         The Application Exchange (APEX) Option Party Pack, Part Deux!

Status of this Memo

Copyright Notice

Abstract

   Application Exchange (APEX), at its core, provides a best-effort
   application-layer datagram service. Options are used to alter the
   semantics of the core service. This memo defines various options to
   change the default behavior of APEX's "relaying mesh".

Table of Contents

1. The attachOverride Option

   Section 5.1 contains the APEX option registration for the
   "attachOverride" option.

   The default behavior of the APEX relaying mesh, in the absence of
   processing options, is to allow at most one application to attach as
   a particular endpoint, on a "first come, first served" basis. The
   "attachOverride" option provides gives preference to the current
   application trying to attach.

   If this option is present in the "attach" operation (c.f., Section
   4.4.1 of [1]) and if any application is already attached as the
   specified endpoint, that endpoint has its attachment terminated
   (c.f., Section 4.4.3 of [1]) concurrently with processing of that
   "attach" operation. The "code" attribute of the resulting
   "terminate" operation is set to 556.

   Note that any data being expected by the previously-attached
   application may instead be delivered to the last application to
   successfully attach. Accordingly, applications should take care to
   properly deal with incoming data having unrecognized transaction-
   identifiers (c.f., Section 6.1.1 of [1]).

This option provides for a new attachment to automatically terminate
any existing attachment for the same endpoint. For example, this
might be helpful when a new attachment is required from a different
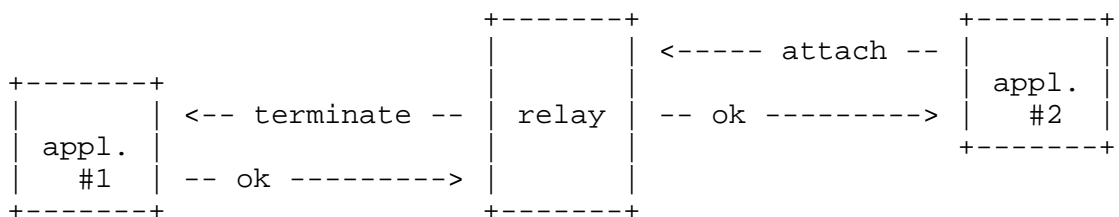device while a previously-used device is still attached e.g.,

```
   +-------+                 +-------+
   |       | -- attach ----->|       |
   | appl. |                 | relay |
   |  #1   | <--------- ok --|       |
   +-------+                 +-------+
```
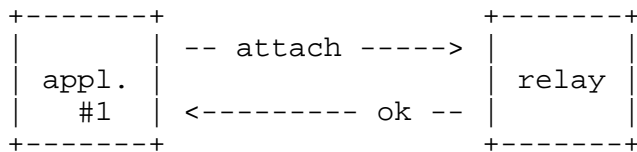
```
   C: <attach endpoint='fred@example.com' transID='1' />
   S: <ok />
```

  ... some time later appl #2 starts on a different computer ...

```
                           +-------+                 +-------+
                           |       | <----- attach --|       |
   +-------+               |       |                 | appl. |
   |       | <-- terminate |-- relay | -- ok --------->|  #2   |
   | appl. |               |       |                 +-------+
   |  #1   | -- ok ------->|       |
   +-------+               +-------+
```

```
           C: <attach endpoint='fred@example.com' transID='2'>
                <option internal='attachOverride' transID='3' />
              </attach>
           S: <ok />
```

```
   C: <terminate transID='1' code='556'>overriden</terminate>
   S: <ok />
```

2. The dataTiming Option

   Section 5.2 contains the APEX option registration for the
   "dataTiming" option. This option contains a "dataTiming" element
   (c.f., Section 6).

   The default behavior of the APEX relaying mesh is "immediate, best
   effort", and expects that all relays and endpoints are able to
   process and transfer data without delay -- in the absence of
   processing options, if a relay is unavailable, then data is silently
   dropped. The "dataTiming" option provides for controlled queuing
   delays in processing, whilst providing reasonable deterministic
   behavior for the originator.

There are two types of delays addressed by the "dataTiming" option:

o delays in transit through the relaying mesh, possibly due to
   intermittent or slow connections, or congested relays; and,

o delays because the intended endpoint is not available to receive
   the data, when used in conjunction with the hold4Endpoint option
   (Section 3).

Accordingly, the "dataTiming" option allows for:

o data to be discarded if not delivered within a finite amount of
   time as specified using the "noLaterThan" attribute (Section 2.1);

o a "statusResponse" message (c.f., Section 5.1 of [1]) to be
   generated if data is not delivered within a known amount of time
   as specified using the "reportAfter" attribute (Section 2.2); and,

o an upper limit on the amount of time for the "statusResponse"
   message to be delivered using the "returnTrip" attribute (Section
   2.1.1), after which the sender may presume the message to be lost.

This option does not provide any functionality with respect to the
priority of the data. Nor does this option have any effect on other
parts of the relaying process.

Further, note that because this option is processed on a per-hop
basis, the originator must set the "targetHop" attribute to the value
"all" and the "mustUnderstand" attribute to the value "true".

2.1 Upper-Bounds on Delivery

The "noLaterThan" attribute of the "dataTiming" option provides for
control over delays that may occur in transit through the relaying
mesh or to the recipient endpoint.

If this option is present in the "data" operation (c.f., Section
4.4.4 of [1]) and the value of the "noLaterThan" attribute is non-
zero, then:

o For Step 5.2 of Section 4.4.4.1 of [1]:

   Immediately prior to sending the data to the next relay, the value
   of the "noLaterThan" attribute is adjusted to reflect the
   processing time of the data at the local relay (e.g., the time
   required to determine the next relay, to successfully issue a
   "bind" operation, and then be ready to immediately issue a "data"
   operation).

If the value of the "noLaterThan" attribute becomes less than or
equal to zero, an error in processing has occurred, the data
element is not sent to the next relay, and if the "reportErrors"
attribute is true, the APEX report service is invoked to send a
timing error report.

o For Step 5.3 of Section 4.4.4.1 of [1]:

If the relay does not receive an "ok" element from the recipient
endpoint within the number of milli-seconds indicated by the value
of the "noLaterThan" attribute, an error in processing has
occurred, and if the "reportErrors" attribute is true, the APEX
report service is invoked to send a timing error report.

Otherwise, if the data is successfully transmitted to the
recipient, and the "returnTrip" attribute is non-zero, the APEX
report service is invoked to send a final hop report.

Note that in some cases, a relay may be able to predict this outcome
without actually connecting to the next relay; if so, a timing error
report may be sent without connecting to the next relay.

2.1.1 Final Hop Report

If the APEX report service (c.f., Section 6.2 of [1]) is invoked to
send a final hop report, it issues a data operation with:

o its originator identifying the report service associated with the
   issuing relay

o its recipient identifying the endpoint address of the originator
   associated with the "dataTiming" option

o a new "dataTiming" option having:

   * its "noLaterThan" attribute equal to the "returnTrip" attribute
      of the original "dataTiming" option

   * and no other attributes present

o its content consisting of a "statusResponse" element having:

   * its "transID" attribute equal to the "transID" attribute of the
      "dataTiming" option

   * and identifying the original recipient with a permanent success
      indicator

For example:

```
                              +-------+              +-------+
                              |       | -- data ------> |       |
                              | relay |              | appl. |
                              |       | <--------- ok -- |  #2   |
                              +-------+              +-------+
```

```
     C: <data content='cid:1@example.com'>
            <originator identity='fred@example.com' />
            <recipient identity='barney@example.com' />
            <option internal='dataTiming' targetHop='all'
                    mustUnderstand='true' transID='86'>
               <dataTiming noLaterThan='10000' returnTrip='20000' />
            </option>
        </data>
     S: <ok />
```

```
       +-------+              +-------+
       |       | <------- data -- |       |
       | appl. |              | relay |
       |  #1   | -- ok --------->  |       |
       +-------+              +-------+
```

```
     C: <data content='#Content'>
            <originator identity='apex=report@example.com' />
            <recipient identity='fred@example.com' />
            <option internal='dataTiming' targetHop='all'
                    mustUnderstand='true' transID='99'>
               <dataTiming noLaterThan='20000' />
            </option>
            <data-content Name='Content'>
               <statusResponse transID='86'>
                  <destination identity='barney@example.com'>
                     <reply code='250' />
                  </destination>
               </statusResponse>
            </data-content>
        </data>
     S: <ok />
```
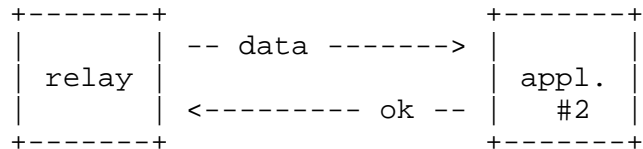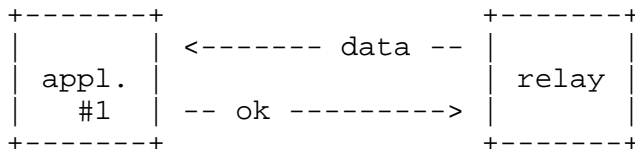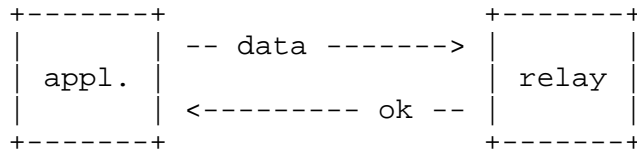
2.1.2 Timing Error Report

   If the APEX report service (c.f., Section 6.2 of [1]) is invoked to
   send a timing error report, it issues a data operation with:

   o its originator identifying the report service associated with the
      issuing relay

   o its recipient identifying the endpoint address of the originator
      associated with the "dataTiming" option

   o its content consisting of a "statusResponse" element having:

      * its "transID" attribute equal to the "transID" attribute of the
         "dataTiming" option

      * and identifying the original recipient with a permanent failure
         indicator

   For example:

```
        +-------+                  +-------+
        |       | -- data ------->  |       |
        | appl. |                   | relay |
        |       | <--------- ok --  |       |
        +-------+                  +-------+
```

```
   C: <data content='cid:1@example.com'>
          <originator identity='fred@example.com' />
          <recipient identity='barney@example.com' />
          <option internal='dataTiming' targetHop='all'
                  mustUnderstand='true' transID='86'>
             <dataTiming noLaterThan='6000' reportErrors='true' />
          </option>
      </data>
   S: <ok />
```

    ... some time later ...

```
         +-------+                  +-------+
         |       | <------- data --  |       |
         | appl. |                   | relay |
         |       | -- ok --------->  |       |
         +-------+                  +-------+
```

```
     C: <data content='#Content'>
            <originator identity='apex=report@example.com' />
            <recipient identity='fred@example.com' />
            <data-content Name='Content'>
               <statusResponse transID='86'>
                  <destination identity='barney@example.com'>
                     <reply code='550' />
                  </destination>
               </statusResponse>
            </data-content>
        </data>
     S: <ok />
```
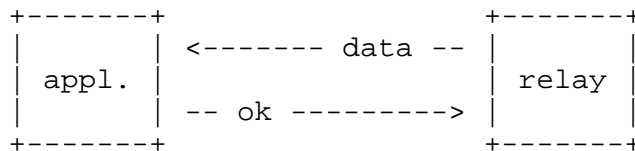
2.2 Reporting on Delayed Delivery

   The "reportAfter" attribute of the "dataTiming" option provides for
   the originator to be notified if delivery is delayed beyond a
   specified time. Delivery of the data is not affected. Note that if
   the value of the "noLaterThan" attribute is non-zero, then it
   provides the operational upper-bounds for the "reportAfter"
   attribute.

   If this option is present in the "data" operation (c.f., Section
   4.4.4 of [1]) and the value of the "reportAfter" attribute is non-
   zero, then:

   o For Step 5.2 of Section 4.4.4.1 of [1]:

      Immediately prior to sending the data to the next relay, the value
      of the "reportAfter" attribute is adjusted to reflect the
      processing time of the data at the local relay (e.g., the time
      required to determine the next relay, to successfully issue a
      "bind" operation, and then be ready to immediately issue a "data"
      operation).

      If the value of the "reportAfter" attribute becomes less than or
      equal to zero, then its value is set to zero and the APEX report
      service is invoked to send a transient timing report; regardless,
      the data element is sent to the next relay.

   o For Step 5.3 of Section 4.4.4.1 of [1]:
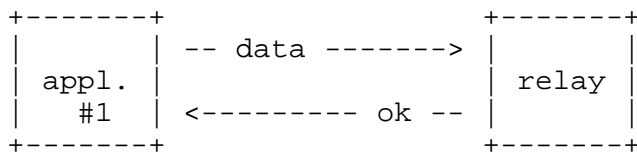
      If the relay does not receive an "ok" element from the recipient
      endpoint within the number of milli-seconds indicated by the value
      of the "reportAfter" attribute, then its value is set to zero and
      the APEX report service is invoked to send a transient timing
      report.

2.2.1 Transient Timing Report

   If the APEX report service (c.f., Section 6.2 of [1]) is invoked to
   send a transient timing report, it issues a data operation with:

   o its originator identifying the report service associated with the
      issuing relay

   o its recipient identifying the endpoint address of the originator
      associated with the "dataTiming" option

   o its content consisting of a "statusResponse" element having:

      * its "transID" attribute equal to the "transID" attribute of the
         "dataTiming" option

      * and identifying the original recipient with a transient success
         indicator

For example:

```
     +-------+                   +-------+
     |       | -- data ------->  |       |
     | appl. |                   | relay |
     |   #1  | <--------- ok --  |       |
     +-------+                   +-------+
```

```
   C: <data content='cid:1@example.com'>
          <originator identity='fred@example.com' />
          <recipient identity='barney@example.com' />
          <option internal='dataTiming' targetHop='all'
                 mustUnderstand='true' transID='86'>
             <dataTiming reportAfter='60000' />
          </option>
       </data>
   S: <ok />
```

... some time later ...

```
                         +-------+                  +-------+
                         |       | <------- data -- |       |
                         | relay |                  | relay |
                         | #n-1  | -- ok --------> |   #n  |
                         +-------+                  +-------+
```

```
   C: <data content='#Content'>
          <originator identity='apex=report@example.com' />
          <recipient identity='fred@example.com' />
          <data-content Name='Content'>
              <statusResponse transID='86'>
                  <destination identity='barney@example.com'>
                       <reply code='350' />
                  </destination>
              </statusResponse>
          </data-content>
       </data>
   S: <ok />
```
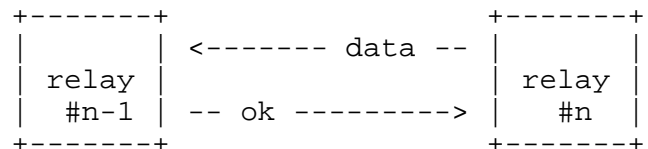
3. The hold4Endpoint Option

   Section 5.3 contains the APEX option registration for the
   "hold4Endpoint" option.

   The default behavior of the APEX relaying mesh, in the absence of
   processing options, is to silently drop data for a recipient if its
   endpoint isn't attached. The "hold4Endpoint" option provides for
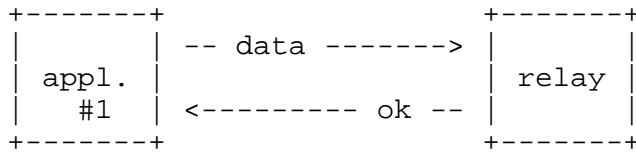   data to be queued if the recipient endpoint is not attached.

If this option is present in the "data" operation (c.f., Section
4.4.4 of [1]), and the value of the "hold4Endpoint" attribute is true
then:

o For Step 5.3 of Section 4.4.4.1 of [1]:

   If the recipient's endpoint is not currently attached, the relay
   will queue the data waiting for an application to attach as that
   endpoint.

Note that in the absence of an upper-bounds on delivery, such as
limits provided by the dataTiming option (Section 2), the data will
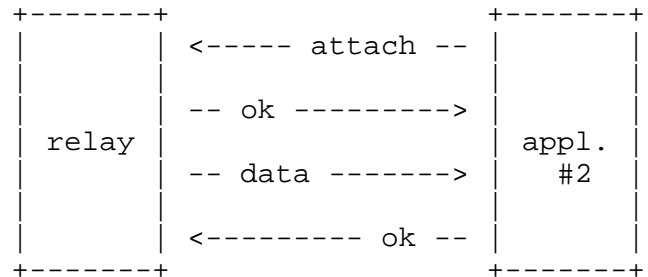be queued indefinitely for the endpoint.

For example:

```
        +-------+                 +-------+
        |       | -- data ------> |       |
        | appl. |                 | relay |
        |  #1   | <--------- ok -- |       |
        +-------+                 +-------+
```

```
    C: <data content='cid:1@example.com'>
           <originator identity='fred@example.com' />
           <recipient identity='barney@example.com' />
           <option internal='hold4Endpoint' />
           <option internal='dataTiming' targetHop='all'
                  mustUnderstand='true' transID='86'>
              <dataTiming noLaterThan='60000' />
           </option>
       </data>
    S: <ok />
```

   ... some time later the recipient's endpoint attaches ...

```
                            +-------+           +-------+
                            |       | <----- attach -- |       |
                            |       |           |       |
                            |       | -- ok ---------> |       |
                            | relay |           | appl. |
                            |       | -- data ------> |  #2   |
                            |       |           |       |
                            |       | <--------- ok -- |       |
                            +-------+           +-------+
```

```
    C: <attach endpoint='barney@example.com' transID='2'>
           <option internal='attachOverride' transID='3' />
       </attach>
    S: <ok />
```

```
    C: <data content='cid:1@example.com'>
           <originator identity='fred@example.com' />
           <recipient identity='barney@example.com' />
           <option internal='hold4Endpoint' />
           <option internal='dataTiming' targetHop='all'
                  mustUnderstand='true' transID='86'>
              <dataTiming noLaterThan='18000' />
           </option>
       </data>
    S: <ok />
```

4. The dataHopping Option

   To detect misconfigurations that cause forwarding loops in the APEX
   relaying mesh, the APEX pubsub service re-introduces a mechanism
   similar to the IP TTL [2] mechanism, in the form of an APEX option.
   Section 5.4 contains the APEX option registration for the
   "dataHopping" option.

   If this option is present in the "data" operation (c.f., Section
   4.4.4 of [1]) and the value of the "noMoreThan" attribute is non-
   zero, then:

   o For Step 5.2 of Section 4.4.4.1 of [1]:

      Immediately prior to sending the data to the next relay, the value
      of the "noMoreThan" attribute is reduced by 1.

      If the value of the "noMoreThan" attribute becomes less than or
      equal to zero, an error in processing has occurred, the data
      element is not sent to the next relay, and if the "reportErrors"
      attribute is true, the APEX report service is invoked to send an
      error report.

   Further, note that because this option is processed on a per-hop
   basis, the originator must set the "targetHop" attribute to the value
   "all" and the "mustUnderstand" attribute to the value "true".

   If the APEX report service (c.f., Section 6.2 of [1]) is invoked to
   send an error report, it issues a data operation with:

   o its originator identifying the report service associated with the
      issuing relay

   o its recipient identifying the endpoint address of the originator
      associated with the "dataHopping" option

   o its content consisting of a "statusResponse" element having:

      * its "transID" attribute equal to the "transID" attribute of the
         "dataHopping" option

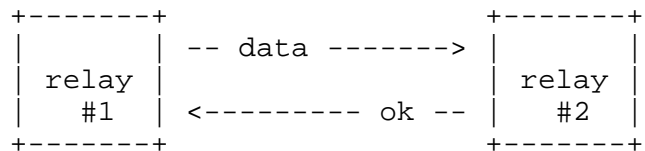   * and identifying the original recipient with a permanent failure
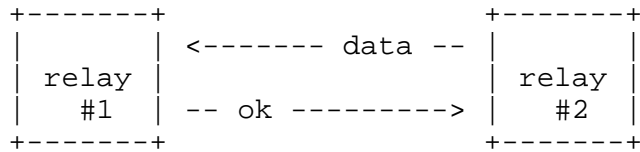   indicator

For example:

```
    +-------+                 +-------+
    |       | -- data ------->|       |
    | appl. |                 | relay |
    |       | <--------- ok --|   #1  |
    +-------+                 +-------+
```

```
C: <data content='cid:1@example.com'>
      <originator identity='appl=pubsub/topic=fred@example.com' />
      <recipient identity='barney@example.com' />
      <option internal='dataHopping' targetHop='all'
              mustUnderstand='true' transID='86'>
         <dataHopping noMoreThan='2' reportErrors='true' />
      </option>
   </data>
S: <ok />
```

```
                          +-------+                 +-------+
                          |       | -- data ------->|       |
                          | relay |                 | relay |
                          |   #1  | <--------- ok --|   #2  |
                          +-------+                 +-------+
```

```
C: <data content='cid:1@example.com'>
      <originator identity='appl=pubsub/topic=fred@example.com' />
      <recipient identity='barney@example.com' />
      <option internal='dataHopping' targetHop='all'
              mustUnderstand='true' transID='86'>
         <dataHopping noMoreThan='1' reportErrors='true' />
      </option>
   </data>
S: <ok />
```

relay #2 determines that further relaying is necessary:

```
   +-------+                +-------+
   |       | <------- data -- |       |
   | relay |                | relay |
   |  #1   | -- ok --------> |  #2   |
   +-------+                +-------+
```

```
C: <data content='#Content'>
      <originator identity='apex=report@example.com' />
      <recipient identity='appl=pubsub/topic=fred@example.com' />
      <data-content Name='Content'>
         <statusResponse transID='86'>
            <destination identity='barney@example.com'>
               <reply code='550' />
            </destination>
         </statusResponse>
      </data-content>
   </data>
S: <ok />
```

5. Initial Registrations

   The APEX option registration template is defined in Section 7.1 of
   [1].

5.1 Registration: The attachOverride Option

   Option Identification: attachOverride

   Present in: APEX's "attach" element

   Contains: nothing

   Processing Rules: c.f., Section 1

   Contact Information: c.f., the "Authors' Addresses" section of this
      memo

5.2 Registration: The dataTiming Option

   Option Identification: dataTiming

   Present in: APEX's "data" element

   Contains: dataTiming (c.f., Section 6)

   Processing Rules: c.f., Section 2

   Contact Information: c.f., the "Authors' Addresses" section of this
      memo

5.3 Registration: The hold4Endpoint Option

   Option Identification: hold4Endpoint

   Present in: APEX's "data" element

   Contains: nothing

   Processing Rules: c.f., Section 3

   Contact Information: c.f., the "Authors' Addresses" section of this
      memo

5.4 Registration: The dataHopping Option

   Option Identification: dataHopping

   Present in: APEX's "data" element

   Contains: dataHopping (c.f., Section 6)

   Processing Rules: c.f., Section 4

   Contact Information: c.f., the "Authors' Addresses" section of this
      memo

6. The APEX Party Pack DTD

```
   <!--
     DTD for the APEX option party pack, as of 2001-05-14


     Refer to this DTD as:

       <!ENTITY % APEXPARTY PUBLIC "-//IETF//DTD APEX PARTY//EN" "">
       %APEXPARTY;
     -->

   <!ENTITY % APEXCORE PUBLIC "-//IETF//DTD APEX CORE//EN"
   %APEXCORE;

   <!--
     DTD data types:

          entity          syntax/reference       example
          ======          ================       =======
       hopcount
          HOPS            0..255                 17

       milli-seconds
          MILLISECS       0..2147483647          60000
     -->

   <!ENTITY  % HOPS      "CDATA">
   <!ENTITY  % MILLISECS
                          "CDATA">


   <!ELEMENT dataHopping EMPTY>
   <!ATTLIST dataHopping
           noMoreThan  %HOPS;           "0"
           reportErrors
                      (true|false)      "false">

   <!ELEMENT dataTiming  EMPTY>
   <!ATTLIST dataTiming
           noLaterThan %MILLISECS;      "0"
           returnTrip  %MILLISECS;      "0"
           reportAfter %MILLISECS;      "0"
           reportErrors
                      (true|false)      "false">
```

7. Security Considerations

   Consult [1]'s Section 11 for a discussion of security issues.

   In addition:

   o The dataTiming option (Section 2) may be used to expose private
      network topology. Accordingly, an administrator may wish to
      choose to disable this option except at the ingress/egress points
      for its administrative domain.

   o The hold4Endpoint option (Section 3) may be used to facilitate
      denial-of-service attacks. Accordingly, an administrator may wish
      to impose administrative limits on this attribute (e.g., always
      require that the "dataTiming" option also be present with a
      short-lived "noLaterThan" attribute).

References

   [1]   Rose, M., Klyne, G. and D. Crocker, "The Application Exchange
          Core", RFC 3340, July 2002.

   [2]   Postel, J., "Internet Protocol", STD 5, RFC 791, September
          1981.

   [3]   Newman, D., "Deliver By SMTP Service Extension", RFC 2852, June
          2000.

Appendix A. Acknowledgements

   The authors gratefully acknowledge the contributions of Chris Newman
   and Bob Wyman. Further, the dataTiming option is similar in function
   to "Deliver By" SMTP service extension defined by Dan Newman in [3].

Appendix B. IANA Considerations

   The IANA completed the registrations specified in Section 5.

Authors' Addresses

   Graham Klyne
   Clearswift Corporation
   1310 Waterside
   Arlington Business Park
   Theale, Reading RG7 4SA
   UK

   Phone: +44 11 8903 8903
   EMail: Graham.Klyne@MIMEsweeper.com

   Marshall T. Rose
   Dover Beach Consulting, Inc.
   POB 255268
   Sacramento, CA 95865-5268
   US

   Phone: +1 916 483 8878
   EMail: mrose@dbc.mtview.ca.us

   Michael F. Schwartz
   Code On The Road, LLC

   EMail: schwartz@CodeOnTheRoad.com
   URI: http://www.CodeOnTheRoad.com

   Eric Dixon

   EMail: edixon@myrealbox.com

   Huston Franklin

   EMail: huston@franklin.ro

   Jay Kint

   EMail: d20@icosahedron.org

      Darren New
      5390 Caminito Exquisito
      San Diego, CA 92130
      US

      Phone: +1 858 350 9733
      EMail: dnew@san.rr.com

      Scott Pead

      EMail: spead@fiber.net