A Method for Deriving Stable IPv6 Interface Identifiers from Amateur Radio Callsigns

Abstract

This document defines a method for generating stable IPv6 Interface
Identifiers for amateur packet radio nodes. This method is meant to
be an alternative to hardware address based Interface Identifier
generation such that the benefits of stable addressing may be
achieved even on nodes that have unstable, changing, or experimental
networking hardware. Instead of a physically-derived address, this
method utilizes a unique hash of the node's government-assigned
callsign as the basis for its Interface Identifier.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document specifies the steps an amateur packet radio node takes
in order to generate a stable and unique IPv6 Interface Identifier
(IID) [RFC2460]. The resulting Interface Identifier SHALL be used in
conjunction with processes such as (but not limited to) Stateless
Address Autoconfiguration (SLAAC) [RFC4862], DHCPv6 [RFC3315], or
manual configuration to configure IPv6 connectivity on the node.

Callsign-derived Interface Identifier generation requires minimal
manual configuration, and when paired with SLAAC may allow a mobile
amateur packet radio node to automatically connect to, and
communicate with any compliant amateur radio network provided that
the node has been configured with a callsign, and is communicating
on the correct radio frequency.

2. Terminology

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].

3. Foreword on Node Identification

Amateur packet radio nodes generally identify themselves with a
string of ASCII characters comprised of:

  1. The station's government-assigned callsign

2. A dash (-)

   3. A number ranging from 0 to 15, inclusive. This will be referred
      to for the remainder of this document as the node's "ID".
      Stations that do not use a node ID generally will use a "0" in
      this place.

   For example, a node operated under the callsign "VA3ZZA" with the
   node ID of "5" would identify itself on-air as "VA3ZZA-5".

4. The Algorithm

   To determine a 64 bit long [RFC4291] Interface Identifier for an
   amateur packet radio node in conformance with this specification,
   the following steps MUST be be taken:

   1. Compute the SHA-256 hash of the node's UPPERCASE callsign.

   2. Use the first 60 bits of the hash as the first 60 bits of the
      Interface Identifier.

   3. Use the final 4 bits of the address' Interface Identifier to
      store the nodes's ID number.

   Addresses generated using this method will look like the following:

```
AAAA:AAAA:AAAA:AAAA:BBBB:BBBB:BBBB:BBBC
|                   | |                 |+- C: Node ID
|                   | +----------------+-- B: Callsign Hash
+-----------------+-------------------- A: Network Prefix
```

4.1. Using the Interface Identifier

   It is recommended to use the generated Interface Identifier with
   Stateless Address Autoconfiguration to automatically determine the
   node's IPv6 prefix and establish routes to other hosts in a radio
   network.

   While SLAAC is the recommended method of configuration, it is not
   required. Amateur packet radio networks may also use alternate
   address configuration mechanisms such as DHCPv6 or manual
   configuration as the participants see fit.

4.2. Resolving SLAAC Duplicate Address Detection Conflicts

   As a side effect of basing the Interface Identifier on an existing
   globally unique identifier, DAD [RFC4862] conflicts should be non-
   existent.

If a conflict is encountered, the node operator SHOULD contact the operator of the offending station, and/or the appropriate regulatory authority about possible unauthorized use of a callsign.

4.3. Benefits of this method

This method of Interface Identifier generation has the following benefits:

*Callsigns are uniquely assigned to stations by existing governing bodies. Using them as the basis of address creation will ensure a unique base hash for each station.

*Hashing callsigns instead of trying to plainly hex encode them allows support for excessively long callsigns.

*Encoding the station ID in the final nibble of the address allows for up to 16 nodes under the same callsign to be assigned addresses within the same /124. This allows address-based access control logic to operate on a whole callsign (first 60 bits of the interface ID) at once, an ability not possible if the ID was also hashed.

4.4. Drawbacks of this method

While it is possible for one node to correlate another's IP address to its station callsign via a lookup table, ideally the raw callsign could be encoded directly into the IPv6 address. Doing so would both allow for a node to easily determine the callsign of a sending station without additional metadata embedded in the received packet, and allow the source address on outgoing packets to be used to satisfy legal station identification requirements.

Unfortunately, this is not feasible due to many governments assigning temporary "special event callsigns" to stations. These special callsigns often do not follow the general length restrictions on permanent callsigns, raising the possibility that a station will be assigned a callsign longer than is possible to encode directly in an IPv6 address, thus being un-addressable.

5. Privacy Considerations

The International Telecommunication Union requires all stations operating in the amateur service to self-identify when transmitting. Various countries also impose further requirements such as the interval and method by which stations must identify themselves.

The legal requirement to identify all transmissions nullifies any privacy benefits gained from other privacy-aware addressing methods.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

This document should not affect the security of the Internet.

8. References

8.1. Normative References

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing
           Architecture", RFC 4291, DOI 10.17487/RFC4291, February
           2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless
           Address Autoconfiguration", RFC 4862, DOI 10.17487/
           RFC4862, September 2007, <https://www.rfc-editor.org/
           info/rfc4862>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6
           (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460,
           December 1998, <https://www.rfc-editor.org/info/rfc2460>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins,
           C., and M. Carney, "Dynamic Host Configuration Protocol
           for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July
           2003, <https://www.rfc-editor.org/info/rfc3315>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/
           RFC2119, March 1997, <https://www.rfc-editor.org/info/
           rfc2119>.

Author's Address

Evan Pratten

Email: evan@ewpratten.com
URI: https://ewpratten.com