An Extension of Format for IPv6 Scoped Addresses

<draft-ietf-ipngwg-scopedaddr-format-01.txt>

JINMEI, Tatuya
Toshiba Corporation/KAME Project
jinmei@{isl.rdc.toshiba.co.jp, kame.net}

ONOE, Atsushi
Sony Corporation/KAME Project
onoe@{sm.sony.co.jp, kame.net}

Main Changes from the 00 draft

- Define a specific format
  - As a result of cooperation with MSR for unification
- Backgrounds
  - Not just an impromptu idea, but a requirement from operation
- Omitting Scope Identifiers
- Combinations of Delimiter Characters
  - prefix, in URL's format
- Moved Implementation related issues to appendices
  - extensions to the basic API
  - KAME's implemenation with precise description

Backgorounds

- Network admins have to cope even with link-locals
  - when a routing trouble occurs, global addresses might not be used to fix the trouble.
  - should use a router’s link-local to telnet to a bad router
- EBGP peering
  - using specific ISP’s global addresses would be unfair
  - using link-local would be better in a neutral IX
  - link-local addresses should be specified in a router’s configuration file
- We should disambiguate links of such link-local addresses;
  - a router usually connects to multiple links
  - we can’t just type "fe80::1"

Proposed Format

- <scoped_address>%<scope_id>

- scoped_address: IPv6 scoped address
  - unicast: link-locals, site-locals
  - multicast: all non-globals
  - should not be used for global addresses, but we didn't prohibit using.
    - an implementation MAY allow users to use the format for globals

- %: delimiter character
  - seems to have less conflicts than other candidates

- scope_id:
  - decimal number
  - (optional) implementation dependent notation
  - MUST uniquely specify a single zone of a scope

Examples

- Using numerical identifiers
  - fe80::1234%1
  - fec0::5678%2
  - ff02::9abc%5
  - ff08::def0%10
- Using interface names as identifiers
  - fe80::1234%ne0
  - fec0::5678%ether2
  - ff02::9abc%pvc1.3
  - ff08::def0%interface10
  - Note: there might be some confusion in use of interface names for identifiers of a larger scope.
    - When more than one interface belongs to a same site, which interface should be used?
    - A mapping function from an address to a name would encounter a same kind of problem.

Omitting Scope Identifiers

- An implementation SHOULD support a "default" zone
  - a user can ommit the explicit scope portion
- An implementation MAY even omit the whole functionality of the extension when there is no ambiguity
  - e.g. on an end host with a single interface

Combinations of Other Delimiter Characters

- With prefixes:
  - <scoped_addr>%<scope_id>/<prefix_length>
  - e.g. fec0:0:0:1::/64(site ID = 3) -> fec0:0:0:1::%3/64

- in URL's:
  - [<scoped_addr>%<scope_id>]
  - e.g. http://[fec0:0:0:2::1234%10]:80/index.html

Extensions to API

- Extensions to getaddrinfo() and getnameinfo()
  - basically no changes from the previous draft
  - precise description of KAME's implementation was added as an appendix

- Revise the basic API
  - A new draft will soon be available

Implementations

- MSR
  - supports the extended format and the extension to API
  - will be incorporated to Windows 2000

- KAME
  - Use interface names as <scope_id> for link-locals
    - assuming one-to-one mapping between I/Fs and links
  - Use numeric identifers for other scopes
    - need more experience for those scopes
  - Also included in FreeBSD 4.0, NetBSD current, OpenBSD current

Other Related Issues

- Using common identifier in a zone
  - e.g. fec0::1234%foo.com within the site for foo.com
  - would be useful, but it is hard to distribute the common identifier
    - need an MZAP like protocol?
    - for now, we'd leave this issue outside the scope.

- new proposal for site-locals
  - introduce a unique site identifier embedded in a site-local address
    - e.g. fec0:1::abcd ("1" is the unique ID)
    - can appear on a wire
  - not a replacement of our draft
    - we can't use this approach to link-locals, for which we need our extended format
    - both proposals can coexit
  - we can discuss both two proposals separately

Next Step

- Clarifications from related documents
  - scoped address architecture draft
  - scoped routing draft
  - revised basic API draft

- Authors hope that the draft proceeds to an RFC status
  - not sure about the appropriate category
    - INFORMATIONAL / EXPERIMENTAL / PS?