Challenges faced by SIPv6 in the Transition from IPv4 to IPv6 for VOIP Services

Mzikayifani Mthombeni [1], Odhiambo Marcel O[2]
University of South Africa, Department of Electrical Engineering, Private Bag X6, Florida, 1710,
Tel: +27 11 256 0690, Fax: +27 11 471 3054
email: [1]mzi.mthombeni@gmail.com, [2]ohangamo@unisa.ac.za

Abstract- The Transition from IPv4 to IPv6, has required modification of SIP (Session Initiation Protocol), which is the protocol that carries VoIP (Voice over Internet Protocol) traffic. Currently available solutions require the co-existence of IPv4 addresses. This work proposes a scheme that will enable IPv4 hosts to communicate with IPv6 host without the modification of either end hosts nor require a pool of IPv4 addresses. The proposed solution is an enhancement of SBIIT (Socket identifier Based IPv4/IPv6 Translator) which is an improvement of the NAPT-PT and SLIIT (Shim Layer based IPv4-IPv6 Translation) based approaches. The enhancements of SLIIT enable SBIIT to support SIP traffic by translating addresses in both the packet header and payload through using the protocols embedded in the IP addresses of the payloads. The design to implement SBIIT will be done using a network processor IXP 435 from Intel which will be used as a Translator interoperating seamlessly between v4 and v6 hosts.

Index Terms— SIP, VoIP, SBIIT, Translation, NAT-PT

I. INTRODUCTION

IPv6 has been rapidly adopted due to the depletion of IPv4 address [1]. IPv6 has however not been able to cater for all the possible short-falls encountered during the transition (from IPv4 to IPv6) , as its primary focus was on having a new IP Addressing scheme that will provide sufficient IP addresses for every user on earth.

One problem that has been overlooked in the IPv4/IPv6 transition was its deficiency to adequately deal with the translation of SIP based VoIP traffic [2]. Translation of SIP traffic happens at the application layer, unlike other forms of traffic that get translated at the lower layers. In the light of continuous coexistence of IPv4 and IPv6 for quite some time, a scheme needs to be devised to allow addressing schemes to operate with minimum or no modifications to the end-user systems and, also not be dependent on the availability of IPv4 addresses. Currently available techniques require the coexistence of IPv4 and IPv6 namely: i) Dual Stack; ii) Tunneling, and iii) Translation [2,3].

The need for inter domain communication (IPv4/IPv6) will therefore leave Translation as the only feasible technique to allow communication, that does not required the aforementioned dependencies.
To implement this scheme, we propose the enhancement of SBIIT (Socket identifier Based IPv4/IPv6 Translator) which was previous explored by Davis et al [9] and in SLIIT [6,9]. Our contribution would be an improvement of the translation based approach which currently uses a 4-tuple complete socket identifier corresponding to a connection and make it a 5- tuple socket identifier. This will be done by using 16bits and adding the identification field

The design and implementation of the improved solution will be carried out using the networking processor; Intel, the IXP 435 that caters for Voice and Video Traffic [4]. The "Translator" will enable a name assigned to a v4 host to be resolved by v6 host during inter communication, seamlessly, and vice versa. In SIP (i.e. VoIP) traffic where addresses are embedded not only in the headers but also in the payload. We will ensure that those IPs are translated as well.

II. RELATED WORK

Currently available techniques for Translation that allow for communication between IPv4 only devices and IPv6-only devices are as follows:
NAT-PT (Network Address Translation-Protocol Translation) as specified in [10] has the downfall that it is dependent on the availability of IPv4 in the pool
NAPT-PT (Network Address Port Translation-Protocol Translation) as also highlighted in [7] has the demerit of that on IPv6 side, there can only be one server running.
SLIIT [9], is a technique that overcame the problems faced by the above translation methods, by using a 4-tuple for maintaining the state of a connection. Included in the 4-tuple are the IPv4 address, port of the IPv4 host, port of the IPv6 and IPv6 address. This quadruple connection identifier allowed for multiple servers per service to run simultaneously on both ends with IPv4 and IPv6 hosts.

The disadvantage of the above techniques is that they work at the network layer and as a consequence fail to translate SIP traffic that has its IP address and port embedded in the application layer [5]. Fortunately SBIIT has SIP-ALG that does the address translation at the application layer thereby enabling IPv4-IPv6 VoIP communication without the modification of end hosts [6]. The DNS machine does not have to be located behind the Translator, but can be situated anywhere in the network, this is an added advantage of SBIIT [8,9].

III. DESIGN AND IMPLEMENTATION

The IPv4-IPv6 Translator is dual-stack host, which means it is capable of handling both IPv4 and IPv6 traffic, while

connected to the LAN. The Network Architecture is shown in Figure 1.
The DNS server enables names based communication when queries for communication are done by a source IP to a destination IP, for example the IP 74.125.233.19 is translated to the matching website URL name by the DNS.

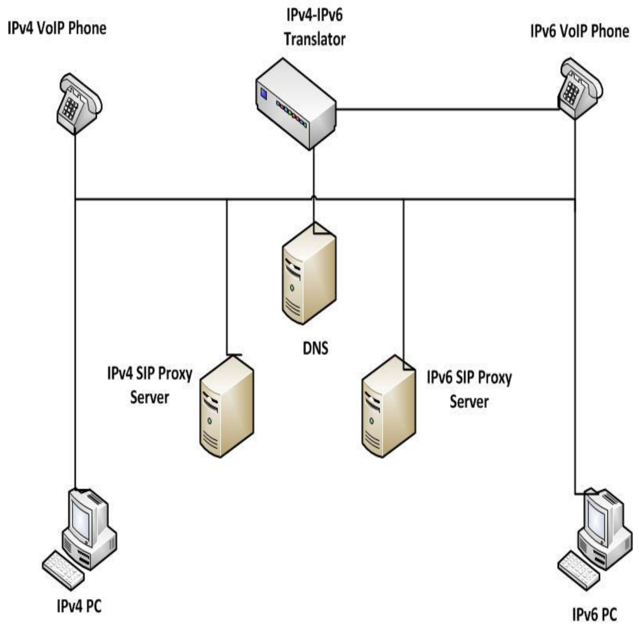

Figure 1: Network Architecture

The DNS queries are routed through the Translator, and the DNS should be configured as the Translators' corresponding address. The DNS-ALG (DNS- Application Level Gateway), in the Translator will forward the DNS queries to the actual DNS Server [9]. The translation involves header conversion, which is made up of network address translation and protocol (IP header) translation. The protocol used for translation is SIIT (Stateless IP/ICMP Translation)[7].

The primary purpose of the translator is to make the end hosts (IPv4 and IPv6) believe they are communicating with hosts in their own addressing scheme (i.e. IPv4 to IPv4, IPv6 to IPv6). To cater for this the appropriate modules in the Translator, kick in to translate the embedded addresses as shown by the DNS ALG and SIP ALG, in [9]

IV. IMPLEMENTATION SCHEDULE

The implementation will be done using a network processor (Intel IXP435) for the translator code. With open source Linux used for the Operating System. While the Translator application will be implemented using C for the user space.

For the SIP Proxy, the SER (SIP Express Router) will be considered in comparison to the Kamailio SIP Router, with a likely implementation bias towards the later, than the former due to its merits.

V. CONCLUSION

This paper describes SBIIT and how we intend to use it to solve the problem faced by IPv4 hosts when communicating with IPv6 hosts, without the need of IPv4 addresses and modifying hosts at either side. The solution will focus on SIP traffic mainly.
The Implementation will be done using a Network Processor, as a Translator between the two addressing schemes. To measure the success of the solution we will test if translation indeed happens, by means of running a “ping” from an IPv4 host to an IPv6, to ensure that traffic goes across from one side to another, when we get ping replies. A VoIP call will be made from either side, to test that the components within the communication path are appropriately converting traffic as they should be. Future research areas include looking at how Real-time Transport Protocol (RTP) for media transfer can be translated, from IPv4 to IPv6, if it will need a translator and how it would be affected by propagation delays.

REFERENCES:

[1] Timothy Rooney. IP Address Management Principles and Management. Wiley Publishing. 2011.

[2] Hoeher .T, and Tomic .S. White Paper. 2006. SIP Collides with IPv6. Vienna, Austria: Telecommunications Research Center Vienna.

[3] Chen. J, Jia.Z, Li.X. A New Design of Embedded IPv4/IPv6 Dual- Stack Protocol. NCIS '11 Proceedings of the 2011 International Conference on Network Computing and Information Security - Volume 02 Pages 163-167. May 2011

[4] Luo .W, Yan .B, Li .X, Mao .W . (2008). Network-Processor-Based IPv4/IPv6 Translator: Implementation and Fault Tolerance, provide details of this report, 2008

[5] Zhou .B, ALG Consideration of SIP, March 2010.

[6] Whai-En Chen, Huang Ya-Lin, Yi-Bing Lin. An effective IPv4–IPv6 translation mechanism for SIP applications in next generation networks**.**. International Journal of Communication Systems. Vol. 23, Issue 8 , August 2010

[7] Li.X et al, "IP/ICMP Translation Algorithm", IETF RFC 6145, April 2011

[8] Arko.J et.al, "IPv4 Run-Out and IPv4-IPv6 Co-Existence Scenarios", IETF RFC 6127, May 2011

[9] Davis A.K , Vasunden. K, Kuri.J, Dagale .H, IPv4-IPv6 Translator for VoIP and Video Conferencing, September 2010

[10] Arko.J et.al,"Guidelines for Using IPv6 Transition Mechanisms during IPv6 Deployment", IETF RFC 6180, May 2011

Mzikayifani Mthombeni received a BTech in Electronic Engineering from Durban University Technology. He is currently pursuing an MTech in Electronic Engineering at University of South Africa. His research interests are in VoIP and MPLS.