Empirical Analysis of IPv6 Transition Technologies Using the IPv6 Network Evaluation Testbed

Marius Georgescu(✉), Hiroaki Hazeyama, Youki Kadobayashi, and Suguru Yamaguchi

Nara Institute of Science and Technology, Nara, Japan
{liviumarius-g,hiroa-ha,suguru}@is.naist.jp, youki-k@is.aist.nara.ac.jp
http://ipv6net.ro/

Abstract. IPv6 is yet to become more than a worthy successor of IPv4, which remains, for now, the dominant Internet Protocol. Behind this fact is the complicated transition period through which the Internet will have to go, until IPv6 will completely replace IPv4. This transition has presented the Internet Community with numerous challenges. One of these challenges is to decide which transition technology is more feasible for a particular network scenario. As an answer, this article is proposing the IPv6 Network Evaluation Testbed (IPv6NET), a research project whose ultimate goal is to obtain feasibility data in order to formulate a coherent, scenario-based IPv6 transition strategy. The paper presents the overview of IPv6NET, the testing methodology and empirical results for a specific network scenario. The scenario was introduced by the IETF and it was dedicated to an Enterprise Network which is using IPv6 as backbone technology. The Enterprise needs to convey communication tjo IPv4 capable nodes through the IPv6-only infrastructure. A suitable IPv6 transition implementation, covering multiple transition technologies, was tested in relation with this scenario. The presented empirical feasibility data includes network performance data such as: latency, throughput, packet loss, CPU load, and operational capability data, such as: configuration, troubleshooting and applications capability.

Keywords: IPv6 transition · IETF IPv6 scenario · 464 scenario · Enterprise Networks · IPv6NET · Asamap · MAPe, MAPt · DSLite · 464XLAT

1 Introduction

The Internet community found in IPv6 an answer for the continual expansion of the Internet, threatened by the limitations of IPv4. IPv6 uses an 128 bit address, extending the address space to $2^{128} \approx 3.4 \cdot 10^{38}$ unique IP addresses, enough for many years to come. However the light aura of IPv6 has dimmed since 1998, mainly because it is not able to communicate directly with its predecessor, IPv4. This introduced the Internet Community with a great challenge, usually called

V.C.M. Leung et al. (Eds.): TridentCom 2014, LNICST 137, pp. 216–228, 2014.
DOI: 10.1007/978-3-319-13326-3_21

the transition to IPv6. The transition is represented by the stages the Internet will have to withstand until IPv6 will completely replace IPv4.

Given the complexity of the current IPv4-dominated Internet, the Transition to IPv6 will be a long and complex process. So far, only a small number of production networks are IPv6 capable. The APNIC Labs IPv6 deployment report shows that only about 1.7 % of the users worldwide are currently using IPv6. IPv6 transition scenarios have been researched within the IETF by the v6ops and Softwire Working Groups. The scenarios were dedicated to four main types of networks: ISP Networks, Enterprise Networks, 3GPP Networks and Unmanaged Networks. The IETF ngtrans Working Group has made many efforts to propose and analyze viable transition mechanisms. Many transition mechanisms have been proposed and implemented. All have advantages and disadvantages considering a certain transition scenario, but no transition mechanism can be considered most feasible for all the scenarios. This opens many research opportunities. One of them is a scenario-based analysis of IPv6 transition implementations, and represents the ultimate goal of our research.

In this paper, we are proposing the IPv6 Network Evaluation Testbed (IPv6NET), which is dedicated to measuring the feasibility of transition mechanisms in a series of scenario-based network tests. As a study case, the article is focusing on one of the scenarios introduced by the IETF for Enteprise Networks in [4], targeting enterprises using an IPv6-only network infrastructure but with IPv4-capable nodes, which need to communicate over the IPv6 infrastructure.

The paper is organized as follows: section 2 presents related literature, section 3 introduces the IPv6NET concept and the testing methodology, in section 4 the empirical results are introduced and the feasibility of the tested implementation is analyzed in relation with the specific scenario, section 5 discusses our approach and lastly section 6 states the conclusions and future work.

2 Related Work

There are a variety of articles dedicated to IPv6 transition experimental environments in current literature. They can be generally classified into closed environments and open environments. The closed environments are usually small scale, local environments, which are isolated from production networks or the Internet. In [12], the performance of Linux operating systems is evaluated in relation to an IPv4-v6 Configured Tunnel and a 6to4 Tunnel. Four workstations were employed to build the testbed. In [14], differences in bandwidth requirements for common network applications like: remote login, web browsing, voice communication, database transaction, and video streaming are analyzed over 3 types of networks: IPv4-only, IPv6-only and a 6to4 tunneling mechanism. The environment was built using the OPNET simulator. Also based on the OPNET simulator was the testbed presented in [7], which analyzed the performance of transition mechanisms over a MPLS backbone. A common trait of the above mentioned closed environments, is the thorough performance analysis, which resulted in quantifiable data like: CPU and memory utilization, throughput, end-to-end delay, jitter and execution time.

However, before transition mechanisms are applied in a large scale environment, a systematic and quantitative performance analysis should be performed. This gets us to the second group of experimental environments, namely, open environments. They can be defined as experimental networks connected to a large scale production network or the Internet. [2] describes the lessons learned from deploying IPv6 in Google's heterogeneous corporate network. The report presents numerous operational troubles like: the lack of dual-stack support of the customer-premises equipments (CPE), or the immature IPv6 support of operating systems and applications. One of their conclusions was that the IPv6 transition can affect every operational aspect in a production environment, hence interoperability considerations have to be made. In [1], experiences with IPv6-only Networks are presented. NAT64 and DNS64 technologies are tested in two open environments: an office and a home environment. Common applications like: web browsing, streaming, instant messaging, VoIP, online gaming, file storage and home control were tested. Application issues in relation to the NAT64/DNS64 technology are identified, for example: Skype's limitation to connect to IPv6 destinations, or the lack of network operational diagnostics for certain standalone games. Experiences with IPv6-only Networks from previous WIDE Camp events in [9] present many meaningful interoperability data such as IPv6 capability of OSes, applications and network devices. Many operational issues have been identified. Some examples are: long fall-back routine, low DHCPv6 capability of certain OSes, lack of IPv6 support in some network devices, DNS64 overload, inappropriate AAAA replies or inappropriate selection of DNS resolvers. Considering these examples we can conclude that open environment testing has the potential of exposing interoperability issues, which can otherwise get overlooked.

Combing the advantages of the two testing methods can lead to a complete feasibility analysis. Hence the IPv6NET project is considering both methods for testing.

3 Testing Methodology on IPv6NET

The IPv6 Network Evaluation Testbed (IPv6NET) is dedicated to quantifying the feasibility of IPv6 transition implementations in relation to a specific network scenarios. IPv6NET has two main components: the testing component and the infrastructure component. The testing component has the following building blocks: a specific network scenario, an associated network template and a test methodology. The infrastructure component is represented by the implementations under test and the network test environment. As mentioned, we are considering building both closed and open environments.

The scenario targeted in this article was introduced by the IETF in [4] as Scenario 3. It is dedicated to an enterprise which decided to use IPv6 as the main protocol for network communications. Some applications and nodes, which are IPv4-capable would need to communicate over the IPv6 infrastructure. In order to achieve this, the Enterprise would need to apply an IPv6 transition technology, which would allow both protocols to coexist in the same environment. For simplicity, the technologies suitable for this specific scenario could be referred to as 464 technologies.

3.1 IPv6NET Feasibility Indicators and Metrics

This subsection presents some clarifications regarding the semantics used for the methodology associated with IPv6NET throughout this paper. For the empirical feasibility analysis presented in this article, we are using the term *feasibility indicator* as a generic classifier for performance metrics. For closed environment testing, the proposed feasibility indicator was *network performance*. Network performance indicates the technical feasibility of each technology in relation with existing computer network standards. To quantify network performance, we have used well established metrics, such as : *round-trip-delay, jitter, throughput, packet loss* and *CPU load*. For open-environment testing, we have proposed as feasibility indicator *operational capability*, which is showing how a certain technology fits in with the existing environment or how it manages to solve problems. To our best knowledge, there are no associated metrics for operational feasibility of network devices in current literature. Consequently we have introduced the following three metrics:

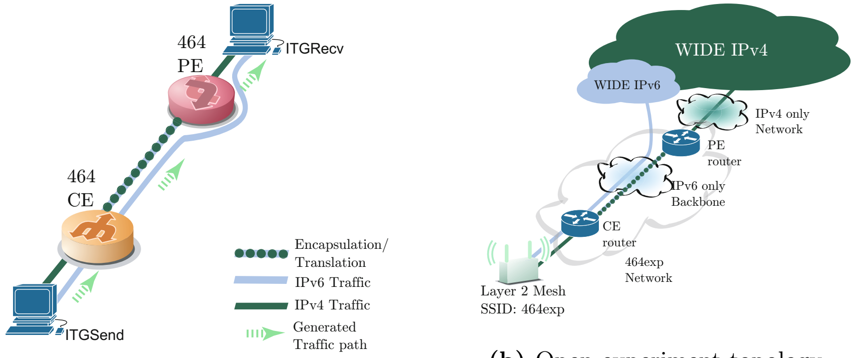
- *configuration capability*: measures how capable a network implementations is in terms of contextual configuration or reconfiguration
- *troubleshooting capability*: measures how capable a network implementation is at isolating and identifying faults
- *applications capability*: measures how capable a device is at ensuring compatibility with common user-side protocols

Details about the measurement process for these three metrics, as well as other methodology and infrastructure details, are presented in the following subsections.

3.2 Closed Environment

Infrastructure. The basic, small scale template for 464 technologies is composed of a set of network routers, a Customer Edge (CE) router which would encapsulate/translate the IPv4 packets in IPv6 packets, and a Provider Edge (PE) router, which would handle the decapsulation/translation from IPv6 back to IPv4. The IPv4-only backbone would be used for forwarding the IPv4 traffic. The IPv6 traffic would be directly forwarded by the IPv6 backbone. The closed experiment's design, presented in fig. 1a, follows the basic network template, including one Customer Edge (CE) machine and one Provider Edge (PE) machine.

Multiple technologies can be considered suitable for the 464 scenario: MAPe [15], MAPt [10], DSLite [6], 464XLAT [11]. Some implementations supporting these technologies have been proposed. One of those is the asamap vyatta distribution, which covers 4 of those technologies: MAPe, MAPt, DSLite and 464XLAT. Both 464 PE and 464CE machines have used as Operating System the asamap vyatta distribution.



(a) Closed experiment topology

(b) Open experiment topology

Fig. 1. Experimental setup

The closed experiment has used as underlaying infrastructure the StarBED, a large scale general purpose network testbed, administered by the National Institute of Information and Communications Technology (NICT) of Japan. Four computers were used for this experiment: two for the devices under test (DUT), 464 PE and 464 CE, and two for the testing platform. The testing platform computers have used Ubuntu 12.04.3 server as base operating system. One of the computers preformed the ITGSend function, generating the traffic, while the other ran the ITGRecv function, receiving the generated traffic.

Methodology. The experimental workload was represented by the amount of traffic inserted into the experimental network. We have considered the combinations of frame size and frame rates displayed in Table 1. These have been recommended in RFC5180, IPv6 Benchmarking Methodology for Network Interconnect Devices, [13] as maximum frame rates × frame sizes for 10 Mbps Ethernet. 10 Mbps rates represent the first experimental baseline. For future tests we intend to expand to 100 Mbps as well as 1000Mbps. The traffic was generated using the Distributed Internet Traffic Generator (D-ITG) [3].

As other important parameters affecting the network performance we have considered: the IP version, IPv4 and IPv6, the upper layers protocols, UDP and TCP, the IPv6 transition technology and the IPv6 transition implementation. A full factorial design was employed, hence $12 \times 2 \times 2 \times 4 \times 1 = 192$ experiments were conducted. As recommended by RFC2544 [5], the duration of each experiment was 60 seconds after the first timestamp is sent. Each test was repeated 20 times and the reported value is the average of the recorded values.

3.3 Open Environment

Infrastructure. The open experiment topology, presented in fig. 1b also follows the basic, small scale 464 network template. The major difference is that

Table 1. *Framesize × Framerate*

No	Frame size	Frame rate	No	Frame size	Frame rate
1	64	14880	7	1518	812
2	128	8445	8	1522	810
3	256	4528	9	2048	604
4	512	2349	10	4096	303
5	1024	1197	11	8192	152
6	1280	961	12	9216	135

the testing platform was replaced by open up-link and down-link connections. The open environment was part of a bigger experimental network, which supplied Internet access to participants at the WIDE Camp 1309, a networking event, held between September 10 and September 13 2013, at Shinsu-Matsushiro Royal Hotel, Nagano, Japan. The 464 network consisted of two virtual machines, the Customer Edge machine (CE) and the Provider Edge machine (PE). The two machines have ran on a virtual environment constructed using a Dell PowerEdge R805 server and the Citrix Barebone XenServer 6.0 as hypervisor. The base implementation for all four tested transition technologies, MAPe, MAPt, DSLite, 464XLAT has been the asamap vyatta distribution. The technologies have been tested sequentially during the four days of the event. On the up-link, the IPv4 and IPv6 traffic was routed by a dual-stack core router. WIDE Camp participants were able to connect to the environments trough a single SSID, *464exp*, handled by the Layer 2 Cisco WiFi Mesh.

Methodology. For operational capability we have used as metrics: *configuration capability*, *troubleshooting capability* and *applications capability.* As measurement method for configuration capability, we have considered a number of configuration tasks, which have been inspired by the abstracted guidelines presented in [8]. The tasks can be organized in three generic groups, *initial setup*,*reconfiguration* and *confirmation.* For an easier referencing we have associated each task with a task code in accordance with the respective group association.

1. IinitialSetup1: Configure an encapsulation/translation virtual interface using a command line interface or a graphical user interface
2. IinitialSetup2: Save the current temporary configuration commands in a file which can be loaded at start-up
3. IinitialSetup3: Self configuration according to contextual configuration details
4. InitialSetup4: Display warnings in the case of misconfiguration and reject the mis-configured command
5. InitialSetup5: Display warnings in the case of missing command and reject saving the temporary configuration
6. InitialSetup6: Display contextual configuration commands help
7. Reconfiguration1: Convert current configuration settings to configuration commands

8. Reconfiguration2: Back-up and restore the current configuration
9. Confirmation1: Show the current configuration
10. Confirmation2: Show abstracted details for the 464 virtual interface

The configuration capability was measured as a ratio between the number of successfully completed configuration tasks and the total number of tasks. Similarly, for troubleshooting capability we have proposed a number of troubleshooting tasks. The tasks follow the fault isolation, fault determination and root cause analysis (RCA) guidelines presented in [8]. Consequently the tasks can be organized into the three generic categories: *fault isolation*, *fault determination* and root cause analysis RCA. For easy referencing, these tasks as well were associated with group codes:

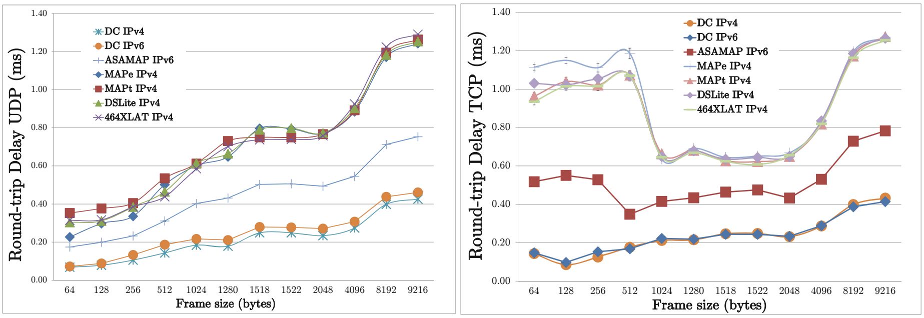
1. FaultIsolation1: Capture and analyze IPv4 and IPv6 packets
2. FaultIsolation2: Send and receive contextual ICMP messages
3. FaultDetermination1: Identify a mis-configured contextual route
4. FaultDetermination2: Identify a mis-configured contextual line in the virtual 464 interface configuration
5. FaultDetermination3: Perform self-check troubleshooting sequence
6. RCA1: Log warning and error messages
7. RCA2: Display log
8. RCA3: Display in the user console the critical messages with contextual details
9. RCA4: Log statistical network interface information
10. RCA5: Display detailed statistical network interface information

The troubleshooting capability was also measured as a ratio of successful tasks over total number of troubleshooting tasks. To measure applications capability, we have tested a non-exhaustive list of common user applications in relation with the 464 transition technologies. The measurement result is presented as a ratio between the number of successfully tested application and the total number of applications.

4 Empirical Results

4.1 Closed Experiment Results

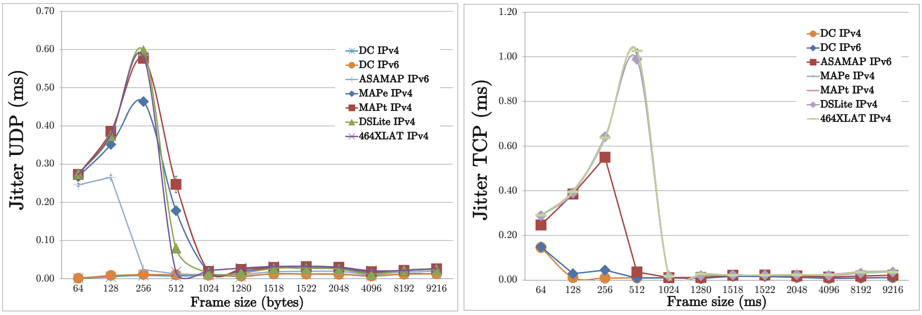
The network performance of the devices under test (DUTs) was compared with a Direct Connection setup in which the two test platform servers were connected directly. The results have been graphed as a function of frame size and the error bars present the margin of error for the mean, calculated at a 99% level of confidence. The latency results, composed of end-to-end delay 2 and jitter 3 show a slightly better performance for 464XLAT, by comparison with the rest of the technologies. Also, in average, translation-based technologies (MAPt, 464XLAT) had a better performance than encapsulation-based technologies (MAPe, DSLite).



(a) UDP

(b) TCP

Fig. 2. Delay results



(a) UDP

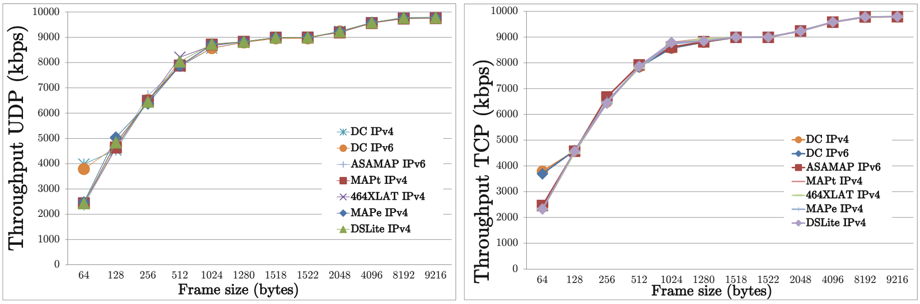
(b) TCP

Fig. 3. Jitter results

The average throughput results, presented in fig. 4, show a similar performance for the four technologies. The overall average shows a small lead for DSLite and encapsulation-based technologies.

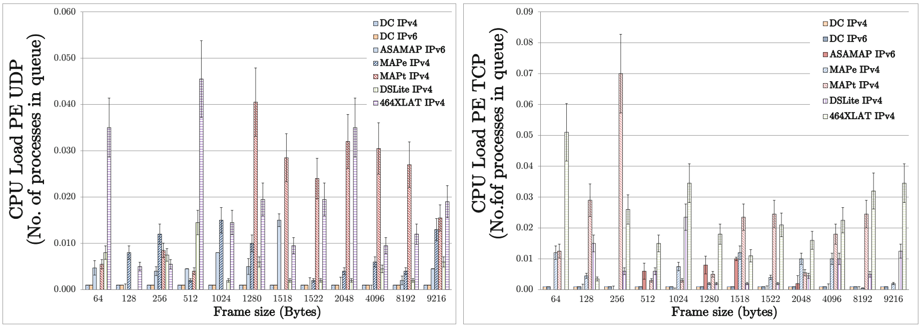
The loss rates, with the exception of some outliers for translation-based technologies over UDP (MAPt and 464XLAT), are very close to 0. For the outliers, the maximum loss-rate is approximately 0.003 %, considered negligible in most cases.

The average CPU load for the provider edge (PE) router, presented in fig. 5, shows a higher average CPU load for translation-based technologies(464XLAT and MAPt). By contrast, the average CPU load of the customer edge (CE) router, shown in fig. 6, is higher for the encapsulation-based technologies. As an overall MAPe seems to have the smallest impact on CPU load. Also notable is that encapsulation-based technologies outperformed the translation-based ones from this standpoint.



(a) UDP **(b)** TCP

Fig. 4. Throughput results



(a) UDP **(b)** TCP

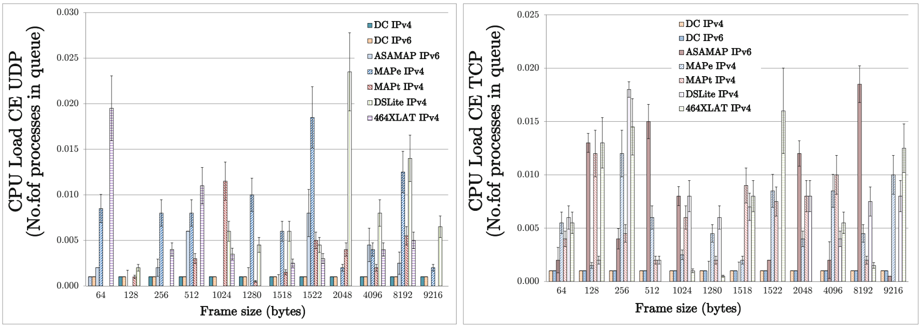
Fig. 5. CPU Load PE results

Considering the overall average of these measurements, the best performance was achieved by MAPe followed closely by DSLite, MAPt and 464XLAT. Also notable was the the IPv6-only connection outperformed all of the 464 technologies.

4.2 Open Experiment Results

During the four days of the WIDE Camp 1309 event, we had the chance to test the operational capability of the asamap implementation. The results for configuration and troubleshooting capability have been summarized in table 2.

Regarding the configuration capability, most of the tasks have been completed successfully. However, a self-configuration setup sequence is not yet available for the asamap implementation. Given the complexity of the transition technologies, a guided self-configuring setup would be a beneficial feature. For the troubleshooting capability as well, most of the tasks have been completed successfully. Two of the troubleshooting tasks couldn't be completed: FaultDetermination3: Displaying critical messages with associated details and RCA3:



(a) UDP

(b) TCP

Fig. 6. CPU Load CE results

Table 2. Operation capability results

Operational Capability		Asamap			
		MAPe	MAPt	464XLAT	DSLite
Configuration Capability	IinitialSetup1	Pass	Pass	Pass	Pass
	IinitialSetup2	Pass	Pass	Pass	Pass
	IinitialSetup3	Fail	Fail	Fail	Fail
	IinitialSetup4	Pass	Pass	Pass	Pass
	IinitialSetup5	Pass	Pass	Pass	Pass
	InitialSetup6	Pass	Pass	Pass	Pass
	Reconfiguration1	Pass	Pass	Pass	Pass
	Reconfiguration2	Pass	Pass	Pass	Pass
	Confirmation1	Pass	Pass	Pass	Pass
	Confirmation2	Pass	Pass	Pass	Pass
Configuration capability result		9/10 = 0.9	9/10 = 0.9	9/10 = 0.9	9/10 = 0.9
Troubleshooting Capability	FaultIsolation1	Pass	Pass	Pass	Pass
	FaultIsolation2	Pass	Pass	Pass	Pass
	FaultDetermination1	Pass	Pass	Pass	Pass
	FaultDetermination2	Pass	Pass	Pass	Pass
	FaultDetermination3	Fail	Fail	Fail	Fail
	RCA1	Pass	Pass	Pass	Pass
	RCA2	Pass	Pass	Pass	Pass
	RCA3	Fail	Fail	Fail	Fail
	RCA4	Pass	Pass	Pass	Pass
	RCA5	Pass	Pass	Pass	Pass
Troubleshooting capability result		8/10 = 0.8	8/10 = 0.8	8/10 = 0.8	8/10 = 0.8

self-check sequence. Regarding the first one, some critical messages are displayed in the user console. However these are hard to interpret and understand. We believe this feature needs improvement. As for the second one, a self-check

Table 3. Applications capability results

Applications			Asamap			
			MAPe	MAPt	464XLAT	DSLite
Win 7 / Win 8 / Ubuntu 12.04 / Android 2.3	Browsing	Chrome	Pass	Pass	Pass	Pass
		Firefox	Pass	Pass	Pass	Pass
		Dolphin	Pass	Pass	Pass	Pass
	E-mail	Outlook	Pass	Pass	Pass	Pass
		Thunderbird	Pass	Pass	Pass	Pass
		Aquamail	Pass	Pass	Pass	Pass
	IM&VoIP	Skype	Pass	Pass	Pass	Pass
		Facebook	Pass	Pass	Pass	Pass
		Google+	Pass	Pass	Pass	Pass
		VoIP Buster	Pass	Pass	Pass	Pass
		Viber	Pass	Pass	Pass	Pass
		DigiOriunde	Pass	Pass	Pass	Pass
	VPN	OpenVPN	Pass	Pass	Pass	Pass
		Spotflux	Pass	Pass	Pass	Pass
	Cloud	Dropbox	Pass	Pass	Pass	Pass
		GDrive	Pass	Pass	Pass	Pass
	FTP	Filezilla	Pass	Pass	Pass	Pass
	Troubleshooting	puTTY	Pass	Pass	Pass	Pass
		WinSCP	Pass	Pass	Pass	Pass
		ConnectBot	Pass	Pass	Pass	Pass
Applications capability result			20/20 = 1	20/20 = 1	20/20 = 1	20/20 = 1

sequence is not available yet. This would represent a substantial improvement of the troubleshooting capability.

As for applications capability, inspired by [1], during the WIDE Camp event we have tested a non-exhaustive list of common applications. The full list of applications and the results are presented in table 3. To summarize we didn't encounter any applications troubles for any of the four technologies.

5 Discussion

IPv6 transition scenarios and IPv6 transition technologies have already been introduced for some time to the Internet Community. However the worldwide deployment rate of IPv6 is still very low. Given the complexity and the diversity of transition technologies, one of the biggest challenges is understanding which technology to use in a certain network scenario.

This article is proposing an answer to that challenge in the form of a network evaluation testbed, called IPv6NET. The contribution of this paper is represented by the detailed testing methodology associated with IPv6NET and the empirical feasibility results, which to our best knowledge represent a first in current literature.

Analyzing the empirical results, we found that one transition technology is *more feasible* than the rest, namely *MAPe*. We have also identified possible performance trends in IPv6 transition technologies benchmarking, for example encapsulation-based technologies seem to have better throughput performance and translation-based technologies better latency performance. A limitation of this method is represented by the lack of control data, since there is no similar alternative system to act as comparison base for the empirical results. We are planning to solve this by comparing the current open source based measurement system with existing commercial network benchmarking tools.

The empirical results can serve as a direct guideline to network operators faced with a similar transition scenario. One limitation of this approach is represented by the diversity and complexity of existing production networks by comparison with the presented scenario. However, by using the detailed methodology, any interested party could potentially implement it, and obtain customized feasibility data. The methodology can also serve as guideline for other researchers interested in joining this effort. Coping with a large number of technologies and their future developments may very well be solved by research collaboration. It can transform this project in an exhaustive IPv6 transition resource.

6 Conclusion

In this article we have introduced IPv6NET, a project aiming to empirically analyze the feasibility of IPv6 transition technologies in relation with specific network scenarios. From the methodology standpoint IPv6NET combines two types of testing environments, closed environments for thorough network performance data, and open environments for operational data. The network performance results, obtained in the close experiment, indicate MAPe as most feasible transition technology for the 464 scenario. However the other three technologies, DSLite, MAPt and 464XLAT follow it closely. As performance general guidelines, for latency, the translation-based technologies (464XLAT, MAPt) had a better performance. For throughput and CPU load the results were in favor of encapsulation-based technologies (MAPe, DSLite). Also a notable thing was that the IPv6-only connection outperformed all the 464 transition technologies.

In terms of applications capability we did not experience any application troubles. The operational capability results indicate that asamap had a good performance as well. Considering the overall operational results, we can safely conclude that the asamap vyatta distribution is a feasible implementation for the 464 network scenario.

For future work, we consider as first step increasing the scale of the network template. Regarding the open environment methodology, we are considering adding security as a feasibility indicator and proposing an associated metric. Another future step is proposing an unique general feasibility indicator (GFI), associated to each transition technology, which would help better centralize and compare the the results.

Acknowledgments. The authors would like to thank Mr. Masakazu Asama for providing the vyatta asamap distribution, upon which the experimental networks were implemented.

References

1. Arkko, J., Keranen, A.: Experiences from an IPv6-Only Network. RFC 6586 (Informational) (April 2012)
2. Babiker, H., Nikolova, I., Chittimaneni, K.K.: Deploying ipv6 in the google enterprise network lessons learned. In: Proceedings of the 25th International Conference on Large Installation System Administration, LISA 2011, p. 10. USENIX Association, Berkeley (2011)
3. Botta, A., Dainotti, A., Pescapè, A.: A tool for the generation of realistic network workload for emerging networking scenarios. Computer Networks **56**(15), 3531–3547 (2012)
4. Bound, J.: IPv6 Enterprise Network Scenarios. RFC 4057 (Informational) (June 2005)
5. Bradner, S. McQuaid, J.: Benchmarking methodology for network interconnect devices (1999)
6. Durand, A., Droms, R., Woodyatt, J., Lee, Y.: Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion. RFC 6333 (Proposed Standard) (August 2011)
7. Grayeli, P., Sarkani, S., Mazzuchi, T.: Performance analysis of ipv6 transition mechanisms over mpls. International Journal of Communication Networks and Information Security 4(2) (2012)
8. Harrington, D.: Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions. RFC 5706 (Informational) (November 2009)
9. Hazeyama, H., Hiromi, R., Ishihara, T., Nakamura, O.: Experiences from IPv6-Only Networks with Transition Technologies in the WIDE Camp Spring 2012. draft-hazeyama-widecamp-ipv6-only-experience-01.txt (March 2012)
10. Hazeyama, H., Hiromi, R., Ishihara, T., Nakamura, O.: Experiences from IPv6-Only Networks with Transition Technologies in the WIDE Camp Spring 2012. draft-hazeyama-widecamp-ipv6-only-experience-01.txt (March 2012)
11. Mawatari, M., Kawashima, M., Byrne, C.: 464XLAT: Combination of Stateful and Stateless Translation. RFC 6877 (April 2013)
12. Narayan, S., Shang, P., Fan, N.: Network performance evaluation of internet protocols ipv4 and ipv6 on operating systems. In: Proceedings of the Sixth International Conference on Wireless and Optical Communications Networks, WOCN 2009, pp. 242–246. IEEE Press, Piscataway (2009)
13. Popoviciu, C., Hamza, A., Van de Velde, G., Dugatkin, D.: Ipv6 benchmarking methodology for network interconnect devices (2008)
14. Sasanus, S., Kaemarungsi, K.: Differences in bandwidth requirements of various applications due to ipv6 migration. In: Proceedings of the International Conference on Information Network 2012, ICOIN 2012, pp. 462–467. IEEE Computer Society, Washington, DC (2012)
15. Troan, O., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., Taylor, T.: Mapping of Address and Port with Encapsulation (MAP). draft-ietf-softwire-map-08 (August 2013)