Lemonade
Internet-Draft
Updates: 2554, 3207
(if approved)
Intended status: Standards Track
Expires: October 19, 2007

T. Finch
University of Cambridge

April 17, 2007

The QUICKSTART SMTP service extension (full profile) draft-fanf-smtp-quickstart-b

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on October 19, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This memo specifies modifications to SMTP's start-up sequence which reduce the number of round trips between the client and the server before message transmission starts. This can significantly reduce the delay for interactive message submission over slow links. (This is the more efficient version of the proposal.)

Document revision

$Cambridge: hermes/doc/qsmtp/draft-fanf-smtp-quickstart-b.xml,v 1.44 2007/04/17 06:46:21 fanf2 Exp $

Table of Contents

1. Introduction 3
2. Terminology 4
3. QUICKSTART SMTP service extension definition 4
4. Extended server greeting 4
5. The QHLO command 5
6. After establishing a connection 7
7. After establishing a security layer 7
8. Backwards compatibility 8
9. Changes to the STARTTLS command 9
10. Changes to the AUTH command 9
11. IANA Considerations 9
12. Security considerations 10
13. References 11
13.1. Normative references 11
13.2. Informative references 11
Appendix A. Example round trip summaries 12
Appendix B. Changes from previous versions 16
Author's Address 17
Intellectual Property and Copyright Statements 18

1. Introduction

SMTP can require many round trips between the client and server before message transmission starts. This is a particular problem for message submission over slow links, where the resulting delay can easily be two seconds or more. The QUICKSTART SMTP service extension introduces some changes which eliminate round trips and therefore reduce the delay before messages can be transmitted. It is based on the PIPELINING service extension [RFC2920]. The key features of the extension are:

- Earlier announcement of the list of SMTP service extensions supported by the server.
- The QHLO command, which is a quicker replacement for the EHLO command.
- The ability to pipeline the STARTTLS and AUTH commands.
- A mechanism to allow the client to issue commands without always having to wait for the server's list of supported extensions.

In a normal message sumbission connection, the client sends the first MAIL command in its 9th packet; with QUICKSTART the first MAIL command can be as early as the client's 3rd packet, which is as early as possible with TCP and TLS. Appendix A illustrates these counts in more detail.

Although SMTP QUICKSTART is probably most useful in conjunction with message submission [RFC4409], it MAY also be used with MTA-to-MTA SMTP - though attention should be paid to the security considerations in Section 12.

1.1. Procedural Rubric

Comments and discussion about this draft should be directed to the <lemonade@ietf.org> mailing list - the working group dealing with enhancements to Internet email to support diverse service environments.

This draft should be read in conjunction with [quickstart-a] which describes an alternative profile of these ideas. Draft -A is simpler to implement, whereas draft -B uses extra state to save even more round trips. Draft -A introduces a QTLS command whereas draft -B is generalised to work with security layers set up using the existing

STARTTLS and AUTH commands.

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].

The metalinguistic notation used in this memo corresponds to the "Augmented Backus-Naur Form" used in other Internet mail system memos. The reader who is not familiar with that syntax should consult the ABNF specification [RFC4234]. Rules not defined in this specification are either defined in the ABNF core rules or in [RFC2821]. Metalanguage terms used in running text are surrounded by pointed brackets (e.g., <qhlo-id>) for clarity.

3. QUICKSTART SMTP service extension definition

The QUICKSTART SMTP service extension is defined as follows:

- The name of the service extension is "QUICKSTART".
- The EHLO keyword value associated with the extension is "QUICKSTART". It has one parameter <qhlo-id> described in Section 4.
- The extension defines the "QHLO" command in Section 5 et seq.
- Changes to the STARTTLS command (updating [RFC3207]) are described in Section 9. Changes to the AUTH command (updating [RFC2554]) are described in Section 10.
- There are no additional parameters to the MAIL or RCPT commands defined by this extension and their maximum lengths are not increased.

4. Extended server greeting

An SMTP server that supports QUICKSTART MUST respond to the client's connection with an extended greeting according to the syntax below. This greeting has a similar format to the SMTP EHLO reply, that is, the first line is the same as an un-extended greeting, and subsequent lines list the SMTP service extensions supported by the server. The list MUST include the QUICKSTART keyword and the PIPELINING keyword [RFC2920]. The server SHALL give exactly the same list in its greeting as it does in reply to the EHLO command.

Syntax:

```
qsmtp-greet  =      "220-"   Domain [ SP ehlo-greet ] CRLF
                *(  "220-"   ehlo-line                CRLF  )
                    "220" SP ehlo-line                CRLF

ehlo-line    =/ "QUICKSTART" SP qhlo-id

qhlo-id      =  esmtp-value
```

The QUICKSTART keyword has a <qhlo-id> parameter which is a case-sensitive token that identifies the list of service extensions and associated parameters in which it was included. The server MUST state a different <qhlo-id> for different lists of extensions or different extension parameters. For example, if the server only supports AUTH PLAIN after TLS negotiation, it will state a different <qhlo-id> in its service extension lists before and after STARTTLS. Section 12 includes some security considerations for the <qhlo-id>.

A client that supports QUICKSTART SHOULD keep a cache of supported service extensions and <qhlo-id>s. This cache SHOULD be keyed by a combination of the server's IP address and the security context. For example, if the client connects to a server then issues EHLO, STARTTLS, and EHLO commands, it will cache two separate service extension lists corresponding to two security contexts: after the connection is established in cleartext, and after TLS is negotiated. More details of the cache are described below.

The extended greeting allows a client to find out the server's supported extensions one round trip earlier than in normal SMTP.

5. The QHLO command

An SMTP client that wishes to use QUICKSTART MUST start the SMTP session by issuing a QHLO command. The QHLO command identifies the client, states that it supports QUICKSTART and possibly other SMTP service extensions, allows the server to verify that the client has read the server's list of supported service extensions correctly, and together with a 250 reply from the server confirms that the client and server are in the initial state.

Syntax:

```
qhlo          =  "QHLO" SP Domain SP qhlo-id CRLF

qhlo-ok-rsp   =  *(  "250-" [    text ] CRLF  )
                     "250"  [ SP text ] CRLF

qhlo-short-no =  *(  "504-" [    text ] CRLF  )
                     "504"  [ SP text ] CRLF

qhlo-long-no  =      "520-" Domain [ SP ehlo-greet ] CRLF
                 *(  "520-" ehlo-line                 CRLF  )
                     "520 " ehlo-line                 CRLF
```

The QHLO command has two parameters. The first <Domain> parameter identifies the client and has the same semantics as the parameter to the EHLO command. The second <qhlo-id> parameter MUST be the same as the <qhlo-id> that would have been given by the server in its response to EHLO or was given in its greeting; it allows the server to verify that the client has read or cached the server's list of supported service extensions correctly.

If the <qhlo-id> is correct, the server SHOULD give a <qhlo-ok-rsp> reply. If the <qhlo-id> does not match the server SHALL give a <qhlo-short-no> reply or a <qhlo-long-no> reply as described below. Replies to QHLO SHOULD NOT include enhanced status codes [RFC2034] [RFC3463].

The QHLO command MAY be pipelined. If the server does not give a 250 reply to QHLO then it SHOULD reject subsequent commands other than NOOP, QHLO, EHLO, HELO, or QUIT with "503 Bad sequence of commands" replies (because the client expects these commands to be interpreted

following a <qhlo-ok-rsp>). If the server supports ENHANCEDSTATUSCODES [RFC2034] [RFC3463], the status code to be returned SHOULD be 5.5.1.

The client MUST be prepared for its QHLO command to be rejected. (This can be caused by server configuration changes or client mobility.) If its QHLO command is rejected for any reason, the client MUST discard all cached lists of supported service extensions for this server for all security contexts. It MUST check the server's replies to any subsequent pipelined commands as usual. The client SHOULD recover as described below.

The QHLO command is typically used first after establishing an initial connection, and again after establishing a security layer using either the STARTTLS command [RFC3207] or the AUTH command [RFC2554] with a suitable SASL mechanism. The requirements vary

slightly between these two situations, so they are described separately in the next two sections.

Simple pipelining of the QHLO command instead of the non-pipelined EHLO command allows one round trip to be saved. It is possible to save more round trips if the client can issue commands before it receives the list of service extensions supported by the server. The next two sections describe how this works.

6. After establishing a connection

If the client has not cached the server's list of supported service extensions, then it MUST wait for the server's greeting, then check it against the syntax specified in Section 4. If the list indicates that the server does not support QUICKSTART then the client SHALL proceed without using any QUICKSTART features. If the server does support QUICKSTART then the client SHOULD cache the list of supported service extensions given in the server's greeting, and issue a QHLO command as described in Section 5.

The rest of this section describes what happens when the client has previously cached the server's list of supported extensions. In this situation the client MAY issue its QHLO command as soon as a connection is established, without waiting for the server's greeting. This saves one round trip.

A server that supports QUICKSTART MUST permit clients to issue a pipelined sequence of commands starting with QHLO before the server sends its <qsmtp-greet> (which it MUST still do as usual). The server SHALL check the client's <qhlo-id> as described in Section 5. If it is incorrect the server MUST reject the QHLO command with a 504 <qhlo-short-no> reply.

If its QHLO command is rejected then the client SHOULD recover by checking for the correct supported service extension list in a <qsmtp-greet> given by the server. It SHOULD store this replacement list in the cache. It SHOULD re-issue its QHLO command with a corrected <qhlo-id> from the greeting.

If the server did not give a <qsmtp-greet> then the client should proceed as described in Section 8.

7. After establishing a security layer

When a security layer has been established using the STARTTLS [RFC3207] or AUTH [RFC2554] commands, the connection is reset to its

initial state after the server's greeting.

If the client has not cached the server's list of supported service extensions, then it MUST issue an EHLO command to obtain the list. If the list indicates that the server does not support QUICKSTART then the client SHALL proceed without using any QUICKSTART features. If the server does support QUICKSTART then the client SHOULD cache the list of supported service extensions given in the server's reply to EHLO.

The rest of this section describes what happens when the client has previously cached the server's list of supported extensions. In this situation the client MAY issue a QHLO command as soon as the security layer is established, instead of an EHLO command. This saves one round trip.

The server SHALL check the client's <qhlo-id> as described in Section 5. If it is incorrect the server MUST reject the QHLO command with a 520 <qhlo-long-no> reply. This reply MUST NOT include enhanced status codes [RFC2034] [RFC3463]. It includes the server's list of supported service extensions which MUST be the same as the server would have given in its 250 reply if the client said EHLO instead of QHLO. This allows the client to obtain a corrected <qhlo-id> as soon as possible.

If its QHLO command is rejected with a <qhlo-long-no> then the client SHOULD examine the 520 reply given by the server for the correct supported service extension list. It SHOULD store this replacement list in the cache. It SHOULD re-issue its QHLO command with a corrected <qhlo-id> from the 520 reply.

If the server did not give a <qhlo-long-no> then the client should proceed as described in Section 8.

8. Backwards compatibility

If the client's first command after connection establishment is not QHLO, the server MAY check (to the best of its ability) that the command was issued after the client received the server's greeting. If the client did not wait then the server MAY treat it as abusive.

A client that issues a QHLO command based on its cache MUST be prepared to deal with errors that indicate the server no longer supports QUICKSTART, i.e. a missing <qsmtp-greet> or <qhlo-long-no>. As with any rejection of QHLO, in this situation the client MUST discard its cached list of the server's supported service extensions. Since there is no replacement list to store in the cache, the

client's subsequent connections to the server will not use QUICKSTART. The client MUST be prepared for the server to get upset (e.g. drop the connection) because the client did not wait for its greeting or because it issued a TLS client hello when the server was expecting an SMTP command; the client should then re-try with a new SMTP connection. Otherwise the client MAY recover within the same connection by issuing an EHLO command and proceeding without using QUICKSTART features.

9. Changes to the STARTTLS command

If the server supports QUICKSTART and STARTTLS [RFC3207], then the client MAY start the TLS handshake immediately after issuing the STARTTLS command, without waiting for the server's reply to STARTTLS.

The sequence of operations from the client's point of view is: send the STARTTLS command immediately followed by the TLS client hello; then receive the reply to the STARTTLS command, and if that reply has a 220 code, receive the TLS server hello; then proceed with TLS as usual. This is illustrated in Appendix A.

If the server wishes to reject the STARTTLS command, it MUST discard any pipelined TLS client hello before giving its reply.

This change allows one round trip to be saved.

10. Changes to the AUTH command

If the server supports QUICKSTART and AUTH [RFC2554], and the client uses a SASL mechanism which can be completed in one round trip (such as EXTERNAL [RFC4422] or PLAIN [RFC4616]) then the client MAY issue the the AUTH command at any point in a pipelined group. If the authentication fails, the server SHOULD reject all subsequent commands other than AUTH, NOOP, HELO, EHLO, QHLO, or QUIT with a "530 Authentication failure" reply (because the client expects these commands to be interpreted following a successful authentication). If the server supports ENHANCEDSTATUSCODES [RFC2034] [RFC3463], the status code to be returned SHOULD be 5.7.0.

This change allows one round trip to be saved. It is also suggested in [RFC4468].

11. IANA Considerations

This memo defines a new SMTP service extension keyword, "QUICKSTART"

in Section 3.

IANA maintains a registry of "WITH protocol types" for use in the "with" clause of the Received header trace fields in an Internet message. Most of the contents of this registry are set out in [RFC3848]. This specification updates the registry as follows:

- The new keyword "QSMTP" indicates that the client used the QUICKSTART extension, that is, it used the QHLO command to start the SMTP session instead of EHLO.

- The new keyword "QSMTPA" indicates that the client used QUICKSTART with the SMTP AUTH [RFC2554] extension, and that it authenticated successfully.

- The new keyword "QSMTPS" indicates that the client used QUICKSTART, and that it successfully completed TLS negotiation to provide a strong transport encryption layer by using the QTLS or STARTTLS commands.

- The new keyword "QSMTPSA" indicates that the client used QUICKSTART and both TLS and AUTH were successfully negotiated (the combination of QSMTPS and QSMTPA).

12. Security considerations

Like the EHLO command, the QHLO command includes a client host name parameter for logging and tracing purposes. This can useful to distinguish different clients behind a NAT, for example. However clients commonly state an incorrect host name, so it SHOULD NOT be relied on. SMTP servers SHOULD use all available client identifiers for logging and tracing, such as its IP address, reverse DNS, TLS certificate, and SMTP AUTH credentials.

It is increasingly popular for SMTP servers (especially MX hosts) to use heuristics based on protocol conformance to identify abusive clients and reject email from them. For example, the server can delay its greeting and see if the client waits to receive it before issuing commands. This heuristic can still be applied in the presence of full support for QUICKSTART, by checking that the client's early first command is not QHLO before deciding that it is abusive.

QUICKSTART requires the client to demonstrate that it has received the server's list of supported extensions, by echoing the <qhlo-id>. This can be faked by abusive clients if the <qhlo-id> is too easy to guess. The server can defend itself against abuse by making the

<qhlo-id> depend on more than just the list of supported extensions and their parameters: for example it might use a digest of the list, the server and client IP addresses, and a secret.

Of course, since QUICKSTART is most useful for interactive message submission, and high latency is not such a problem for automated message relay, MX hosts can simply elect not to support QUICKSTART.

13. References

13.1. Normative references

[RFC2034] Freed, N., "SMTP Service Extension for Returning Enhanced Error Codes", RFC 2034, October 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2554] Myers, J., "SMTP Service Extension for Authentication", RFC 2554, March 1999.

[RFC2821] Klensin, J., "Simple Mail Transfer Protocol", RFC 2821, April 2001.

[RFC2920] Freed, N., "SMTP Service Extension for Command Pipelining", STD 60, RFC 2920, September 2000.

[RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, February 2002.

[RFC3463] Vaudreuil, G., "Enhanced Mail System Status Codes", RFC 3463, January 2003.

[RFC3848] Newman, C., "ESMTP and LMTP Transmission Types

Registration", RFC 3848, July 2004.

[RFC4234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

13.2. Informative references

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[RFC4409] Gellens, R. and J. Klensin, "Message Submission for Mail", RFC 4409, April 2006.

[RFC4422] Melnikov, A. and K. Zeilenga, "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.

[RFC4468] Newman, C., "Message Submission BURL Extension", RFC 4468, May 2006.

[RFC4616] Zeilenga, K., "The PLAIN Simple Authentication and Security Layer (SASL) Mechanism", RFC 4616, August 2006.

[old-quickstart-01]
Finch, T., "The QUICKSTART SMTP service extension", Internet Draft draft-fanf-smtp-quickstart-01.txt, February 2007.

[quickstart-a]
Finch, T., "The QUICKSTART SMTP service extension (simple profile)", Internet
Draft draft-fanf-smtp-quickstart-a-00.txt, April 2007.

Appendix A. Example round trip summaries

In the following examples, a line prefixed with C: indicates the start of a packet from the client to the server, and S: indicates one from the server to the client. Subsequent lines without a prefix are pipelined into the same packet. The C: or S: may be followed by a protocol name - TCP, SMTP or TLS - which indicates a change in the top-most protocol active in the connection. The rest of the line describes a protocol element.

The following figure outlines the sequence of packets at the start of a non-QUICKSTART message submission [RFC4409], including the TCP handshake [RFC0793], starting ESMTP [RFC2821], securing it using STARTTLS [RFC3207] [RFC4346], authenticating [RFC2554] [RFC4422] [RFC4616], and starting to send the message using pipelining [RFC2920]. The MAIL command appears in the 9th packet from the client, and pipelining is not possible before this point.

```
     C: TCP  SYN
     S:      SYN/ACK
     C:      ACK
     S: SMTP <greeting>
     C:      EHLO <Domain>
     S:      <ehlo-ok-rsp>
     C:      STARTTLS
     S:      220 OK
     C: TLS  client hello
     S:      server hello, certificate, etc.
     C:      key exchange, change cipher, finish handshake
     S:      change cipher, finish handshake
     C: SMTP EHLO <Domain>
     S:      <ehlo-ok-rsp>
     C:      AUTH PLAIN <creds>
     S:      235 OK
     C:      MAIL FROM:<...>
             RCPT TO:<...>
             ...
```

The following shows a successful QUICKSTART connection with a TLS session cache hit. The MAIL command is in the client's 3rd packet.

```
     C: TCP  SYN
     S:      SYN/ACK
     C:      ACK
        SMTP QHLO <Domain> <qhlo-id>
             STARTTLS
        TLS  client hello
     S: SMTP <qsmtp-greet>
             250 OK
             220 OK
        TLS  server hello, change cipher, finish handshake
     C:      change cipher, finish handshake
        SMTP QHLO <Domain> <qhlo-id>
             AUTH PLAIN <creds>
             MAIL FROM:<...>
             RCPT TO:<...>
             ...
     S:      250 OK
             235 OK
             250 OK
             ...
```

If the client has not cached the server's <qhlo-id>s then the negotiation is as follows. The MAIL command is in the client's 6th packet.

```
     C: TCP  SYN
     S:      SYN/ACK
     C:      ACK
     S: SMTP <qsmtp-greet>
     C:      QHLO <Domain> <qhlo-id>
             STARTTLS
        TLS  client hello
     S: SMTP 250 OK
             220 OK
        TLS  server hello, certificate, etc.
     C:      key exchange, change cipher, finish handshake
     S:      change cipher, finish handshake
     C: SMTP EHLO <Domain>
     S:      <ehlo-ok-rsp>
     C:      AUTH PLAIN <creds>
             MAIL FROM:<...>
             RCPT TO:<...>
             ...
     S:      235 OK
```

```
               250 OK
               ...
```

If the QUICKSTART handshake fails, the server rejects the initial pipelined commands and discards the pipelined TLS client hello, and the client renegotiates. The MAIL command is still in the client's 6th packet.

```
     C: TCP  SYN
     S:      SYN/ACK
     C:      ACK
        SMTP QHLO <Domain> <qhlo-id>
             STARTTLS
        TLS  client hello
     S: SMTP <qsmtp-greet>
             504 bad QHLO ID
             503 bad sequence of commands
     C:      QHLO <Domain> <qhlo-id>
             STARTTLS
        TLS  client hello
     S: SMTP 250 OK
             220 OK
        TLS  server hello, certificate, etc.
     C:      key exchange, change cipher, finish handshake
     S:      change cipher, finish handshake
     C: SMTP EHLO <Domain>
     S:      <ehlo-ok-rsp>
     C:      AUTH PLAIN <creds>
             MAIL FROM:<...>
             RCPT TO:<...>
             ...
     S:      235 OK
             250 OK
             ...
```

Appendix B. Changes from previous versions

This memo is derived from [old-quickstart-01] by separating it into two drafts, this one describing the more advanced profile of the ideas, and [quickstart-a] describing the simpler profile.

The QUICKSTART extension no longer has any optional features.

PIPELINING is now a required prerequisite for QUICKSTART.

Clarify why commands are rejected after a failed AUTH.

Require consistency between the server's various lists of supported service extensions.

Some ABNF fixes.

Better consistency with existing SMTP reply codes.

This profile omits the QTLS command, and the simple form of the QHLO command.

Author's Address

Tony Finch

University of Cambridge Computing Service
New Museums Site
Pembroke Street
Cambridge CB2 3QH
ENGLAND

Phone: +44 797 040 1426
Email: dot@dotat.at
URI: http://dotat.at/

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).