# The U and G bits in IPv6 Interface Identifiers

draft-carpenter-6man-ug-01

**Brian Carpenter**
**Sheng Jiang**

*IETF 86*
**March 2013**

# Background

- RFC 4291 assumes that the normal case is to transform a MAC address into an IID, preserving the IEEE u and g bits (inverting u).

  - For unicast, u could be 0 or 1, g should be 0 (but the algorithm does not check)

- Numerous other forms of IID invented, e.g.:

  - temporary addresses (RFC 4941)

  - CGAs and HBAs

  - stable privacy addresses

  - 4rd mapped addresses

# Inconsistencies (1)

- In CGAs and HBAs, u = g = 0.

- In temporary addresses, u = 0 but g is variable. stable-privacy-addresses proposes the same.

- 4rd proposes u = g = 1.

- Reserved Subnet Anycast Addresses have u = 0, g = 1.

- Under /127 prefixes, u and g are both variable.

- The idea that these bits have semantics derived from IEEE MAC addresses is clearly bogus.

# Inconsistencies (2)

- In any case, there is evidence from the field that even in IEEE MAC addresses, duplicate addresses are widespread, so the u bit is untrustworthy.

- We can conclude that the state of the u and g bits conveys no meaning in an IID; they are "just bits".

- *Note:* ILNP does have the constraint that its Node Identifiers must be unique within a given site, but as we have just shown, the state of the u bit does not in any way guarantee this.

# The problem

- Whenever a new IID format is proposed, there is confusion caused by
  a) the implication in RFC 4291 that all IIDs are in Modified EUI-64 format
  b) the statement in RFC 4291 that

  ```
  The use of the universal/local bit in the
  Modified EUI-64 format identifier is to allow
  development of future technology that can take
  advantage of interface identifiers with
  universal scope.
  ```

- a) is false and b) is based on a false premise.

# Residual usefulness of u and g bits

- If an IID is known or guessed to have been created according to RFC 4291, it could be transformed back into a MAC address. This can be helpful during fault diagnosis.

- If each method of IID creation specifies the values of u and g, and each new method is carefully designed, these bits reduce the chances of duplicate IIDs. (But DAD remains essential.)

# Proposed updates to RFC 4291 (1)

- The EUI-64 to IID transformation defined in RFC 4291 MUST be used for all cases where an IID is derived from an IEEE address.

- Specifications of other forms of IID will either specify explicitly how the u and g bits are set, or will simply include them as part of a field within the IID.

- The u and g bits in an IID have no semantics. The whole IID should be viewed as opaque by third parties.

# Proposed updates to RFC 4291(2)

- In the following statement, the reference to "Modified EUI-64" applies only to IIDs actually derived from an IEEE address:

  **For all unicast addresses, except those that start with the binary value 000, Interface IDs are required to be 64 bits long and to be constructed in Modified EUI-64 format.**

- This statement is deleted:

  **The use of the universal/local bit in the Modified EUI-64 format identifier is to allow development of future technology that can take advantage of interface identifiers with universal scope.**

# Questions? Discussion?

- Does 6man want to adopt this draft?