Received: 16 December 2023 | Revised: 25 July 2024 | Accepted: 22 August 2024

DOI: 10.1002/dac.5982

RESEARCH ARTICLE

WILEY

Equivalence and difference of the dual device under test setup and the single device under test setup of RFC 8219

Gábor Lencse[1] | **Szabolcs Szilágyi**[2]

[1]Department of Telecommunications, Széchenyi István University, Győr, Hungary

[2]Department of IT Systems and Networks, University of Debrecen, Debrecen, Hungary

Correspondence
Gábor Lencse, Department of Telecommunications, Széchenyi István University, 1 Egyetem tér, H-9026 Győr, Hungary.
Email: lencse@sze.hu

Summary

RFC 8219 has defined a comprehensive benchmarking methodology for the IPv6 transition technologies. It recommends two kinds of measurement setups: The dual device under test (DUT) setup facilitates the benchmarking of the customer edge (CE) and provider edge (PE) devices together using a legacy RFC 2544 or RFC 5180 Network Performance Tester, whereas the single DUT setup requires a separate technology-specific tester for the benchmarking of each device. As such, special-purpose testers do not exist for the vast majority of the IPv6 transition technologies; the only viable solution can be the usage of the dual DUT setup. In this paper, we investigate if the two kinds of measurement setups provide the same or different results; moreover, we examine how the single DUT measurement results can be estimated from the dual DUT measurement results. To that end, we make theoretical considerations and also perform IPv4 packet forwarding and stateless IP/ICMP translation (SIIT) measurements using both measurement setups and analyze the results of the throughput and latency measurements. It was found that the throughput results of the dual DUT setup could approximate well those of the single DUT setup and their differences followed the predictions of our theoretical considerations. However, the latency results did not always follow the theoretical expectations.

KEYWORDS

benchmarking, IPv6 transition technologies, performance analysis, SIIT

1 | INTRODUCTION

The Internet is undergoing a transition from IPv4 to IPv6, which has been a slow process for several reasons and will likely as not take at least another decade.[1] As IPv4 and IPv6 are incompatible with each other, IETF has standardized several *IPv6 transition technologies*.[2] They can be used in various communication scenarios. For example, in the current state of the transition to IPv6, the IPv4-as-a-Service (*IPv4aaS*) technologies enable the Internet service providers (ISPs) to use only IPv6 in their access and core networks (to save costs) and still continue providing an IPv4 Internet access for their customers. The five most important such technologies are 464XLAT, DS-Lite, MAP-E, MAPT-T, and Lw4o6.[3] They all have their advantages and disadvantages[4] that network operators need to consider when choosing the most appropriate IPv4aaS implementations for their networks. Performance and especially scalability can be important decision factors when they choose the technology and its given implementation to use. RFC 8219[5] defined a benchmarking methodology for IPv6 transition technologies in 2017. However, the network operators still cannot find published measurement results of the performance of different implementations of the various IPv4aaS solutions. One of the chief obstacles to performing RFC 8219 compliant measurements is the fact that the single device under test (DUT) setup requires a technology-specific tester (see Section 2.2 for details). One mitigation for the problem can be the usage of the dual DUT setup. In theory, it can be applied to eliminate the need for a technology-specific tester, allowing instead the use of a commercially available RFC 2544[6] or RFC 5180[7] compliant Network Performance Tester. However, RFC 8219 mentions that "one of the limitations of the dual DUT setup is the inability to reflect asymmetries in behavior between the DUTs." In addition to that, we shall point out further potential and actual differences.

Throughput is a widely used performance characteristic, which can be measured in various ways. For example, Kovács used **iperf**, and he also measured the download time of a large file and then calculated the throughput as file size per download time.[8] The results of these two kinds of measurements were in good agreement. Al-Imareen and Lencse used **iperf3** (the latest version of **iperf**) in both.[9,10] The application of **iperf3** is very common, and it is even known as "the industry standard tool and the most common tool to use for Ethernet throughput measurement."[11] However, it has a serious limitation. The measurement results may depend on both the performance of DUT and of the measurement environment, that is, the computers used for the execution of the **iperf** server and the **iperf** client. This limitation also applies to the results obtained by other methods, for example, by downloading a file and measuring its download time. This issue has been addressed by RFC 2544, which laid down the methodology for benchmarking network interconnect devices in an objective way. (See Section 2 for a short summary of the details.) Therefore, we have chosen **siitperf**,[12] a high-performance RFC 2544, RFC 4814,[13] RFC 5180, and RFC 8219 compliant measurement tool.

The aim of this current effort is to investigate if the two kinds of measurement setups provide the same or different results. Moreover, we examine how one can estimate the single DUT measurement results from the dual DUT measurement results, which could open up the possibility of benchmarking the five most important IPv4aaS technologies without special testers specific to each given technology. To that end, we make theoretical considerations and perform IPv4 packet forwarding and stateless IP/ICMP translation (SIIT)[14] performance measurements using both measurement setups and analyze the results. The significance of our current research is that it would be advantageous if one could use the dual DUT setup, which does not require a special-purpose tester and estimate the results of the single DUT setup, which characterize the performance of the DUT better.

The remainder of this paper is structured as follows: In Section 2, we summarize all necessary background information on the state-of-the-art benchmarking methodology defined by a series of IETF RFCs, as well as the basics of the operation of our testing tool and its improvements in a nutshell. In Section 3, we make different theoretical considerations about the equivalence and difference of the dual DUT setup and single DUT setup. In Section 4, we perform IPv4 packet forwarding tests and analyze our results. In Section 5, we make some more sophisticated SIIT measurements and analyze the results. In Section 6, we present our conclusions and highlight our plans for future research.

2 | BACKGROUND INFORMATION ON BENCHMARKING

In short, the aim of benchmarking is to produce reasonable and reproducible measurement data that can sufficiently characterize the performance of the tested device.

2.1 | Benchmarking methodology for network interconnect devices

RFC 2544[6] has specified all important aspects of the performance measurements of network interconnect devices. They include *test setup*, *types of measurements*, *measurement procedures*, and *frame sizes*.

In the simplest case, the measurement setup is made up of two devices, the *tester* and the *DUT*. They both have two network interfaces, which are connected so that the tester can send *test frames* through the DUT and receive them back.

The most important performance characteristic is the *throughput*. It is defined as the highest constant frame rate at which the DUT is able to forward all test frames without frame loss. It has to be measured using bidirectional traffic, and test trial must be at least 60 s long. In practice, throughput is usually measured using a binary search.

Latency is measured at the frame rate previously determined by the throughput measurement.

Frame loss rate is determined at various frame rates decrements in steps of a maximum 10% starting from the theoretical maximum frame rate for the media until there is no frame loss in two consecutive measurements.

There are standard frame sizes, with which the above measurements are performed, for example, 64, 128, and 256 bytes.

The appendix of the RFC also specifies the *test frame format*. If IP packets are forwarded, then the transport layer protocol should be UDP, and the source and destination port numbers have fixed values in the original version. This configuration was changed by RFC 4814,[13] which—among others—requires the usage of pseudorandom source and destination port numbers.

RFC 5180[7] extended the benchmarking methodology for devices using IPv6, but it excluded IPv6 transition technologies from its scope.

2.2 | Benchmarking methodology for IPv6 transition technologies

RFC 8219[5] has defined a comprehensive benchmarking methodology for IPv6 transition technologies. To that end, it has classified the high number of IPv6 transition technologies into a low number of categories regarding the solution they apply for access and core network traversal. These categories are:

1. Dual stack
2. Single translation
3. Double translation
4. Encapsulation

As dual stack means that both IPv4 and IPv6 protocol stacks are implemented, the dual stack devices do not need any new methodology: they can be benchmarked as described by RFC 2544 and RFC 5180.

Single translation technologies should be benchmarked according to the single DUT setup shown in Figure 1. Although the arrows are unidirectional, testing with bidirectional traffic is required and testing with unidirectional traffic is optional. IPv6 is used on the one side and IPv4 is used on the other. Care should be taken with the minimum and maximum frame sizes as frame size is changed due to the translation between IPv6 and IPv4. Otherwise, the situation is fairly simple, when, for example, a SIIT[14] gateway is benchmarked.

However, the single DUT setup can also be used for benchmarking one of the building blocks of a double translation or encapsulation technology solution. This task can be demanding regarding the abilities of the tester; for example, when the Address Family Transition Router (AFTR) of DS-Lite[15] is benchmarked, then, for example, on the left side, the tester sends IPv6 packets that contain encapsulated IPv4 packets and it receives IPv4 packets on the right side. As

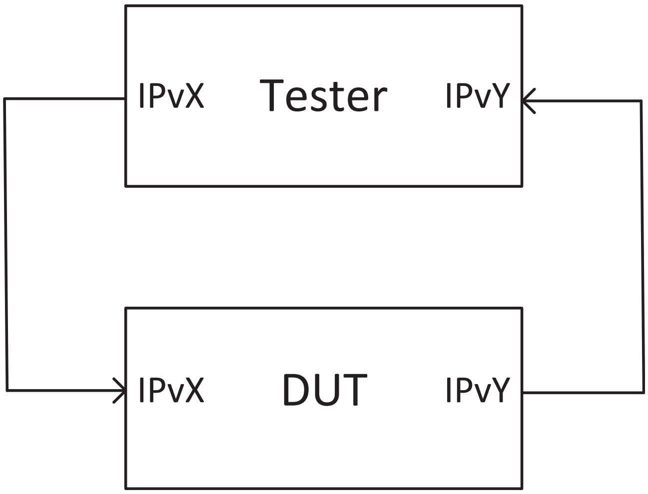


FIGURE 1 Single DUT test setup, where $X \neq Y$, $X, Y \in \{4, 6\}$.[5]

for the traffic in the reverse direction, the tester sends IPv4 packets on the right side and receives IPv6 packets that contain encapsulated IPv4 packets on the left side. And the traffic is different when the B4 element of DS-Lite is benchmarked. Again, for benchmarking the customer edge (CE) and border relay (BR) elements of MAP-E[16] or MAP-T,[17] a technology-specific traffic has to be used. It means that a special-purpose tester is needed for benchmarking each element of every single IPv6 transition technology. Using the dual DUT setup can mitigate this problem (see explanation below).

Double translation and encapsulation technologies should be benchmarked according to the dual DUT setup shown in Figure 2. In the case of the five aforementioned IPv4aaS technologies, it means that the tester has to send and receive only IPv4 traffic. In the case of 464XLAT and MAP-T, the IPv4 packets are translated to IPv6 packets and then back to IPv4 packets. In the case of the other three IPv4aaS technologies, the IPv4 packets are encapsulated into IPv6 packets, and they are then de-capsulated. We note that in the case of some other IPv6 transition technologies, the tester may need to send and receive IPv6 packets that are encapsulated into IPv4 packets by the given IPv6 transition technology which are then de-capsulated.

As for the benchmarking procedures, RFC 8219 has adopted the throughput and frame loss rate measurement procedures of RFC 2544 without any change. Thus, these tests can be performed according to the dual DUT setup using legacy RFC 2544 or RFC 5180 compliant testers.

RFC 8219 has redefined the *latency* benchmarking procedure to use at least 500 timestamps (instead of a single one) to achieve more precise results; thus, strictly speaking, that measurement cannot be performed using a legacy tester.

2.3 | RFC 8219 compliant testers

To the best of our knowledge, our **siitperf** is the world's first free software RFC 8219 compliant SIIT tester implementation. Its original version literally followed the test frame format described in the appendix of RFC 2544 including the fixed port numbers.[12] It was then enabled to use RFC 4814 pseudorandom port numbers.[18] Its most current version can also be used for benchmarking stateful NAT44/NAT64 gateways.[19] Its core business logic was written in C/C++ using DPDK to ensure high enough performance.

From the very beginning, **siitperf** was designed to be a flexible tool for research purposes. Therefore, it uses parameters even where the RFCs recommend constants. For example, RFC 2544 requires testing first with a single IP address pair and then with 256 destination networks, but **siitperf** allows any number of destination networks to be set from 1 to 256. It also allows the setting of the same IP version at both sides, meaning it can be used to measure IPv4 or IPv6 packet forwarding performance, too.

Our team is working on the implementation of other RFC 8219 compliant testers, for example, the design of a MAP-T tester was reported in Al-hamadani and Lencse.[20]

3 | THEORETICAL CONSIDERATIONS OF EQUIVALENCE OR DIFFERENCE OF THE TWO SETUPS

In this section, we make various considerations regarding what factors may cause a difference between the results produced using the single or the dual DUT setup.

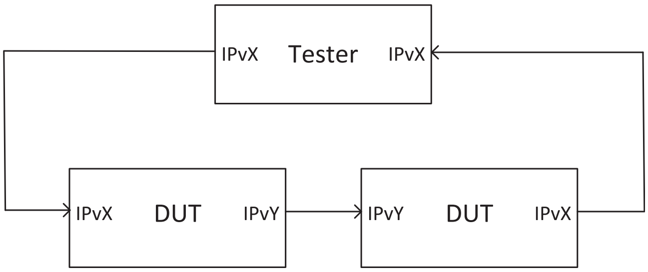


FIGURE 2 Dual DUT test setup, where $X \neq Y$, $X, Y \in \{4, 6\}$.[5]

3.1 | Asymmetry of the operation of the two DUTs

As for the asymmetry of the operation of the two DUTs, we have actual experience with the Jool implementation of the 464XLAT and MAP-T technologies.[21] We measured the packet forwarding performance of the CLAT and PLAT components of 464XLAT together, and then, we repeated the measurement by logging the CPU utilization of the two devices to discover which was causing the bottleneck. It was the PLAT, which was accurately reflected in our results. We also measured the packet forwarding performance of the CE and BR components of MAP-T together and then used the same method to find the bottleneck, which was the CE. Thus, this measurement system was found unsuitable for measuring the performance of the BR component. This meant a significant limitation for us as we were interested in the scalability of the BR, but we were unable to measure it: We found only a lower bound of it. (In a real-world application scenario, one provider-side device serves a high number of client-side devices. This is why we are usually interested in the performance and scalability of the provider-side devices.)

3.2 | Different number of routing table entries

In the simplest case, when routers are benchmarked according to the single DUT setup, the router needs to have at least two entries in its routing table (one for each network); however, if two routers are benchmarked together, the routers need to have at least three entries (again, one for each network, see Section 4.4). The situation is similar, when more than one destination networks are used. (See Section 4.4 for the details and the performance consequences of the difference.)

3.3 | The issue of the identical devices

In theory, the information technology (IT) industry produces completely identical devices, including computer hardware. For example, if buying two instances of a given type of CPU, they should have exactly the same performance. This applies to all other devices, too. However, RFC 2544 throughput is defined as the highest frame rate at which the device can forward all packets without frame loss. In practice, the measured throughput can be different, even if theoretically identical devices are used. (See Section 4.3 for our results demonstrating the issue.)

3.4 | Question of expected results

Let us suppose that we know some performance characteristics of a device measured according to the single DUT setup. Let it be just a router for simplicity. If we neglect all the aforementioned issues, what performance characteristics do we expect, if two of them are benchmarked together?

For example, the throughput should be roughly the same. The latency measured using the dual DUT setup should be the double of the latency measured using the single DUT setup. But what about the frame loss rate? Should it be the same or the double? Our prediction depends on the model we have in mind. For example:

1. If frame loss is caused by some random events, then we expect the frame loss rate to be doubled.
2. If frame loss is caused by the exhaustion of the processing capacity, the frame loss should be the same.

With these uncertainties, it is not easy to make an accurate prediction even when estimating the results of the dual DUT setup on the basis of the results of the single DUT setup. And our aim is usually the opposite.

4 | IPv4 PACKET FORWARDING MEASUREMENTS

The primary aim of these measurements was to check how the higher number of routing table entries influence the performance of the dual DUT setup compared to the single DUT setup. As a preliminary test, we also compared the IPv4 packet forwarding performance of three theoretically identical devices with the aim of detecting potential differences and to also select the two that had a similar performance.

4.1 | Hardware and software environment

The measurements were done at the University of Debrecen using the equipment of the Faculty of Informatics.

As for the tester device, we used a computer with the following relevant parameters: Asustek ROG STRIX Z390-H GAMING motherboard, Intel Core i7-9700KF CPU, 4 × 16 GB 1333 MHz DDR4 RAM, and dual-port Intel 82599ES 10GbE network interface card (NIC).

In order to save on the power budget, we switched off the last two of the eight cores of the CPU using the following commands:

```
echo 0 > /sys/devices/system/cpu/cpu6/online
echo 0 > /sys/devices/system/cpu/cpu7/online
```

As a consequence, the remaining CPU cores could operate at 5.1 GHz clock frequency during our measurements.

We used Ubuntu 18.04.6 LTS operating system with 4.15.0-197-generic Linux kernel. The DPDK version was: 17.11.10-0ubuntu0.2. We reserved 16 huge pages of 1 GB size and CPU Cores 2, 3, 4, and 5 for **siitperf** using the following kernel command line parameters:

```
isolcpus=2,3,4,5 hugepagesz=1G hugepages=16
```

We note that we used two of the reserved CPU cores for sending test frames (one in each direction) and the other two for receiving test frames (one in each direction).

As for the DUTs, we used a computer with the following relevant parameters: Gigabyte Technology Z77-D3H motherboard, Intel Core i7-3770K CPU, 4 × 4 GB 800 MHz DDR3 RAM, and dual-port Intel 82599ES 10GbE NIC.

In order to ensure stable and reliable results, we disabled hyper threading and set the clock frequency of the four CPU cores to fixed 2.5 GHz with the help of the **tlp** Linux package using the **idle** = **poll** kernel command line parameters and the following settings in the **/etc/tlp.conf** file:

```
CPU_SCALING_GOVERNOR_ON_AC=performance
CPU_SCALING_MIN_FREQ_ON_AC=2500000
CPU_SCALING_MAX_FREQ_ON_AC=2500000
```

We used Debian 11.4 operating system with 5.10.0-16-amd64 Linux kernel.

4.2 | Self-test of the tester

Before performing any actual tests, we executed a self-test of the tester to exclude the possibility that its insufficient performance influences the results of our measurements. To that end, the two 10 Gbps Ethernet network interfaces of the tester were looped back with a direct cable as shown in Figure 3.

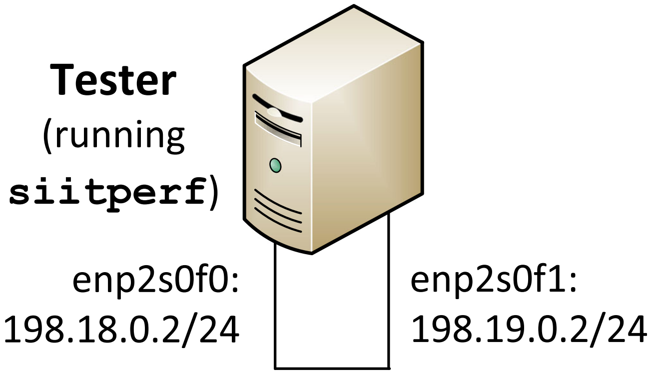


FIGURE 3 Measurement setup for the self-test of the tester.

We performed an IPv4 throughput measurement using 64-byte frame size and bidirectional traffic. The binary search was executed 20 times, and the measured throughput is shown in Table 1. We note that other Network Performance Testers usually report the total number of forwarded frames per second, but **siitperf** reports *the number of frames per second per direction*, so the results are to be understood *per direction*. Our results show that the tester had a high enough performance and its results were stable.

We did not test any other frame sizes, because we always generated 64 bytes long frames with **siitperf** in all our tests.

4.3 | IPv4 throughput of the DUTs

We measured the IPv4 packet forwarding performance of the three identical DUTs according to the single DUT test setup shown in Figure 4. The value of x was always 0 in these measurements, x will have a role in Section 4.4. The 10 Gbps Ethernet network interfaces were interconnected with direct cables.

TABLE 1 Self-test results of the tester using bidirectional traffic, IPv4, and 64-byte frame size.

Average (fps/direction)	7,177,245
Minimum (fps/direction)	7,170,898
Maximum (fps/direction)	7,184,569

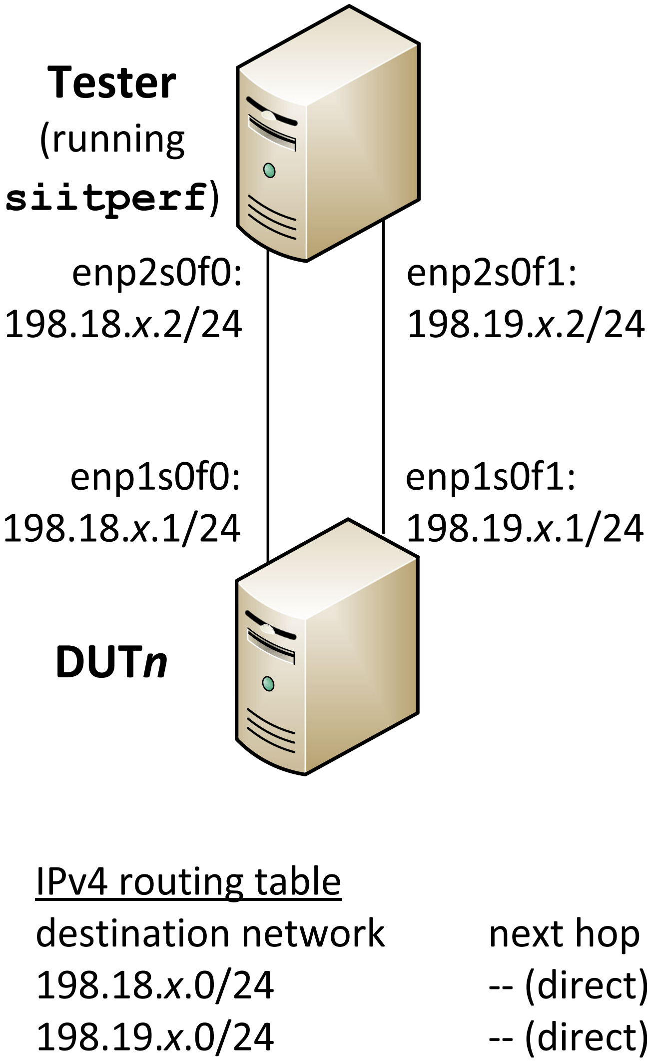


FIGURE 4 Single DUT test setup for IPv4 packet forwarding rate measurements.

There are two further important details worth mentioning: receive-side scaling and non-zero-loss acceptance criterion.

The aim of *receive-side scaling*[22] (also called *multi-queue receiving*) is that the packet receiving performance of the system increases nearly linearly with the number of its CPU cores. To that end, the interrupts of the incoming packets are distributed among the CPU cores of a contemporary multi-core CPUs. To facilitate this, modern NICs are able to compute a hash value from the *four tuple* (source IP address, destination IP address, source port number, and destination port number) of the incoming packets and use it to assign the packets to the receive queues and their interrupts to the CPU cores. As in our current tests, we always used the same IP addresses, only the port numbers could ensure entropy for the hash function. By default, only the source and destination IP addresses are used to compute the hash value; therefore, we had to set the proper hash computing by the following commands on the DUT1, DUT2, and DUT3 devices:

```
ethtool -N enp1s0f0 rx-flow-hash udp4 sdfn
ethtool -N enp1s0f1 rx-flow-hash udp4 sdfn
```

We note that no such settings were used with the tester because DPDK uses poll mode driver and a single core is able to handle more than seven million packets as we measured before. (But their IPv6 equivalents were used for the IPv6 interfaces during SIIT testing.)

RFC 2544 defines throughput as the highest rate at which the DUT can forward *all* packets. However, commercial Network Performance Testers usually have a parameter called *loss tolerance* and it can be set to a value higher than 0. During our preliminary measurements, we observed that sometimes the binary search failed due to a low number of missing test frames. Even though using the zero-loss criterion meant that the difference of the minimum and maximum of the results was only a few percent of the median; nevertheless, very much consistent results were needed to be able to point out small differences; therefore, 0.01% loss tolerance was used for our measurements. (Namely, a single frame may be missing from every 10,000 test frames.) Earlier the 99.99% acceptance criterion was used for benchmarking SIIT implementations[23] and authoritative DNS servers,[24] and we also analyzed the issue in Lencse et al.[25]

Our results are shown in Table 2. Here, the [minimum, maximum] interval of the results of DUT1 does not even overlap with those intervals of DUT2 and DUT3. Thus, we successfully demonstrated the case when the results of theoretically identical hardware produce different results. For our further tests, we selected DUT2 and DUT3 as two devices that produce sufficiently similar results.

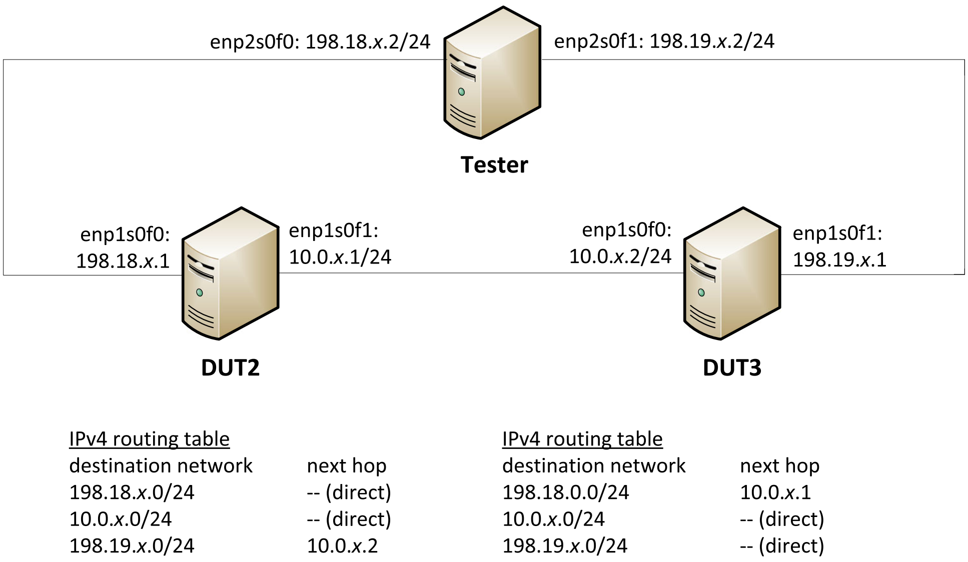
4.4 | Comparison of the results of the single DUT setup and of the dual DUT setup

We used the topology shown in Figures 4 and 5 for the single DUT setup measurements and for the dual DUT setup measurements, respectively. In both cases, *x* was used to express the different number of destination networks. The tests were performed using 1, 2, 4, 8, 16, 32, 64, 128, and 256 destination networks. It can also be seen from the two figures that in the case of the dual DUT setup, the routing tables contained 50% more rules than in the case of the single DUT setup.

The results are shown in Figure 6. The color bars show the median values, and the error bars show the minimum and maximum values. (The error bars are usually hardly visible due to the consistent nature of the results of the 20 repetitions of measurements.) The results completely support our theoretical consideration in Section 3.2: 50% more routing table entries causes lower performance in the case of the dual DUT setup compared to the results of the single DUT setup with the same number of destination networks, and the doubling of the number of destination networks decreases further the performance of the system.

TABLE 2 IPv4 packet forwarding performance using bidirectional traffic, 64-byte frame size, and 0.01% loss tolerance.

Which DUT?	DUT1	DUT2	DUT3
Median (fps/direction)	2,005,495	1,980,792	1,979,037
Minimum (fps/direction)	1,991,307	1,976,377	1,972,637
Maximum (fps/direction)	2,016,670	1,982,971	1,985,908



IPv4 routing table

destination network	next hop
198.18.*x*.0/24	-- (direct)
10.0.*x*.0/24	-- (direct)
198.19.*x*.0/24	10.0.*x*.2

IPv4 routing table

destination network	next hop
198.18.0.0/24	10.0.*x*.1
10.0.*x*.0/24	-- (direct)
198.19.*x*.0/24	-- (direct)

FIGURE 5 Dual DUT test setup for IPv4 packet forwarding rate measurements.

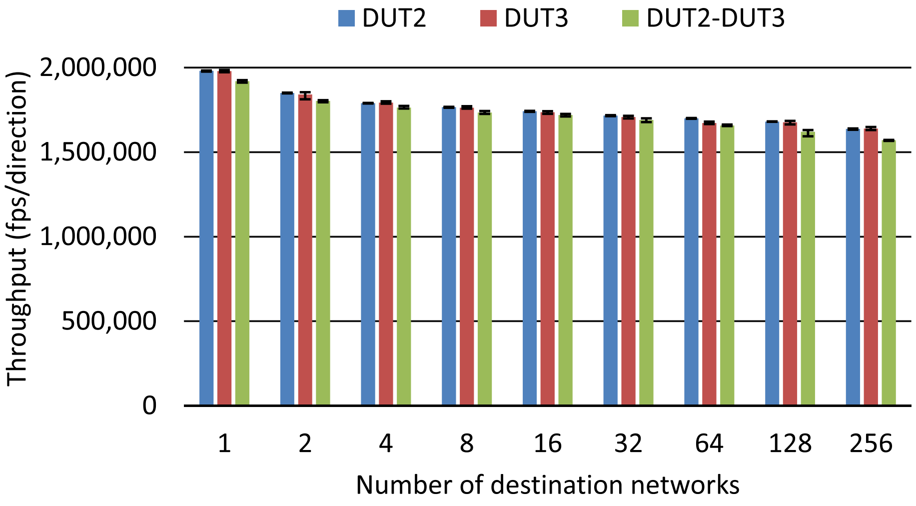


FIGURE 6 IPv4 packet forwarding performance using bidirectional traffic, 64-byte frame size, and 0.01% loss tolerance.

5 | MEASUREMENTS WITH SIIT GATEWAYS

SIIT is a fairly simple IPv6 transition technology. With an appropriate trick, we could measure its throughput and frame loss rate characteristics using an RFC 5180 compliant tester.[26] Now, we can also measure the *latency* of SIIT gateways using **siitperf**.

5.1 | Test setups

We used DUT2 and DUT3 to measure their performance characteristics one by one according to the single DUT setup and together according to the dual DUT setup. For an easy-to-follow presentation, first, we disclose our dual DUT setup, and then, we derive the two single DUT setups from it.

Figure 7 shows the dual DUT setup for benchmarking SIIT gateways. From the tester's point of view, the setup is exactly the same as shown in Figure 5. The tester is not even aware of the fact that the traffic between DUT2 and DUT3 is IPv6. The two DUTs use Explicit Address Mapping[27] between several IPv4 and IPv6 prefixes as shown in the middle of Figure 7. (As before, *x* may mean different number of networks in the IPv4 addresses. The role of *xx* is the same in

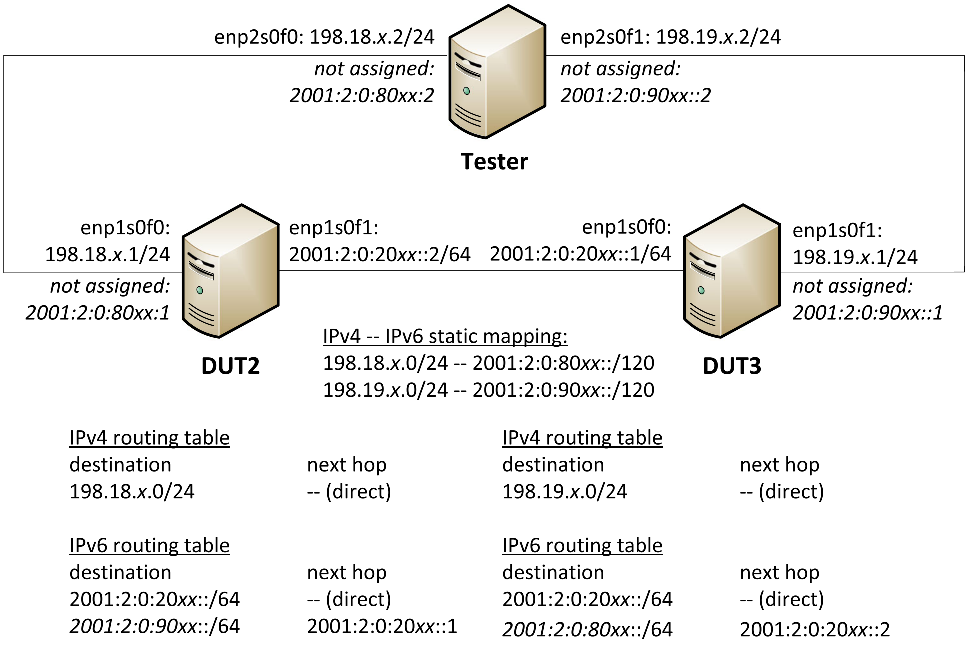


FIGURE 7 Dual DUT test setup for benchmarking SIIT gateways, DUT2 and DUT3 implement stateless NAT46 and stateless NAT64, respectively.

```
/sbin/modprobe jool_siit
jool_siit instance add "benchmarking" --netfilter
for (( i=0; i<$1; i++ ))
do
    H=$(printf "%.2x" $i)
    jool_siit -i "benchmarking" eamt add 2001:2:0:80$H::/120 198.18.$i.0/24
    jool_siit -i "benchmarking" eamt add 2001:2:0:90$H::/120 198.19.$i.0/24
done
jool_siit -i "benchmarking" eamt display
```

FIGURE 8 Script for setting Jool for dual DUT measurements.

the IPv6 addresses.) The IPv4 or IPv6 addresses that are actually assigned to the network interfaces are always written above the lines. The IPv6 peers of the IPv4 addresses that are *not assigned* to the interfaces are written below the line using an italic font to show how IPv4 the interfaces are represented in the IPv6 address space.

SIIT was implemented by Jool.[28] Its version 4.1.5 was used. We performed all the necessary settings by scripts. As a sample, we included the script one for Jool settings in Figure 8. It took a single parameter, the number of networks, and it set everything for Jool. We set the IP addresses and the routing rules also by similar scripts.

The two single DUT setups were derived from the dual DUT setup shown in Figure 7 in a very simple way. First, DUT3 was removed, and DUT2 was benchmarked as a stateless NAT46 translator as shown in Figure 9. Then, DUT2 was removed from the dual DUT setup, and DUT3 was benchmarked as a stateless NAT64 translator as shown in Figure 10. As for Jool, the same script was used as shown in Figure 8, and the other scripts were also derived from those of the dual DUT setup in a natural way.

5.2 | Throughput

As elaborated in Section 4.3, we used 0.01% loss tolerance. We compared the throughput of the three test systems in Figure 11. The results are shown as a function of the number of destination networks. Compared to Figure 6, three salient differences can be observed here:

1. The throughput results are significantly lower.

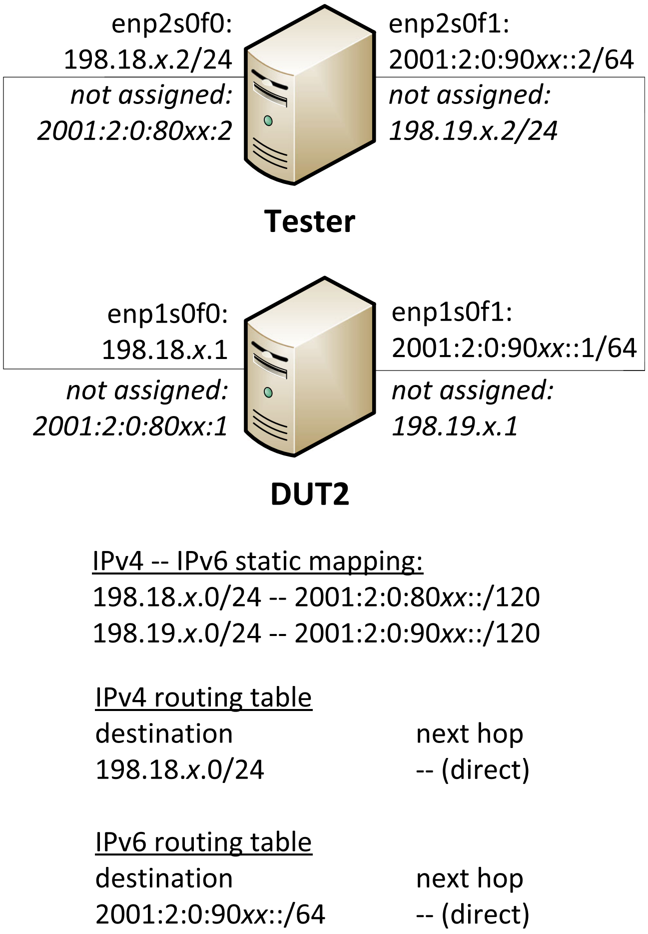


FIGURE 9 Single DUT test setup for benchmarking SIIT gateways, DUT2 implements stateless NAT46.

2. The performance degradation caused by the increasing number of destination networks is significantly higher.
3. From about 32 destination networks, the difference between the single DUT results and the dual DUT results disappears.

The first one has a very simple explanation: SIIT is a more computation intensive task than IPv4 packet forwarding. (Since a new, different version IP packet header has to be built.)

The second one reveals a scalability problem of Jool. (At 256 destination networks, Jool had to store and handle 512 IPv4–IPv6 mappings. Perhaps the lookup of the mappings is not implemented efficiently.)

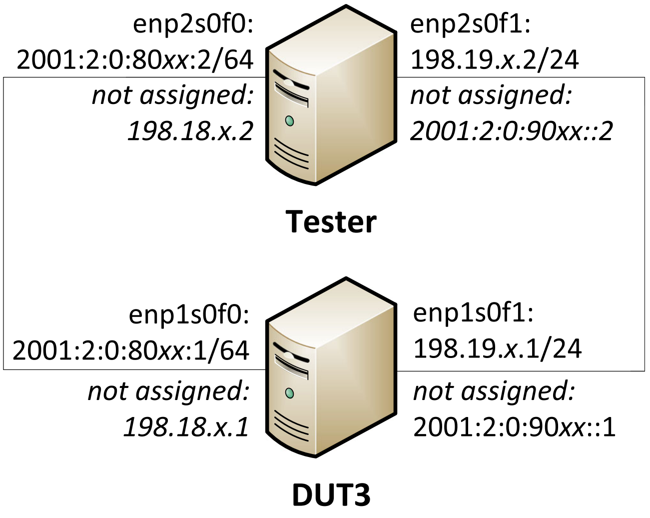
The third most probably happened because the computation cost of the routing decisions became negligible compared to that of the SIIT operation.

All in all, our results show that the throughput of SIIT gateways can be approximately determined by benchmarking two identical SIIT gateways together according to the dual DUT setup using an RFC 2544 compliant legacy Network Performance Tester.

5.3 | Latency

We measured the latency according to the single DUT setup for DUT2 and DUT3 individually and also together according to the dual DUT setup.

We calculated the typical latency for both directions according to Section 7.2 of RFC 8219. We note that **siitperf** calls the direction “forward,” if it follows the direction of the arrows in Figure 1, and it calls the opposite direction “reverse.” The typical latencies in the forward and in the reverse direction are shown in Figures 12 and 13, respectively.



IPv4 -- IPv6 static mapping:
198.18.*x*.0/24 -- 2001:2:0:80*xx*::/120
198.19.*x*.0/24 -- 2001:2:0:90*xx*::/120

IPv4 routing table

destination	next hop
198.19.*x*.0/24	-- (direct)

IPv6 routing table

destination	next hop
2001:2:0:80*xx*::/64	-- (direct)

FIGURE 10 Single DUT test setup for benchmarking SIIT gateways, DUT3 implements stateless NAT64.

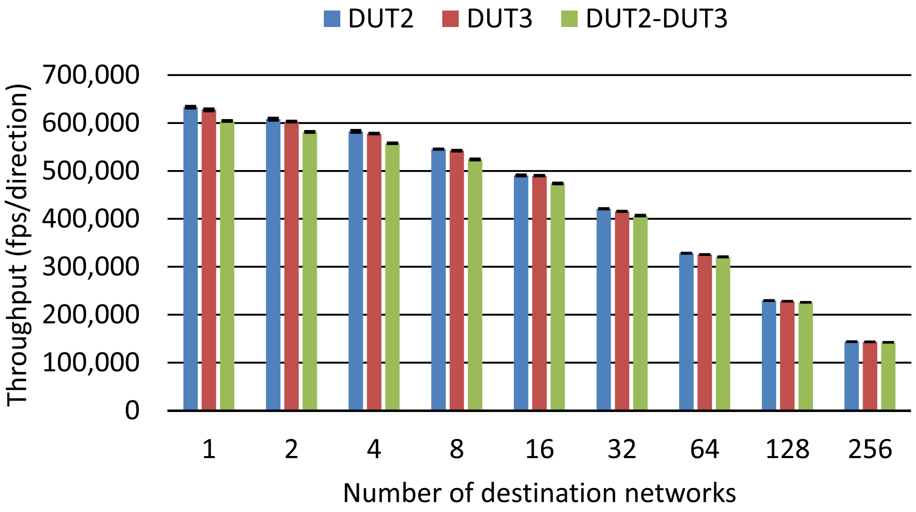


FIGURE 11 SIIT packet forwarding performance using bidirectional traffic, 64/84-byte frame size, and 0.01% loss tolerance.

From one to four destination networks, the results comply with our expectations that the latency measured according to the dual DUT setup is approximately the double of the latencies measured according to the single DUT setup. Yet from eight destination networks, the latency of the dual DUT setup becomes systematically higher. Finding its root cause is beyond the scope of our current paper.

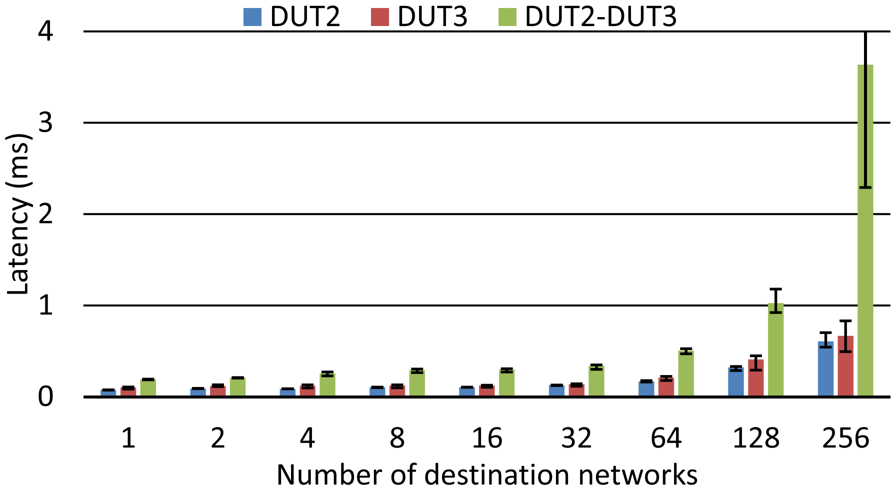


FIGURE 12 SIIT typical latency in the “forward” direction using bidirectional traffic and 64/84-byte frame size.

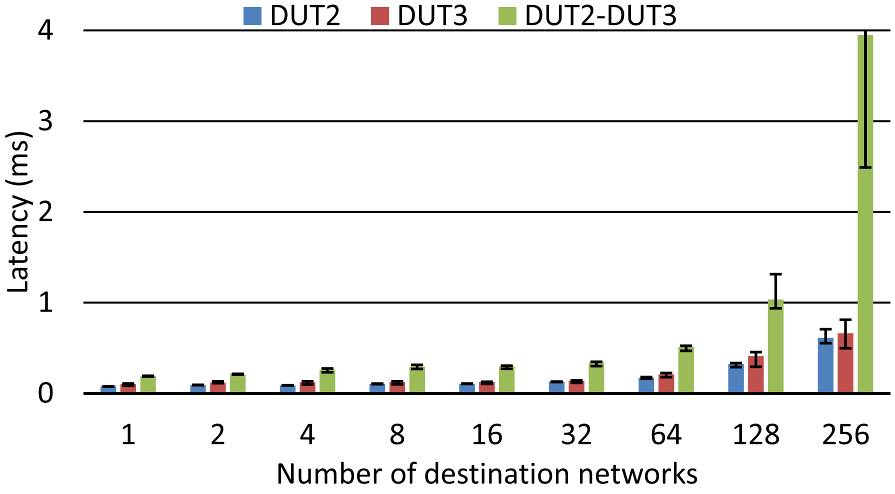


FIGURE 13 SIIT typical latency in the “reverse” direction using bidirectional traffic and 64/84-byte frame size.

6 | CONCLUSION AND FUTURE WORK

We have pointed out several potential differences that can prevent the use of the dual DUT setup instead of the single DUT setup (asymmetry of the corresponding functions, difference in the number of routing rules, and difference in the performance of the theoretically identical hardware). We have shown that in the case of two IPv4 routers and also in the case of two SIIT gateways, the throughput results of the dual DUT setup could approximate well that of the single DUT setup. However, the latency did not always follow the theoretical expectations.

Based on the results of the current paper, we plan to measure the performance and scalability of various implementations of the five IPv4aaS technologies using the method elaborated in Lencse and Bazsó.[29] In short, this method handles the aggregate of the CE and provider edge (PE) devices of the given IPv4aaS technology together. This aggregate is equivalent to a stateful NAT44 gateway that can be benchmarked according to our new methodology defined in Lencse and Shima[30] and validated in Lencse et al.[31] The stateful extension of **siitperf**[19] can be used to carry out the benchmarking measurements. As a demonstration and validation of our proposed measurement method for benchmarking the IPv4aaS technologies according to the dual DUT setup of RFC 8219, we also compared the scalability of the Jool implementation of 464XLAT and MAP-T technologies in Lencse and Bazsó.[29]

ACKNOWLEDGMENTS

The authors would like to thank Bertalan Kovács, Sándor Répás, Norbert Nagy, and Ádám Bazsó for reading and commenting on the manuscript. The authors thank Natasha Bailey, Széchenyi István University, for the English language proofreading of the manuscript.

CONFLICT OF INTEREST STATEMENT

On behalf of all authors, the corresponding author states that there is no conflict of interest.

DATA AVAILABILITY STATEMENT

Research data are not shared. (The data produced are the measurement results available in the paper.)

ORCID

Gábor Lencse https://orcid.org/0000-0001-5552-3237

REFERENCES

1. Nikkhah M, Guérin R. Migrating the internet to IPv6: an exploration of the when and why. *IEEE/ACM Trans Netw*. 2016;24(4):2291-2304. doi:10.1109/TNET.2015.2453338
2. Lencse G, Kadobayashi Y. Comprehensive survey of IPv6 transition technologies: a subjective classification for security analysis. *IEICE Trans Commun*. 2019;E102.B(10):2021-2035. doi:10.1587/transcom.2018EBR0002
3. Palet Martinez J, Liu HM-H, Kawashima M. Requirements for IPv6 customer edge routers to support IPv4-as-a-Service. IETF RFC 8585. 2019. 10.17487/RFC8585.
4. Lencse G, Palet Martinez J, Howard L, Patterson R, Farrer I. Pros and cons of IPv6 transition technologies for IPv4-as-a-Service (IPv4aaS). IETF RFC 9313. 2022. 10.17487/RFC9313.
5. Georgescu M, Pislaru L, Lencse G. Benchmarking methodology for IPv6 transition technologies. IETF RFC 8219. 2017.
6. Bradner S, McQuaid J. Benchmarking methodology for network interconnect devices. IETF RFC 2544. 1999. 10.17487/RFC2544.
7. Popoviciu C, Hamza A, de Velde GV, Dugatkin D. IPv6 benchmarking methodology for network interconnect devices. IETF RFC 5180. 2008. 10.17487/RFC5180.
8. Kovács Á. Evaluation of the aggregation capability of the MPT network layer multipath communication library and multipath TCP. *Acta Polytech Hung*. 2019;16(6):129-147. doi:10.12700/APH.16.6.2019.6.9
9. Al-Imareen N, Lencse G. Effect of path QoS on throughput aggregation capability of the MPT network layer multipath communication library. *Infocommun J*. 2023;15(2):14-20. doi:10.36244/ICJ.2023.2.3
10. Al-Imareen N, Lencse G. On the impact of packet reordering in MPT-GRE multipath networks. In: *2023 46th International Conference on Telecommunications and Signal Processing (TSP), Virtual Conference*. IEEE; 2023:82-86. doi:10.1109/TSP59544.2023.10197737
11. Microchip. Using Linux iPerf to assess Ethernet performance. 2023. Microchip Technical Support Portal. 000014855. [Online]. Available: https://microchipsupport.force.com/s/article/Using-Linux-iPerf-to-assess-Ethernet-performance
12. Lencse G. Design and implementation of a software tester for benchmarking stateless NAT64 gateways. *IEICE Trans Commun*. 2021; E104-B(2):128-140. doi:10.1587/transcom.2019EBN0010
13. Newman D, Player T. Hash and stuffing: overlooked factors in network device benchmarking. IETF RFC 4814. 2008. 10.17487/RFC4814.
14. Bao C, Li X, Baker F, Anderson T, Gont F. IP/ICMP translation algorithm. IETF RFC 7915. 2016. 10.17487/RFC7915.
15. Durand A, Droms R, Woodyatt J, Lee Y. Dual-stack lite broadband deployments following IPv4 exhaustion. IETF RFC 6333. 2011. 10.17487/RFC6333.
16. Troan O, Dec W, Li X, et al. Mapping of address and port with encapsulation (MAP-E). IETF RFC 7597. 2015. 10.17487/RFC7597.
17. Li X, Bao C, Dec W, Troan O, Matsushima S, Murakami T. Mapping of address and port using translation (MAP-T). IETF RFC 7599. 2015. 10.17487/RFC7599.
18. Lencse G. Adding RFC 4814 random port feature to siitperf: design, implementation and performance estimation. *Int J Adv Telecommun Electrotech Signals Syst*. 2020;9(3):18-26. doi:10.11601/ijates.v9i3.291
19. Lencse G. Design and implementation of a software tester for benchmarking stateful NATxy gateways: theory and practice of extending siitperf for stateful tests. *Comput Commun*. 2022;172(1):75-88. doi:10.1016/j.comcom.2022.05.028
20. Al-hamadani A, Lencse G. Towards implementing a software tester for benchmarking MAP-T devices. *Infocommun J*. 2022;14(3):45-54. doi:10.36244/ICJ.2022.3.6
21. Lencse G, Nagy N. Towards the scalability comparison of the Jool implementation of the 464XLAT and of the MAP-T IPv4aaS technologies. *Int J Commun Syst*. 2022;35(18):e5354. doi:10.1002/dac.5354
22. Herbert T, de Bruijn W. Scaling in the Linux networking stack. n.d. [Online]. Available: https://www.kernel.org/doc/Documentation/networking/scaling.txt
23. Lencse G, Shima K. Performance analysis of SIIT implementations: testing and improving the methodology. *Comput Commun*. 2020; 156(1):54-67. doi:10.1016/j.comcom.2020.03.034
24. Lencse G. Benchmarking authoritative DNS servers. *IEEE Access*. 2020;8:130224-130238. doi:10.1109/ACCESS.2020.3009141

25. Lencse G, Kovács Á, Shima K. Gaming with the throughput and the latency benchmarking measurement procedures of RFC 2544. *Int J Adv Telecommun Electrotech Signals Syst*. 2020;9(2):10-17. doi:10.11601/ijates.v9i2.288
26. Lencse G. Benchmarking stateless NAT64 implementations with a standard tester. *Telecommun Syst*. 2020;75(3):245-257. doi:10.1007/s11235-020-00681-x
27. Anderson T, Potter AL. Explicit address mappings for stateless IP/ICMP translation. IETF RFC 7757. 2016. 10.17487/RFC7757.
28. NIC Mexico. Jool: SIIT and NAT64. n.d. [Online]. Available: https://www.jool.mx/en/about.html
29. Lencse G, Bazsó Á. Benchmarking methodology for IPv4aaS technologies: comparison of the scalability of the Jool implementation of 464XLAT and MAP-T. *Comput Commun*. 2024;219:243-258. doi:10.1016/j.comcom.2024.03.007
30. Lencse G, Shima K. Benchmarking methodology for stateful NATxy gateways using RFC 4814 pseudorandom port numbers. 2024. InternetDraft. https://datatracker.ietf.org/doc/html/draft-ietf-bmwg-benchmarking-stateful
31. Lencse G, Shima K, Cho K. Benchmarking methodology for stateful NAT64 gateways. *Comput Commun*. 2023;210:256-272. doi:10.1016/j.comcom.2023.08.009

How to cite this article: Lencse G, Szilágyi S. Equivalence and difference of the dual device under test setup and the single device under test setup of RFC 8219. *Int J Commun Syst*. 2024;e5982. doi:10.1002/dac.5982