2011 Fourth IEEE International Conference on Utility and Cloud Computing

Dot-base62x: A Compact Textual Representation of IPv6 Address for Clouds

Zhenxing Liu, Lu Liu, James Hardy, Ashiq Anjum, Richard Hill, Nick Antonopoulos
Distributed and Intelligent Systems Research Group, School of Computing and Mathematics, University of Derby
Derbyshire, United Kingdom
{z.liu3, j.hardy2}@unimail.derby.ac.uk, {l.liu, a.anjum, r.hill, n.antonopoulos}@derby.ac.uk

Abstract — Cloud computing has dramatically reshaped the whole IT industry in recent years. With the transition from IPv4 to IPv6, services running in Cloud computing will face problems associated with IPv6 addressing: 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, compact notation (27 bytes), without variants or conflicts with http_URL. The proposal is known as dot-base62x as it is an IPv6 address with Base62x and uses the well-known period (or dot) as a group delimiter instead of the colon. The relative merits and demerits of other works that predate this paper have been reviewed and critically 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, Base62x, colon hexadecimal, Text Encoding/Decoding

I. Introduction

Cloud computing paradigm has emerged as an energy-efficient, fault-tolerant and on-demand approach which enables ubiquitous network accesses to a shared pool of flexibly reconfigurable computing resources. Networks, servers, storage, applications and services can be rapidly deployed with minimal management input or service provider interaction [1]. It is also marketed as a fast, low cost method for small and medium-size business to setup an online presence.

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. It is anticipated that there will be a protracted period of change and that dual-stack IP networking will be utilised for a considerable time.

The reasons why IPv6 is necessary and how the new IP scheme copes with the increasing demand from IT industry can be read from Davis' book [2] and other resources [3,4]. One of the most distinct motivators for change is the depletion of IPv4 addresses, i.e. current IPv4 *Class A* address ranges have been fully allocated, restricting the availability of IP addresses for new Internet users and services. A secondary motivator is that IPv6 has significant potential advantages over IPv4. These advantages fall into two categories:

- ✓ *Changes that address fundamental inadequacies of IPv4.*
  Examples are the four-fold increase in address space (from 2^32 to 2^128), the lack of NAT means that true end-to-end communication will become the norm, Stateless and Stateful Address Configuration can simplify network management) and network "noise" reduction due to the removal of broadcast and increased reliance on multicast.
- ✓ *Advanced features introduced with IPv6.* e.g.
  - Network built-in security. IPsec is mandatory and not optional as a measure to ensure connections are confidential.
  - Mobile IPv6. Permits system and content portability by allowing "roaming" while still maintaining a "home" network address at all times.
  - Improved Quality-of-Service (QoS). A new capability is added in IPv6 to enable the labelling of packets belonging to particular traffic flows for which the sender requests special handling.

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.

Firstly, 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 delimiters) in human readable form which is both challenging to remember and prone to mistakes when read, written or deployed.

Furthermore, the current IPv6 notation of "colon hexadecimal" [6] has other issues, e.g. too many variants of text representation with a single IPv6 address [5], a potential ambiguity with current http_URL and the annoyance of being a "two-key" entry on most keyboards.

Bearing these issues in mind, this paper introduces a novel scheme to present an IPv6 address in Base62x with period (or "dot") delimiters as used in IPv4. This scheme will overcome the highlighted issues and offer other benefits after its implementation. This is the key finding of the study.

The remaining sections of this proposal are organized as follows. Section 2 is a literature review of other works that relate to the issues identified above. Section 3 gives a brief introduction to Base62x notation including its definition, algorithm and usage. Then in section 4 the new scheme, dot-base62x is presented and explained in detail with

978-0-7695-4592-9/11 $26.00 © 2011 IEEE
DOI 10.1109/UCC.2011.70

experiments and analysis. Section 5 re-iterates some of the benefits with dot-base62x notation. A conclusion of the proposal is given in section 6 and there are suggestions for future consideration in section 7.

II. LITERATURE REVIEWS

It is accepted that this paper is not the first to make critical comment on current IPv6 text representation and raising issues as described in section 1. It is also a near certainty that this paper will not the last until those issues have been solved in a better and more acceptable way. In other works technicians and engineers have expressed their opinions about the current IPv6 notation with words like "pretty long" [7] "a bit unwieldy" [4], "ugly", "untidy", "awkward" and "difficult to comprehend and work with".

Since the current IPv6 address scheme was introduced by IETF RFC 1884 in 1995 [8], some work has been done to address current IPv6 text representation issues related to excessive length, appearance/forms of representation and the potential ambiguity with existing http_URL. Here are two representative examples worth mentioned to discuss in detail as below.

A. Elz's informational RFC 1924

RFC 1924 [9] invented a method to present an IPv6 address in base85. The base85 system consists of the following characters list in an ascending order:

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

Base85 uses this character set to express any numerical value, including IPv6 address.

As an example of use, a standard RFC 1884/4291 format IPv6 address of

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

Translates it to a base85 representation as

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

The encoded string is clearly much shorter than the original one, but this is the only apparent benefit. Primarily, 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 available 7 bits and will result in over-length and discontiguous binary strings.

Due to its unusual idea on using the so-called "base 85" to express IPv6 address, this suggestion has been argued for a long time. However, it does show that a new direction for achieving a shorter notation for IPv6 address was recognized very early into its development and it explains the logic process of its author who was trying to cope with the issues born with RFC 1884.

B. Translucent Implementation in Traditional Base 64

Parwez [10] proposed *"another brave idea to present a translucent representation of currently implemented IPv6 address with a more compact and end-user friendly format for IT professionals especially for naïve users in networking environments."*

"Simply 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 to 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 to 3; Maximum number of characters are 22 or more precisely 21.33 characters; …"

Some examples of IPv6 address taken directly from this paper are listed below:

NUML.EDU.PK.ISB-#10.10.20.30
encyclopedia.com.US-02
IT------------------.1
IT+++---.1
IEEE-AaBbCcDdEeFfGgHh3
::1

Although these "addresses" are in a format that is unfamiliar with, they appear far less daunting than the Base85 example given earlier.

This 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 predict that this scheme would be readily accepted by academia or industry as it could introduce more complication than the original RFC 1884/4291.

C. On-going Studies on These Issues

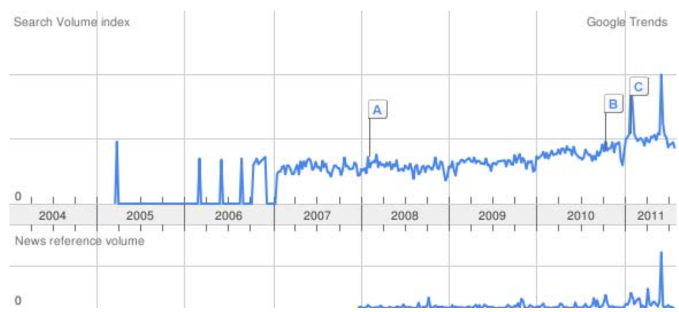

Figure 1. "IPv6 address" searching trends in Google [11]

The well-known problem of depleted of IPv4 addresses means that globally IPv6 addressing is continually attracting more attention than ever before. Most of the papers, publicity, guidance and training, though not directly attempting to resolve addressing issues, will encourage technicians and engineers to challenge, change or accept IPv6. IPv6 is no longer a *future* problem and decisions currently being made are likely to impact networks, communications and the Internet for many years to come. Figure 1 shows the trends of "IPv6 address" in Google's search engine.

III. BASE62X

Base62x was first described in our paper [12]. It is an alternative approach to Base64 for non-alphanumeric characters and is considered 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).

The differences between the original Base64 and Base62x can be seen in Table 1. In the new scheme, the symbols “+”, “/” and “=” are discarded. Instead, the alphabet “x” (or any other one amongst the group of 0-9, a-z and A-Z) is a special tag and subsequently x1 represents number 61, x2 for 62 and x3 for 63. As a result, the new alphabet series are A-Z, a-z (excluding x), 0-9, and x1, x2, x3.

Since there is no symbol used in Base62x index table, it shortens the length of IPv6 address without adversely affecting readability, one of the important requirements of the proposed IPv6 address notation.

TABLE 1. CODES INDEX COMPARISONS

Base62x				***Base64***			
Value	***Enc***	***Value***	***Enc***	***Value***	***Enc***	***Value***	***Enc***
0	0	.		0	A	.	
1	1	.		1	B	.	
2	2	.		2	C	.	
3	3	60	z	3	D	60	8
4	4	61	x1	4	E	61	9
.		62	x2	.		62	+
.		63	x3	.		63	/
.		(tag)	X	.		(pad)	=

Here are some examples of string encoded in Base62x:

aBC OK93
A__B* GLx1VGYe
中文简体 vBYjvfQ7vww0vBsJ

IV. DOT-BASE62X NOTATION OF IPV6 ADDRESS

A. *Definitions*

This proposed new scheme of IPv6 address notation presented by this paper is called *dot-base62x*. The binary and colon-hexadecimal representations of an IPv6 address are shown in Figure 2.

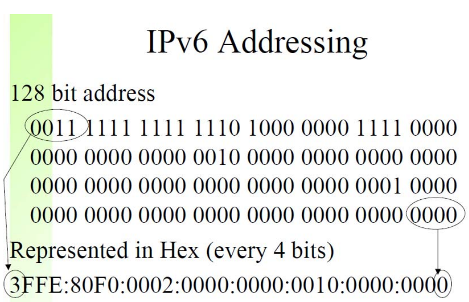

Figure 2. IPv6 address in base 16

This long address is commonly depicted as eight pairs of bytes, but it can also be considered in three sections as shown in Figure 3.

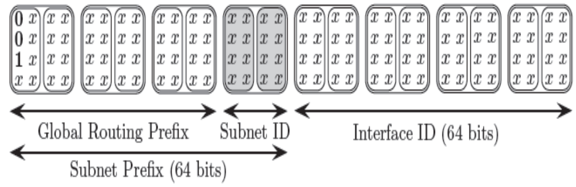

Figure 3. Globally-routed unicast address format address [4]

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, IP6 addressing could be described using the format:

yyy.yyy.yy.yyy.yyy.yy (3.3.2.3.3.2),

where each “y” stands for one byte (8 bits or two characters). After encoding into Base62x, “yyy” (3 bytes, 24bits or six hex characters) will be replaced by “xxxx” (four base62x 6-bit characters) and “yy” (2 bytes, 16 bits or four hex characters) will be replaced by “xxx” (three base62x 6-bit characters) in Base62x.

Therefore, using the proposed new notation scheme, an IPv6 address in Basex62x will be in the general format

xxxx.xxxx.xxx.xxxx.xxxx.xxx (4.4.3.4.4.3),

where each “x” represents any one six-bit character (using the code scheme 0-9, a-z, A-Z, x1-3). 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 dot-base62x notation of IPv6 address and has the following features:

- Encoded in Base62x
- Dot-separated six segments
- Prototype length: 22 codes + 5 dots = 27 characters
- Character range: 0-9, A-Z, a-z
- Case-sensitive

B. *Conversions from/to dot-base62x*

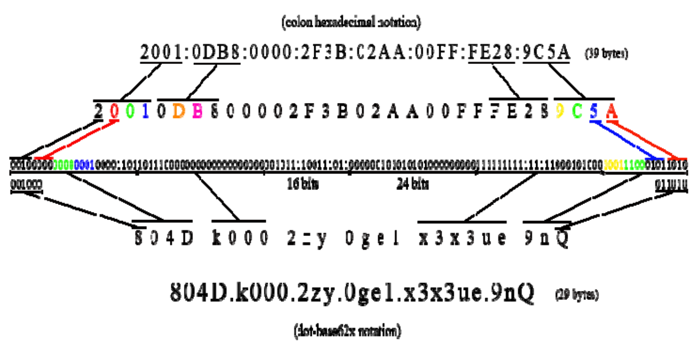

Figure 4. Logic of dot-base62x notation, compared with colon hexadecimal

The process of converting an IPv6 address into dot-base62x can be summarized as these steps:

- S1. Split the given16-byte address into 6 segments as 3:3:2:3:3:2
- S2. Converting each segment into Base62x
- S3. Separate the Base62x encoded string into 4:4:3:4:4:3 as xxxx.xxxx.xxx.xxxx.xxxx.xxx

Figure 4 is a graphical representation of the following example. In the middle of the illustration there is a string

00100000000000000100001101101110000000000000000
00000001011110011101100000010101010100000000
00111111111111111000101000100111000101101 0

which is the binary representation of the IPv6 address

2001:0DB8:0000:2F3B:02AA:00FF:FE28:9C5A.

Firstly, split the binary string into 6 segments by the proportions of 3:3:2:3:3:2,

001000 000000 000100 001101 (3 bytes, 24 bits)
101110 000000 000000 000000 (3 bytes, 24 bits)
0010 111100 111011 (2 bytes, 16 bits)
000000 101010 101000 000000 (3 bytes, 24 bits)
111111 111111 111000 101000 (3 bytes, 24 bits)
1001 110001 011010 (2 bytes, 16 bits)

Secondly, encode each segment using 6-bit Base62x,

804D k000 2zy 0ge0 x3x3ue 9nQ

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

804D.k000.2zy.0ge0.x3x3ue.9nQ

Two more IPv6 addresses conversions have been demonstrated using dot-base62x as below.

Given the IPv6 address in IPv4 style looks as:

128.91.45.157.220.40.0.0.0.0.252.87.212.200.31.255

Its binary format is:

10000000010110110010110110011101110111011100001
01000000000000000000000000000000000000111111
000101011111010100110010000001111111111111 1

When each segment is converted into dot-base62x format, the following string is the result

W5ij.dTme.000.003z.Lx1J8.1x3x3.

As a final example, an IPv6 address is given as

fe80:0000:0000:0000:020c:f1ff:fefd:d2be,

After encoding to dot-base62x, it becomes

x3e00.0000.000.0Wpn.x3x3yx1.DAx2.

C. Comparisons between dot-base62x and colon hexadecimal notation of IPv6 address

The differences between current colon hexadecimal and the proposed dot-base62x notation of IPv6 address have been listed in Table 2 which summarizes a few aspects of these two forms.

No.	**Fields**	**Colon hexadecimal**	**Dot-base62x**
1	Encoding base	Base 16	Base62x
2	Separator	Colon (:)	Dot, full stop (.)
3	Number of separators	7	5
4	Segments/groups	8	6
5	Format address length	39	27
6	Minimum length	2(::)	11(0.0.0.0.0.0)
7	Maximum length	45	49
8	Average length	~37	~28
9	Bits operation	Each 4 bits	Each 6 bits
10	Format	xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx	xxxx.xxxx.yxx.xxxx.xxxx.yxx
11	Example	2001:DB8:0:2F3B:2AA:FF:FE28:9C5A	W5ij.dTme.0.3z.Lx1J8.1x3x3
12	Variants	Multiple forms	Only one
13	Status	IETF RFC	Newly-invented

Table 2. COMPARISONS OF DOT-BASE62X AND COLON HEXADECIMAL

V. ADVANTAGES AND BENEFITS

The whole Internet community and especially Cloud computing which has a major reliance on TCP/IP, will benefit from the proposed scheme in the following aspects of IP-related systems and applications.

A. 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 (39-27)/39= 30.77% when compared to the same address in colon hexadecimal, i.e., from 39 to 27 in bytes. Figure 5 shows the comparisons of lengths of IPv6 encoded in colon hexadecimal and dot-base62x.

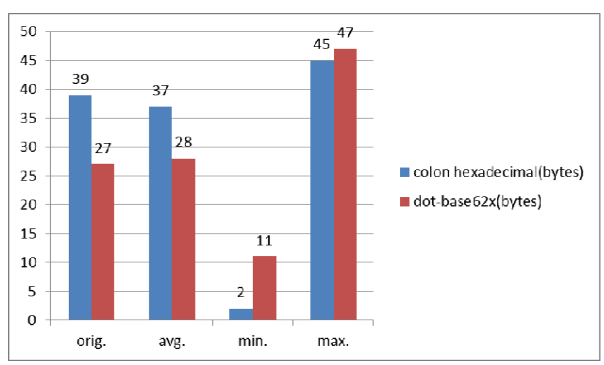

Figure 5. IPv6 address lengths in dot-base62x and hexadecimal

B. *Compact form, less segments, more human-friendly*

The proposed scheme has a more compact form than current colon hexadecimal notation.

Firstly, instead of the eight groups of characters in colon-hexadecimal, there are only six in dot-base62x. This simpler representation makes the scheme more human-friendly.

Secondly, within the six groups there are always two segments, which only consist of three digits, an additional simplification.

Thirdly, an IPv4 address uses 4 character groups, using the dot-62x scheme an IPv6 address has 6 groups which is a more symmetric and aesthetically pleasing form.

Last but not least, the continuous two-key entry requirement of the colon symbol will very quickly lead to many years of tedium and non-standard keyboard mapping that is resolved by returning to the single keystroke period (or dot) separator.

C. *Compatible with IPv4 dot-decimal*

This point is obvious that keeping the identical separator in both versions of IP will maintain consistency in the whole Internet community. People working within the field of networks are already familiar with dot-separated style IP address and will find in more acceptable for the transition from version four to version six if the proposed scheme became widely adopted.

D. *Minimized the number of variants*

IETF RFC 5952 [5] recommends a unified text representation to avoid confusions caused by multiple output forms of colon hexadecimal notation from a single IPv6 address.

Dot-base62x notation avoids this issue by introducing only one method to compress a given single IPv6 address, the identical method which has been used with IPv4. The method is to always suppress the leading zeros. Therefore, any single IPv6 in dot-basex62 has one and only one textual representation in the same way that an IPv4 is only written in one form.

E. *Avoiding conflict with exist http_URL*

Clearly, with the exception of IPv4 itself, there was no intention for IPv6 address notation to conflict with other existing RFC standards. However, the colon symbol serves as a "port" identifier part in current http_URL scheme, which could lead to confusion between a colon hexadecimal address and http_URL.

The current remedy for this conflict introduces further complication by enclosing the IPv6 address in a pair of square brackets before using it as an IP address in http_URL, e.g., http://[2001:db8:0000:0:1::1]:8080/file/to/path?query. Dot-base62x has no such problem, by abandoning colon in its output form and instead using the "dot" as in IPv4, a greater degree of compatibility is maintained. This in turn means more existing IPv4 systems and applications can be made to be seamlessly compatible with IPv6 addresses in dot-base62x.

VI. CONCLUSION

The Internet has revolutionized human history 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.

Literature reviews show that current colon hexadecimal notation of IPv6 address has the following issues when being deployed in cloud computing.

- ✗ Too long. Usually it has 39 characters, sometimes up to 45 characters.
- ✗ Too many variants. A single IPv6 can have several variations in appearance which can cause confusion.
- ✗ Colon (:) conflict with http_URL.
- ✗ Incompatible with IPv4.

A new scheme, named as *dot-base62x*, has been proposed by this study as a means to encode IPv6 addresses in Base62x and separate the encoded string with five dots. The proposed dot-base62x has the following advantages and benefits compared with current colon hexadecimal notation.

- ✓ Shorter notation.
- ✓ Compact shape, less segments, more human-friendly.
- ✓ Compatible with IPv4 dot-decimal.
- ✓ Minimized the number of variants.
- ✓ Avoiding conflict with exist http_URL.

From what we have discussed, we reasonably conclude that current colon hexadecimal notation of IPv6 address is not the best of choice and with high confidence the proposed dot-base62x textual representation is recommended for consideration of adopting in IPv6 address in the coming IT era of Cloud computing.

VII. RECOMMENDATIONS AND FUTURE WORK

IPv6 is unlikely to be the final addressing scheme used on the Internet. For this proposed scheme itself, there are a few options, recommendations and further works to be done.

A. *Fixed-width or various length of IPv6 in Base62x*

Due to three double-digital characters being added in its index table, the length of strings encoded in dot-base62x may vary in a small range without any other compressing involved. This may be a major concern with dot-base62x when compared with colon hexadecimal.

Taking compressing and better compatibility into consideration, we recommend that keeping its varying length is a better choice from a long term and developmental point of view. As varying lengths are unavoidable in all schemes discussed, it is not considered significant that the length may extend 27 bytes in common use to 47 bytes in very rare extremes.

B. *Ambiguous Characters*

Dot-base62x uses all the possible alphanumeric characters in its output form, so it is likely that sometimes one of its output forms consists of these potentially ambiguous characters:

"0" (zero) and "o/O" (letter O)

"1" (one) and "l" (letter L, slightly higher than number one)

"2" (two) and "z/Z" (letter Z)

Though such study goes beyond this report, it is necessary to advise a set of hints to write or display these illegible characters if some practical methods have been found in future work addressing this annoying issue.

C. *Integration of IPv4 addressing*

Dot-base62x uses six "dot separated" groups of characters to fully identify an address. The last four groups are local subnet and specific device. It may be possible to interpret an IPv4 address as a locally sourced dot-base62x address, expediting the change to IPv6 by minimizing equipment changes, and consequently simplifying the program and greatly reducing the associated costs.

ACKNOWLEDGMENT

The work reported in this paper has been supported by the RLTF HECloud Programme (Ref: RLTFD005), Major Program of the National Natural Science Foundation of China, under contact No. 90818028 and China 973 Fundamental R&D Program (No. 2011CB302600).

REFERENCES

[1] Liu, Z., Lallie, H. S., Liu, L. "A Hash-based Secure Interface on Plain Connection", Proceedings of CHINACOM'11, Harbin, China, ICST.OTG & IEEE Press, August 2011.

[2] Davies, J. Understanding IPv6, Second Edition, Microsoft Press, ISBN-10: 0735624461, 978-0735624467, 2008, pp43-45, 50, 92

[3] Huston, G. IPv4 Address Report, daily generated, Retrieved in August 2011, http://www.potaroo.net/tools/ipv4/index.html, 2011

[4] Stockebrand, B. IPv6 in Practice -- A Unixer's Guide to the Next Generation Internet, Springer-Verlag Berlin Heidelberg, ISBN 3-540-24524-3, 978-3-540-24524-7, , 2007, pp22-29

[5] Kawamura, S., Kawashima, M. A Recommendation for IPv6 Address Text Representation, IETF RFC, retrieved in August 2011, http://tools.ietf.org/html/rfc5952

[6] Hinden, R., Deering, S. IP Version 6 Addressing Architecture, IETF RFC 4291, 2005, retrieved in August 2011, http://www.rfc.net/rfc4291.html

[7] Kozierok, C. M. The TCP/IP guide: a comprehensive, illustrated Internet protocols reference, No Starch Press, ISBN 159327047X, 9781593270476, pp7, 379

[8] Hinden, R., Deering, S. IP Version 6 Addressing Architecture, IETF RFC 1884, 1995, retrieved in August 2011, http://tools.ietf.org/html/rfc1884

[9] Elz, R. "A Compact Representation of IPv6 Addresses", IETF RFC 1924, 1996, retrieved in August 2011, http://tools.ietf.org/html/rfc1924

[10] Parwez, R., Haq, E. et al. Translucent Implementation of IPv6 Addressing Scheme in Communication Networks, International Journal for Infonomics (IJI), 2010, Volume 3, Issue 3

[11] Google. Google Trends, retrieved in August 2011, http://www.google.com/trends?q=ipv6+address&ctab=0&geo=all&date=all

[12] Liu, Z., Liu L., et al. Base62x: An Alternative Approach to Base64 for non-Alphanumeric Characters, Proceedings of ICNC & FSKD, Shanghai, China, July 2011, IEEE Computer Society Press