Internet-Draft
Expires: 26 April, 2007

Pars Mutaf
Institut National des Telecommunications
Evry, France
26 October, 2006

Human-regenerable IPv6 interface identifiers and addresses
<draft-mutaf-ipv6humid-02.txt>

Status of this Memo

Copyright Notice

Abstract

This document defines the human-regenerable IPv6 interface identifiers and addresses.

1. Introduction

This document defines the human-regenerable IPv6 interface identifiers and addresses. They are constructed from easily remembered or guessed character strings input by the user. Human-regenerable addresses can bring convenience in some ad-hoc scenarios, i.e. when a centralized DNS service is not available.

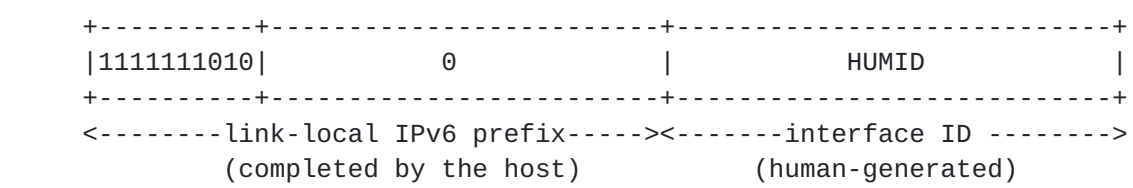
This document does not specify the type of character strings that can be used in human-regenerable IPv6 addresses. Human names, pseudonyms, other simple strings with different meanings, or short phrases may be used depending on user imagination and needs.

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].

2. Human-regenerable IPv6 addresses

This document refers to the type of IPv6 addresses where the leftmost 64 bits of a 128-bit address form the subnet prefix and the rightmost 64 bits of the address form the interface identifier [IP6ARCH]. Human-regenerable addresses are based on a construct called HUMID (HUMan-regenerable interface ID). HUMIDs are constructed by computing a cryptographic hash of an easily remembered or guessed character string, such as a "human name". For example, a hand-held personal device e.g. cellular phone or personal digital assistant can be given the HUMID constructed from its user's first and last names. In some cases, pseudonyms or other simple strings may also be used, depending on user imagination and needs.

On the same link, a host can easily contact another host at its HUMID (modulo human typing erroneous name). The link-local prefix is constant, well-known and specified in [IP6ARCH]. A hosts on the same link can be reachable at its IPv6 address constructed as follows:



The HUMID interface identifier is constructed using the string input by the user, and the prefix is completed by host. The HUMID can be based on the user's first and last name, for example. The resultant address is human-regenerable. Any user who knows the name, can regenerate the same IPv6 address and reach the host. Similarly, the link-local prefix will be completed by the host.

If well-known, other subnet prefixes than the link-local IPv6 prefix MAY be used.

The following steps are required for address configuration:

- The user input character string, possibly pre-formatted as described in Appendix A, MUST be input to the SHA-1 [SHA] hash function. The HUMID interface identifier is obtained by taking the leftmost 64 bits of the 160-bit SHA-1 hash value, and setting bit 6 (the left-most bit is numbered 0) to zero. This creates a HUMID with the universal/local bit indicating local significance only.

- The 64-bit subnet prefix and the 64-bit HUMID are concatenated to form a 128-bit IPv6 address with the subnet prefix on the left and interface identifier on the right, as specified in [IP6ARCH].

The resultant IPv6 address MUST be tested for local uniqueness using the Duplicate Address Detection (DAD) mechanism specified in [SAA]. Upon detecting a duplicate address, the user MUST be notified. The user MUST try another character string.

3. Usage example: Ad-hoc Network

In the following discussion, a wireless ad-hoc networking scenario is described and the address/name configuration complexity is analyzed. It is shown that human-regenerable IPv6 addresses bring convenience.

Users:

Alice, Franck, Mary, David, Eric, Carol

Each user has a personal portable device equipped with ad-hoc mode wireless interface e.g. 802.11. Each user may need communicate with any other user, and hence IPv6 addresses and/or names have to be configured and distributed in a convenient way.

Solution #1:

```
Alice  configures a simple IPv6 address fe80::1
Franck configures a simple IPv6 address fe80::2
Mary   configures a simple IPv6 address fe80::3
David  configures a simple IPv6 address fe80::4
Eric   configures a simple IPv6 address fe80::5
Carol  configures a simple IPv6 address fe80::6
```

This approach is not practical. The users need to communicate (orally, for example) with each other during address configuration. Otherwise address collisions are very probable. It is unclear who should configure which address. It is also difficult to remember who configured which address.

Solution #2:

```
Alice  configures a unique address fe80::212 :3fff:fe79:f820
Franck configures a unique address fe80::86f7:e437:faa5:a7fc
```

```
Mary   configures a unique address fe80::e9d7:1f5e:e7c9:2d6d
David  configures a unique address fe80::84a5:1684:1ba7:7a5b
Eric   configures a unique address fe80::3c36:3836:cf4e:1666
Carol  configures a unique address fe80::58e6:b3a4:14a1:e090
```

This approach has two problems. Firstly, the addresses are hard to remember and hard to pronounce. Each address can be given a simple name and stored in each device (for example, in the /etc/hosts file in Unix systems). However, it is unclear how the addresses can be distributed. Secondly, this method is vulnerable to loss of state. It is very hard (if not impossible) to recover from loss of state.

Solution #3:

Each user configures an address from his/her human name. Assuming that the users know each other, they can easily regenerate each other's address at any time in a stateless manner. (See Appendix B for practical details on using human-regenerable addresses).

This document defines and recommends Solution #3.

4. Security considerations

Human-regenerable addresses are not different from a regular IPv6 address. They can be impersonated. For example, sending a short message to a destination user's human-regenerated address is not secure. There is no guarantee that the message will be received by the intended destination. Human-regenerable address ownership of a responder can be verified during a voice session, since the initiator will recognize the responder's voice. The secure use of human-regenerable addresses is therefore better ensured with interactive voice communication.

5. Conclusion

This document described the human-regenerable IPv6 interface identifiers and addresses. In some ad-hoc scenarios, human-regenerable IPv6 addresses can bring convenience.

Human-regenerable IPv6 addresses may have other applications that are not described in this document. For example, interface identifiers constructed from human name may be used as a replacement for the stateful "phone book" in cellular hosts. Currently cellular device identifiers are not publicly available, for privacy reasons probably. There is also anectodical evidence that users often loose the list of their correspondent nodes (accidentally upon loss of state, or when their device is lost/stolen/changed). A special protocol may be built ontop of human-regenerable IPv6 addresses, in order to overcome these difficulties: initiator can generate a request to the host "John Smith", and John Smith's host can return its phone number (if accepted by John Smith). The design of this protocol (for example, handling of name collisions) falls beyond the scope of this document.

References

[IP6ARCH] Hinden, R. and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture", RFC 3513, April 2003.

[SAA] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", draft-ietf-ipv6-rfc2462bis-08 (work in progress), May 2005.

[SHA] National Institute of Standards and Technology, "Secure Hash Standard", Federal Information Processing Standards Publication FIPS PUB 180-1, April 1995, <http://www.itl.nist.gov/fipspubs/fip180-1.htm>.

[DNS] Mockapetris, P., "DOMAIN NAMES - CONCEPTS AND FACILITIES", RFC 1035, November 1987.

Author's Address

Pars Mutaf
Institut National des Telecommunications
Email: pars.mutaf@int-evry.fr

Appendix A. Character string disambiguation

A same character string may be entered in different ways (by different users, or by the same user at different times). For example, the following string may be entered as:

Jean-Francois Le Roux
Jean Francois Le Roux
jean-francois le roux
jean francois le roux
Jean-Francois Leroux
Jean Francois Leroux
jean-francois leroux
jean francois leroux

A user cannot be reproached for entering the same character string differently. More than one string representations may be considered correct, especially in the computer world. One user may use a particular upper case character, but another user may not. A user may use a white space between two

words, but another user may use a point or dash, etc. When input to a cryptographic hash function (as described above), these nuances will yield different outputs and foil the interoperability of human-regenerable identifiers addresses.

This document therefore recommends the following techniques:

Special characters that lead to doubtful concatenation (white space, dash, point, etc.) SHOULD be removed. A character SHOULD be removed by being replaced by the character on the right (if any). All upper case letters SHOULD be converted to their lower case counterpart. When these rules are applied to the above given examples, they all become:

```
jeanfrancoisleroux
```

regardless of how they are entered by the user. This facilitates the successful matching of a HUMID generated by different users. String ending characters (such as '\0' in C language) may also lead to doubt. They SHOULD NOT be input to the hash function.

Appendix B. Applying human-regenerable addresses

This section presents a possible technique for applying human-regenerable IPv6 addresses. Other techniques, using a graphical user interface for example, may also be applied. In this section, it is assumed that a command called "humid" is available. The command takes a character string and outputs a link-local IPv6 address as follows (Unix jargon):

```
commandline> humid "hello world"
fe80::6adf:b183:a4a2:c94a
commandline>
```

An IPv6 address can be configured as follows (assuming that network the network interface is "eth0"):

```
commandline> ifconfig eth0 inet6 add `humid "hello world"`
commandline>
```

The host "hello world" can be pinged as follows:

```
commandline> ping6 -I eth0 `humid "hello world"`
```

The host "hello world" can be reached through ssh, as follows:

```
commandline> ssh `humid "hello world"`%eth0
```

Alternatively, the humid command can be used for bootstrapping the /etc/hosts file and using standard name resolution routines. For example the following command:

```
commandline> echo "alpha `humid alpha`" >> /etc/hosts
```

will add the entry

```
alpha fe80::be76:331b:95df:c399
```

to the /etc/hosts file. At this point the host "alpha" can be reached using standard name resolution routines.