

Research Article

A Case Study of IPv6 Network Performance: Packet Delay, Loss, and Reordering

Fuliang Li,¹ Xingwei Wang,¹ Tian Pan,² and Jiahai Yang³

¹*School of Computer Science and Engineering, Northeastern University, Shenyang, China*

²*Department of ICE, Beijing University of Posts and Telecommunications, Beijing, China*

³*Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China*

Correspondence should be addressed to Xingwei Wang; wangxw@mail.neu.edu.cn

Received 22 June 2017; Accepted 12 September 2017; Published 17 October 2017

Academic Editor: Filippo Cacace

Copyright © 2017 Fuliang Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Internet Protocol (IP) is used to identify and locate computers on the Internet. Currently, IPv4 still routes most Internet traffic. However, with the exhausting of IPv4 addresses, the transition to IPv6 is imminent, because, as the successor of IPv4, IPv6 can provide a larger available address space. Existing studies have addressed the notion that IPv6-centric next generation networks are widely deployed and applied. In order to gain a deep understanding of IPv6, this paper revisits several critical IPv6 performance metrics. Our extensive measurement shows that packet delay and loss rate of IPv6 are similar to IPv4 when the AS-level paths are roughly the same. Specifically, when the link utilization exceeds a threshold, for example, 0.83 in our study, variation of packet delay presents a similar pattern with the variation of link utilization. If packet delay of a path is large, packet-loss rate of that path is more likely to fluctuate. In addition, we conduct a first-ever analysis of packet reordering in IPv6 world. Few IPv6 probe packets are out-of-order and the reordering rate is 2.3×10^{-6} , which is much lower than that of 0.79% in IPv4 world. Our analysis consolidates an experimental basis for operators and researchers of IPv6 networks.

1. Introduction

As a communication protocol, IP provides an identification and location system for computers on the Internet. Internet Protocol version 4 (IPv4) [1] has been used successfully for the past years. As the successor of IPv4, Internet Protocol version 6 (IPv6) [2] was developed by the Internet Engineering Task Force (IETF) [3] since 1998. Currently, IPv4 still routes most Internet traffic [4]. However, due to the predictable exhaustion of IPv4 addresses [5], the transition to IPv6 is imperative, because, compared with IPv4, IPv6 can provide a larger available address space.

To try and encourage the adoption of IPv6, more than 400 organizations and institutions participated in the World IPv6 Day on 8th June 2011 [6]. To further promote the deployment and development of IPv6, World IPv6 Launch was held on 6th of June 2012 [7]. Google has announced that over 10% of users access Google through IPv6 [8]. CAIDA has shown that, from January 2014 to January 2015, the number

of IPv6 ASes increased by 23% and the number of links between them increased by 29% [9]. Existing studies have addressed that IPv6-centric next generation networks are widely deployed and applied [10–12]. With the development of IPv6, applications such as video conference, IPTV, VoIP, and online games will become more and more prevalent in IPv6 networks, so keeping network availability as well as good performance is crucial. Investigating the performance of IPv6 network is the basis of guaranteeing the network with a good performance. Therefore, many studies have been performed to analyze IPv6 network performance, as well as figure out the differences between IPv4 and IPv6 [13–17]. However, our understanding of IPv6 network cannot catch up with the speed of its development. A revisiting of current IPv6 network performance is needed.

In this paper, we revisit IPv6 performance and make a comparison with IPv4 from several metrics, including network reachability, packet delay, packet loss, and packet reordering. A preliminary version of this paper has been

published in ICNC 2016 [18]. Our contributions and main findings are summarized as follows.

(1) We design a performance probing method for IPv6 network based on *OneProbe* [19], a new TCP tool for reliable and metric-rich path monitoring. Using this method, we can persistently observe and analyze performance of IPv6 network from different angles. Analysis results can help us fully understand IPv6 networks and provide an essential basis for IPv6 network operators and researchers.

(2) We conduct a comparison study on IPv6 packet delay (round trip time, i.e., RTT). The average RTT of IPv6 is 247.71 milliseconds from our probing host to the remote servers, while the average RTT of IPv4 is 244.48 milliseconds, which is similar to the value of IPv6. However, in most cases, IPv6 RTT is higher than IPv4 RTT in Asia Pacific region. We explore the reasons for the difference by dividing the AS-level paths into three parts, that is, *domestic-as*, *middle-as*, and *target-domestic-as*. We find that the connectivity between *middle-as* and *target-domestic-as* and the length of *middle-as* are the dominant factors that cause the difference in RTT between IPv4 and IPv6. In addition, in our measurement, we find that when the link utilization exceeds 0.83, variation of packet delay presents a similar pattern with the variation of link utilization.

(3) We make a comparison analysis between IPv4 and IPv6 in packet-loss rate. The average packet-loss rate of IPv6 is 0.25%, while the average packet-loss rate of IPv4 is 0.33%, which is a bit higher than that of IPv6. We also investigate the relationship between packet delay and packet-loss rate. Packet-loss rate of a path is more likely to fluctuate when the packet delay of that path becomes large.

(4) We conduct a first-ever analysis of packet reordering in IPv6 world. We find that the average packet reordering rate of IPv6 is $2.3 * 10^{-6}$, while the average packet reordering rate of IPv4 is 0.79%. The reason for the difference is that fragmentation in IPv6 is not encouraged and only the hosts can conduct packet fragmentation in IPv6 network, which could reduce the possibility of packet reordering.

The remainder of the paper is organized as follows. The related work is presented in Section 2. We describe the methodologies used to measure IPv6 network performance in Section 3. Section 4 presents our analysis results, including reachability, packet delay, packet-loss rate, and packet reordering. Our conclusions are presented in Section 5.

2. Related Work

Related work is summarized from two aspects. We begin with discussion on packet delay and packet-loss analysis in IPv6 world. Then we show the studies that have been conducted to characterize packet reordering in IPv4 world.

Wang et al. [13] compared IPv6 and IPv4 performance from the perspective of end users and found that it still has headroom to improve IPv6 performance, including increasing its connectivity and reducing its packet-loss rate. Muniyappa [14] carried out a simulation-based study comparing the use of IPv4 versus IPv6 on simple campus networks and found that performances of IPv4 and IPv6 are almost the same and the difference is negligible. Shiwani et al.

[15] measured throughput, delay, and jitter of IPv6 on a test-bed implementation. Zhou et al. [16] found that IPv6 paths have a higher delay and loss than their IPv4 counterparts, mainly due to its poor performance in IPv6-in-IPv4 tunnels. Nikkhah et al. [17] and Dhamdhare et al. [20] found that IPv6 delay is similar to IPv4 performance when the AS-level paths are identical. Czyn et al. [21] concluded that IPv6 RTT is getting better according to a long-term observation. Livadariu et al. [22] revealed that IPv6 routing system is less stable than IPv4, and IPv6 performance is comparable to that over IPv4. Bajpai and Schönwälder [23] found that TCP connection establishment times of IPv6 are relatively higher than that of IPv4 in the cases when CDN caches are present for IPv4, but are largely absent for IPv6. Lin et al. [24] designed and implemented a high-performance IPv6 address lookup engine in GPU-accelerated software routers. In this paper, we revisit several critical performance metrics of IPv6 network, including reachability, delay, and packet-loss rate. In addition, we also explore the reasons for the differences between IPv4 and IPv6. Furthermore, we analyze the relationship between delay and link utilization, as well as capture the variation pattern and find a fitting model for the delay when it fluctuates greatly.

Packet reordering in IPv6 world is less understood. However, packet reordering has great influence on transmission efficiency. Stevens [25] found that when the TCP receiver gets out-of-order packets, it sends duplicate ACKs to trigger fast retransmission algorithm at the sender, which may cause unnecessary retransmission in transport layer. Laor and Gendel [26] found that packet reordering results in limited speed of packet transmission as well as throughput degradation. Wang et al. [27] proposed a single-point reorder-judging algorithm to measure packet reordering and found a threshold to help distinguish reordering and loss on some heavily reordering paths. Zhou and Van Mieghem [28] analyzed the end-to-end packet reordering by tracing UDP packets between 12 test-boxes in RIPENCC and found that packet reordering has a significant influence on the UDP performance but does not have a significant impact on UDP delay. Bohacek et al. [29] mentioned that packet reordering is generally ascribed to transient conditions, pathological behavior, and erroneous implementation. Bennett et al. [30] refuted the widely held belief that packet reordering in the Internet is an uncommon behavior caused by incorrect or malfunctioning network components. Przybylski et al. [31] discussed the factors affecting ordered packet delivery and revealed that the main protocols of modern gigabit networks cannot cope well with reordering.

In addition, packet reordering also greatly affects network intrusion detection and prevention systems that need to maintain the state of each flow [32–34]. Out-of-order packets must be buffered until all the packets are reassembled in the correct order. However, the buffer may overflow when the packet reordering occurs frequently. The attackers may also craft ambiguous traffic to breach the defense of such systems. For example, an attacker may send TCP segments with different payloads for the same sequence number space. In this paper, we conduct a first-ever study on IPv6 packet reordering. This can not only let us know exactly whether

TABLE 1: Parameters of OneProbe.

Parameters	z (bytes)	M (bytes)	m (bytes)	e (seconds)	r (pkt/s)
Values	65535	750	750	300	2

IPv6 packets are disordered, but also provide a basis for investigating TCP behaviors in IPv6 world.

3. Methodologies

In this section, we first take a look at the probing tool, *OneProbe*. Then we describe our probing method.

3.1. Overview of OneProbe. *OneProbe* is a reliable and metric-rich path monitoring method based on TCP [19]. We use *HTTP/OneProbe*, which sends legitimate *HTTP GET* request in the TCP data probes to induce *HTTP* response messages to measure delay, packet-loss rate, and packet reordering rate. Each probe of *OneProbe* consists of two customized back-to-back packets, applied to measure the performance of the forward link. When probes arrive at remote-ends, they will induce remote endpoints to send back two back-to-back packets, which are used to measure the performance of the reverse link. In this study, the values of the forward link and reverse link are merged to evaluate packet-loss rate and packet reordering rate. When using *OneProbe*, we should set some parameters with appropriate values. As shown in Table 1, z refers to the size of the packets that will be captured for performance analysis. M represents the size of the probing packets, while m represents the size of the response packets. Parameter of e is used to control how long the probing will last. We denote r the sending rate of the probing packets. In our experiment, we send 2 probing packets in one second. In addition to these primary parameters, we also need a source address, a destination address, and its corresponding URL (Uniform Resource Locator) to launch a *OneProbe* process.

3.2. Probing Method. Based on *OneProbe*, we design a probing method, which can persistently evaluate the interdomain performance of an IPv6 network. Our probing method mainly contains three phases, that is, obtaining URLs, classifying URLs, and probing.

(1) Obtaining URLs. We first find the URLs that meet the requirements of *OneProbe*. To guarantee the accuracy of probing, the object of each URL is well over 10 Kbytes [19]. Because of probing such kind of URLs, we can induce sufficient number of response packets. We download top 1 M websites from Alexa [35] and obtain desirable URLs by crawling these websites. In addition, to make comparison between IPv4 and IPv6, only the URLs supporting both IPv4 and IPv6 access are chosen in our study.

(2) Classifying URLs. For the sake of analysis, we divide the URLs into five groups in terms of the five Regional Internet Registries (RIRs) (RIR is an organization that manages the allocation and registration of IP addresses and autonomous

system numbers. RIR divides the world into five RIRs, including African Network Information Center (AFRINIC), American Registry for Internet Numbers (ARIN), Asia-Pacific Network Information Centre (APNIC), Latin America and Caribbean Network Information Centre (LACNIC), and Réseaux IP Européens Network Coordination Centre (RIPENCC)), that is, APNIC, ARIN, AFRINIC, LACNIC, and RIPENCC. To this end, we explore the addresses of these destination URLs, and match these address to the prefixes that have been assigned to the five RIRs [36].

(3) Probing. Our probing host locates in CERNET2 [37]. To avoid cross impact caused by too many synchronous probing packets, we choose no more than 15 URLs from each RIR randomly and launch no more than 15 *OneProbe* processes synchronously at each time. Note that these URLs belong to different websites and servers. For each RIR, we use *OneProbe* to probe the destination URLs enabled with both IPv6 and IPv4 for 10 minutes (5 minutes for IPv6, 5 minutes for IPv4) in one polling cycle. That is to say, it costs 50 minutes to probe all the five RIRs one after another. Therefore, we denote an hour as a polling cycle.

4. Performance Analysis

According to the probing methods depicted in Section 3, we conduct a case study on IPv6 performance from four aspects, that is, reachability, delay, packet loss, and packet reordering. In this study, one-week probing data is gathered and used. There are 168 polling cycles during the one-week observation.

4.1. Reachability. Reachability is a basic performance metric for network operation and management. There are many reasons that may cause network interruption, such as routing policy adjustment, security policy control, physical link failure, and intermediate node failure. We use *interrupts* and *interrupt rate* to describe the reachability. *Interrupts* denotes the number of disconnects between the probing client and probed destination URLs. $\text{Interrupt rate} = \text{interrupts} / (\text{the number of polling cycles} * \text{the number of URLs})$.

As shown in Table 2, RIPENCC of IPv6 has the most number of *interrupts*. *Interrupt rates* of APNIC and LACNIC are lower than those of the other three RIRs, while, as shown in Table 3, *interrupt rates* of IPv4 in APNIC and LACNIC are higher than those of the other three RIRs. In APNIC and LACNIC, IPv6 performs better than IPv4. However, the total *Interrupts* of IPv6 are 676, while the number of IPv4 is 437. Overall, the reachability of IPv4 is better than that of IPv6.

4.2. Packet Delay. We use round trip time (RTT) to describe the time overhead (i.e., delay) of a packet that goes to the destination and returns to the source.

TABLE 2: Reachability of IPv6.

Regions	APNIC	ARIN	RIPENCC	AFRINIC	LACNIC
Interrupts	20	136	272	228	20
Interrupt rate	0.4%	2.7%	5.4%	6.2%	0.54%

TABLE 3: Reachability of IPv4.

Regions	APNIC	ARIN	RIPENCC	AFRINIC	LACNIC
Interrupts	287	18	24	8	100
Interrupt rate	5.7%	0.36%	0.48%	0.2%	2.1%

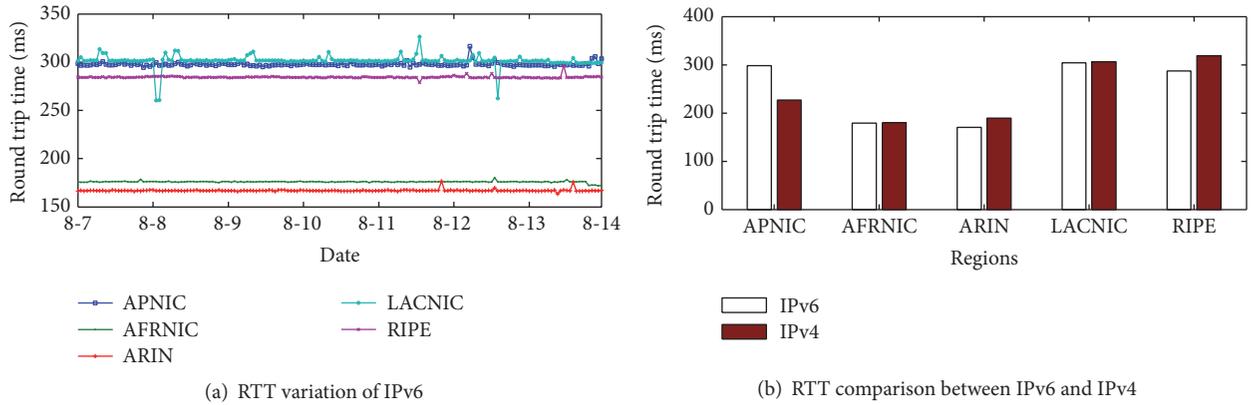


FIGURE 1: RTT variation across the five RIRs.

4.2.1. Results Analysis. Figure 1(a) shows the variation of IPv6 RTT across the five RIRs. RTT of ARIN is approximately 160 milliseconds, and RTT of AFRINIC is around 180 milliseconds. RTT of APNIC and LACNIC are similar to each other. However, RTT of APNIC is more stable than that of LACNIC. The reason for the instability is that some URLs of LACNIC with high RTT were interrupted during our measurement; thus their RTT values cannot be measured in some polling cycles. These unreachable URLs were excluded when calculating the average RTT. As a result, the average RTT of LACNIC in some polling cycles decreases. RTT of RIPENCC fluctuates at 280 milliseconds and it is relatively stable. Results show that RTT of LACNIC, APNIC, and RIPENCC are higher than that of ARIN and AFRINIC. Through further analysis, we find that the number of hops between probing client and destination URLs in ARIN and AFRINIC is smaller than that of the other three RIRs. This is in accordance with the rule that less number of the hops usually causes smaller RTT.

According to the aggregation analysis results, we make a comparison between IPv6 and IPv4 in RTT. As depicted in Figure 1(b), in the regions of AFRINIC, ARIN, RIPENCC, and LACNIC, RTT of IPv6 is similar to that of IPv4. This is because, for a destination URL in these regions, the AS-level paths of IPv6 and IPv4 are similar. However, we find that IPv6 RTT is higher than IPv4 RTT in APNIC. We try to explain the reason for the differences by exploring AS-level paths between the probe client and the destination URLs.

4.2.2. Reason Analysis. As shown in Figure 2, we divide AS-level paths into three parts, that is, *domestic-as*, *middle-as*, and *target-domestic-as*. *Domestic-as* denotes the ASes which locate in the same country/area as the probe client, while *target-domestic-as* denotes the ASes locating in the same country/area as the destination URLs. The remaining ASes that may locate in any countries or areas are *middle-as*.

We calculate the RTT for each destination URL in APNIC. As depicted in Figure 3, for most destination URLs, such as URL {1~6, 8~12, 14, 15}, IPv6 RTT is larger than IPv4 RTT. For 80% of these URLs, the length of IPv6 *middle-as* is the same as that of IPv4 *middle-as*. For the rest of the URLs, IPv6 *middle-as* has one more AS in comparison with IPv4 *middle-as*. When a probing packet begins to enter *middle-as*, IPv6 RTT is similar to IPv4 RTT. However, when the probing packet enters *target-domestic-as* from *middle-as*, the growth of IPv6 RTT is more obvious than that of IPv4 RTT. After the probing packet enters *target-domestic-as*, the increase of IPv4 RTT is similar to that of IPv6 RTT. Therefore, the path between *middle-as* and *target-domestic-as* causes the difference between IPv6 RTT and IPv4 RTT. That is to say, the connectivity between *middle-as* and *target-domestic-as* in IPv4 network is better than that in IPv6 network, resulting in IPv4 RTT being smaller than IPv6 RTT for most destination URLs.

As shown in Figure 3, IPv6 RTT of URL {7, 13} is smaller than IPv4 RTT. (1) For URL 7, the length of IPv6 *middle-as* (i.e., the number of ASes in *middle-as*) is the same as to

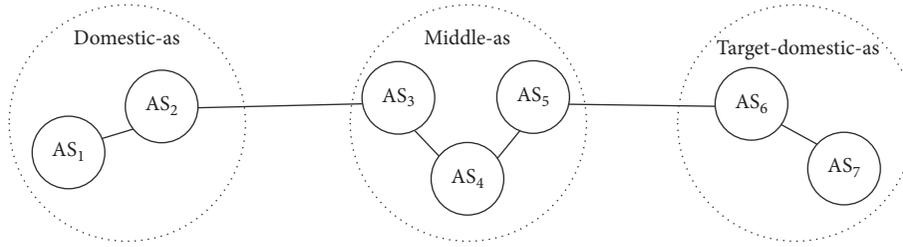


FIGURE 2: Division of AS-level paths.

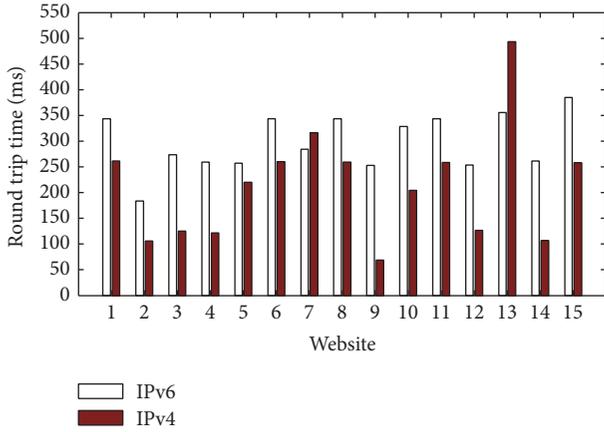


FIGURE 3: RTT distribution in APNIC.

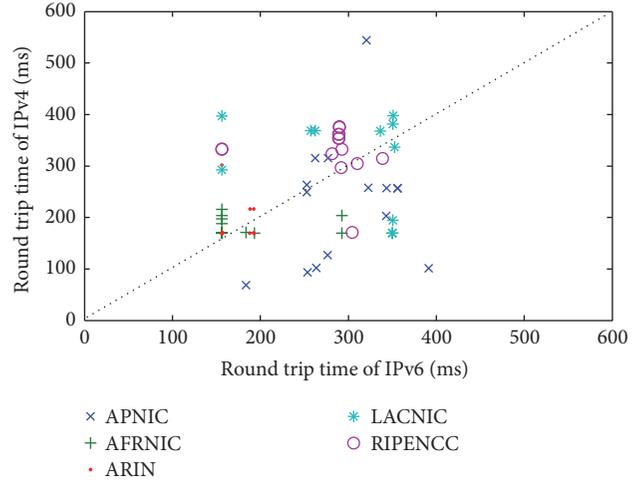


FIGURE 4: RTT distribution of IPv4/IPv6 in one polling cycle.

that of IPv4 *middle-as*. However, *middle-as* contributes 120 milliseconds to IPv6 RTT, while it adds 150 milliseconds to IPv4 RTT. This is the reason for the smaller RTT of IPv6 in comparison with IPv4. (2) For URL 13, the length of IPv6 *middle-as* is 1, while the length of IPv4 *middle-as* is 3. The increase of IPv4 RTT in *middle-as* is much larger than IPv6 RTT. Thus, for this destination URL, the length of *middle-as* causes the difference between IPv4 RTT and IPv6 RTT.

Wang et al. [13] analyzed the delay of more than 3600 websites in APNIC, ARIN, and RIPENCC. We conduct a similar analysis across all the five RIRs. We use scatter diagram to describe the RTT distribution of IPv4/IPv6 in one polling cycle. As depicted in Figure 4, for each destination URL, IPv6 RTT is represented across the *x*-axis and IPv4 RTT is represented across the *y*-axis. We find that RTT of ARIN almost locates between 150 milliseconds and 200 milliseconds for both IPv6 and IPv4. The values of ARIN are most clustered among the five RIRs. Results imply that the paths between the probing client and the destination URLs of ARIN are relatively simple. RTT of APNIC shows the most dispersive distribution, which is caused by the diversified paths between the probing client and the destination URLs.

We use *Traceroute* to explore AS-level paths between the probing client and the destination URLs. The total number of ASes between the probing client and the destination URLs is denoted as AS_{total} . Note that there are some duplicated ASes in AS_{total} . The number of unique ASes in AS_{total} is represented as AS_{unique} . AS_{differ} denotes the ratio of AS_{unique} and AS_{total} ; that is, $AS_{differ} = AS_{unique} / AS_{total}$. We calculate

the AS_{differ} of the five RIRs. The results are 91.7% (APNIC), 66.7% (RIPENCC), 40% (LACNIC), 27.3% (AFRINIC), and 13.3% (ARIN). Among the five RIRs, the AS_{differ} of APNIC is highest, which reveals that paths between the probing client and the destination URLs of APNIC are most diversified. This could explain why RTT distribution of APNIC is most dispersive. The AS_{differ} of ARIN is lowest, which could explain that RTT of ARIN is most clustered.

4.2.3. Stability Analysis. In addition, we conduct a comparison analysis between IPv4 RTT and IPv6 RTT without distinguishing regions. During the one-week observation, the average RTT of IPv6 is 247.71 milliseconds, while the average RTT of IPv4 is 244.48 milliseconds. As depicted in Figure 5, IPv6 RTT is relatively stable over time, while IPv4 RTT fluctuates greatly. This can be confirmed by the standard deviation, which is calculated based on the average RTT of each hour. The standard deviation of IPv6 is 1.4 milliseconds, while the standard deviation of IPv4 is 3.3 milliseconds.

We also utilize weighted moving average to further compare the stability and predictability of IPv6 RTT with IPv4 RTT. The average RTT of the five RIRs in each polling cycle is regarded a data point. That is to say, there are 168 data points during the whole observation. For a window size W , we predict the next data point by computing moving average for the previous consecutive W data points. We compute the moving average error for different window sizes. We find that

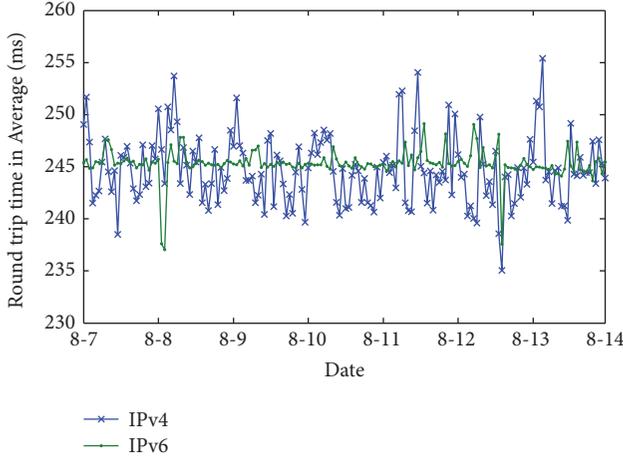


FIGURE 5: RTT comparison between IPv4 and IPv6.

when the window size equals to 2, the moving average error of IPv6 RTT is below 1% in most cases, while the moving average error of IPv4 fluctuates greatly with a maximum of 4.5%. Results reveal that stability and prediction accuracy of IPv6 RTT are much better than those of IPv4.

The more obvious fluctuation of IPv4 RTT may be caused by some overutilized links in IPv4 networks. In order to investigate the relationship between delay variation and link utilization, we probe the RTT of two backbone nodes of CERNET2. These two nodes (N_1 and N_2) directly connect another node (N_3) where we deploy the probing host. The link from N_3 to N_1 is labeled as L_1 , and the link from N_3 to N_2 is labeled as L_2 . Then we calculate the link utilization for L_1 and L_2 . Note that the link utilization denotes the ratio between the average traffic rate (in each five minutes) and the bandwidth of the physical link. For the clarity of description, we calculate and present the average link utilization of each five minutes.

As shown in Figures 6(a) and 6(b), RTT of N_2 shows a similar pattern with the utilization of L_2 . RTT of N_2 is high and fluctuant when the utilization of L_2 is large, while RTT of N_2 is low and stable when the utilization of L_2 is small. However, not all the links present the strong relationship between the delay and link utilization, such as N_1 and L_1 . Although the link utilization of L_1 has obvious fluctuations, RTT of N_1 is relatively stable. Therefore, we infer that the RTT fluctuates greatly only when the link utilization exceeds a threshold.

As depicted in Figure 7(a), we find that RTT of N_1 is concentrated around 20 milliseconds except two abnormal points. The maximum link utilization of L_1 is 0.8223 during the one-week observation. As shown in Figure 7(b), when the link utilization is below 0.7, RTT is pretty steady around 13 milliseconds except one abnormal point. When the link utilization varies between 0.7 and 0.83, RTT begins to fluctuate and generates some high values. Once the link utilization is 0.83 or above 0.83, RTT of N_2 greatly fluctuates from 20 to 80 milliseconds. In addition, when the link utilization exceeds 0.83, most points are distributed from 50 to 60 milliseconds. As depicted in Figure 7(b), the number of points in the right

of the gray line accounts for 49.4% of the total points. RTT of N_1 is relatively stable because the maximum link utilization of L_1 is 0.8223, which is smaller than 0.83. According to the above analysis, we could conclude that when the link utilization exceeds a threshold, for example, 0.83 in our case study, the RTT fluctuates greatly.

4.2.4. Model Analysis. If the RTT fluctuates significantly when the link is heavily used, we should consider the influence on the real-time applications, such as video conference, IPTV, VoIP, and online games. If we can predict the changes of RTT, we can make some reliable routing policies for these delay-sensitive applications. We take N_2 as an example and try to find a fitting model to depict and predict RTT changes.

We apply *R-square*, which is an important metric to evaluate the goodness of a fitting model or accuracy of a prediction model. As depicted in (1), y_i ($1 \leq i \leq 168$) denotes the average RTT of each polling cycle during the one-week observation. Then, \hat{y} denotes the modelling or prediction value of y_i , and \bar{y} represents the average value of y_i . The value of *R-square* is between 0 and 1. The more closely the value approaches to 1, the more accurately the model can explain the RTT changes. Through repeating attempts, we find that RTT variation of N_2 follows the model of *Fourier*. We also find that when the number of *Fourier* functions equals to 7, we could obtain the best fitting results. The value of *R-square* is 0.92. The general fitting model is shown in (2). The specific coefficient parameters of this model are shown in Table 4.

$$R\text{-square} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (1)$$

$$\begin{aligned} f(x) = & a_0 + a_1 * \cos(x * w) + b_1 * \sin(x * w) \\ & + a_2 * \cos(2 * x * w) + b_2 \\ & * \sin(2 * x * w) + a_3 \\ & * \cos(3 * x * w) + b_3 \\ & * \sin(3 * x * w) + a_4 \\ & * \cos(4 * x * w) + b_4 \\ & * \sin(4 * x * w) + a_5 \\ & * \cos(5 * x * w) + b_5 \\ & * \sin(5 * x * w) + a_6 \\ & * \cos(6 * x * w) + b_6 \\ & * \sin(6 * x * w) + a_7 \\ & * \cos(7 * x * w) + b_7 \\ & * \sin(7 * x * w). \end{aligned} \quad (2)$$

According to the fitting model shown in (2), we get the fitted curve for the RTT variation of N_2 . As shown in Figure 8, this model can depict the RTT changes over time. For each polling cycle, we extract a value from the fitted curve. These

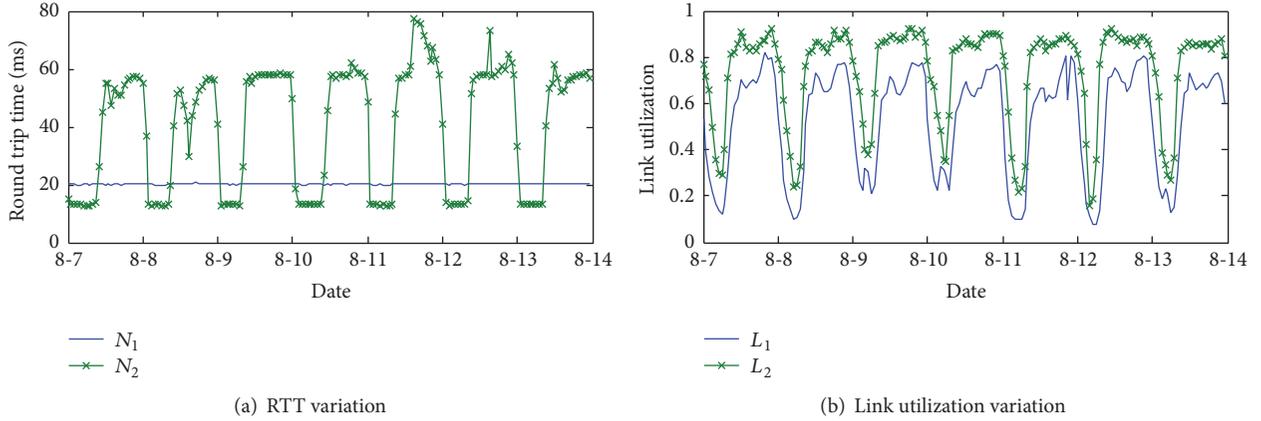


FIGURE 6: Relationship between the delay and link utilization.

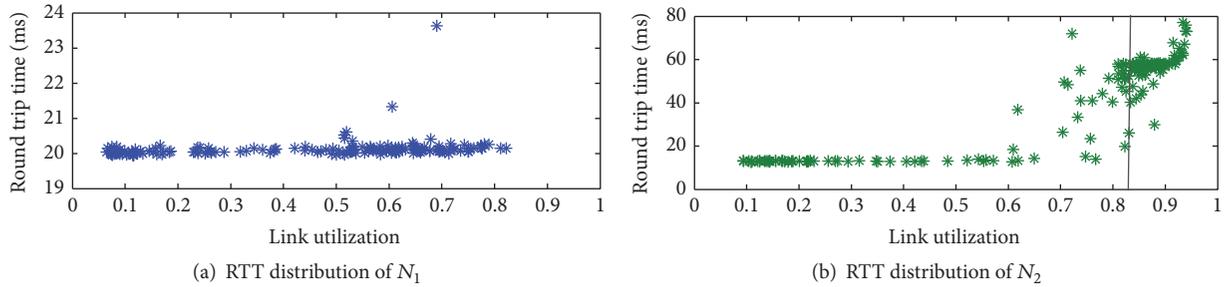


FIGURE 7: RTT distribution across link utilization.

TABLE 4: Coefficient parameters of the general fitting model.

Sequence number	a_i	b_i	a_0	w
$i = 1$	1.48	-25.66		
$i = 2$	9.183	-0.808		
$i = 3$	0.769	-3.611		
$i = 4$	4.82	-0.096	0.263	40.52
$i = 5$	0.893	1.272		
$i = 6$	1.228	0.298		
$i = 7$	0.622	1.434		

values are considered as the predicted RTT. We also use *R-square* to evaluate the prediction accuracy. The value of *R-square* is 0.74, which illustrates that this model can predict the RTT to a certain extent. In this study, we take N_2 as an example to present the property of RTT variation when the link utilization exceeds a threshold. And the significance is that we could make some reliable routing policies in advance, which can decrease or avoid the influence on delay-sensitive applications caused by delay fluctuation.

4.3. Packet-Loss Rate. Figure 9(a) shows the variation of IPv6 packet-loss rate across the five RIRs. During the observation period, we find that packet-loss rate of LACNIC fluctuates greatly. The maximum packet-loss rate of LACNIC is close to 2%. Packet-loss rate of ARIN is higher than that of AFRINIC,

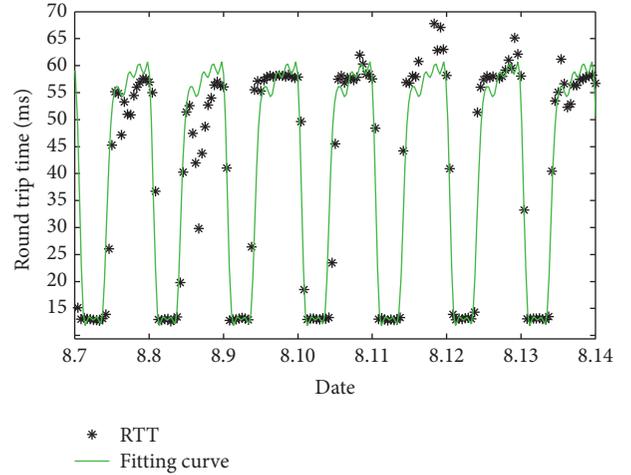


FIGURE 8: Fitting curve of the RTT variation of N_2 .

but both of them are stable at around 0.5%. Packet-loss rates of APNIC and RIPENCC keep below 0.3%. Packet-loss rates of APNIC, RIPENCC, and LACNIC are more fluctuant, but are lower than those of ARIN and AFRINIC in most cases. However, as depicted in Figure 1(a), RTT of APNIC, RIPENCC, and LACNIC is larger than that of ARIN and AFRINIC. Therefore, we may conclude that delay does not affect the value of packet-loss rate, but if delay of a path is large, packet-loss rate of that path is more likely to fluctuate.

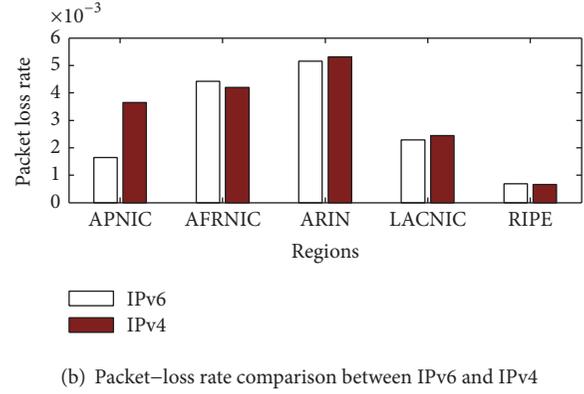
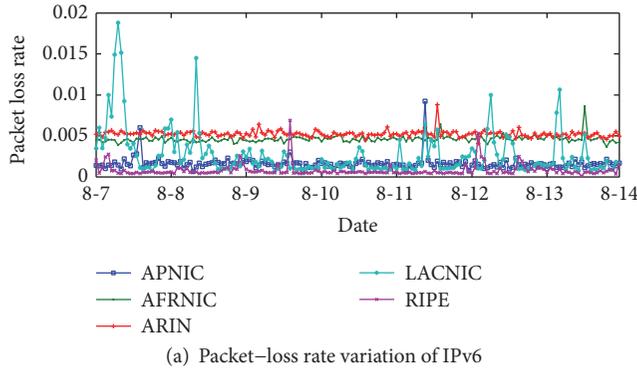


FIGURE 9: Packet-loss rate variation across the five RIRs.

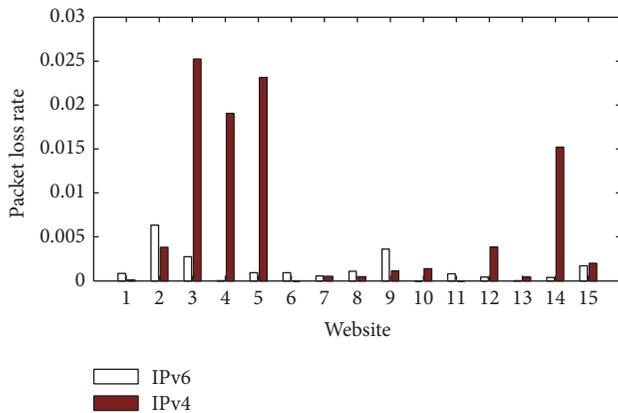


FIGURE 10: Packet-loss rate distribution of APNIC.

We make a comparison analysis between IPv4 and IPv6 in packet-loss rate. As depicted in Figure 9(b), in the regions of AFRNIC, ARIN, RIPENCC, and LACNIC, packet-loss rate of IPv6 is relatively close to that of IPv4, which is in accordance with the RTT analysis results. But packet-loss rate of IPv6 is lower than that of IPv4 in APNIC, which is opposite to the RTT analysis results. This confirms that delay does not affect the value of packet-loss rate. We then try to find the reason why there is a difference between IPv6 packet-loss rate and IPv4 packet-loss rate in APNIC. We calculate the packet-loss rate for each destination URL in APNIC. As depicted in Figure 10, packet-loss rates of IPv4 and IPv6 are close to each other for most URLs. However, some URLs present obvious differences between IPv4 and IPv6. (1) For URL {3~5, 12, 14}, packet-loss rate of IPv4 is much higher than that of IPv6. (2) For URL {2, 9}, packet-loss rate of IPv4 is lower than that of IPv6. Further analysis reveals that packet-loss rates in *domestic-as* are roughly the same for these URLs and the differences are mainly caused by *middle-as*. For URL {3~5, 12, 14}, IPv4 is encountered with higher packet-loss rate than IPv6 in *middle-as*, while for URL {2, 9}, IPv6 is encountered with higher packet-loss rate than IPv4 in *middle-as*.

In addition, we conduct a comparison analysis between IPv4 packet-loss rate and IPv6 packet-loss rate without

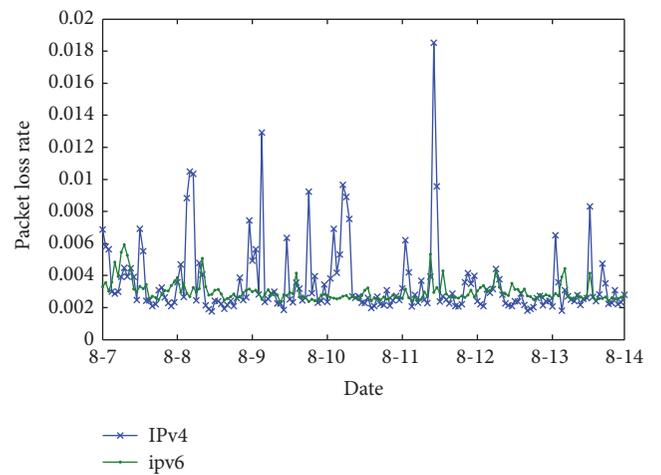


FIGURE 11: Packet-loss rate comparison between IPv4 and IPv6.

distinguishing regions. The average packet-loss rate of IPv6 is 0.25%, while the average packet-loss rate of IPv4 is 0.33%. As depicted in Figure 11, we notice that IPv6 packet-loss rate tends to be stable over time, while IPv4 packet-loss rate fluctuates greatly. The standard deviation of IPv6 packet-loss rate is $6.55 * 10^{-4}$, while the counterpart of IPv4 packet-loss rate is $2.44 * 10^{-3}$. This result confirms that IPv6 packet-loss rate is relatively low and fluctuates smoothly.

We utilize weighted moving average error to further compare the stability and predictability of IPv6 packet-loss rate with IPv4. The average packet-loss rate of the five RIRs in each polling cycle is regarded a data point. We find that when the window size equals 2, the moving average error of IPv6 packet-loss rate is almost below 50%, while the moving average error of IPv4 packet-loss rate is often higher than 50% and fluctuates greatly. Results reveal that stability and prediction accuracy of IPv6 packet-loss rate is much better than that of IPv4.

4.4. Packet Reordering. Packet reordering refers to a packet that does not arrive in expected order. There are many reasons resulting in out-of-order packets, for example, network

TABLE 5: Packet reordering rate comparison between IPv4 and IPv6.

Type	APNIC	AFRNIC	ARIN	LACNIC	RIPENCC
IPv6 ($*10^{-6}$)	8.1329	0	0.1686	3.2457	0
IPv4 ($*10^{-2}$)	0.7243	0.7343	0.7857	1.3629	0.8186

congestion, timeout-retransmission, and multipath transmission. In this paper, we conduct a first-ever analysis of packet reordering in IPv6 world. As shown in Table 5, during the observation period, there are no IPv6 out-of-order packets in AFRNIC and RIPENCC. Only, in some polling cycles, IPv6 has some out-of-order packets in ARIN, LACNIC, and APNIC. We also notice that packet reordering rate of IPv6 is much smaller than that of IPv4. In IPv4 world, packet reordering rates present differences among the five RIRs. Packet reordering rate of LACNIC is above 1%, while packet reordering rates of the other RIRs are below 0.8%.

We also conduct a comparison analysis without distinguishing regions. The average packet reordering rate of IPv6 is $2.3 * 10^{-6}$, while the average packet reordering rate of IPv4 is 0.79%. Two reasons cause the differences in packet reordering rate between IPv4 and IPv6. (1) Only end hosts can perform packet fragmentation in IPv6 network. And fragmentation is not encouraged in most cases, while, in IPv4 network, fragmentation can be performed by both hosts and routers, which will cause many packet fragments and increase the probability of packet reordering. (2) In addition, IPv6 simplifies its basic header to accelerate packet processing, which can further reduce the probability of packet reordering.

5. Conclusions and Future Remarks

IPv6-centric next generation network is experiencing fast development. But our understanding of IPv6 cannot keep up with the growth of IPv6. In this paper, we propose a probing method, which can persistently measure the interdomain performance of an IPv6 network. We collect one-week measurement data and revisit several critical performance metrics for the studied IPv6 network. Our main findings include (1) packet delay and loss of IPv6 being similar to its counterpart of IPv4 when the AS-level paths are roughly the same. Packet delay presents a strong correlation with the link utilization. When the link utilization exceeds a threshold, for example, 0.83 in our study, variation of packet delay presents a similar pattern with the variation of link utilization; (2) the performance of *middle-as* and the length of *middle-as* are the dominant reasons for the differences (in delay and packet-loss rate) between IPv6 and IPv4. In addition, packet delay does not affect the value of packet-loss rate, but if packet delay of a path is large, packet-loss rate of that path is more likely to fluctuate over time; (3) few IPv6 probes are out-of-order and the reordering rate is $2.3 * 10^{-6}$, which is much lower than the rate of 0.79% in IPv4 world.

In this paper, we only deploy probing client in a single source location, while it would be more interesting to deploy probing clients in multiple geographical locations in the future.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

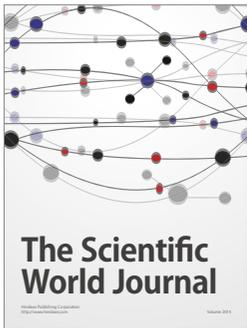
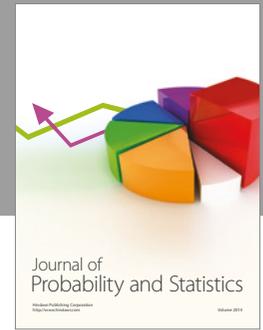
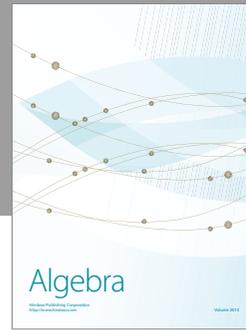
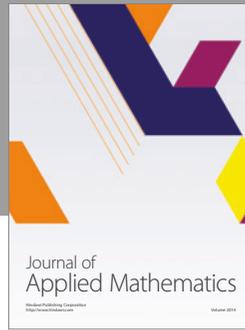
Acknowledgments

The authors would like to acknowledge the support from CNGI-NOC staff. This work is supported by the National Natural Science Foundation of China under Grant nos. 61602105 and 61572123; China Postdoctoral Science Foundation under Grant no. 2016M601323; the Fundamental Research Funds for the Central Universities Project under Grant no. N150403007; CERNET Innovation Project under Grant no. NGII20160126.

References

- [1] "Internet Protocol," RFC 791, 1981, <https://tools.ietf.org/html/rfc791>.
- [2] S. Deering and R. Hinden, "Internet Protocol version 6 (IPv6)," RFC 2460, 1998, <https://www.ietf.org/rfc/rfc2460>.
- [3] The Internet Engineering Task Force (IETF), <https://www.ietf.org/>.
- [4] "Trace Statistics for CAIDA Passive OC48 and OC192 Traces," http://www.caida.org/data/passive/trace_stats/.
- [5] IPv4 address exhaustion, https://en.wikipedia.org/wiki/IPv4_address_shortage.
- [6] *World IPv6 Day*, Internet Society, 2011, <http://isoc.org/wp/worldipv6day/>.
- [7] *World IPv6 Launch*, 2012, <http://www.worldipv6launch.org/>.
- [8] <https://www.google.com/intl/en/ipv6/statistics.html>.
- [9] http://www.caida.org/research/topology/as_core_network/2015/.
- [10] F. Li, C. An, J. Yang, J. Wu, and H. Zhang, "A study of traffic from the perspective of a large pure IPv6 ISP," *Computer Communications*, vol. 37, pp. 40–52, 2014.
- [11] F. Li, J. Yang, J. Wu, Z. Zheng, H. Zhang, and X. Wang, "Configuration analysis and recommendation: Case studies in IPv6 networks," *Computer Communications*, vol. 53, pp. 37–51, 2014.
- [12] F. Li, J. Yang, X. Wang, T. Pan, C. An, and J. Wu, "Characteristics analysis at prefix granularity: A case study in an IPv6 network," *Journal of Network and Computer Applications*, vol. 70, pp. 156–170, 2016.
- [13] Y. Wang, S. Ye, and X. Li, "Understanding current IPv6 performance: A measurement study," in *Proceedings of the 10th IEEE Symposium on Computers and Communications, ISCC 2005*, pp. 71–76, esp, June 2005.
- [14] V. K. Muniyappa, *erformance Analysis of IPv4 versus IPv6 in a Simple Campus Network*, Long Beach, 2012.
- [15] S. Shiwani, G. N. Purohit, and N. Hemrajani, "Performance Analysis of IPv4 v/s IPv6 in Virtual Environment using

- UBUNTU,” in *Proceedings of the International Conference on Computer Communication and Networks*, pp. 86–91, 2011.
- [16] X. Zhou, M. Jacobsson, H. Uijterwaal, and P. Van Mieghem, “IPv6 delay and loss performance evolution,” *International Journal of Communication Systems*, vol. 21, no. 6, pp. 643–663, 2008.
- [17] M. Nikkhah, R. Guérin, Y. Lee, and R. Woundy, “Assessing IPv6 through web access a measurement study and its findings,” in *Proceedings of the 7th ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, pp. 1–26, December 2011.
- [18] F. Li, X. Wang, T. Pan, and J. Yang, “Packet delay, loss and reordering in IPv6 world: a case study,” in *Proceedings of the International Conference on Computing, Networking and Communications (ICNC '16)*, pp. 1–6, February 2016.
- [19] X. Luo, E. W. W. Chan, and R. K. C. Chang, “Design and Implementation of TCP Data Probes for Reliable and Metric-Rich Network Path Monitoring,” in *Proceedings of the Design and Implementation of TCP Data Probes for Reliable and Metric-Rich Network Path Monitoring*, 2009.
- [20] A. Dhamdhere, M. Luckie, B. Huffaker, K. Claffy, A. Elmokashfi, and E. Aben, “Measuring the deployment of IPv6: topology, routing and performance,” in *Proceedings of the ACM Internet Measurement Conference (IMC '12)*, pp. 537–550, Boston, Mass, USA, November 2012.
- [21] J. Czyz, M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil, and M. Bailey, “Measuring IPv6 Adoption,” in *ACM SIGCOMM Computer Communication Review*, vol. 44, pp. 87–98, 2014.
- [22] I. Livadariu, A. Elmokashfi, and A. Dhamdhere, “Characterizing IPv6 control and data plane stability,” in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM 2016*, pp. 1–9, April 2016.
- [23] V. Bajpai and J. Schönwälder, “IPv4 versus IPv6-Who connects faster?” in *Proceedings of the 2015 14th IFIP Networking Conference, IFIP Networking 2015*, pp. 1–9, May 2015.
- [24] F. Lin, G. Wang, J. Zhou, S. Zhang, and X. Yao, “High-performance IPv6 address lookup in GPU-accelerated software routers,” *Journal of Network and Computer Applications*, vol. 74, pp. 1–10, 2016.
- [25] W. Stevens, “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms,” RFC Editor RFC2001, 1997.
- [26] M. Laor and L. Gendel, “The effect of packet reordering in a backbone link on application throughput,” *IEEE Network*, vol. 16, no. 5, pp. 28–36, 2002.
- [27] Y. Wang, G. Lu, and X. Li, “A study of Internet packet reordering,” in *Proceedings of the International Conference on Information Networking*, Lecture Notes in Computer Science, pp. 350–359, 2004.
- [28] X. Zhou and P. Van Mieghem, “Reordering of IP packets in Internet,” in *Proceedings of the Passive and Active Network Measurement*, pp. 237–246, 2004.
- [29] S. Bohacek, J. P. Hespanha, J. Lee, C. Lim, and K. Obraczka, “TCP-PR: TCP for persistent packet reordering,” in *Proceedings of the 23th IEEE International Conference on Distributed Computing Systems*, pp. 222–231, May 2003.
- [30] J. C. R. Bennett, C. Partridge, and N. Shectman, “Shectman. Packet reordering is not pathological network behavior,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 789–798, 1999.
- [31] M. Przybylski, B. Belter, and A. Binczewski, “Shall we worry about packet reordering?” *Computational Methods in Science and Technology*, vol. 11, no. 2, pp. 141–146, 2005.
- [32] S. Dharmapurikar and V. Paxson, “Robust TCP Stream Reassembly in the Presence of Adversaries,” in *Proceedings of the USENIX Security Symposium*, 2005.
- [33] G. Varghese, J. Andrew Fingerhut, and F. Bonomi, “Detecting evasion attacks at high speeds without reassembly,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 327–338, 2006.
- [34] M. Vutukuru, H. Balakrishnan, and V. Paxson, “Efficient and robust TCP stream normalization,” *Security and Privacy*, pp. 96–110, 2008.
- [35] Sites Top, <http://www.alexa.com/topsites>.
- [36] Statistics, <https://www.nro.net/statistics>.
- [37] J. Wu, J. H. Wang, and J. Yang, “CNGI-CERNET2: an IPv6 Deployment in China. ACM SIGCOMM Computer Communication Review,” *CNGI-CERNET2: an IPv6 Deployment in China*, vol. 41, no. 2, pp. 48–52, 2011.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

