                 **Network Time Protocol I-Do Extension Field**
                         **draft-stenn-ntp-i-do-00**

Abstract

   The first implementation of NTPv4 was released in 2003.  NTPv4 is
   defined by RFC 5905 [RFC5905].  It contains a public-key security
   protocol, autokey, which is defined by RFC 5906 [RFC5906].  Until
   very recently, autokey has been the only defined "user" of NTP packet
   Extension Fields.  New proposals for extension fields are being
   written and there is currently no convenient way to learn if a remote
   instance of NTP supports any extension fields or not.  This proposal
   contains a method to tell a remote instance of NTP what we support,
   and ask what they support.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF).  Note that other groups may also distribute
   working documents as Internet-Drafts.  The list of current Internet-
   Drafts is at http://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time.  It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on September 15, 2016.

Table of Contents

## 1.  Introduction

   The first implementation of NTPv4 was released in 2003.  NTPv4 is
   defined by RFC 5905 [RFC5905].  It contains a public-key security
   protocol, autokey, which is defined by RFC 5906 [RFC5906].  Until
   very recently, autokey has been the only defined "user" of NTP packet
   Extension Fields.  New proposals for extension fields are being
   written and there is currently no convenient way to learn if a remote
   instance of NTP supports any extension fields or not.  This proposal
   contains a method to tell a remote instance of NTP what we support,
   and ask what they support.

## 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 RFC 2119 [RFC2119].

## 2.  The I-Do Extension Field

   If an incoming packet contains an unrecognized extension field, one
   of two things will happen.  Either that extension field will be
   ignored, or the entire packet will be dropped.  If an extension field
   is present there ordinarily SHOULD be a MAC following the extension
   field.  Some extension fields are unable to be "signed" by a MAC,
   regardless of whether or not that MAC is a traditional MAC or an
   extension field MAC.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-------------------------------+
|           Field Type          |          Field Length         |
+-------------------------------+-------------------------------+
|            I-Do 1             |             ...               |
+-------------------------------+-------------------------------+
|            I-Do N             |           Padding             |
+-------------------------------+-------------------------------+
```

                NTP Extension Field: REFID Suggestion

Field Type: TBD (Recommendation for IANA: 0x0007 (I-Do, MAC
required), 0x2007 (I-Do, MAC OPTIONAL), 0x4007 (I-Do Response, MAC
required), 0x6007 I-Do Response, MAC OPTIONAL))

Field Length: as needed

Payload: An enumeration of the suppported base Field Types, followed
by any padding, 0x0000, needed to fill the payload to the desired
32-bit boundary.

Example: A system that wants to advertise support for Autokey and
I-Do, sending to a system that wants to advertise support for I-Do,
NTS, and MAC-As-Extension-Field

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-------------------------------+
|    Field Type (0x2007)        |     Field Length (0x0008)     |
+-------------------------------+-------------------------------+
|             0x0007            |             0x0002            |
+-------------------------------+-------------------------------+
```

                   NTP Extension Field: I-Do

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---------------+---------------+-------------------------------+
|    Field Type (0x6007)        |     Field Length (0x000a)     |
+-------------------------------+-------------------------------+
|             0x0003            |             0x0004            |
+-------------------------------+-------------------------------+
|             0x0007            |             0x0000            |
+-------------------------------+-------------------------------+
```

                NTP Extension Field: I-Do Response

The sender of any I-Do extension field MUST send an extension field
with a Field Type of 0x0007 (I-Do, MAC required) or 0x2007 (I-Do, MAC
OPTIONAL) and SHOULD include a paylod with any 0x0000 padding values
after enumerating the supported base Extension Field Types.  The
responding system MUST reply with an extension field with a Field
Type of 0x4007 (I-Do Response, MAC required) or 0x6007 (I-Do
Response, MAC OPTIONAL), and SHOULD include a paylod with any 0x0000
padding values after enumerating the supported base Extension Field
Types.

The following information is included here until it is specified in a
better location.  If the Field Type does not have bit 0x2000 set,
there MUST be a MAC included later in the packet for this field to be
accepted.  If the Field Type has bit 0x2000 set, the presence of a
MAC later in the packet is OPTIONAL.

Any system that receives an I-Do extension field as either an "offer"
or a "response" SHOULD scan the entire payload looking for nonzero
values that specify the capabilities of the remote association.

Any system that receives an I-Do "offer", 0x0007 or 0x2007, SHOULD
reply with an I-Do "response", 0x4007 or 0x6007.

Any system that sends an I-Do "offer" or "response" may send as few
or as many of its supported Field Types as it chooses.  At any
subsequent time, either side may re-negotiate the list of supported
field types it is prepared to accept from the other system by sending
a new I-Do extension field.

The most-recently received I-Do list replaces any previous I-Do list.

## 3.  Acknowledgements

The author wishes to acknowledge the contributions of Joey
Saccadonuts.

## 4.  IANA Considerations

This memo requests IANA to allocate NTP Extension Field Types 0x0007
(I-Do), 0x2007 (I-Do, MAC OPTIONAL), 0x4007 (I-Do Response), and
0x6007 (I-Do Response, MAC OPTIONAL)for this proposal.

## 5.  Security Considerations

Additional information TBD

6.  Normative References

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

   [RFC5905]  Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch,
              "Network Time Protocol Version 4: Protocol and Algorithms
              Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010,
              <http://www.rfc-editor.org/info/rfc5905>.

   [RFC5906]  Haberman, B., Ed. and D. Mills, "Network Time Protocol
              Version 4: Autokey Specification", RFC 5906,
              DOI 10.17487/RFC5906, June 2010,
              <http://www.rfc-editor.org/info/rfc5906>.

Author's Address

   Harlan Stenn
   Network Time Foundation
   P.O. Box 918
   Talent, OR  97540
   US

   Email: stenn@nwtime.org