ms for the IPv4 2.5 percentile delays, respectively. For 90% of the IPv6 tunnel paths, the median delay is less than 151.4 ms, and the maximum median delay of the paths is 186.2 ms, while it is 132.6 and 158.2 ms for the IPv4 counterparts, respectively.

- Figure 10(b) shows 97.5 percentile delay of the IPv6 tunnel paths and their corresponding IPv4 values. The results indicate that all the IPv6 tunnel paths have larger 97.5 percentile delays than their corresponding IPv4 paths. The worst performance is found on paths with IPv6-in-IPv4 tunnels. In general, for 90% of the IPv6 tunnel paths, the 97.5 percentile delay is less than 196.3 ms, and the maximum 97.5 percentile delay of the paths is 367.4 ms, while it is 135.1 and 162.1 ms for their IPv4 counterparts, respectively.
- Figure 10(c) shows the scatter plot of the IPv6 tunnel path median delays *versus* the IPv4 median delays. The data points are approximately classified into three groups by R, the ratio of the IPv6 over the IPv4 one-way delay: group A for the paths with small R ($R \leqslant 1.25$) within the same continent; group B for the paths with equal R ($R \leqslant 1.25$) between different continents; and group C for the paths with large R ($R > 1.25$). Our results show that both the IPv6 native paths and tunnel paths have about half of the paths with a larger median delay

DOI: 10.1002/dac

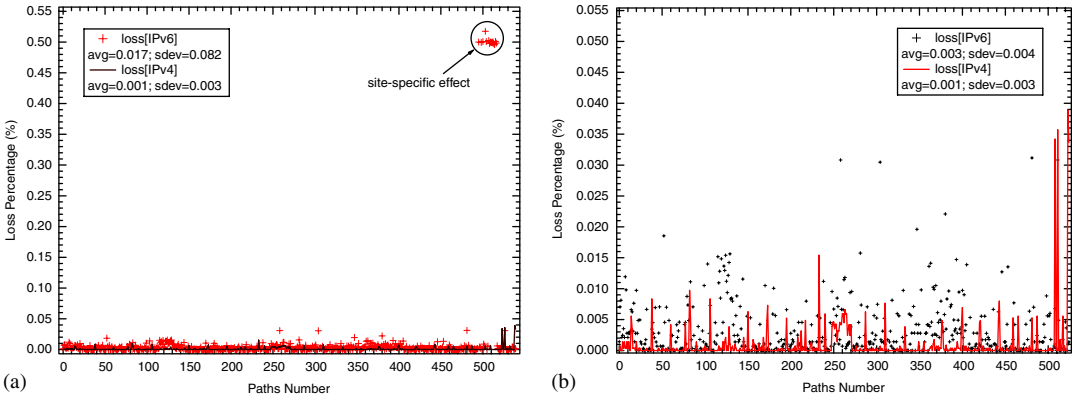

Figure 11. Loss percentages of all the paths over a day: (a) all measurements and (b) no site-specific effect.

than those of their IPv4 counterparts. For example, for the native paths, about 53% of the paths are of group *A*, and 47% of group *C*; while for the IPv6 tunnel paths, 1.6% of paths are of group *A*, 50.4% of group *B*, and 48% of group *C*.

In short, we found that, in comparison with their IPv4 counterparts, IPv6 paths with tunnels perform worse than the IPv6 native paths. The worst performance is found with IPv6 paths that contain IPv6-in-IPv4 tunnels.

4.3.3. IPv6 paths loss performance. Figure 11 shows the measured loss results of IPv6 traffic and IPv4 traffic over one day (September 19, 2004). The results indicate that in most cases, IPv6 packets suffered a little higher loss than those IPv4 counterparts. For 90% of the IPv6 native paths, the packet loss is less than 0.009%, and the maximum loss of native paths is 0.51%, while it is 0.004% and 0.039% for the IPv4 counterparts, respectively. For 90% of the IPv6 tunnel paths, the packet loss is less than 0.35%, and the maximum loss of the paths is 0.50%, while it is 0.002 and 0.008% for the IPv4 counterparts, respectively. Some IPv6 paths have a high packet loss, and all those paths contain a site located in Hungary. However, some other paths do not experience such high loss rate. We suspect that the difference in loss may be caused by routing problems of the nodes on that site. Moreover, not all routers put into the software path the packets that they cannot handle in hardware. In those cases, the packets are simply dropped [13]. Similar site-specific behaviors have also been found by Wang *et al.* [4].

4.4. Delay of single source–destination paths over a day

This section studies the delay performance of a typical IPv6 tunnel path and a typical IPv6 native path over one day. First, Figure 12(a) shows the delay performance of a typical IPv6 tunnel path and its IPv4 counterpart (from a testbox located in Amsterdam to a testbox located in Dublin) in September 2004, where delay is on the *Y*-axis and the packet sequence number on the *X*-axis. The tunnel discovery tool detected that IPv6 packets were transferred in an IPv6-in-IPv4 tunnel (MTU 1480) from the source to the destination. Figure 12(a) illustrates that IPv6 packets on this path had a larger delay and delay variation.

DOI: 10.1002/dac

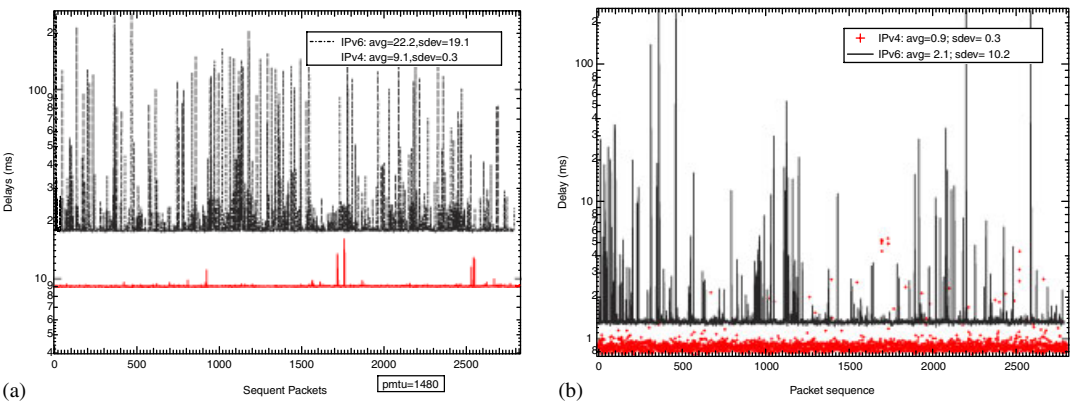

Figure 12. Tunnel and native IPv6 path delays performance over a day (September 19, 2004): (a) tunnel path and (b) native path.

IPv6			
hops	IP address	Host name	AS num(s)
1	2001:610:240:2::1	gw6.e01.ripe.net	3333
2	2001:7f8:1::a501:2702:1	e0-0-0.6b2.AMS7.Alter.net	1200/5417
3	2001:600:8:5::2	intermax-ipv6.customer.alter.net	12702
4	2001:600:4:8e5::2	tu.6r001.cwt.esat.net	12702
5	2001:7c8:2:8::4	fe0-0.6rt501.cwt.esat.net	2110
6	2001:7c8:2:9::3	fe0-0.6rt515.cwt.esat.net	2110
7	2001:7c8:a1:1::c178:c976	tt25.ripe.net	2110
IPv4			
1	193.0.0.238	g0013.nikrtr.ripe.net	3333
2	195.69.144.108	ixp1.nl-ams2.eu.bt.net	1200/5417
3	166.49.163.189	t2a1-ge8-0.nl-ams2.eu.bt.net	5400
4	166.49.153.130	166-49-153-130.eu.bt.net	5400
5	193.95.129.19	vlan3.rt002.cwt.esat-x.com	2110
6	193.95.130.154	vlan53.rt501.cwt.esat.net	2110
7	193.95.130.242	vlan515.rt515.cwt.esat.net	2110
8	193.120.201.118	tt25.ripe.net	2110

Figure 13. Traceroute information between a source–destination path in both IPv6 and IPv4 protocols.

Second, Figure 12(b) shows the delay performance of a typical IPv6 native path and its IPv4 counterpart (both testboxes are located in Amsterdam) over the same day. The corresponding traceroutes show that IPv4 and IPv6 have the same AS path. Once again, we observe that similar high peaks appear in the IPv6 performance. In general, it is expected that IPv6 and IPv4 will compete for the computing resources. The higher IPv6 average delay might confirm that IPv6 packets do have a lower priority than IPv4 ones and suffer a longer processing time.

DOI: 10.1002/dac

Generally, the current IPv6 packets using 6in4 tunnels experience higher delays than IPv4 packets, because tunnels produce additional overhead in the packet size and processing time on the gateway. This partially does explain the large fluctuation of the delay in the graph. This fluctuation can also have many other causes, including queue length variations and variations in the processing time needed to handle the packets in the routers. While IPv4 routing is mostly done in hardware, IPv6 packets more often suffer from a larger variation in the processing time due software routing. It is also possible that IPv6 is given lower priority than IPv4 by ISPs as IPv4 is still much more important. Further, IPv6 and IPv4 packets may take different paths between the source and the destination, including IPv6 paths that do not have tunnels. This might be due to different peering agreements for IPv4 and IPv6. The traceroutes in Figure 13 show both IP level and AS level routing for a native IPv6 path and its corresponding IPv4 path: IPv6 behaves different from IPv4: At hops 3 and 4, IPv6 reaches a different AS than IPv4. At hop 5, their traceroutes reached the same AS again. We found that it is common for IPv6 paths to go through different ASes than IPv4. Some observations from the RIPE TTM Web site do tell that some IPv6 paths followed a less optimal route compared with IPv4.

5. CONCLUSION

A detailed measurement study of the delay and loss evolution in IPv6 networks based on the RIPE infrastructure has been presented.

Concerning the delays over one day, native IPv6 paths have small 2.5 percentile and median end-to-end delay, and comparable delay to their IPv4 counterparts. IPv6 tunnel paths have relatively large 2.5 percentile and median end-to-end delay, and about half of the paths have significantly more delay compared with their IPv4 counterparts. The worst performance came from IPv6-in-IPv4 tunnels. For the 97.5 percentile delay, IPv4 by far outperforms IPv6 for both native and tunnel IPv6 paths. For the delay we can summarize as follows:

$$\text{IPv4 delay} \leqslant \text{IPv6 native delay} \ll \text{IPv6 tunnels delay}$$

As demonstrated earlier, the IPv4 delay performance shows a slight improvement over the 2 years. No such trend is seen for either IPv6 native or IPv6 tunnel paths. Instead, we see big variations in the delay performance.

Concerning loss, IPv4 again performs best. However, when comparing IPv6 native and tunnel paths, no certain difference can be demonstrated. For packet loss, we summarize as follows:

$$\text{IPv4 loss} \leqslant \text{IPv6 native loss} \approx \text{IPv6 tunnels loss}$$

It is always difficult to explain all these behaviors without a detailed view on all involved networks. However, some causes of the longer delay for IPv6 can be explained as follows:

(1) The lack of routers with IPv6 hardware-optimized implementation. Hardware-based IPv6 tunneling implementations are virtually non-existing today. However, some software-based IPv6 routers can perform quite well [13].
(2) Other viable causes for the longer delay for IPv6 are:

- Less optimal paths are used for IPv6, especially when tunnels are used.
- There are different or fewer peering agreements for IPv6 between ISPs.

DOI: 10.1002/dac

- Network management and monitoring of IPv6 networks are not as advanced as for IPv4 networks. ISPs do not invest equally on IPv6 network management as they do on IPv4. Also, the experience with IPv4 is larger (e.g. more traffic engineering).
- IPv6 has a lower priority in the routers. IPv6 is still seen as experimental and should never degrade the performance of the more important IPv4 traffic.

It is expected that IPv4 and IPv6 will coexist for a while, and that IPv6 tunnels play a key role in the transient phase. Software-based routers and tunnels provide a quick way to deploy IPv6. The drawback, on the other hand, is that IPv6 tunneling degrades the traffic performance, mainly in terms of larger delay. Clearly, our results suggest that for a better IPv6 quality, we should only use native IPv6 and hardware-based routers everywhere. Nevertheless, the performance quality of software-based routers and tunnels is still acceptable for a successful transition phase.

ACKNOWLEDGEMENTS

This work was performed while X. Zhou was with the Delft University of Technology, The Netherlands. His work was supported by the NWO SAID project.

REFERENCES

1. Deering S, Hinden R. *RFC 2460*: *Internet Protocol*, *Version* 6 (*IPv*6) *Specification*, IETF, December 1998.
2. Adam Y, Fillinger B, Astic I, Lahmadi A, Brigant P. Deployment and test of IPv6 services in the VTHD network. *IEEE Communications Magazine* 2004; **42**(1):98–104. DOI: 10.1109/MCOM.2004.1262168.
3. Cho K, Luckie M, Huffaker B. Identifying IPv6 network problems in the dual-stack world. *Proceedings of the ACM SIGCOMM Workshop on Network Troubleshooting*, Portland, OR, U.S.A., September 2004; 283–288. DOI: 10.1145/1016687.1016697.
4. Wang Y, Ye S, Li X. Understanding current IPv6 performance: a measurement study. *Proceedings of 10th IEEE Symposium on Computers and Communications* (*ISCC'05*), Cartagena, Murcia, Spain, June 2005; 71–76. DOI: 10.1109/ISCC.2005.151.
5. Zhou X, Van Mieghem P. Hopcound sand E2E delay: IPv6 Versus IPv4. *Proceedings of the Workshop on Passive and Active Measurement* (*PAM'05*), Lecture Notes in Computer Science, vol. 3431/2005, March 2005; 345–348. DOI: 10.1007/b135479.
6. Georgatos F, Gruber F, Karrenberg D, Santcroos M, Susanj A, Uijterwaal H, Wilhem R. Providing active measurements as a regular service for ISP's. *Proceedings of the Workshop on Passive and Active Measurement* (*PAM'01*), Amsterdam, The Netherlands, April 2001.
7. Durand A. Managing 100+ million IP addresses. *The North American Network Operators' Group* (*Nanog 37*) *Meeting*, San Jose, CA, U.S.A., June 2006.
8. Carpenter B, Moore K. *RFC 3056*: *Connection of IPv6 Domains via IPv4 Clouds*, IETF, February 2001.
9. Carpenter B, Jung C. *RFC 2529*: *Transmission of IPv6 over IPv4 Domains without Explicit Tunnels*, IETF, March 1999.
10. Conta A, Deering S. *RFC 2473*: *Generic Packet Tunneling in IPv6 Specification*, IETF, December 1998.
11. Hanks S, Li T, Traina P. *RFC 1701*: *Generic Routing Encapsulation* (*GRE*), IETF, October 1994.
12. Tsirtsis G, Srisuresh P. *RFC 2766*: *Network Address Translation—Protocol Translation* (*NAT-PT*), IETF, February 2000.
13. Popoviciu C, Levy-Abegnoli E, Grossetete P. *Deploying IPv6 Networks*. Cisco Press: Indiana, U.S.A., February 2006, Print ISBN-10: 1-58-705210-5.
14. Srivastava V, Wargo C, Lai S. Aviation application over IPv6: performance issues. *Proceedings of the IEEE Aerospace Conference*, vol. 3(6–13), Springfield, VA, U.S.A., March 2004; 1661–1670. DOI: 10.1109/AERO.2004.1367941.
15. Van Mieghem P. *Data Communications Networking*. Techne Press: Amsterdam, 2006.

16. Almes G, Kalidindi S, Zekauskas M. *RFC 2679*: *A One-way Delay Metric for IPPM*, IETF, September 1999.
17. Almes G, Kalidindi S, Zekauskas M. *RFC 2680*: *A One-way Packet Loss Metric for IPPM*, IETF, September 1999.
18. Baccelli F, Machiraju S, Veitch D, Bolot J. The role of PASTA in network measurement. *ACM SIGCOMM Computer Communication Review*, vol. 36(4), Pisa, Italy, October 2006; 231–242. DOI: 10.1145/1159913.1159940.
19. Colitti L, Di Battista G, Patrignani M. Discovering IPv6-in-IPv4 tunnels in the Internet. *Proceedings of the IEEE/IFIP Network Operations and Management Symposium* (*NOMS'04*), vol. 1, Seoul, Korea, April 2004; 613–626.
20. Yajnik M, Moon S, Kurose J, Townsley D. Measurement and modeling of the temporal dependence in packet loss. *Proceedings of Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies* (*INFOCOM'99*), vol. 1, New York, U.S.A., March 1999; 345–352. DOI: 10.1109/INFCOM.1999.749301.

AUTHORS' BIOGRAPHIES

Xiaoming Zhou received the MSc degree in 2002 and the PhD degree in Electrical Engineering in 2006 from Delft University of Technology in the Netherlands. He received a scholarship to analyze Internet performance in 2002. In January 2003, he began his PhD work at Network Architectures and Services (NAS) group at TU Delft. The main theme of the research was Internet measurement and analysis at the network and application layer. Currently, he is a research scientist in Philips Research, Eindhoven, The Netherlands. His main research interests are the convergence of different networks, and exploring the new product concepts for Philips. E-mail: xiaoming.zhou@philips.com.

Martin Jacobsson graduated in Computer Science from University of Linköping, Sweden in 2002. In 2003, he joined the Wireless and Mobile Communications group led by professor Niemegeers at Delft University of Technology as a doctoral candidate. He has participated in several Dutch and European research projects. His PhD research includes *ad hoc* and self-organization wireless networking techniques in combination with future infrastructure-based networks for personal networks. He has also been involved in developing performance measurement tools for IP networks for TeliaSonera. E-mail: m.jacobsson@ewi.tudelft.nl.

Henk Uijterwaal holds a PhD in Physics and worked at High Energy Physics institutes in the Netherlands, Germany, U.S.A., and U.K. He joined the RIPE NCC in 1997 and is currently manager of New Projects group. He was responsible for the development of the Test Traffic Measurements Service, from the initial concept until it became a production service. He is also one of the chairs of the IETF IP Performance Metrics (IPPM) Working group. His main research interests are network performance, routing, and security. E-mail: henk@ripe.net.

Piet F. A. Van Mieghem is professor at the Delft University of Technology with a chair in telecommunication networks and chairman of the basic unit Network Architectures and Services (NAS). His main research interests lie in new Internet-like architectures for future, broadband, and QoS-aware networks and in the modeling and performance analysis of network behavior and complex infrastructures. Professor Van Mieghem received a Master's and PhD in Electrical Engineering from the K. U. Leuven (Belgium) in 1987 and 1991, respectively. Before joining Delft, he worked at the Interuniversity Micro Electronic Center (IMEC) from 1987 to 1991. During 1993–1998, he was a member of the Alcatel Corporate Research Center in Antwerp, where he was engaged in performance analysis of ATM systems and in network architectural concepts of both ATM networks (PNNI) and the Internet.

He was a visiting scientist at MIT (department of Electrical Engineering, 1992–1993) and, in 2005, he was a visiting professor at ULCA (department of Electrical Engineering). He was member of the editorial board of the journal Computer Networks from 2005 to 2006.

He is the author of two books: Performance Analysis of Communications Networks and Systems, Cambridge University Press (2006) and Data Communications Networking, Techne Press, Amsterdam (2006). E-mail: P.F.A.VanMieghem@tudelft.nl.

DOI: 10.1002/dac