IPv6 Maintenance Working Group	S. Kawamura	
Internet-Draft	NEC BIGLOBE, Ltd.	
Intended status: Standards Track	M. Kawashima	
Expires: May 14, 2010	NEC AccessTechnica, Ltd.	
	November 10, 2009	

TOC

A Recommendation for IPv6 Address Text Representation
draft-ietf-6man-text-addr-representation-02

Abstract

As IPv6 network grows, there will be more engineers and also non-engineers who will have the need to use an IPv6 address in text. While the IPv6 address architecture RFC 4291 section 2.2 depicts a flexible model for text representation of an IPv6 address, this flexibility has been causing problems for operators, system engineers, and users. This document will describe the problems that a flexible text representation has been causing. This document also recommends a canonical representation format that best avoids confusion. It is expected that the canonical format is followed by humans and systems when representing IPv6 addresses as text, but all implementations must accept and be able to handle any legitimate RFC4291 format.

Status of this Memo

Copyright Notice

Table of Contents

---

1. Introduction

TOC

A single IPv6 address can be text represented in many ways. Examples are shown below.

```
2001:db8:0:0:1:0:0:1

2001:0db8:0:0:1:0:0:1

2001:db8::1:0:0:1

2001:db8::0:1:0:0:1

2001:0db8::1:0:0:1

2001:db8:0:0:1::1

2001:db8:0000:0:1::1

2001:DB8:0:0:1::1
```

All the above point to the same IPv6 address. This flexibility has caused many problems for operators, systems engineers, and customers. The problems will be noted in Section 3 (Problems Encountered with the Flexible Model). Also, a canonical representation format to avoid problems will be introduced in Section 4 (A Recommendation for IPv6 Text Representation).

TOC

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

---

2. Text Representation Flexibility of RFC4291

TOC

Examples of flexibility in Section 2.2 of [RFC4291] (Hinden, R. and S. Deering, “IP Version 6 Addressing Architecture,” February 2006.) are described below.

2.1. Leading Zeros in a 16 Bit Field

TOC

> 'It is not necessary to write the leading zeros in an individual field.'

In other words, it is also not necessary to omit leading zeros. This means that, it is possible to select from such as the following example. The final 16 bit field is different, but all these addresses mean the same.

> 2001:db8:aaaa:bbbb:cccc:dddd:eeee:0001
>
> 2001:db8:aaaa:bbbb:cccc:dddd:eeee:001
>
> 2001:db8:aaaa:bbbb:cccc:dddd:eeee:01
>
> 2001:db8:aaaa:bbbb:cccc:dddd:eeee:1

2.2. Zero Compression

TOC

> 'A special syntax is available to compress the zeros. The use of "::" indicates one or more groups of 16 bits of zeros.'

It is possible to select whether or not to omit just one 16 bits of zeros.

> 2001:db8:aaaa:bbbb:cccc:dddd::1

> 2001:db8:aaaa:bbbb:cccc:dddd:0:1

In case where there are more than one zero fields, there is a choice of how many fields can be shortened. Examples follow.

> 2001:db8:0:0:0::1
>
> 2001:db8:0:0::1
>
> 2001:db8:0::1
>
> 2001:db8::1

In addition, [RFC4291] (Hinden, R. and S. Deering, “IP Version 6 Addressing Architecture,” February 2006.) in section 2.2 notes,

> 'The "::" can only appear once in an address.'

This gives a choice on where, in a single address to compress the zero. Examples are shown below.

> 2001:db8::aaaa:0:0:1
>
> 2001:db8:0:0:aaaa::1

2.3. Uppercase or Lowercase

TOC

[RFC4291] (Hinden, R. and S. Deering, “IP Version 6 Addressing Architecture,” February 2006.) does not mention about preference of uppercase or lowercase. Various flavors are shown below.

> 2001:db8:aaaa:bbbb:cccc:dddd:eeee:aaaa
>
> 2001:db8:aaaa:bbbb:cccc:dddd:eeee:AAAA
>
> 2001:db8:aaaa:bbbb:cccc:dddd:eeee:AaAa

---

3. Problems Encountered with the Flexible Model

TOC

TOC

3.1. Searching

3.1.1. General Summary

TOC

A search of an IPv6 address if conducted through a UNIX system is usually case sensitive and extended options to allow for regular expression use will come in handy. However, there are many applications in the Internet today that do not provide this capability. When searching for an IPv6 address in such systems, the system engineer will have to try each and every possibility to search for an address. This has critical impacts especially when trying to deploy IPv6 over an enterprise network.

3.1.2. Searching Spreadsheets and Text Files

TOC

Spreadsheet applications and text editors on GUI systems, rarely have the ability to search for a text using regular expression. Moreover, there are many non-engineers (who are not aware of case sensitivity and regular expression use) that use these application to manage IP addresses. This has worked quite well with IPv4 since text representation in IPv4 has very little flexibility. There is no incentive to encourage these non-engineers to change their tool or learn regular expression when they decide to go dual-stack. If the entry in the spreadsheet reads, 2001:db8::1:0:0:1, but the search was conducted as 2001:db8:0:0:1::1, this will show a result of no match. One example where this will cause problem is, when the search is being conducted to assign a new address from a pool, and a check was being done to see if it was not in use. This may cause problems to the end-hosts or end-users. This type of address management is very often seen in enterprise network s and also in ISPs.

3.1.3. Searching with Whois

TOC

The "whois" utility is used by a wide range of people today. When a record is set to a database, one will likely check the output to see if the entry is correct. If an entity was recorded as 2001:db8::/48, but the whois output showed 2001:0db8:0000::/48, most non-engineers would think that their input was wrong, and will likely retry several times or make a frustrated call to the database hostmaster. If there was a need to register the same address on different systems, and each system showed a different text representation, this would confuse people even more. Although this document focuses on addresses rather than prefixes, this is worth mentioning since problems encountered are mostly equal.

3.1.4. Searching for an Address in a Network Diagram

TOC

Network diagrams and blue-prints contain IP addresses as allocated to system devices. In times of trouble shooting, there may be a need to search through a diagram to find the point of failure (for example, if a traceroute stopped at 2001:db8::1, one would search the diagram for that address). This is a technique quite often in use in enterprise networks and managed services. Again, the different flavors of text representation will result in a time-consuming search, leading to longer MTTR in times of trouble.

3.2. Parsing and Modifying

TOC

3.2.1. General Summary

TOC

With all the possible text representation ways, each application must include a module, object, link, etc. to a function that will parse IPv6 addresses in a manner that no matter how it is represented, they will mean the same address. This is not too much a problem if the output is to be just 'read' or 'managed' by a network engineer. However, many system engineers who integrate complex computer systems to corporate customers will have difficulties finding that their favorite tool will not have this function, or will encounter difficulties such as having to rewrite their macro's or scripts for their customers.

3.2.2. Logging

TOC

If an application were to output a log summary that represented the address in full (such as 2001:0db8:0000:0000:1111:2222:3333:4444), the output would be highly unreadable compared to the IPv4 output. The address would have to be parsed and reformed to make it useful for human reading. Sometimes, logging for critical systems is done by mirroring the same traffic to two different systems. Care must be taken that no matter what the log output is, the logs should be parsed so they will mean the same.

3.2.3. Auditing: Case 1

TOC

When a router or any other network appliance machine configuration is audited, there are many methods to compare the configuration information of a node. Sometimes, auditing will be done by just comparing the changes made each day. In this case, if configuration was done such that 2001:db8::1 was changed to 2001:0db8:0000:0000:0000:0000:0000:0001 just because the new engineer on the block felt it was better, a simple diff will tell you that a different address was configured. If this was done on a wide scale network, people will be focusing on 'why the extra zeros were put in' instead of doing any real auditing. Lots of tools are just plain 'diff's that do not take into account address representation rules.

3.2.4. Auditing: Case 2

TOC

Node configurations will be matched against an information system that manages IP addresses. If output notation is different, there will need to be a script that is implemented to cover for this. An SNMP GET of an interface address and text representation in a humanly written text file is highly unlikely to match on first try.

3.2.5. Verification

TOC

Some protocols require certain data fields to be verified. One example of this is X.509 certificates. If an IPv6 address was embedded in one of the fields in a certificate, and the verification was done by just a simple textual comparison, the certificate may be mistakenly shown as being invalid due to a difference in text representation methods.

3.2.6. Unexpected Modifying

TOC

Sometimes, a system will take an address and modify it as a convenience. For example, a system may take an input of 2001:0db8:0::1 and make the output 2001:db8::1 (which is seen in some RIR databases).

If the zeros were input for a reason, the outcome may be somewhat unexpected.

3.3. Operating

TOC

3.3.1. General Summary

TOC

When an operator sets an IPv6 address of a system as 2001:db8:0:0:1:0:0:1, the system may take the address and show the configuration result as 2001:DB8::1:0:0:1. A distinguished engineer will know that the right address is set, but an operator, or a customer that is communicating with the operator to solve a problem, is usually not as distinguished as we would like.

3.3.2. Customer Calls

TOC

When a customer calls to inquire about a suspected outage, IPv6 address representation should be handled with care. Not all customers are engineers nor have the same skill in IPv6 technology. The NOC will have to take extra steps to humanly parse the address to avoid having to explain to the customers that 2001:db8:0:1::1 is the same as 2001:db8::1:0:0:0:1. This is one thing that will never happen in IPv4 because IPv4 address cannot be abbreviated.

3.3.3. Abuse

TOC

Network abuse is reported along with the abusing IP address. This 'reporting' could take any shape or form of the flexible model. A team that handles network abuse must be able to tell the difference between a 2001:db8::1:0:1 and 2001:db8:1::0:1. Mistakes in the placement of the "::" will result in a critical situation. A system that handles these incidents should be able to handle any type of input and parse it in a correct manner. Also, incidents are reported over the phone. It is unnecessary to report if the letter is an uppercase or lowercase. However, when a letter is spelled uppercase, people tend to clarify that it is uppercase, which is unnecessary information.

3.4. Other Minor Problems

TOC

3.4.1. Changing Platforms

TOC

When an engineer decides to change the platform of a running service, the same code may not work as expected due to the difference in IPv6 address text representation. Usually, a change in a platform (e.g. Unix to Windows, Cisco to Juniper) will result in a major change of code, but flexibility in address representation will increase the work load.

3.4.2. Preference in Documentation

TOC

A document that is edited by more than one author, may become harder to read.

3.4.3. Legibility

TOC

Capital case D and 0 can be quite often misread. Capital B and 8 can also be misread.

4. A Recommendation for IPv6 Text Representation

TOC

A recommendation for a canonical text representation format of IPv6 addresses is presented in this section. The recommendation in this document is one that, complies fully with [RFC4291] (Hinden, R. and S. Deering, “IP Version 6 Addressing Architecture,” February 2006.), is implemented by various operating systems, and is human friendly. The recommendation in this document SHOULD be followed by systems when generating an address to represent as text, but all implementations MUST accept any legitimate [RFC4291] (Hinden, R. and S. Deering, “IP Version 6 Addressing Architecture,” February 2006.) format. It is advised that humans also follow these recommendations when spelling an address.

TOC

4.1. Handling Leading Zeros in a 16 Bit Field

Leading zeros should be chopped for human legibility and easier searching. Also, a single 16 bit 0000 field should be represented as just 0. Place holder zeros are often cause of misreading.

4.2. "::" Usage

TOC

4.2.1. Shorten As Much As Possible

TOC

The use of "::" should be used to its maximum capability (i.e. 2001:db8::0:1 is not considered as clean representation).

4.2.2. Handling One 16 Bit 0 Field

TOC

"::" should not be used to shorten just one 16 bit 0 field for it would tend to mislead that there are more than one 16 bit field that is shortened.

4.2.3. Choice in Placement of "::"

TOC

When there is an alternative choice in the placement of a "::", the longest run of consecutive 16 bit 0 fields should be shortened (i.e. latter is shortened in 2001:0:0:1:0:0:0:1). When the length of the consecutive 16 bit 0 fields are equal (i.e. 2001:db8:0:0:1:0:0:1), the former is shortened. This is consistent with many current implementations. One idea to avoid any confusion, is for the operator to not use 16 bit field 0 in the first 64 bits. By nature IPv6 addresses are usually assigned or allocated to end-users from a prefix of 32 bits or longer (typically 48 bits or longer).

TOC

4.3. Lower Case

Recent implementations tend to represent IPv6 address as lower case. It is better to use lower case to avoid problems such as described in section 3.3.3 and 3.4.3.

---

5. Text Representation of Special Addresses

TOC

Addresses such as IPv4-Mapped IPv6 addresses, ISATAP [RFC5214] (Templin, F., Gleeson, T., and D. Thaler, “Intra-Site Automatic Tunnel Addressing Protocol (ISATAP),” March 2008.), and IPv4-translated addresses [RFC2765] (Nordmark, E., “Stateless IP/ICMP Translation Algorithm (SIIT),” February 2000.) have IPv4 addresses embedded in the low-order 32 bits of the address. These addresses have special representation that may mix hexadecimal and decimal notations. In cases where there is a choice of whether to express the address as fully hexadecimal or hexadecimal and decimal mixed, and if the address type can be distinguished as having IPv4 addresses embedded in the lower 32 bits solely from the 128bits of the address field itself, mixed notation is the better choice. However, there may be situations where hexadecimal representation is chosen to meet certain needs. Addressing those needs is out of the scope of this document. The text representation method noted in Section 4 (A Recommendation for IPv6 Text Representation) should be applied for the leading hexadecimal part (i.e. ::ffff:192.0.2.1 instead of 0:0:0:0:0:ffff:192.0.2.1).

---

6. Notes on Combining IPv6 Addresses with Port Numbers

TOC

When IPv6 addresses and port numbers are represented in text combined together, there seems to be many different ways to do so. Examples are shown below.

- [2001:db8::1]:80
- 2001:db8::1:80
- 2001:db8::1.80
- 2001:db8::1 port 80
- 2001:db8::1p80
- 2001:db8::1#80

The situation is not much different in IPv4, but the most ambiguous case with IPv6 is the second bullet. This is due to the "::"usage in IPv6 addresses. This style is not recommended for its ambiguity. The [] style as expressed in [RFC3986] (Berners-Lee, T., Fielding, R., and L. Masinter, “Uniform Resource Identifier (URI): Generic Syntax,” January 2005.) is recommended. Other styles are acceptable when cross-platform portability does not become an issue.

---

7. Conclusion

TOC

The recommended format of text representing an IPv6 address is summarized as follows.

> (1) omit leading zeros in a 16 bit field
>
> (2) when using "::", shorten consecutive zero fields to their maximum extent (leave no zero fields behind).
>
> (3) "::" used where shortens address the most
>
> (4) "::" used in the former part in case of a tie breaker
>
> (5) do not shorten one 16 bit 0 field, but always shorten when there are two or more consecutive 16 bit 0 fields
>
> (6) use lower case

Hints for developers are written in the Appendix section.

---

8. Security Considerations

TOC

None.

---

9. IANA Considerations

TOC

None.

TOC

10. Acknowledgements

The authors would like to thank Jan Zorz, Randy Bush, Yuichi Minami, Toshimitsu Matsuura for their generous and helpful comments in kick starting this document. We also would like to thank Brian Carpenter, Akira Kato, Juergen Schoenwaelder, Antonio Querubin, Dave Thaler, Brian Haley, Suresh Krishnan, Jerry Huang, Roman Donchenko, Heikki Vatiainen for their input. Also a very special thanks to Ron Bonica, Fred Baker, Brian Haberman, Robert Hinden, Jari Arkko, and Kurt Lindqvist for their support in bringing this document to the light of IETF working groups.

---

11. References

TOC

11.1. Normative References

TOC

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC4291]	Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture," RFC 4291, February 2006 (TXT).

11.2. Informative References

TOC

[RFC2765]	Nordmark, E., "Stateless IP/ICMP Translation Algorithm (SIIT)," RFC 2765, February 2000 (TXT).
[RFC3986]	Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," STD 66, RFC 3986, January 2005 (TXT, HTML, XML).
[RFC4038]	Shin, M-K., Hong, Y-G., Hagino, J., Savola, P., and E. Castro, "Application Aspects of IPv6 Transition," RFC 4038, March 2005 (TXT).
[RFC5214]	Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)," RFC 5214, March 2008 (TXT).

Appendix A. For Developers

TOC

We recommend that developers use display routines that conform to these rules. For example, the usage of getnameinfo() with flags argument

NI_NUMERICHOST in FreeBSD 7.0 will give a conforming output, except for the special addresses notes in Section 5 (Text Representation of Special Addresses). The function inet_ntop() of FreeBSD7.0 is a good C code reference, but should not be called directly. See [RFC4038] (Shin, M-K., Hong, Y-G., Hagino, J., Savola, P., and E. Castro, “Application Aspects of IPv6 Transition,” March 2005.) for details.

Appendix B. Prefix Issues

TOC

Problems with prefixes are just the same as problems encountered with addresses. Text representation method of IPv6 prefixes should be no different from that of IPv6 addresses.

Authors' Addresses

TOC

	Seiichi Kawamura
	NEC BIGLOBE, Ltd.
	14-22, Shibaura 4-chome
	Minatoku, Tokyo 108-8558
	JAPAN
Phone:	+81 3 3798 6085
Email:	kawamucho@mesh.ad.jp
	Masanobu Kawashima
	NEC AccessTechnica, Ltd.
	800, Shimomata
	Kakegawa-shi, Shizuoka 436-8501
	JAPAN
Phone:	+81 537 23 9655
Email:	kawashimam@necat.nec.co.jp