

ADAPTIVE REPRESENTATION OF IPv6 ADDRESS

DEEPTHI S, PRASHANTI G & SANDHYA RANI K

Assistant Professor, Computer Science and Engineering, Vignan's Lara Institute of Technology and Science, Guntur,
Andhra Pradesh, India

ABSTRACT

Introduction of IPv6 made drastic changes in IT Industry in recent years due to new innovation of Cloud computing. As IPv6 deployment increases, there will be a dramatic increase in the need to use IPv6 addresses and so many services running in Cloud computing will face problems associated with IPv6 addressing[6]: the notation is too long (39 bytes), there are too many variants of a single IPv6 address and a potential conflict may exist with conventional http_URL notation caused by the use of the colon (:). This paper proposes a new scheme to represent an IPv6 address with a shorter, more compact notation (26 bytes), without variants or conflicts with http_URL. The proposal is known as adaptive representation of IPv6 address with Base64 and uses the well-known period (or dot) as a group delimiter instead of the colon and non symbolic usage of the Base 64. The paper mainly concentrated on reducing the no of segments in the IPv6 address.

This is an Enhancement for Non Symbolic Representation of IPv6 address. Here we discussed regarding merits and demerits of other works that predate this paper have been reviewed and evaluated. Cloud computing, as a continuously emerging mainstream of network-based applications, is likely to be a forerunner in the use of IPv6 as the base protocol. As a result, Cloud computing will benefit most from the new, compact and user-friendly textual representation of IPv6 address proposed by this paper.

KEYWORDS: IPv6 Address, Cloud Computing, Base64, Colon Hexadecimal, Uni Cast Address, Text Conversions

INTRODUCTION

Cloud computing is a new computing paradigm where in computer processing is being performed through internet by a standard browser. Cloud computing builds on established trends for driving the cost out of the delivery of services while increasing the speed and agility with which services are deployed. It shortens the time from sketching out application architecture to actual deployment. It incorporates virtualization, on-demand deployment, Internet delivery of services, and open source software. Cloud computing relies on the infrastructure of Internet; as a consequence, it will be significantly affected by the transition from current IPv4 to next generation IPv6[10].

It is notable that all cloud computing service modes

Software as a Service: In this model an application is hosted as a service to customer who accesses it via the Internet.

Platform as a Service: PaaS services include application design, development, testing, deployment and hosting. In this not only services but server, memory and other platforms can be used and subscriber needs to pay as per terms and conditions.

Infrastructure as a Service: The Cloud Infrastructure services delivers the platform virtualization which shows only the desired features and hides the other ones using the environment in which servers, software or network equipment are fully outsourced as the utility computing which will based on the proper utilization of the resources by using the Principle of

reusability that includes the virtual private server offerings for the tier data center and many tie attributes which is finally assembled up to form the hundreds of the virtual machines.

These are made possible with the underlying support of TCP/IP. Without a reliable, efficient networking Infrastructure it is unlikely that cloud computing would be able to develop. It is anticipated that there will be a protracted Period of change and that dual-stack IP networking will be utilized for a considerable time. The reasons why IPv6 is necessary, it is aimed to resolve a number of shortcomings of the current version (IPv4)[12]. The main issue at that time was IPv4 address space exhaustion, as well as the lacking of with auto configuration, mobility, flow Labeling and security. Flow labeling and security issues with IPv4 have been addressed as far as the addresses are concerned in the meantime, and these are no longer an argument for a changeover. Although the 32-bit address space of IPv4 permits over 4.2 billion addresses, previous and current allocation practices have limited the number of publicly useable IPv4 addresses to a few hundred million. For this the Internet Engineering Task Force (IETF) has proposed a set of mechanisms (i.e. NAT [4]) to alleviate the scarcity of public IPv4 addresses. For example, NATs have been deployed to translate a public IPv4 address into a set of private IPv4 addresses.

All these solutions are makeshift measures to extend the life of the IPv4 address space. They will not ultimately overcome the scarcity of the IPv4 public address space, or many other limitations of today’s communications. the demand for new IP addresses is continuously increasing and it is speculated that after the depletion of IPv4as shown in figure 1[11].In order to address the limitations of IPv4, the Internet Engineering Task Force (IETF) developed the IP version 6 (IPv6) suite of protocols and standards. The increasing number of Internet users, systems, and the convergence of services into common infrastructure will drive the demand for IPv6. Percentage of IPv6-enabled Autonomous Systems registered in each RIR is showed in figure 2. The commercial opportunities that IPv6 can provide for wireless devices, peer-to-peer networking and the “smart home” are also driving the move to the new technology. The wireless market requires a low latency, always on, auto-roaming always reachable IP service. Percentage of IPv6-enabled Autonomous Systems registered in each RIR is showed in figure 2.

Projected IANA Exhaustion: 30/07/2011

Projected RIR Exhaustion: 15/03/2012

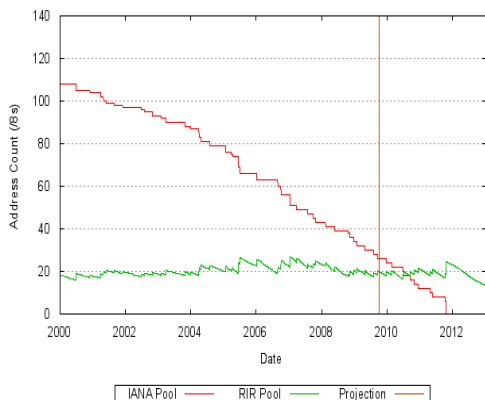


Figure 1: IPv4 Consumption: Projection

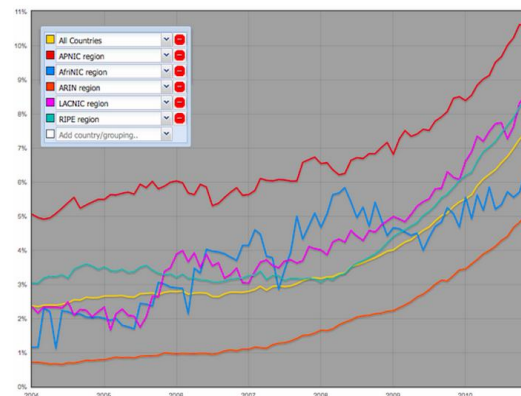


Figure 2: Percentage of IPv6-Enabled Autonomous Systems Registered in Each RIR

Benefits over Using IPv6

Internet Protocol version 6 (IPv6) is the next-generation network protocol, which has been standardized to replace the current IPv4. It holds great promise to become the backbone of the future of the Internet and offers a significant improvement over IPv4 in terms of scalability, security, mobility and convergence.

Larger Address Space

Probably the most commonly known advantage of IPv6 over IPv4 is its enlarged address space. While IPv4 address is 32-bit long, IPv6 uses a 128-bit address, the four fold increase in length results in approximately 8×10^{28} more addresses being available and resolving the public address depletion issues. The basic framework of the IPv6 protocol was standardized by the Internet Engineering Task Force (IETF) in the 1990s. However, there is still ongoing development of certain advanced aspects of the protocol. Internet Protocol IPv6 provides higher efficiency like having enough level of IPs, stronger security and mobility. In fact it is good to evaluate the performance benefits that we can get from IPv6 protocol in compare to the IPv4 protocol. We can upgrade the existing IPv4 infrastructure to the next generation Internet Protocol (IPv6) and get its advantages using the transition mechanisms [6]. IPv6 which also provides Network built-in security the means to encrypt traffic even within a local network.

Improved Quality-of-Service (QoS)

A new capability is added in IPv6 to enable the labeling of packets belonging to particular traffic flows for which the sender requests special handling. This is done through the “flow label” component in the IPv6 header [5]. Mobile IPv6. The IPv6 standards include a feature called “Mobile IPv6”.

The new features of IPv6 result in a number of business benefits:

Lower Network Administration Costs:

The auto configuration and hierarchical addressing features of IPv6 will make networks easy to manage.

Optimized for Next-Generation Networks:

Getting rid of NAT re-enables the peer-to-peer model and helps with deploying new applications (e.g., communications and mobility solutions, such as VoIP.) [3]

Protection of Company Assets:

Integrated IP security (IPSec) makes IPv6 inherently secure and provides for a unified security strategy for the entire network.

Investment Protection:

The transition and translation suite of protocols helps with easy and planned migration from IPv4 and IPv6, while allowing for coexistence in the transition phase.

IPv6 Addressing and Architecture

Whilst we believe that IPv6 will begin a new and improved communications era for the whole IT industry, we also accept that IPv6 itself is not perfect. First, it is obvious that with such a large address space (3.4×10^{38} or 340 undecillion addresses) a significant number of characters will be required to represent any single address.

A full IPv6 address consists of 32 bytes or a string of 39 characters (including 7 delimiters) in human readable form which is both challenging to remember and prone to mistakes when read, written or deployed. A longer notation means more buffer space is needed when saving, there is an increased cost in bandwidth and latency time during transit, and more computing power is used when reading/writing, searching/parsing, etc.

Second, the current IPv6 notation of “colon hexadecimal” [5] has another issue that there are too many variants of text representation for a single IPv6 address [6]. With such a degree of flexibility in representing an address, it might become

prone to misinterpretation in both human and computer environments (searching, parsing and modifying, logging and operating).

As an example, this IPv6 address:

FE80:0000:0000:0000:0260:97FF:FE8F:64AA

Third, the use of the colon (:) separator in place of the dot (.) presents both a potential ambiguity with current http_URL/Windows UNC and the annoyance of being a “two-key” entry on most. It is unpredictable that how many systems and applications will be affected by this incompatibility[5].

Bearing these issues in mind and considering the increasing demands of cloud computing, this paper introduces a novel scheme to present an IPv6 address in non symbolic Base64with period (or “dot”) delimiters as used in IPv4.

LITERATURE SURVEY

This paper introduced new mechanism which is not the first and it has future enhancement. In order to present our work we made a lot of study regarding existing methods. Those are having some difficulties on representation and remembrance. Our paper has made modifications which overcome them. The models we studied are described below. Papers aiming to shorten IPv6 address notation

RFC 4291

Text Representation of Addresses: As IPv6 deployment increases, there will be a dramatic increase in the need to use IPv6 addresses in text. The RFC4291 includes the IPv6 addressing model, text representations of IPv6 addresses, definition of IPv6 unicast addresses, anycast addresses, and multicast addresses, and an IPv6 node's required addresses. One conventional way of representing text in string ways is given below. The preferred form is x:x:x:x:x:x:x, where the 'x's are one to four hexadecimal digits of the eight 16-bit pieces of the address[13].

Examples

ABCD:EF01:2345:6789:ABCD:EF01:2345:6789.

2001:DB8:0:0:8:800:200C:417A

Note that it is not necessary to write the leading zeros in an individual field, but there must be at least one numeral in every field (except for the case described in 2.).For example, the following are legal representations of the 60-bit

Prefix 20010DB80000CD3 (Hexadecimal)

2001: 0DB8:0000:CD30:0000:0000:0000:0000/60

2001: 0DB8:: CD30:0:0:0/60

2001: 0DB8:0:CD30::/60

This way we can compress our IPv6 address but the problem is "::-" can only appear once in an address. It can be used to compress 3 segments with zeros. And also some are using for 4 segments and even more with all 0's.

RFC 1924

The New Encoding Format: The new standard way of writing IPv6 addresses is to treat them as a 128 bit integer, encode that in base 85 notation, then encode that using 85 ASCII characters.

Why 85?

2^{128} is 340282366920938463463374607431768211456.

85^{20} is 387595310845143558731231784820556640625, and thus in 20 digits of base 85 representation all possible 2^{128} IPv6 addresses can clearly be encoded.

84^{20} is 305904398238499908683087849324518834176, clearly not sufficient, 21 characters would be needed to encode using base 84, this wastage of notational space cannot be tolerated. This wastage of notational space cannot be tolerated. The character set to encode the 85 base85 digits, is defined to be, in ascending order[4]:

'0'..'9', 'A'..'Z', 'a'..'z', '!', '#', '\$', '%', '&', '(',
)', '*', '+', '-', ';', '<', '=', '>', '?', '@', '^', '_',
'`', '{', '|', '}', and '~'.

This set has been chosen with considerable care.

Converting an IPv6 Address to Base 85

The conversion process is a simple one of division, taking the remainders at each step, and dividing the quotient again, then reading up the page, as is done for any other base conversion.

For example, consider the address shown above

1080:0:0:0:8:800:200C:417A

In decimal, considered as a 128 bit number, that is

21932261930451111902915077091070067066.

As we divide that successively by 85 the following remainders emerge:

51, 34, 65, 57, 58, 0, 75, 53, 37, 4, 19, 61, 31, 63, 12, 66, 46, 70, 68, 4.

Thus in base85 the address is:

4-68-70-46-66-12-63-31-61-19-4-37-53-75-0-58-57-65-34-51.

Then, when encoded as specified above, this becomes:

4)+k&C#VzJ4br>0wv%Yp

This procedure is trivially reversed to produce the binary form of the address from textually encoded format. But it is an order of magnitude harder to read, use and understand.

It also necessitates the user to learn a whole new alphabet. Finally, 85 is not a "bit-boundary" number, base85 therefore does not fully utilize all of the required 7 bits and as a result will produce over-length and discontinuous binary strings.

Transparent Implementation in Traditional Base 64

Currently implemented IPv6 address with a more compact and end-user friendly format for IT professionals especially for naïve users in networking environments. imply it can be said that presenting IPv6 address in base 64, the transformation goes under rules: "The character set to encode the base64 IPv6 address is: 0–9, a to z, A to Z, . (dot)

and—(hyphen); Case sensitive IP scheming; each character represents 6-bits; Last character has to be among 0–3; Maximum number of characters are 22 or more precisely 21.33 characters; . . .”

The following example of Base64 data is from [5], with corrections.

NUML: EDU: PK: ISB _ 10:10:20:30

Although these “addresses” are in a format that is unfamiliar, the reduced character set makes them appear far less daunting than the Base85 example given earlier. The base64 scheme attempts to solve the address length issue by introducing more symbols in a similar manner to Base85 scheme. Both schemes shorten the address representation but in doing so they sacrifice readability and manageability. It is difficult to suggest that this scheme would be readily accepted by academia or industry as it could introduce more complication than the original RFC 1884/4291.

Proposals to Resolve Colon-Related Conflicts Extra Square Brackets in Domain Part of http_URL

IETF RFC 2732 [11], “Format for Literal IPv6 Addresses in URL’s” was created to address the colon-related issue. RFC 2732 narrates the situation where why this is necessary and how to handle it with one more pair of square brackets.

“The textual representation defined for literal IPv6 addresses in [ARCH] is not directly compatible with URL’s. Both use “:” and “.” characters as delimiters.

This document defines the format for literal IPv6 Addresses in URL’s for implementation in World Wide Web browsers. The goal is to have a format that allows easy “cut” and “paste” operations with a minimum of editing of the literal address.” Microsoft registered and reserved the second-level domain “ipv6-literal.net” on the Internet. IPv6 addresses are transcribed as a hostname or sub domain name within this name space, in the following fashion

2aa1: da8 : 65b3 : 8d3 : 1369 : 0a8e : 570 : 7448 is written as

2aa1 - da8 - 65b3 - 8d3- 1369- 0a8e- 570- 7448 literal: net:

As proposed, this method involves more cost and greater complexity, which means it, is not the best choice. There is a very clear requirement to resolve all of these problems preferably in a single method and without introducing further compound confusion.

Even though having lot of problems in using IPv6 but using is increased. The IPv6 Adoption is shown in Figure 3. We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6 [15].

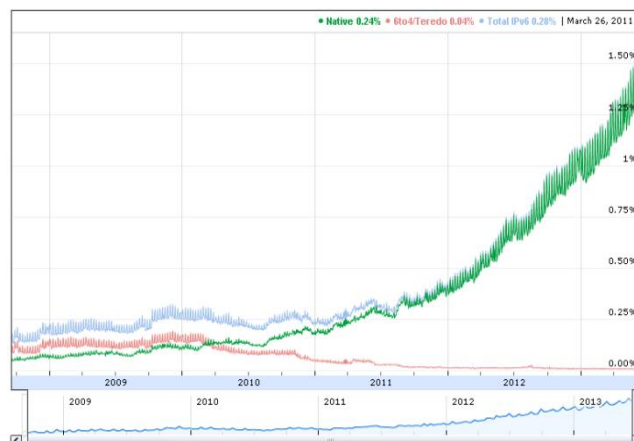


Figure 3: Percentage of Users that Access Google over IPv6

Non Symbolic Base64

A piece of work in the field of base encoding which strives to overcome some issues with conventional base 64 system. As discussed in RFC 4648, traditional base 64 needs three more symbols (“+”, “/”, “=”) to organize its algorithm and representation. As noted by the RFC, this could be problematic in some scenarios. Encoded into structures that mandate other requirements. For base 16 and base 32, this determines the use of upper- or lowercase alphabets.

For base 64, the non alphanumeric characters (in particular, “/”) may be problematic in file names and URLs. Used as identifiers. Certain characters, notably “+” and “/” in the base 64 alphabet, are treated as word-breaks by legacy text search/index tools.

There are nine other groups of variants [15] substituted into traditional base 64 as an attempt to resolve the problems introduced by these otherwise controversial symbols. Non Symbolic Base64 was first described in a paper [16] where it is proposed as an alternative approach to Base64[16] for non-alphanumeric characters and is intended to be an improved implementation of Base64.

It does not use any symbols in its representation, only case sensitive alphabetical (a-z, A-Z) and numeric characters (0–9).Some of the differences between the original Base64 and non Symbolic Base 64 can be seen in

Table 1 and Table 2. In the new Scheme, the symbols “+”, “/” and “=” are not used. Instead, the character “A¹” is a special tag and Subsequently A¹ represents number 62[14], B¹ for 63. As a result, the new alphabet series is 0–9, A-Z, a-z and A¹, B¹

Table 1: Base 64 Representation

value	encoding	value	encoding	value	encoding	value	encoding
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v	pad	=
14	O	31	f	48	w		
15	P	32	g	49	x		
16	Q	33	h	50	y		

Table 2: Non Symbolic Base 64 Representation

value	encoding	value	encoding	value	encoding	value	encoding
0	0	17	H	34	Y	51	p
1	1	18	I	35	Z	52	q
2	2	19	J	36	a	53	r
3	3	20	K	37	b	54	s
4	4	21	L	38	c	55	t
5	5	22	M	39	d	56	u
6	6	23	N	40	e	57	v
7	7	24	O	41	f	58	w
8	8	25	P	42	g	59	x
9	9	26	Q	43	h	60	y
10	A	27	R	44	i	61	z
11	B	28	S	45	j	62	A ¹
12	C	29	T	46	k	63	B ¹
13	D	30	U	47	l		
14	E	31	V	48	m		
15	F	32	W	49	n		
16	G	33	X	50	o		

Since there is no symbol used in n\Non Symbolic Base 64 index table, it shortens the length of IPv6 address without adversely affecting readability, one of the important requirements of the proposed IPv6 address notation.

Non Symbolic Base64 Notation of IPv6 Address

The base 64 to shorten the representation of IPv6 addresses. For an IP address, base 16 is the closest option next to base 10, afterwards there are base 32, base 64 and base 85 in Elz’s informal RFC paper. This proposed new scheme of IPv6 address notation presented by this paper is called Non Symbolic Base64 the binary and colon-hexadecimal representations of an IPv6 address are shown below.

IPv6 Addressing

128 bits

```
0100 1111 1010 1010 0001 1010 1110 0011
0000 0000 1010 0111 0110 0001 0000 0111
0000 0000 0000 0000 0001 1011 1111 1111
0000 0000 0010 0010 0111 1000 1001 1000
```

Represented in Hex(4 bit)

4FAA:1BE2:00A7:6107:0000:1BFF:0022:7898

This long address is commonly depicted as eight pairs of bytes, but it can also be considered in three sections as shown in Figure 4. The general format for IPv6 global unicast addresses is as follows[1]:

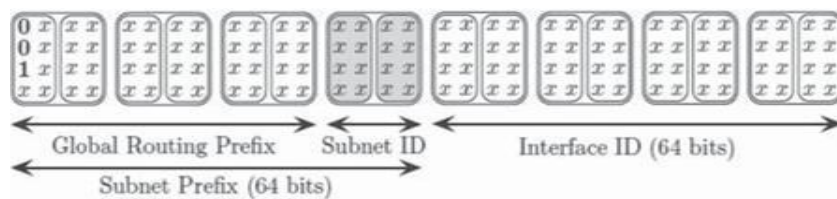


Figure 4: Global Unicast Addresses

The first half of the address is a 64-bit subnet prefix comprising of a six byte (48 bits) Global Routing Prefix and a two byte (16 bits) Subnet ID. The second part of the address is another 64 bits known as the Interface ID and is used mainly in a unicast addressing. For the purpose of this paper, IPv6 addressing could be described using

The format is having following features:

- Encoded in Non Symbolic Base64.
- Dot-separated six segments
- **Prototype Length:** 24 codes + 4 dots = 28 characters
- **Character Range:** 0–9, A-Z, a-z
- Case-sensitive

Conversions to/from Non Symbolic Base64

The process of converting an IPv6 address into Non Symbolic Base64 can be summarized as these steps:

- S1. Divide the given 16-byte address into 5 segments as 4:2:2:4:4
 yyyy:yy:yy:yyyy:yyyy (4:2:2:4:4)
- S2. Convert each segment into Non symbolic Base64
- S3. Separate the Non symbolic Base64 encoded string into 6:4:4:6:6 as
 xxxxxx:xxxx:xxxx:xxxxxx:xxxxxx (6:4:4:6:6)

This method has shorter notation An IPv6 address length reduction of 28.8% or so in theory and which uses 5 segments each segment of 6 bits total 28 bytes are required for representation. Even though some method having 27 bytes of representation but it used five dot representations. Here we got one byte extra but as a human being remembering 5 segments is better than remembering 6 segments. So definitely it has an advantage. This can be enhanced in future.

PROPOSED METHOD

According to Non Symbolic Base64 Method where we have 5 segments. Here in this proposed method we tried a new method which consists only four segments and those are constructed according the global unicast addressing as shown in Figure 4. This method which reduces the 28 Bytes to 26 Bytes definitely a better representation of IPv6 address. This is enhancement of our previous paper called as Non SymbolicBase64: Effective Representation of IPv6 Address.

Here we will see Procedure:

yyyy:yy:yy:yyyyyyyy (4:2:2:8);

Where each “y” stands for one byte (8 bits or two characters). After encoding into Non Symbolic Base64, “yyyy” (4 bytes, 32bits or 6 hex characters) will be replaced by “xxxxxx” (four Non Symbolic Base64, 6-bit characters) and “yy” (2 bytes, 16 bits or four hex characters) will be replaced by “xxx” (three Non Symbolic Base64, 6-bit characters), yyyyyyyy (8 bytes, 64 bits or 11 hex characters) will be replaced by “xxxxxxxxxxx” (11Non Symbolic Base64, 6-bit characters)in Adaptive Representation of IPv6 Address.[1]

Therefore, using the proposed new notation scheme, an IPv6 address in Non Symbolic Base64 will be in the general format xxxxxx:xxxx:xxxx:xxxxxxxxxxx (6:4:4:11) where each “x” represents any one six-bit character (using the code scheme 0–9, a-z, A-Z, A¹B¹)[16].

As before, the first half of the address indicates the subnet prefix and the second half indicates the interface ID. The first 3-digit group indicates the subnet ID. The proposed scheme is known as Non Symbolic Base64 notation of IPv6 address and has the Following Features:

- Encoded in Non Symbolic Base64.
- Dot-separated six segments
- **Prototype Length:** 23 codes + 3 dots = 26 characters
- **Character Range:** 0–9, A-Z, a-z
- Case-sensitive

Conversions to/from Non Symbolic Base64

The process of converting an IPv6 address into Non Symbolic Base64 can be summarized as these steps:

- S1. Divide the given 16-byte address into 4 segments as 4:2:2:8
- S2. Convert each segment into Non symbolic Base64
- S3. Separate the Non symbolic Base64 encoded string into 6:4:4:11 as

xxxxxx:xxxx:xxxx:xxxxxxxxxxx (6:4:4:11)

001101101001010111000000100010110000000000

10001001010110011110100100000101011110000

0001010110011111111111111111001100110101100

The above there is a string is the Binary representation of IPv6 address

3695:C08B:0011:2B3D:20AF:00AC:FFFF:99AC

Non Symbolic Base64 Conversion is as Follows for Adaptive Representation of IPv6 Address

Firstly, divide the binary string into 4 segments by the proportions of 4:2:2:8,

0011 0110 100101 011100 000010 001011 (4 Bytes,32 Bits)

0000 000000 010001 (2 Bytes, 16 Bits)

0010 101100 111101 (2 Bytes, 16 Bits)

0010 000010 101111 000000 001010 110011 111111 111111 111001 100110 101100(8 Bytes, 64 Bits)

Secondly, encode each segment using 6-bit Non Symbolic Base64,

36bS2B 00H 2iz 22i0ApB¹B¹vci

Thirdly, add the period (or dot) as a delimiter,

36bS2B. 00H .2iz .22i0ApB¹B¹vci

Using the steps listed, a conversion program was written to automate the process of converting an IPv6 address from base 16 to Non Symbolic Base64

Advantages and Benefits of Adaptive Representation of IPv6 Address

The whole Internet community and especially Cloud computing, which will have a major reliance on IPv6, will benefit from the proposed scheme in the following aspects of IP-related systems and applications. Shorter notation the original objective of this study was to find a shorter textual representation for IPv6 addressing. The length of an IPv6 address encoded in dot-base62x has a theoretical reduction in length of $(39-26)/39 = 33.3\%$ when compared to the same address in colon hexadecimal, i.e., from 39 to 28 in bytes.

CONCLUSIONS

This research has introduced an effective way of textual representation of IPv6 addresses. The growth of Internet has increased use of ipv6 addresses in recent decades and it will continue to contribute to and reshape the world for many years to come. Cloud computing as the mainstream services of future IT applications will encounter many scenarios where IP addresses are used in plain text representation rather than binary mode. This study reviews the development of current Internet addressing with a primary focus on potential IP evolution.

The literature survey finds some failures those are

- Too long.
- Too many variants.
- Colon (:)
- Incompatible with IPv4.

A new scheme, named as Adaptive Representation of IPv6, has been proposed by this study as a means to encode IPv6 addresses in Non Symbolic Base64 and separate the encoded string with four dots. The proposed method has shorter notation An IPv6 address length reduction of 33.3% or so in theory and which uses 4 segments each segment of 6 bits. a human being remembering 4 segments is better than remembering 5 segments. So definitely it has an advantage.

ACKNOWLEDGEMENTS

We take this opportunity to acknowledge those who have been great support and inspiration through the research work. Our sincere thanks to Mrs.B.RenukaDevi Head of the department of CSE to her diligence and motivation. Special thanks to Vignan's Lara Institute of Science and Technology, for giving us such a nice opportunity to work in the great environment and for providing the necessary facilities during the research and encouragement from time to time. Thanks to our colleagues who have been a source of inspiration and motivation that helped to us. And to all other people who directly or indirectly supported and help us to fulfill our task. Finally, we heartily appreciate our family members for their motivation, love and support in our goal.

REFERENCES

1. Dot-base62x: building a compact and user-friendly text presentation scheme of ipv6 addresses for cloud computing Zhenxing Liu*, Lu Liu*, James Hardy, Ashiq Anjum, Richard Hill and Nick Antonopoulos.
2. Liu Z, Lallie HS, Liu L (2011) "A Hash-based Secure Interface on Plain Connection", Proceedings of CHINACOM'11.ICST.OTG & IEEE Press, Harbin, China.
3. Davies J: Understanding IPv6, Second Edition, Microsoft Press, Redmond, Washington. ISBN-10: 0735624461, 978-0735624467, 2008, pp43-45, 50, 92.
4. RFC 1924 A Compact Representation of IPv6 Addresses. <http://tools.ietf.org/html/rfc1924>
5. RFC 5952 Recommendation for IPv6 Address Text Representation. <http://tools.ietf.org/pdf/rfc5952.pdf>
6. Establishing Link between IPv6 Networks Using IPv4 Clouds1Junaid Qayyum, 2Fayyaz Gul .
7. Illustrating the Impediments for Widespread Deployment of IPv6 ALA HAMARSHEH and MARNIX GOOSSENS.
8. RFC 3513 Internet Protocol Version 6 (IPv6) Addressing Architecture. <http://www.ietf.org/rfc/rfc3513.txt>.
9. Performance Analysis of IPv6 Transition Mechanisms over MPLS .Parisa Grayeli, Shahram Sarkani ,and Thomas Mazzuchi.
10. IPv4 and IPv6 Interoperability Framework for Mobile Cloud Computing Networks Deployment Yvette E. Gelogo1, Sunguk Lee2 and Byungjoo Park1*vv
11. Huston G: IPv4 Address Report, daily generated, Retrieved in August 2011, <http://www.potaroo.net/tools/ipv4/index.html>, 2011.
12. Stockebrand B (2007) IPv6 in Practice—A Unixer's Guide to the Next Generation Internet. Springer, Berlin Heidelberg, ISBN 3-540-24524-3, 978-3-540-24524-7.
13. Hinden R, Deering S: IP Version 6 Addressing Architecture, IETF RFC 4291, 2005, retrieved in August 2011,<http://www.rfc.net/rfc4291.html>.
14. RFC 4648 Base-N Encodings. <http://www.ietf.org/rfc/rfc4648.txt>
15. Google collects statistics about IPv6 adoption in the Internet, <http://www.google.com/ipv6/statistics.html>.
16. Non Symbolic Base64: An Effective Representation of IPv6 Address , Deepthi.S, Prashanti.G, Sandhya Rani.K

