SMTP
Internet-Draft
Intended status: Standards Track
Expires: August 5, 2007

T. Finch
University of Cambridge

February 2007

The QUICKSTART SMTP service extension
draft-fanf-smtp-quickstart-00

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 August 5, 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.

Document revision

$Cambridge: hermes/doc/qsmtp/draft-fanf-smtp-quickstart.xml,v 1.20 2007/02/24 10:50:04 fanf2 Exp $

Table of Contents

1. Introduction 4
2. Terminology 5
3. QUICKSTART SMTP service extension definition 6
4. Extended server greeting 7
5. The QHLO command 8
6. The QTLS command 9
7. Changes to the AUTH command 11
8. Issuing commands early 12
9. Changes to the STARTTLS command 14
10. Backwards compatibility 15
11. IANA Considerations 16
12. Security considerations 17
13. References 18
13.1. Normative references 18
13.2. Informative references 18
Appendix A. Example round trip summaries 20
Author's Address 27
Intellectual Property and Copyright Statements 28

1. Introduction

The start of an SMTP connection is not currently optimized to reduce the number of round trips between the client and server. This is a particular problem for message submission over slow links, where the resulting delay can easily be more than a second. The QUICKSTART SMTP service extension introduces some changes which eliminate round trips and therefore reduce the delay before messages can be transmitted. The key features of the extension are:

- Earlier announcement of the list of SMTP service extensions supported by the server.
- The QHLO and QTLS commands, which are quicker replacements for the EHLO and STARTTLS commands.
- 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 extension list.

Each of these features has advantages by itself, and they can be used together to provide greater improvements. QUICKSTART is designed with two profiles in mind: the basic profile uses the QHLO and QTLS commands and the pipelined AUTH command to bring the first MAIL command forward from the client's 9th packet to the 5th; the full profile uses the <qhlo-id> and pipelined STARTTLS to bring it as early as the 3rd. 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.

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 BNF" 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., <list-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".
- The QUICKSTART EHLO keyword value has four OPTIONAL parameters ("QTLS", "AUTH", "QHLO=<qhlo-id>", and "STARTTLS") to which more can be added by future specifications.
- The extension defines two additional commands: "QHLO" and "QTLS".
- 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 response, 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 SHOULD include PIPELINING [RFC2920].

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

Syntax:

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

ehlo-line    =/ "QUICKSTART" *( SP qsmtp-param )

qsmtp-param  =  "QHLO=" qhlo-id / "QTLS" /
                "AUTH" / "STARTTLS" /
                esmtp-param

qhlo-id      =  esmtp-value
```

The QUICKSTART keyword MAY be followed by parameters that indicate which OPTIONAL features of this specification are supported by the server. Clients MUST ignore QUICKSTART parameters that they do not understand. The following parameters are defined in this specification:

- The "QTLS" parameter indicates that the server supports the QTLS command specified in Section 6.
- The "AUTH" parameter indicates that the server supports pipelining of the AUTH command as specified in Section 7.
- The "QHLO=<qhlo-id>" parameter indicates that the server allows the client to issue commands before receiving the server's list of supported extensions, as described in Section 8.
- The "STARTTLS" parameter indicates that the server supports pipelining of the STARTTLS command as specified in Section 9.

5. The QHLO command

The QHLO command has two forms. Support for the basic form defined in this section is REQUIRED. The full form is OPTIONAL, and is defined in Section 8.

An SMTP client that supports QUICKSTART MUST first ensure that the server supports it by checking the server's greeting against the syntax specified in Section 4. The client then issues QHLO as its first command instead of EHLO. The basic form of QHLO has the same semantics as the EHLO command, except that it MAY be pipelined if the server also supports PIPELINING [RFC2920], and the server's 220 response to QHLO does not include its list of supported extensions.

The pipelined QHLO command allows a client to save one round trip by not having to wait for the server's response before issuing further commands.

Syntax:

```
qhlo         =  "QHLO" SP Domain CRLF

qhlo-ok-rsp  =  "220" SP domain [ SP ehlo-greet ] CRLF
```

6. The QTLS command

Support for the QTLS command is OPTIONAL, and is indicated by the presence of the QTLS parameter after the QUICKSTART keyword in the list of extensions supported by the server. If the server supports both STARTTLS [RFC3207] and QUICKSTART then it SHOULD also support QTLS.

Syntax:

```
qtls  =  "QTLS" CRLF
```

The QTLS command has no parameters. After issuing the QTLS command, the client immediately starts the TLS handshake. The server MUST NOT issue an SMTP response to the QTLS command before the TLS handshake. Therefore it is not possible for the server to report problems (as with the 454 response to the STARTTLS command) except by closing the connection - which it SHOULD NOT do. Instead, the server SHOULD ensure that it can support TLS (e.g. by initializing its TLS library) before listing the extensions it supports, rather than lazily in reaction to a QTLS or STARTTLS command.

After the TLS handshake has been completed, the SMTP protocol is reset to the state at the start of a connection, so the server proceeds to issue a QUICKSTART extended 220 greeting <qsmtp-greet>.

If the server requires that the client performs a TLS negotiation before it accepts any commands, then it SHOULD accept the QHLO and QTLS commands before TLS negotiation, as well as the NOOP, EHLO, STARTTLS, and QUIT commands specified in [RFC3207]

Apart from these differences, the QTLS command is the same as the STARTTLS command, and the other requirements specified in [RFC3207] apply.

The QTLS command saves one round trip before the TLS handshake by omitting the server's response, and saves another after the TLS handshake by announcing the list of extensions supported by the server earlier than is typical with STARTTLS.

Clients will generally want to issue the QTLS command at the start of a connection, immediately after the QHLO command. The most efficient way of doing so is to pipeline QHLO, QTLS, and the TLS handshake, but then it can be difficult to disentangle the server's response to QHLO from its side of the TLS handshake. Therefore servers MUST allow clients to issue the QTLS command without issuing QHLO or EHLO first, though they MAY require that the client does not issue the QHLO command before receiving the server's greeting. This is illustrated

in Appendix A.

7. Changes to the AUTH command

This section describes an OPTIONAL change to the behaviour of the AUTH command [RFC2554], support for which is indicated by the presence of the AUTH parameter after the QUICKSTART keyword in the list of extensions supported by the server. This change is also specified by [RFC4468], so if the server lists support for BURL and QUICKSTART and PIPELINING then it MUST include the QUICKSTART AUTH parameter.

If the client uses a SASL mechanism which can be completed in one round trip, such as EXTERNAL [RFC4422] or PLAIN [RFC4616], then it MAY pipeline the AUTH command. 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. If the server supports ENHANCEDSTATUSCODES [RFC2034] [RFC3463], the status code to be returned SHOULD be 5.7.0.

This change allows another round trip to be saved.

8. Issuing commands early

It is possible to save more round trips if the client can issue commands before receiving the list of extensions supported by the server. There are opportunities to do this at the very start of the TCP connection, after the STARTTLS command [RFC3207], and after the AUTH command negotiates a security layer [RFC2554]. These are called "QUICKSTART points" in this section.

Support for this is OPTIONAL, and is indicated by the presence of the "QHLO=<qhlo-id>" parameter after the QUICKSTART keyword in the list of extensions supported by the server. The server MUST also support PIPELINING [RFC2920].

The <qhlo-id> parameter value is a case-sensitive token which 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 if any of their parameters changes. This generally implies that the server will state different <qhlo-id>s at different QUICKSTART points. For example, if the server only supports AUTH PLAIN after TLS negotiation, it will state a different <qhlo-id> in its responses to EHLO before and after STARTTLS. If the server's SIZE parameter [RFC1870] changes in response to available disk space, its <qhlo-id> will change too. Section 12 includes some security considerations for the <qhlo-id>.

The client uses the <qhlo-id> with the full form of the QHLO command.

Syntax:

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

The <qhlo-id> in the client's QHLO command MUST match the <qhlo-id> issued by the server in its <qsmtp-greet>, or the <qhlo-id> that the server would have issued in its <ehlo-ok-rsp> response to an EHLO command from the client. If the <qhlo-id> matches, the server SHOULD respond with a <qhlo-ok-rsp> response. If the <qhlo-id> does not match there are two possible responses: if the server has already listed its supported service extensions (e.g. in a <qsmtp-greet>) it responds with a "520 Please use the correct QHLO ID" response; otherwise (e.g. after STARTTLS or AUTH) it responds with a <qhlo-long-no>, which includes the list of extensions supported by the server.

Syntax:

```
qhlo-long-no  =  (    "521-"   domain [ SP ehlo-greet ] CRLF
                    *( "521-"   ehlo-line                CRLF )
                       "521" SP ehlo-line                CRLF   )
```

If the server responds to QHLO with a 520 or 521 code, it SHOULD reject subsequent commands other than NOOP, QHLO, or QUIT with "503 Bad sequence of commands" responses. If the server supports ENHANCEDSTATUSCODES [RFC2034] [RFC3463], the status code to be returned SHOULD be 5.5.1. The client SHOULD recover from this state by issuing a QHLO command containing the correct <qhlo-id>.

The client MAY issue commands early at a QUICKSTART point if it has previously connected to the server and the server included QUICKSTART QHLO=<qhlo-id> in its list of supported extensions at that point. The client's first command at the QUICKSTART point MUST be the full form of QHLO. The client MUST have remembered the details of the extensions that the client is interested in and which are supported by the server at that point, including the <qhlo-id>. The client MUST NOT confuse <qhlo-id>s obtained from different QUICKSTART points. The client MUST NOT assume that the server's current <qhlo-id> will be the same as on the previous connection, therefore the client MUST be prepared for its QHLO being rejected. Note that this implies that the client SHOULD NOT pipeline an early QHLO command and a QTLS command, since the server cannot reject the QTLS command except by closing the connection; instead it SHOULD use STARTTLS as described in Section 9.

9. Changes to the STARTTLS command

This section describes an OPTIONAL change to the behaviour of the STARTTLS command [RFC3207] which is indicated by the presence of the STARTTLS parameter after the QUICKSTART keyword in the list of extensions supported by the server. If the server lists both the

QTLS and QHLO=<qhlo-id> parameters, then it SHOULD also list the STARTTLS parameter.

The advantages of the QTLS command are that it allows the client to initiate a TLS handshake faster without getting muddled by SMTP responses from the server, and it allows the client to receive the list of extensions supported by the server inside TLS sooner. When the client uses the QHLO=<qhlo-id> feature to issue commands early, it isn't possible to avoid interleaving SMTP responses with the TLS handshake, and it is no longer necessary to get the server's list of supported extensions. Furthermore, the QTLS command's lack of error recovery becomes a problem. Therefore full QUICKSTART uses a pipelined STARTTLS command.

If the server supports QUICKSTART STARTTLS, then the client MAY start the TLS handshake immediately after issuing the STARTTLS command, without waiting for the server's response to STARTTLS. If the server wishes to reject the STARTTLS command, it MUST discard any pipelined TLS client hello before issuing its response.

10. Backwards compatibility

If the first command issued by the client is EHLO, then the server MUST treat it as normal extended SMTP conversation as specified in [RFC2821]. In this case the client MUST disregard the list of extensions provided in the server's greeting. For example, if the server lists PIPELINING in its greeting, the client MUST NOT assume that PIPELINING will be listed in the server's response to EHLO.

If the first command issued by the client is not QTLS nor QHLO nor EHLO, then the server MUST treat it as an un-extended SMTP conversation as specified in [RFC0821].

In the above situations, the server MAY require that the client's first command is issued after the client has received the greeting.

A client that issues commands early (as described in the two previous sections) MUST be prepared to deal with errors that indicate the server no longer supports QUICKSTART, such as 50X or 55X SMTP responses [RFC2821], or 5.5.X, 5.3.X, or 5.7.X enhanced status codes [RFC3463]. If the client issued a QTLS or pipelined STARTTLS command, it SHOULD disconnect and re-try using ESMTP without QUICKSTART; otherwise it MAY recover by issuing an EHLO command and re-trying.

Similarly, a client that issues commands early MUST be prepared to deal with a 421 or 4.3.2 "Service not available" error. In this situation the client SHOULD disconnect and re-try some time later.

11. IANA Considerations

This memo defines a new SMTP service extension keyword, "QUICKSTART" in Section 3. Four parameters are initially defined for the keyword. Further parameters MAY be defined by future specifications and MUST conform to the <esmtp-param> syntax; parameter keywords starting with X are for Private Use, and other parameter keywords SHOULD be allocated according to the Specification Required policy [RFC2434].

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.

The full QUICKSTART protocol requires the client to demonstrate that it has previously communicated with the server, 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 pump-and-dump 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

[RFC0821] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, August 1982.

[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.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[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.

[RFC1870] Klensin, J., Freed, N., and K. Moore, "SMTP Service Extension for Message Size Declaration", STD 10, RFC 1870, November 1995.

[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.

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 the use of a pipelined QHLO command, but no other QUICKSTART features. (It is not very realistic.) The MAIL command is in the client's 8th packet.

```
C: TCP  SYN
S:      SYN/ACK
C:      ACK
S: SMTP <qsmtp-greet>
C:      QHLO <domain>
        STARTTLS
S:      220 OK
        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 the use of the QTLS and QHLO commands, to illustrate a basic QUICKSTART connection. The MAIL command is in the client's 6th packet.

```
      C: TCP  SYN
      S:      SYN/ACK
      C:      ACK
      S: SMTP <qsmtp-greet>
      C:      QTLS
         TLS  client hello
      S:      server hello, certificate, etc.
      C:      key exchange, change cipher, finish handshake
      S:      change cipher, finish handshake
         SMTP <qsmtp-greet>
      C:      QHLO <domain>
              AUTH PLAIN <creds>
      S:      220 OK
              235 OK
      C:      MAIL FROM:<...>
              RCPT TO:<...>
              ...
```

Note that the client omitted the QHLO at the start of the previous connection. This simplification avoids the following situation where it has to switch back and forth between SMTP and TLS at the start of the TLS handshake.

```
              ...
      S: SMTP <qsmtp-greet>
      C:      QHLO <domain>
              QTLS
         TLS  client hello
      S: SMTP 220 OK
         TLS  server hello, certificate, etc.
              ...
```

With pipelined AUTH, the MAIL command is in the client's 5th packet.

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

The following shows the use of QTLS with a TLS session cache hit and use of the full QHLO command after TLS is started. This form is simpler than pipelined STARTTLS (as shown below) but slightly less efficient. The MAIL coomand is in the client's 4th packet.

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

The following shows a successful full QUICKSTART connection. The MAIL command is in the client's 4th packet.

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

If the QUICKSTART handshake fails, the server rejects the initial pipelined commands and discards the first TLS client hello, and the client falls back to basic QUICKSTART. The MAIL command is in the client's 5th packet, the same as basic QUICKSTART without the failed handshake.

```
      C: TCP  SYN
      S:      SYN/ACK
      C:      ACK
         SMTP QHLO <domain> <qhlo-id>
              STARTTLS
         TLS  client hello
      S: SMTP <qsmtp-greet>
              520 bad QHLO ID
              503 bad sequence of commands
      C:      QHLO <domain> <qhlo-id>
              QTLS
```

```
           TLS  client hello
       S: SMTP 220 OK
           TLS  server hello, certificate, etc.
       C:      key exchange, change cipher, finish handshake
       S:      change cipher, finish handshake
               <qsmtp-greet>
       C:      QHLO <domain>
               AUTH PLAIN <creds>
               MAIL FROM:<...>
               RCPT TO:<...>
               ...
       S:      220 OK
               235 OK
               250 OK
               ...
```

The following shows a successful full 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>
               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:      220 OK
               235 OK
               250 OK
               ...
```

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).