Benchmarking Methodology Working Group
Internet-Draft
Intended status: Informational
Expires: 2 January 2026

G. Lencse
Sz辿chenyi Istv叩n University
K. Shima
SoftBank Corp.
O. Troan
Cisco

1 July 2025

Recommendations for using Multiple IP Addresses in Benchmarking Tests
draft-lencse-bmwg-multiple-ip-addresses-05

Abstract

RFC 2544 has defined a benchmarking methodology for network interconnect devices. Its test frame format contained fixed IP addresses and fixed port numbers. RFC 4814 introduced pseudorandom port numbers but used a single source and destination IP address pair when testing with a single destination network. This limitation may cause an issue when the device under test uses the Receive-Side Scaling (RSS) mechanism in the packet processing flow. RSS has two implementations: the first only includes the IP addresses, whereas the second also includes the port numbers in the tuple used for hashing. Benchmarking tests that use a single IP address pair and RFC 4814 pseudorandom port numbers are biased against the first type of RSS implementation because traffic is not distributed among the processing elements. This document recommends the usage of pseudorandom IP addresses in a similar manner as RFC 4814 did with the port numbers.

If accepted, this document updates all affected RFCs, including RFC 2544, RFC 4814, RFC 5180, RFC 8219.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 2 January 2026.

Table of Contents

1. Introduction

[RFC2544] has defined a comprehensive benchmarking methodology for network interconnect devices, which is still in use. It was amended by several RFCs, which did not formally update it. [RFC4814] introduced pseudorandom port numbers (instead of fixed ones). [RFC5180] addressed IPv6 specificities and added technology updates but declared IPv6 transition technologies out of scope. [RFC8219] addressed the IPv6 transition technologies.

Contemporary IPv4 or IPv6 packet-forwarding devices typically have multiple packet processing elements and the packets are load-balanced across them. In the case of hardware routers, packets are shared among multiple Application-Specific Integrated Circuit (ASIC) pipelines, whereas in the case of software routers, packets are distributed among multiple CPU cores. In both cases, certain packet fields, e.g., source and destination IP addresses and/or port numbers are usually used as input fields of a hash function to ensure per-flow consistency. Software routers typically use Receive-Side Scaling (RSS). It has two types of implementations: the first one only includes the IP addresses, whereas the second one also includes the port numbers into the tuple used for hashing [RSS2014]. [RFC4814] compliant testers work properly in the second case; however, the pseudorandom port numbers cannot provide entropy if the Device Under Test (DUT) follows the first type of RSS implementation; therefore, these devices produce poor benchmarking results in [RFC4814] compliant laboratory tests, whereas they can exhibit high performance in production environments where the usage of multiple IP addresses ensures the entropy for the proper operation of their RSS implementation. Therefore, the conditions of laboratory tests should be improved to ensure unbiased performance testing. To that end, this document examines how the usage of multiple IP addresses can be introduced in the performance testing of network interconnect devices using IPv4 or IPv6 addresses, observing the limitations of the ranges of special purpose IPv4 and IPv6 addresses reserved for benchmarking measurements. Practical recommendations are given for the usage of pseudorandom source and destination IP addresses in the case of both IPv4 and IPv6 following the approach of RFC 4814 regarding the port numbers and also considering the effect of the growing number of ARP or NDP table entries.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Recommendation for Multiple IP Addresses

First, the potential ranges are examined in the case of IPv4 and IPv6, then considerations regarding the "optimal" number of IP addresses and the usage of gateways are made.

2.1. Potential Ranges for IPv4 Addresses

The 198.18.0.0/15 IPv4 address range was reserved for benchmarking tests. It is divided into two halves: 198.18.0.0/16 and 198.19.0.0/16 are to be used on the two sides of the test setup. Considering the requirement of [RFC2544] regarding the IP addresses, the test suite SHOULD be first run with a single source and destination address pair. Then, the destination networks should be random using the 16-23 bits of the network addresses mentioned above. The Device Under Test (DUT) is assigned the first address of each network, and the Tester can be assigned, for example, the second address from each network. That is, 198.18.R.1/24 and 198.19.R.1/24 are assigned to the DUT; 198.18.R.2/24 and 198.19.R.2/24 are assigned to the Tester, where R is pseudorandom in the 0-255 interval.

The above framework provides rules on the design of how multiple IP addresses can be used and the scarcity of the IPv4 addresses imposes serious limitations. It means that when, e.g., the very first networks (198.18.0.0/24 and 198.19.0.0/24) are used at each side of the test setup, the maximum range of the IP addresses assigned to the Tester can be 198.18.0.2/24-198.18.0.254/24 and 198.19.0.2/24-198.19.0.254/24, as shown in Figure 1. When 256 destination networks are used, then the 16-23 bits identifying the destination networks also contribute to the entropy provided to the hash function. When only a single destination network is used, then the 16-23 bits can also be leveraged to generate a higher number of IP addresses, thus their ranges can be: 198.18.0.2/16-198.18.255.254/16 and 198.19.0.2/16-198.19.255.254/16 as shown in Figure 2. In both cases, the Tester and the DUT are in the same networks, that is, they are connected directly without using gateways. Conversely, the corresponding network interfaces of the Tester and the DUT may be connected through gateways. Then the network interface of the DUT uses an IP address from a different network than the corresponding interface of the Tester, and the network interface of the DUT sends the packets to the gateway, which is set as the next hop router towards the network assigned to the corresponding interface of the Tester, as shown in Figure 3. (It is noted that [RFC1918] private IP addresses were used due to the insufficient IP address range reserved for benchmarking.)

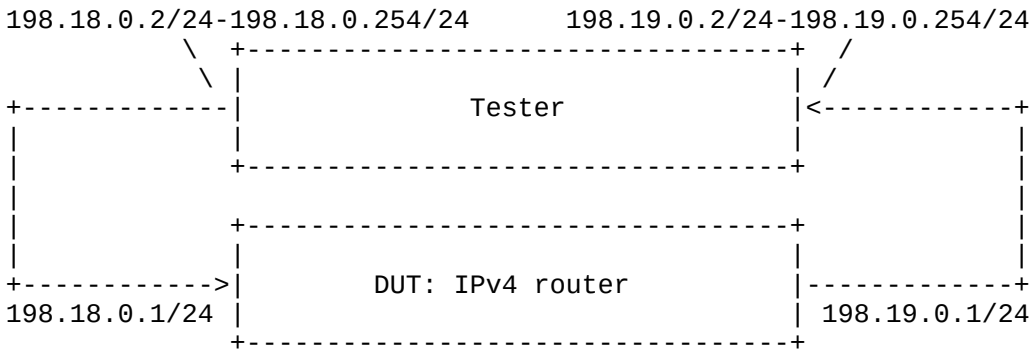


Figure 1: Test setup for benchmarking IPv4 routers when using multiple destination networks (Tester and DUT are directly connected)

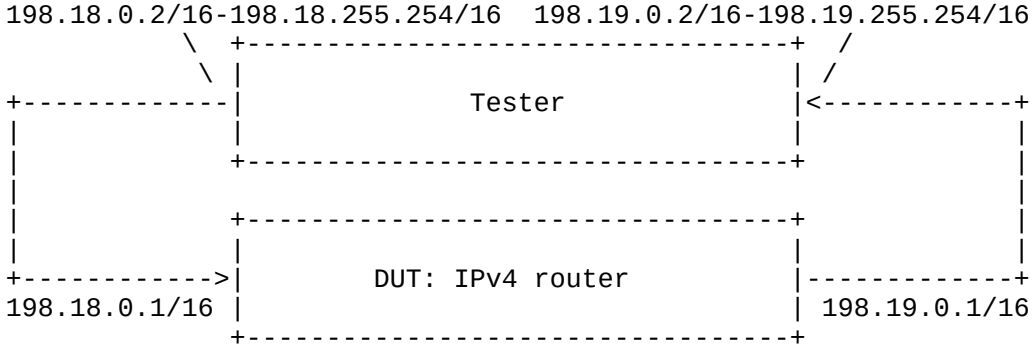
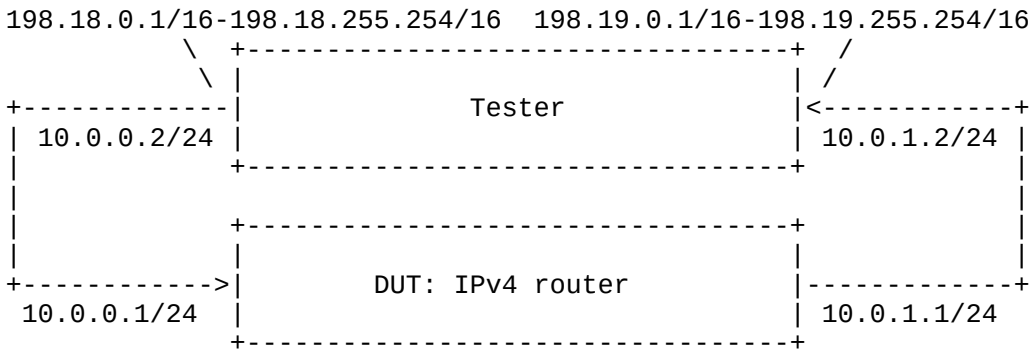


Figure 2: Test setup for benchmarking IPv4 routers when using a single destination network (Tester and DUT are directly connected)



routes in the DUT:

destination network	next hop
198.18.0.0/16	10.0.0.2
198.19.0.0/16	10.0.1.2

Figure 3: Test setup for benchmarking IPv4 routers when using a single destination network (gateways are used)

2.2. Potential Ranges for IPv6 Addresses

The 2001:2::/48 IPv6 address range, which was reserved for benchmarking tests, is large enough. If it is split into two halves to be used on the two sides of the test setup as 2001:2::/49 and 2001:2:8000::/49, the ranges are abundant. Even if their first /56 subnets (2001:2::/56 and 2001:2:8000::/56) are enough to ensure 256 networks on each side of the test setup. As these networks are of /64 size, their host ID parts are vastly abundant. For convenience considerations, we recommend the usage of their 96-111 bits to generate potentially 65536 different IP addresses, as shown in Figure 4 in the case when a single destination network is used and the Tester and the DUT are connected directly, without gateways. (When needed, the 256 destination networks can be described by the 56-63, bits as mentioned before.) Figure 5 shows the case when gateways are used.

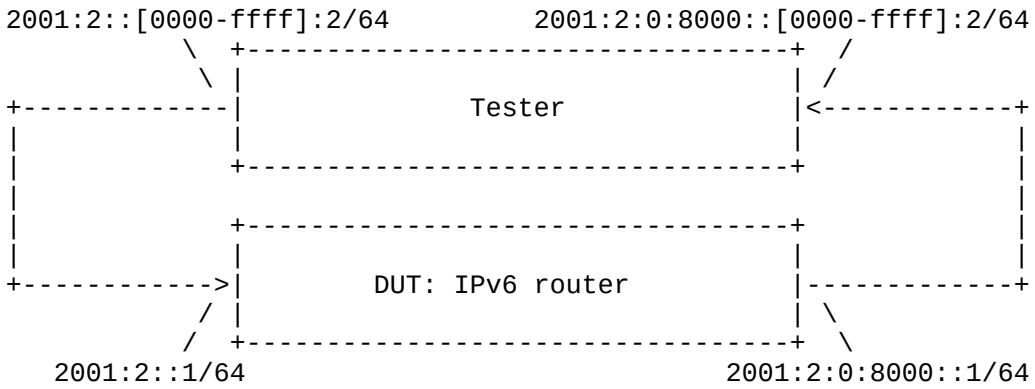
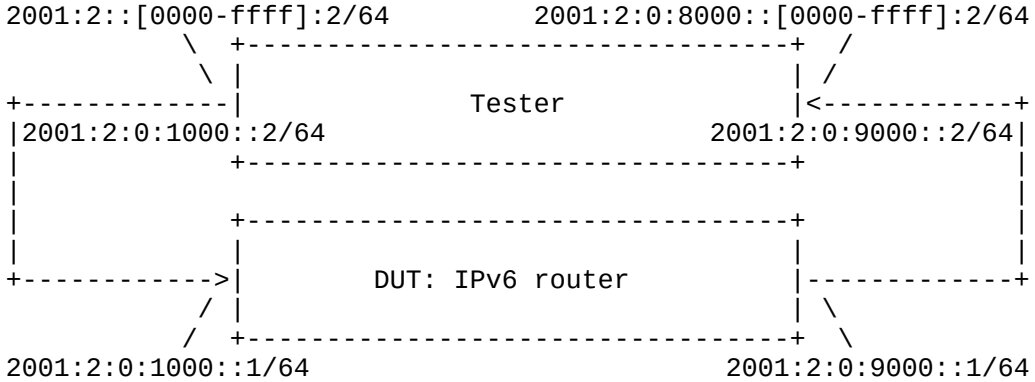


Figure 4: Test setup for benchmarking IPv6 routers (Tester and DUT are directly connected)



routes in the DUT:

destination network	next hop
2001:2::/64	2001:2:0:1000::2/64
2001:2:0:8000::/64	2001:2:0:9000::2/64

Figure 5: Test setup for benchmarking IPv6 routers (gateways are used)

2.3. Considerations for the IP Address Ranges to be Used

On the one hand, the more IP addresses are used, the more entropy is ensured and thus the most even distribution of the load over the processing elements can be expected. However, on the other hand, the usage of multiple IP addresses has its costs when no gateways are used: multiple Address Resolution Protocol (ARP for IPv4) or Neighbor Discovery Protocol (NDP, for IPv6) table entries are used. Increasing them over a few thousand may have a deteriorating effect on the performance of the DUT. This effect does exist when gateways are used because the DUT sends the packets to the gateways, and thus only their IPv4 or IPv6 addresses needed to be stored in the ARP or NDP table of the DUT.

It is noted that under typical operating conditions, a router is not connected directly to a high number of devices. If it is a backbone router, it is connected directly to several other routers. If it is a local router, it is connected directly to a single upstream router (or, at most, a few of them) and (through a switch) to the local hosts, the number of which is unlikely to be higher than a few thousand. In both cases, a high number of different IP addresses may provide entropy for hashing without causing pressure on the ARP or NDP tables of the router.

3. Implementation of the Recommended Solution

The recommended solution has been implemented in siitperf [SIITPERF] as a proof of concept. Multiple IPv4 and IPv6 addresses are supported from commit number 165cb7f on September 6, 2023. The details of the implementation can be found in [LEN2024].

4. Recommendation for Testing with Multiple IP Addresses

Based on the theoretical considerations in Section 2.3 that

1. it is desirable to use as high number of IP addresses as possible

2. routers do not need to handle a high number of neighbors under typical operating conditions

the usage of the maximum possible number of IP addresses and the test setups with gateways shown in Figure 3 and Figure 5 are recommended.

It is noted that the maximum possible number of IPv4 addresses are 253 and 65533 when 8 bits and 16 bits are available. In the case of IPv6, it is 65536.

This recommendation has been justified by the measurement results mentioned in Appendix A.

5. Considerations for Stateful Tests

Stateful technologies like stateful NAT44 or stateful NAT64 are out of the scope of this document. They are covered by [RFC9693].

6. Acknowledgements

The authors would like to thank Boris Hassanov, Giuseppe Fioccola, and Libin Liu for their review and comments.

7. IANA Considerations

This document does not make any request to IANA.

8. Security Considerations

The usage of high number of different IP addresses may exhaust the ARP/NDP table of the DUT. As such, it may be considered by the DUT as a kind of Denial of Service attack.

For further security considerations, please refer to Section 13 of [RFC8219].

9. References

9.1. Normative References

[RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <https://www.rfc-editor.org/info/rfc1918>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, DOI 10.17487/RFC2544, March 1999, <https://www.rfc-editor.org/info/rfc2544>.

[RFC4814] Newman, D. and T. Player, "Hash and Stuffing: Overlooked Factors in Network Device Benchmarking", RFC 4814, DOI 10.17487/RFC4814, March 2007, <https://www.rfc-editor.org/info/rfc4814>.

[RFC5180] Popoviciu, C., Hamza, A., Van de Velde, G., and D. Dugatkin, "IPv6 Benchmarking Methodology for Network Interconnect Devices", RFC 5180, DOI 10.17487/RFC5180, May 2008, <https://www.rfc-editor.org/info/rfc5180>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8219] Georgescu, M., Pislaru, L., and G. Lencse, "Benchmarking Methodology for IPv6 Transition Technologies", RFC 8219, DOI 10.17487/RFC8219, August 2017, <https://www.rfc-editor.org/info/rfc8219>.

[RFC9693] Lencse, G. and K. Shima, "Benchmarking Methodology for Stateful NATxy Gateways", RFC 9693, DOI 10.17487/RFC9693, January 2025, <https://www.rfc-editor.org/info/rfc9693>.

9.2. Informative References

[LEN2024] Lencse, G., "Making stateless and stateful network performance measurements unbiased", Computer Communications, vol. 225, no. 1, pp. 141-155, DOI 10.1016/j.comcom.2024.05.018, 1 September 2024, <https://www.sciencedirect.com/science/article/pii/S0140366424001993>.

[OBSD72CL] de Raadt, T., "OpenBSD 7.2 Changelog", available online, 20 October 2022, <https://www.openbsd.org/plus72.html>.

[RSS2014] Herbert, T. and W. Brujin, "Scaling in the Linux networking stack", Linux Kernel Documentation, available from GitHub, 2014, <https://www.kernel.org/doc/Documentation/networking/scaling.txt>.

[SIITPERF] Lencse, G., "Siitperf: An RFC 8219 compliant SIIT and stateful NAT64/NAT44 tester written in C++ using DPDK", source code, available from GitHub, 2019-2023, <https://github.com/lencsegabor/siitperf>.

Appendix A. Experiments and Results

A.1. Demonstration of the Difference

The effectiveness of the solution was also demonstrated in [LEN2024]. OpenBSD was chosen as the operating system of the DUT. It uses the first type of RSS solution: only the IP addresses are used by the hash function. It was examined how much difference the usage of multiple IP addresses makes in the IPv4 and IPv6 throughput performance. It should be noted that IP packet forwarding under OpenBSD was single-threaded until version 7.1. The ChangeLog of OpenBSD 7.2 [OBSD72CL] states, "Activated parallel IP forwarding, starting 4 softnet tasks but limiting the usage to the number of CPUs."

The test setup for the IPv4 and IPv6 measurements was according to Figure 2 and Figure 4, respectively. However, only 1000 different IP addresses were used at each side of the test setups to limit the potential performance degradation caused by the high number of ARP or NDP table entries.

The DUT was a Dell PowerEdge R730 server with two 3.2GHz Intel Xeon E5-2667 v3 CPUs having 8 cores each, 8x16GB 2133MHz DDR4 SDRAM (accessed quad channel), and Intel X540-T2 10GbE network adapter. Hyper-threading was switched off in the BIOS.

All tests were executed 10 times, and the median, minimum and maximum values of the throughput results were calculated. In the case of IPv4 packet forwarding, the usage of pseudorandom IP addresses caused a highly significant (more than 3-fold) performance increase compared to the case when fixed IP addresses were used. In the case of IPv6, the throughput values were significantly lower, and the increase caused by the usage of pseudorandom IP addresses was only about 50%, but it is still a well-visible difference. All the details of the measurements can be found in [LEN2024].

A.2. Examination of the Effect of the Number of IP Addresses Used

A.2.1. Without Gateways

To examine how the number of IP addresses used affects the throughput, the same hardware described in Appendix A.1 was used, but the DUT had the Debian Linux 11.7 operating system to be able to fully utilize all CPU cores, and the first type of RSS was set using the "ethtool" command. The IPv4 and IPv6 test setups followed Figure 2 and Figure 4, respectively.

Two measurement series were performed: one with IPv4 and the other with IPv6. As for the IPv4 addresses, their number was doubled in the consecutive experiments: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512,

1k, 2k, 4k, 8k, 16k, 32k, and 64k-3 IPv4 addresses were used. Thus, the number of IPv4 address combinations (supplying entropy to RSS) were 1x1, 2x2, 4x4, 8x8, 16x16, 32x32, ... , 16kx16k, 32kx32k, and (64k-3)x(64k-3). The same number of IPv6 addresses were used, except that the last value was exactly 64k.

The trend of the results followed the same pattern for IPv4 and IPv6. The throughput increased steeply during the first few consecutive experiments, then it was nearly constant for a relatively wide range of the number of IP addresses, and finally, it deteriorated significantly.

Although the shallow observer could conclude that the above experienced nearly constant throughput for a wide range of the number of IP addresses, making it easy to choose a "good enough" number, the authors contend the opposite. On the one hand, in the general case, the number of CPU cores of the DUT may be significantly higher than in the above case. On the other hand, the beginning of the performance degradation may depend on several factors, such as ARP or NDP table implementation, the sizes of the given level (L1, L2, etc.) caches of the CPU, etc. Moreover, it is problematic if the testing method needs to be aware of the details of the internal structure and operation of the DUT. Finally, performing a series of measurements (like above) is highly time-consuming, and thus, it should be avoided.

A.2.2. With Gateways

To examine the throughput of IPv4 and IPv6 packet forwarding when gateways are used, the test setups shown in Figure 3 and Figure 5 were employed, respectively. And the test system was the same as in Appendix A.2.1. The number of IPv4 and IPv6 addresses was 64k-2 and 64k, respectively. To have a basis for comparison, the IPv4 and IPv6 packet forwarding throughput measurements were also performed using single IPv4 and IPv6 address pairs plus [RFC2544] pseudorandom port numbers with the second type of RSS setting, where port numbers are also included in the hash function. The two different tests gave approximately the same results in both the IPv4 and the IPv6 measurements.

Appendix B. Change Log

B.1. 00

Initial version.

B.2. 01

Measurement results added.

B.3. 02

Minor update (one cited reference was published).

B.4. 03

Test setups with gateways, measurement results, and final recommendations added. Grammar checking was done.

B.5. 04

Addressed the review comments of Boris Hassanov. Recommendation for Testing with Multiple IP Addresses was moved from Section 6 to Section 2.4.

B.6. 05

Addressed the review comments of Giuseppe Fioccola. Gateway addresses were added to Figures 3 and 5. "Experiments and Results" was moved from Section 4 to the Appendix. Recommendation for Testing with Multiple IP Addresses was moved from Section 2.4 to Section 4. Security considerations were updated.

In the Introduction, hardware and software routers were distiguished.

Ole Troan joined as a co-author.

Authors' Addresses

G叩bor Lencse
Sz辿chenyi Istv叩n University
Gyナ喪
Egyetem tテゥr 1.
H-9026
Hungary
Email: lencse@sze.hu

Keiichi Shima
SoftBank Corp.
1-7-1 Kaigan, Tokyo
105-7529
Japan
Email: shima@wide.ad.jp

URI: https://softbank.co.jp/

Ole Troan
Cisco
Philip Pedersens vei 1
N-1366 Lysaker
Norway
Email: ot@cisco.com