EMAILCORE J. Klensin
Internet-Draft 26 February 2024
Obsoletes: 5321, 1846, 7504, 7505 (if approved)
Intended status: Standards Track
Expires: 29 August 2024

Simple Mail Transfer Protocol
draft-ietf-emailcore-rfc5321bis-27

Abstract

This document is a specification of the basic protocol for Internet electronic mail transport. It (including text carried forward from RFC 5321) consolidates, updates, and clarifies several previous documents, making all or parts of most of them obsolete. It covers the SMTP extension mechanisms and best practices for the contemporary Internet, but does not provide details about particular extensions. The document also provides information about use of SMTP for other than strict mail transport and delivery. This document replaces RFC 5321, the earlier version with the same title, and supersedes RFCs 1846, 7504, and 7505, incorporating all the relevant information in them.

Notes on Reading This Working Draft

// RFC Editor: Please remove this note.

Early versions of this working draft were extensively annotated with information, primarily about changes made over the decade since RFC 5321 appeared, especially when those changes might be controversial or should get careful review. Most of those annotations and associated questions are marked in CREF comments ("//" in the text form). Starting with version -09 of the draft, annotations and notes that were no longer relevant are being pruned to improve readability. In general, any annotations or comments not marked with "[[Note in Draft", in the contents of an "Editor's note", or in the "Errata Summary" appendix (Appendix I.1, are just notes on changes that have already been made and where those changes originated. Completed changes are described in Appendix H. Those that are still in progress or awaiting WG review are listed in Appendix G (and not identified in their title lines as "closed"). As one can tell from the dates (when they are given), this document has been periodically updated over a very long period of time.

As people review or try to use this document, it may be worth paying special attention to the historical discussion in Section 1.2.

This evolving draft should be discussed on the emailcore@ietf.org list.

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 https://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 29 August 2024.

Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction 10
1.1. Transport of Electronic Mail 10
1.2. History and Context for This Document 11
1.3. Document Conventions 12
2. The SMTP Model 12
2.1. Basic Structure 12
2.2. The Extension Model 15
2.2.1. Background 15
2.2.2. Definition and Registration of Extensions 16
2.2.3. Special Issues with Extensions 17
2.3. SMTP Terminology 17
2.3.1. Mail Objects 17

2.3.2. Senders and Receivers 18
2.3.3. Mail Agents and Message Stores 18
2.3.4. Host 19
2.3.5. Domain Names 19
2.3.6. Buffer and State Table 20
2.3.7. Commands and Replies 20
2.3.8. Lines 20
2.3.9. Message Content and Mail Data 21
2.3.10. Originator, Delivery, Relay, and Gateway Systems . . 21
2.3.11. Mailbox and Address 22
2.3.12. Sessions and Transactions 22
2.4. General Syntax Principles and Transaction Model 22
3. The SMTP Procedures: An Overview 25
3.1. Session Initiation 25
3.2. Client Initiation 25
3.3. Mail Transactions 26
3.4. Address Modification and Expansion 29
3.4.1. Forwarding for Address Correction or Updating 29
3.4.2. Aliases and Mailing Lists 30
3.4.2.1. Simple Aliases 30
3.4.2.2. Mailing Lists 30
3.5. Commands for Debugging Addresses 31
3.5.1. Overview 31
3.5.2. VRFY Normal Response 33
3.5.3. Meaning of VRFY or EXPN Success Response 34
3.5.4. Semantics and Applications of EXPN 34
3.6. Relaying and Mail Routing 34
3.6.1. Mail eXchange Records and Relaying 35
3.6.2. Message Submission Servers as Relays 36
3.7. Mail Gatewaying 36
3.7.1. Header Fields in Gatewaying 36
3.7.2. Received Lines in Gatewaying 37
3.7.3. Addresses in Gatewaying 37
3.7.4. Other Header Fields in Gatewaying 38
3.7.5. Envelopes in Gatewaying 38
3.8. Terminating Sessions and Connections 38
4. The SMTP Specifications 39
4.1. SMTP Commands 39
4.1.1. Command Semantics and Syntax 39
4.1.1.1. Extended HELLO (EHLO) or HELLO (HELO) 40
4.1.1.2. MAIL (MAIL) 42
4.1.1.3. RECIPIENT (RCPT) 42
4.1.1.4. DATA (DATA) 44
4.1.1.5. RESET (RSET) 45
4.1.1.6. VERIFY (VRFY) 46
4.1.1.7. EXPAND (EXPN) 46
4.1.1.8. HELP (HELP) 46
4.1.1.9. NOOP (NOOP) 47

4.1.1.10. QUIT (QUIT) 47
4.1.2. Command Argument Syntax 48
4.1.3. Address Literals 50
4.1.4. Order of Commands 51
4.2. SMTP Replies 53
4.2.1. Reply Code Severities and Theory 55
4.2.2. Reply Codes by Function Groups 57
4.2.3. Reply Codes in Numeric Order 59
4.2.4. Some specific code situations and relationships . . . 60
4.2.4.1. Reply Code 502 61
4.2.4.2. "No mail accepted" situations and the 521, 554, 556, and 450 codes 61
4.2.4.3. Reply Codes after DATA and the Subsequent <CRLF>.<CRLF> 62
4.3. Sequencing of Commands and Replies 63
4.3.1. Sequencing Overview 63
4.3.2. Command-Reply Sequences 64
4.4. Trace Information 66
4.4.1. Received Header Field (Time Stamp) 66
4.4.2. Return-path Header Field 67
4.4.3. Return-path, Non-SMTP Systems, and Gateways 68
4.4.4. Additional Trace Fields 68
4.4.5. Trace Information Summary and Analysis 68
4.5. Additional Implementation Issues 70
4.5.1. Minimum Implementation 70
4.5.2. Transparency 71
4.5.3. Sizes and Timeouts 71
4.5.3.1. Size Limits and Minimums 71
4.5.3.1.1. Local-part 72
4.5.3.1.2. Domain 72
4.5.3.1.3. Path 72
4.5.3.1.4. Command Line 72
4.5.3.1.5. Reply Line 72
4.5.3.1.6. Text Line 72
4.5.3.1.7. Message Content 72
4.5.3.1.8. Recipient Buffer 73
4.5.3.1.9. Treatment When Limits Exceeded 73
4.5.3.1.10. Too Many Recipients Code 73
4.5.3.2. Timeouts 74
4.5.3.2.1. Initial 220 Message: 5 Minutes 74
4.5.3.2.2. MAIL Command: 5 Minutes 74
4.5.3.2.3. RCPT Command: 5 Minutes 75
4.5.3.2.4. DATA Initiation: 2 Minutes 75
4.5.3.2.5. Data Block: 3 Minutes 75
4.5.3.2.6. DATA Termination: 10 Minutes. 75
4.5.3.2.7. Server Timeout: 5 Minutes. 75
4.5.4. Retry Strategies 75
4.5.5. Messages with a Null Reverse-Path 77

5. Address Resolution and Mail Handling 78
  5.1. Locating the Target Host 78
  5.2. IPv6 and MX Records 81
6. Problem Detection and Handling 81
  6.1. Reliable Delivery and Replies by Email 81
  6.2. Unwanted, Unsolicited, and "Attack" Messages 82
  6.3. Loop Detection 83
  6.4. Compensating for Irregularities 83
7. Security Considerations 84
  7.1. Mail Security and Spoofing 85
  7.2. Hiding Addresses from Trace 86
  7.3. VRFY, EXPN, and Security 86
  7.4. Mail Rerouting Based on the 251 and 551 Response Codes . 87
  7.5. Information Disclosure in Announcements 87
  7.6. Information Disclosure in Trace Fields 87
  7.7. Information Disclosure in Message Forwarding 88
  7.8. Local Operational Requirements and Resistance to Attacks 88
  7.9. Scope of Operation of SMTP Servers 88
8. IANA Considerations 89
  8.1. SMTP-related Registries 89
    8.1.1. Simple Mail Transfer Protocol (SMTP) Service Extensions 89
      8.1.1.1. Registration Models 89
      8.1.1.2. Registry Information 91
    8.1.2. Address Literal Tags 92
    8.1.3. Mail Transmission Types 92
    8.1.4. Additional Registered Clauses 92
  8.2. New Registry Actions with <<This Document>> 92
  8.3. Specification of Registry Group and Registry Structure . 93
9. Acknowledgments 94
10. References 95
  10.1. Normative References 95
  10.2. Informative References 98
Appendix A. TCP Transport Service 103
Appendix B. Generating SMTP Commands from Internet Message Format Header Fields 103
Appendix C. Placeholder (formerly Source Routes) 105
Appendix D. Scenarios 105
  D.1. A Typical SMTP Transaction Scenario 105
  D.2. Aborted SMTP Transaction Scenario 106
  D.3. Relayed Mail Scenario 106
  D.4. Verifying and Sending Scenario 108
Appendix E. Other Gateway Issues 109
Appendix F. Deprecated Features of RFC 821 109
  F.1. TURN 110
  F.2. Source Routing 110
  F.3. HELO 111

F.4. #-literals 111
F.5. Dates and Years 112
F.6. Sending versus Mailing 112
Appendix G. Other Outstanding Issues 112
G.1. IP Address literals (closed) 113
G.2. Repeated Use of EHLO (closed) 113
G.3. Meaning of "MTA" and Related Terminology (closed) 113
G.4. Originator, or Originating System, Authentication (closed) 113
G.5. Remove or deprecate the work-around from code 552 to 452 (closed) 113
G.6. Clarify where the protocol stands with respect to submission and TLS issues (Closed) 113
G.7. Probably-substantive Discussion Topics Identified in Other Ways (closed or OBE) 114
G.7.1. Issues with 521, 554, and 556 codes (closed) 114
G.7.2. SMTP Model, terminology, and relationship to RFC 5598 (closed) 114
G.7.3. Resolvable FQDNs and private domain names (closed) . 114
G.7.4. Possible clarification about mail transactions and transaction state (closed) 114
G.7.5. Issues with mailing lists, aliases, and forwarding (closed) 114
G.7.6. Requirements for domain name and/or IP address in EHLO (closed) 114
G.7.7. Does the 'first digit only' and/or non-listed reply (c code text need clarification)? (closed) 114
G.7.8. Size limits (closed) 114
G.7.9. Discussion of 'blind' copies and RCPT (closed) . . . 115
G.7.10. Further clarifications needed to source routes? (closed) 115
G.7.11. Should 1yz Be Revisited? (closed) 115
G.7.12. Review Timeout Specifications (closed) 115
G.7.13. Possible SEND, SAML, SOML Loose End (closed) 115
G.7.14. Abstract Update (closed) 115
G.7.15. Informative References to MIME and/or Message Submission (closed) 115
G.7.16. Mail Transaction Discussion (closed) 115
G.7.17. Hop by hop Authentication and/or Encryption (closed) 115
G.7.18. More Text About 554 Given 521, etc. (closed) 115
G.7.19. Minimum Lengths and Quantities (closed) 115
G.8. Enhanced Reply Codes and DSNs (closed) 116
G.9. Revisiting Quoted Strings (closed) 116
G.10. Internationalization (closed) 116
G.11. SMTP Clients, Servers, Senders, and Receivers (closed) . 116
G.12. Extension Keywords Starting in 'X-' (closed) 116
G.13. Deprecating HELO (closed) 116

G.14. The FOR Clause in Trace Fields: Semantics, Security Considerations, and Other Issues (closed) 116
G.15. Resistance to Attacks and Operational Necessity (closed) 116
G.16. Mandatory 8BITMIME (closed) 116
G.17. New tickets created between 2022-01-21 and 2022-03-01 (closed (all of them)) 117
G.18. Approval Required to Register an SMTP Service Extension with IANA (closed) 117
G.19. Inconsistencies between rfc5321bis and IANA registry and related issues (closed) 117
G.20. Side-effects of approval change for Service Extension from "Standards Track or IESG Approved Experimental" to "Specification Required" (closed) 118
G.21. Appendix B and Message Submission (closed) 118
G.22. IANA Registration Model for Registries Other Than Service Extensions (Closed) 118
G.23. Headers Inserted in Mail Transport (closed) 118
G.24. Describing the "Operational Requirements" Loopholes (tentatively closed) 118
G.25. Relocate paragraphs after the first out of Section 3.6.2 and to 3.6.1 (tentatively closed) 118
G.26. Remove SHOULD requirement for use of reply text 118
G.27. Section 4.2.4.3 appears to impose a BCP 14 requirement on users 119
G.28. Misplaced paragraph about special treatment in Section 4.4.3 119
Appendix H. Completed Items Moved from Appendix G 119
H.1. IP Address literals 120
H.2. Repeated Use of EHLO 120
H.3. Meaning of "MTA" and Related Terminology 120
H.4. Originator, or Originating System, Authentication (to A/S) 121
H.5. Possible clarification about mail transactions and transaction state 121
H.6. Remove or deprecate the work-around from code 552 to 452 121
H.7. Resolvable FQDNs and private domain names 121
H.8. Issues with 521, 554, and 556 codes 121
H.9. Requirements for domain name and/or IP address in EHLO (mostly closed, some to A/S) 122
H.10. Does the 'first digit only' and/or non-listed reply code text need clarification? 122
H.11. Size limits 122
H.12. Further clarifications needed to source routes? 122
H.13. Should 1yz Be Revisited? 122
H.14. Review Timeout Specifications 123
H.15. Possible SEND, SAML, SOML Loose End 123

H.16. Abstract Update 123
H.17. Informative References to MIME and/or Message Submission 123
H.18. Hop by hop Authentication and/or Encryption 123
H.19. Enhanced Reply Codes and DSNs 123
H.20. SMTP Clients, Servers, Senders, and Receivers 124
H.21. Extension Keywords Starting in 'X-' 124
H.22. Deprecating HELO 124
H.23. Resistance to Attacks and Operational Necessity 125
H.24. New tickets created between 2022-01-21 and 2022-03-01 and subsequently closed. 125
H.25. Mandatory 8BITMIME 125
H.26. Discussion of 'blind' copies and RCPT 126
H.27. Revisiting Quoted Strings (reclassified) 126
H.28. Clarify where the protocol stands with respect to submission and TLS issues (reclassified) 126
H.29. SMTP Model, terminology, and relationship to RFC 5598 . . 127
H.30. More Text About 554 Given 521, etc. 127
H.31. Minimum Lengths and Quantities 127
H.32. Internationalization 127
H.33. Inconsistencies between rfc5321bis and IANA registry and related issues 128
H.34. Approval Required to Register an SMTP Service Extension with IANA 129
H.35. Side-effects of approval change for Service Extension from "Standards Track or IESG Approved Experimental" to "Specification Required" 129
H.36. Issues with mailing lists, aliases, and forwarding . . . 129
H.37. IANA Registration Model for Registries Other Than Service Extensions 130
H.38. Appendix B and Message Submission 130
H.39. The FOR Clause in Trace Fields: Semantics, Security Considerations, and Other Issues 130
H.40. Headers Inserted in Mail Transport 131
H.41. Describing the "Operational Requirements" Loopholes . . . 132
H.42. Relocate paragraphs after the first out of Section 3.6.2 and to 3.6.1 (closed) 132
Appendix I. Detailed Change Descriptions and Logs 132
I.1. RFC 5321 Errata Summary 132
I.2. Changes from RFC 5321 (published October 2008) to the initial (-00) version of this draft 134
I.3. Changes Among Versions of rfc5321bis 135
I.3.1. Changes from draft-klensin-rfc5321bis-00 (posted 2012-12-02) to -01 135
I.3.2. Changes from draft-klensin-rfc5321bis-01 (20191203) to -02 135
I.3.3. Changes from draft-klensin-rfc5321bis-02 (2019-12-27) to -03 135

I.3.4. Changes from draft-klensin-rfc5321bis-03 (2020-07-02) to draft-ietf-emailcore-rfc5321bis-00 136
I.3.5. Changes from draft-ietf-emailcore-rfc5321bis-00 (2020-10-06) to -01 136
I.3.6. Changes from draft-ietf-emailcore-rfc5321bis-01 (2020-12-25) to -02 137
I.3.7. Changes from draft-ietf-emailcore-rfc5321bis-02 (2021-02-21) to -03 137
I.3.8. Changes from draft-ietf-emailcore-rfc5321bis-03 (2021-07-10) to -04 138
I.3.9. Changes from draft-ietf-emailcore-rfc5321bis-04 (2021-10-03) to -05 139
I.3.10. Changes from draft-ietf-emailcore-rfc5321bis-05 (2021-10-24) to -06 139
I.3.11. Changes from draft-ietf-emailcore-rfc5321bis-06 (2021-11-07) to -07 140
I.3.12. Changes from draft-ietf-emailcore-rfc5321bis-07 (2021-12-04) to -08 141
I.3.13. Changes from draft-ietf-emailcore-rfc5321bis-08 (2021-12-31) to -09 142
I.3.14. Changes from draft-ietf-emailcore-rfc5321bis-09 (2022-02-01 to -10 143
I.3.15. Changes from draft-ietf-emailcore-rfc5321bis-10 (2022-03-07) to -11 143
I.3.16. Changes from draft-ietf-emailcore-rfc5321bis-11 (2022-05-24) to -12 145
I.3.17. Changes from draft-ietf-emailcore-rfc5321bis-12 (2022-07-09) to -13 145
I.3.18. Changes from draft-ietf-emailcore-rfc5321bis-13 (2022-09-13) to -14 146
I.3.19. Changes from draft-ietf-emailcore-rfc5321bis-14 (2022-10-23) to -15 146
I.3.20. Changes from draft-ietf-emailcore-rfc5321bis-15 (2022-11-06) to -16 147
I.3.21. Changes from draft-ietf-emailcore-rfc5321bis-16 (2022-12-06) to -17 147
I.3.22. Changes from draft-ietf-emailcore-rfc5321bis-17 (2022-12-31) to -18 148
I.3.23. Changes from draft-ietf-emailcore-rfc5321bis-18 (2023-02-07) to -19 148
I.3.24. Changes from draft-ietf-emailcore-rfc5321bis-19 (2023-07-23) to -20 149
I.3.25. Changes from draft-ietf-emailcore-rfc5321bis-20 (2023-11-16) to -21 149
I.3.26. Changes from draft-ietf-emailcore-rfc5321bis-21 (2023-11-27) to -22 150
I.3.27. Changes from draft-ietf-emailcore-rfc5321bis-22 (2023-12-05) to -23 150

I.3.28. Changes from draft-ietf-emailcore-rfc5321bis-23
(2023-12-10) to -24 150
I.3.29. Changes from draft-ietf-emailcore-rfc5321bis-24
(2024-01-11) to -25 150
I.3.30. Changes from draft-ietf-emailcore-rfc5321bis-25
(2024-01-24) to -26 151
I.3.31. Changes from draft-ietf-emailcore-rfc5321bis-26
(2024-02-09) to -27 151
I.4. Summary of changes from RFC 5321 (published in October
2008) to <<This Document>> 152
Index 153
Author's Address 157

1. Introduction

1.1. Transport of Electronic Mail

The objective of the Simple Mail Transfer Protocol (SMTP) is to transfer mail reliably and efficiently.

SMTP is independent of the particular transmission subsystem and requires only a reliable ordered data stream channel. While this document specifically discusses transport over TCP, other transports are possible. Appendices to RFC 821 [6] describe some of them.

An important feature of SMTP is its capability to transport mail across multiple networks, usually referred to as "SMTP mail relaying" (see Section 3.6). A network consists of the mutually-TCP-accessible hosts on the public Internet, the mutually-TCP-accessible hosts on a firewall-isolated TCP/IP Intranet [37] or hosts in some other LAN or WAN environment utilizing a non-TCP transport-level protocol. Using SMTP, a process can transfer mail to another process on the same network or to some other network via a relay or gateway process accessible to both networks.

In this way, a mail message may pass through a number of intermediate relay or gateway hosts on its path from sender to ultimate recipient. The Mail eXchanger mechanisms of the domain name system (RFC 1035 [7], RFC 974 [20], and Section 5 of this document) are used to identify the appropriate next-hop destination for a message being transported.

1.2. History and Context for This Document

This Internet Standard specification contains material, in many cases including copied exact text, from several documents including some dating back to RFC 821 [6], published over forty years ago. While most of the early features are unchanged, others have been updated or enhanced. This section summarizes the relationship of the present specification to earlier ones leading up to the very similar RFC 5321 [54] of October 2008. Changes between RFC 5321 and <<This Document>> appear in Appendix I.4. This document provides the specification of the basic protocol for Internet electronic mail transport. It consolidates, updates and clarifies, but does not add new or change existing functionality of the following:

* the original SMTP (Simple Mail Transfer Protocol) specification of RFC 821 [6],

* domain name system requirements and implications for mail transport from RFC 1035 [7] and RFC 974 [20],

* the clarifications and applicability statements in RFC 1123 [10],

* the new error codes added by RFC 1846 [25] and later by RFC 7504 [52], obsoleting both of those documents, and

* material drawn from the SMTP Extension mechanisms in RFC 1869 [27].

It also includes editorial, clarification, and correction changes that were made to RFC 2821 [35] to bring that specification to Draft Standard and similar changes to RFC 5321 [54] to bring the current document to Internet Standard as well as changes to the description and specification of IANA registries to align with contemporary practice and thinking.

It may help the reader to understand that, to reduce the risk of introducing errors, large parts of the document essentially merge the earlier specifications listed in the bullet points above rather than providing a completely rewritten, reorganized, and integrated description of SMTP. That strategy and the consequent document organization, had IETF consensus at the time RFC 2821 was written. An index and additional cross-references are provided to assist in the quest for information.

It obsoletes RFCs 5321 [54] (the earlier version of this specification), 1846 [25] and incorporates the substance of 7504 [52] (specification of reply codes), and 7505 [53] (the "Null MX" specification). Although SMTP was designed as a mail transport and

delivery protocol, this specification also contains information that is relevant to its optional use for submission of mail by users and to some aspects of the Post Office Protocol (POP) (RFC 937 [18], RFC 1939 [28]) and IMAP (RFC 9051 [41]) protocols. In general, the separate mail submission protocol specified in RFC 6409 [48] is now preferred to direct use of SMTP for that function; more discussion of that subject appears in that document.

Section 2.3 provides definitions of terms specific to this document. Except when the historical terminology is necessary for clarity, this document uses the current 'client' and 'server' terminology to identify the sending and receiving SMTP processes, respectively. In general, "sender-SMTP" and "SMTP client" are equivalent as are "receiver-SMTP" and "SMTP server".

A companion document, rfc5322bis [16], discusses message header sections and bodies and specifies formats and structures for them. Other relevant documents and their relationships are discussed in a forthcoming Applicability Statement [55].

1.3. Document Conventions

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 [1] and RFC 8174 [2]. As each of these terms was intentionally and carefully chosen to improve the interoperability of email, each use of these terms is to be treated as a conformance requirement.

This document has a long history. To avoid the risk of various errors and of confusing readers and documents that point to this one, most examples and the domain names they contain are preserved from RFC 2821. Readers are cautioned that these are illustrative examples that should not actually be used in either code or configuration files.

2. The SMTP Model

2.1. Basic Structure

The SMTP design can be pictured as:

```
                   +----------+            +----------+
      +------+     |          |            |          |    +------+
      | User |<--> |          |   SMTP     |          |<-->| File |
      +------+     |  Client- |Commands/   | Server-  |    |System|
      +------+     |   SMTP   |  Replies   |    SMTP  |    +------+
      | File |<--> |          |<---------->|          | or
      |System|     |          |  and Mail  |          |    +------+
      +------+     |          |            |          |    |Next- |
                   |          |            |          |    |   hop|
      +-------+    |          |            |          |<-->|SMTP  |
      |Submis-|<-->|          |            |          |    |client|
      |   sion|    +----------+            +----------+    +------+
      |System |
      +-------+     SMTP client            SMTP server
                     (sender)               (receiver)
```

When an SMTP client has a message to transmit, it establishes a two-way transmission channel to an SMTP server. The responsibility of an SMTP client is to transfer mail messages to one or more SMTP servers, or report its failure to do so.

The means by which a mail message is presented to an SMTP client (i.e., between the first two columns above), and how that client determines the identifier(s) ("names") of the domain(s) to which mail messages are to be transferred, are local matters. They are not addressed by this document. In some cases, the designated domain(s), or those determined by an SMTP client, will identify the final destination(s) of the mail message. In other cases, common with SMTP clients associated with implementations of the POP (RFC 937 [18], RFC 1939 [28]) or IMAP (RFC 9051 [41]) protocols, or when the SMTP client is inside an isolated transport service environment, the domain determined will identify an intermediate destination through which all mail messages are to be relayed. SMTP clients that transfer all traffic regardless of the target domains associated with the individual messages, or that do not maintain queues for retrying message transmissions that initially cannot be completed, may otherwise conform to this specification but are not considered fully-capable. Fully-capable SMTP implementations, including the relays used by these less capable ones, and their destinations, are expected to support all of the queuing, retrying, and alternate address functions discussed in this specification. In many situations and configurations, the less-capable clients discussed above SHOULD be using the message submission protocol (RFC 6409 [48]) rather than SMTP.

The means by which an SMTP client, once it has determined a target domain, determines the identity of an SMTP server to which a copy of a message is to be transferred, and then performs that transfer, are

covered by this document. To effect a mail transfer to an SMTP server, an SMTP client establishes a two-way transmission channel to that SMTP server. An SMTP client determines the address of an appropriate host running an SMTP server by resolving a destination domain name to either an intermediate Mail eXchanger host or a final target host.

An SMTP server may be either the ultimate destination or an intermediate "relay" (that is, it may assume the role of an SMTP client after receiving the message) or "gateway" (that is, it may transport the message further using some protocol other than SMTP). SMTP commands are generated by the SMTP client and sent to the SMTP server. SMTP replies are sent from the SMTP server to the SMTP client in response to the commands.

In other words, message transfer can occur in a single connection between the original SMTP-sender and the final SMTP-recipient, or can occur in a series of hops through intermediary systems. In either case, once the server has issued a success response at the end of the mail data, a formal handoff of responsibility for the message occurs: the protocol requires that a server MUST accept responsibility for either delivering the message or properly reporting the failure to do so (see Sections 6.1, 6.2, and 7.8, below).

Once the transmission channel is established and initial handshaking is completed, the SMTP client normally initiates a mail transaction. Such a transaction consists of a series of commands to specify the originator and destination of the mail and transmission of the message content (including any lines in the header section or other structure) itself. When the same message is sent to multiple recipients, this protocol encourages the transmission of only one copy of the data for all recipients at the same destination (or intermediate relay) host.

The server responds to each command with a reply; replies may indicate that the command was accepted, that additional commands are expected, or that a temporary or permanent error condition exists. Commands specifying the sender or recipients may include server-permitted SMTP service extension requests, as discussed in Section 2.2. The dialog is purposely lock-step, one-at-a-time, although this can be modified by mutually agreed upon extension requests such as command pipelining (RFC 2920 [36]).

Once a given mail message has been transmitted, the client may either request that the connection be shut down or may initiate other mail transactions. In addition, an SMTP client may use a connection to an SMTP server for ancillary services such as verification of email addresses or retrieval of mailing list subscriber addresses.

As suggested above, this protocol provides mechanisms for the transmission of mail. Historically, this transmission normally occurred directly from the sending user's host to the receiving user's host when the two hosts are connected to the same transport service. When they are not connected to the same transport service or other circumstances dictate, transmission occurs via one or more relay SMTP servers. A very common case in the Internet today involves submission of the original message to an intermediate, "message submission" server, which is similar to a relay but has some additional properties; such servers are discussed in Section 2.3.10 and at some length in RFC 6409 [48]. An intermediate host that acts as either an SMTP relay or as a gateway into some other transmission environment is usually selected through the use of the domain name service (DNS) Mail eXchanger mechanism.

2.2. The Extension Model

2.2.1. Background

In an effort that started in 1990, approximately a decade after RFC 821 was completed, the protocol was modified with a "service extensions" model that permits the client and server to agree to utilize shared functionality beyond the original SMTP requirements [24]. The SMTP extension mechanism defines a means whereby an extended SMTP client and server may recognize each other, and the server can inform the client as to the service extensions that it supports.

Contemporary SMTP implementations MUST support the basic extension mechanisms. For instance, servers MUST support the EHLO command even if they do not implement any specific extensions and clients SHOULD preferentially utilize EHLO rather than HELO. (However, for compatibility with older conforming implementations, SMTP clients and servers MUST support the original HELO mechanisms as a fallback.) Unless the different characteristics of HELO must be identified for interoperability purposes, this document discusses only EHLO.

SMTP is widely deployed and high-quality implementations have proven to be very robust. However, the Internet community now considers some services to be important that were not anticipated when the protocol was first designed. If support for those services is to be added, it must be done in a way that permits older implementations to continue working acceptably. The extension framework consists of:

* The SMTP command EHLO, superseding the earlier HELO,

* a registry of SMTP service extensions,

* additional parameters to the SMTP MAIL and RCPT commands, and

* optional replacements for commands defined in this protocol, such as for DATA in non-ASCII transmissions (RFC 3030 [38]).

SMTP's strength comes primarily from its simplicity. Experience with many protocols has shown that protocols with few options tend towards ubiquity, whereas protocols with many options tend towards obscurity.

As part of deciding whether to implement and support an extension, regardless of its benefits, each SMTP implementation, each extension must be carefully scrutinized with respect to its implementation, deployment, and interoperability costs. In many cases, the cost of extending the SMTP service will likely outweigh the benefit.

2.2.2. Definition and Registration of Extensions

Especially for extensions intended for general use and expected to interoperate well with multiple implementations, a readily-available, stable, and adequate definition is essential for those evaluating, implementing, or configuring the extension. The information below describes important characteristics of that documentation. In order to make it accessible and to prevent naming conflicts, the IANA maintains a registry of SMTP service extensions [64] and each service extension must be recorded in that registry as specified in Section 8.1.1 and below.

Experience has shown that obtaining broad review and input from the broader community produces much better results than narrower discussions, e.g., only among the designers. While it is usually best to obtain that input prior to registration and to do so formally as part of an IETF Standards Track specification, there is no requirement to do so. An alternate, simplified, registration procedure (see Section 8.1.1.1, Paragraph 6, Item 2) allows extensions to be written and registered that permit modifications after registration, perhaps even after deployment experience. Even when that simplified procedure is used, and although it is not required, it will often be useful for the submitter or IANA to notify a relevant IETF mailing list of the extension request. Registrants may also reach out to selected individuals for advice on the specification and how to best obtain additional useful input. Details of the registration process itself and two available registration models appear in Section 8.1.1 below.

For standards track registrations, the definition and related description of the extension will include, not only the keyword name and syntax for the service extension and other information required for registration, but a detailed description of the purpose of the

extension, what it is expected to accomplish, how its use changes the behavior of client and server SMTP implementations that use it, and how it interacts with other relevant extensions and elements of this specification. To be of maximum use, the alternative procedure should still make that information available.

Any keyword value presented in the EHLO response MUST correspond to an SMTP service extension registered with IANA as described in Section 8.1. A conforming server MUST NOT offer keyword values that are not described in a registered extension.

SMTP Clients MUST ignore any announced extension they do not recognize and, if the announcement involves parsing or other problems that prevent reliable interpretation of the response to the EHLO command, send QUIT and terminate the SMTP session.

2.2.3. Special Issues with Extensions

Extensions that change fairly basic properties of SMTP operation are permitted. The text in other sections of this document must be understood in that context. In particular, extensions can change the minimum limits specified in Section 4.5.3, can change the ASCII character set requirement as mentioned above, or can introduce some optional modes of message handling.

In particular, if an extension implies that the delivery path normally supports special features of that extension, and an intermediate SMTP system finds a next hop that does not support the required extension, it MAY choose, based on the specific extension and circumstances, to requeue the message and try later and/or try an alternate MX host. If this strategy is employed, the timeout to fall back to an unextended format (if one is available) SHOULD be less than the normal timeout for bouncing as undeliverable (e.g., if normal timeout is three days, the requeue timeout before attempting to transmit the mail without the extension might be one day).

2.3. SMTP Terminology

2.3.1. Mail Objects

SMTP transports a mail object. A mail object contains an envelope and content.

The SMTP envelope is sent as a series of SMTP protocol units (described in Section 3). It consists of an originator address (to which error reports should be directed), one or more recipient addresses, and optional protocol extension material. Historically, variations on the reverse-path (originator) address specification

command (MAIL) could be used to specify alternate delivery modes, such as immediate display; those variations have now been deprecated (see Appendix F and Appendix F.6).

The SMTP content is sent in the SMTP DATA protocol unit and has two parts: the header section and the body. If the content conforms to other contemporary standards, the header section consists of a collection of header fields, each consisting of a header name, a colon, and data, structured as in the message format specification (RFC5322bis [16]); the body, if structured, is defined according to MIME (RFC 2045 [30]). The content is textual in nature, expressed using the US-ASCII repertoire [4]. SMTP extensions (such as "8BITMIME", RFC 6152 [51]) may relax this restriction for the content body and content header fields. Two MIME extensions (RFC 2047 [31] and RFC 2231 [34]) define an algorithm for representing header values outside the US-ASCII repertoire, while still encoding them using that repertoire.

2.3.2. Senders and Receivers

In RFC 821, the two hosts participating in an SMTP transaction were described as the "SMTP-sender" and "SMTP-receiver". This document has been changed to reflect current industry terminology and hence refers to them as the "SMTP client" (or sometimes just "the client") and "SMTP server" (or just "the server"), respectively. Since a given host may act both as server and client in a relay situation, "receiver" and "sender" terminology is still used where needed for clarity.

2.3.3. Mail Agents and Message Stores

Additional mail system terminology became common after RFC 821 was published and, where convenient, is used in this specification. In particular, SMTP servers and clients provide a mail transport service and therefore act as "Mail Transfer Agents" (MTAs). "Mail User Agents" (MUAs or UAs) are normally thought of as the sources and targets of mail. At the source, an MUA might collect mail to be transmitted from a user and hand it off to an MTA or, more commonly in recent years, a specialized variation on an MTA called a "Submission Server" (MSA) [48]. At the other end of the process, the final ("delivery") MTA would be thought of as handing the mail off to an MUA (or at least transferring responsibility to it, e.g., by depositing the message in a "message store"). However, while these terms are used with at least the appearance of great precision in other environments, the implied boundaries between MUAs and MTAs often do not accurately match common, and conforming, practices with Internet mail. Hence, the reader should be cautious about inferring the strong relationships and responsibilities that might be implied

if these terms were used elsewhere.

2.3.4. Host

For the purposes of this specification, a host is a computer system attached to the Internet (or, in some cases, to a private TCP/IP network) and supporting the SMTP protocol. Hosts are known by names (see the next section); they SHOULD NOT be identified by numerical addresses, i.e., by address literals as described in Section 4.1.2.

2.3.5. Domain Names

A domain name (or often just a "domain") consists of one or more components, separated by dots if more than one appears. In the case of a top-level domain used by itself in an email address, a single string is used without any dots. This makes the requirement, described in more detail below, that only fully-qualified domain names appear in SMTP transactions on the public Internet, particularly important where top-level domains are involved. These components ("labels" in the DNS terminology of RFC 1035 [7]) are restricted for purposes of SMTP as defined here to consist of a sequence of letters, digits, and hyphens drawn from the ASCII character set [4] and conforming to what RFC 1035 calls the "preferred name syntax", with the exception that leading digits in labels are permitted [8]. Domain names are used as names of hosts and, except where additionally restricted in this document, of other entities in the domain name hierarchy. For example, a domain may refer to a host alias (label of a CNAME RR) or the label of Mail eXchanger records to be used to deliver mail instead of representing a host name. See RFC 1035 and Section 5 of this specification.

The domain name, as described in this document and in RFC 1035 [7], MUST be the entire, fully-qualified name (often referred to as an "FQDN"). Other than an address literal (see Section 4.1.3) where those are permitted, any string that is not a domain name in FQDN form is no more than a reference to be interpreted locally. Such local references for domain names MUST NOT appear in any SMTP transaction (Cf. Section 5). Mechanisms for inferring FQDNs from local references (including partial names or local aliases) are outside of this specification and normally the province of message submission. Due to a history of problems, SMTP servers SHOULD NOT make such inferences (Message Submission Servers [48] have somewhat more flexibility) and intermediate (relay) SMTP servers MUST NOT make them.

When domain names are used in SMTP, and unless further restricted in this document, names that can be resolved to MX RRs or address (i.e., A or AAAA) RRs (as discussed in Section 5) are permitted, as are CNAME RRs whose targets can be resolved, in turn, to MX or address RRs. There are two exceptions to the rule requiring FQDNs:

* The domain name given in the EHLO command MUST be either a primary host name (a domain name that resolves to an address RR) or, if the host has no name, an address literal, as described in Section 4.1.3 and discussed further in the EHLO discussion of Section 4.1.4.

* The reserved mailbox name "postmaster" MAY be used in a RCPT command without domain qualification (see Section 4.1.1.3) and MUST be accepted if so used.

2.3.6. Buffer and State Table

SMTP sessions are stateful, with both parties carefully maintaining a common view of the current state. In this document, we model this state by a virtual "buffer" and a "state table" on the server that may be used by the client to, for example, "clear the buffer" or "reset the state table", causing the information in the buffer to be discarded and the state to be returned to some previous state.

2.3.7. Commands and Replies

SMTP commands and, unless altered by a service extension, message data, are transmitted from the sender to the receiver via the transmission channel in "lines" (defined in Section 2.3.8 below).

An SMTP reply is an acknowledgment (positive or negative) sent in "lines" from receiver to sender via the transmission channel in response to a command. The general form of a reply is a numeric completion code (indicating failure or success) usually followed by a text string. The codes are for use by programs and the text is usually intended for human users. RFC 3463 [12], specifies further structuring of the reply strings, including the use of supplemental and more specific completion codes (see also RFC 5248 [50]).

2.3.8. Lines

Lines consist of zero or more data characters terminated by the sequence ASCII character "CR" (hex value 0D) followed immediately by ASCII character "LF" (hex value 0A). This termination sequence is denoted as <CRLF> in this document. Conforming implementations MUST NOT recognize or generate any other character or character sequence as a line terminator. Limits MAY be imposed on line lengths by

servers (see Section 4).

In addition, the appearance of "bare" "CR" or "LF" characters in text (i.e., either without the other) has a long history of causing problems in mail implementations and applications that use the mail system as a tool. Unless negotiated otherwise using an SMTP extension, SMTP client implementations MUST NOT transmit these characters except when they are intended as line terminators and then MUST, as indicated above, transmit them only as a <CRLF> sequence.

2.3.9. Message Content and Mail Data

The terms "message content" and "mail data" are used interchangeably in this document to describe the material transmitted after the DATA command is accepted and before the end of data indication is transmitted. Message content includes the message header section and the possibly structured message body. In the absence of extensions, both are required to be ASCII (see Section 2.3.1). The MIME specification (RFC 2045 [30]) provides the standard mechanisms for structured message bodies.

2.3.10. Originator, Delivery, Relay, and Gateway Systems

This specification makes a distinction among four types of SMTP systems, based on the role those systems play in transmitting electronic mail. An "originating" system (sometimes called an SMTP originator) introduces mail into the Internet or, more generally, into a transport service environment. A "delivery" SMTP system is one that receives mail from a transport service environment and passes it to a mail user agent or deposits it in a message store that a mail user agent is expected to subsequently access. A "relay" SMTP system (usually referred to just as a "relay") receives mail from an SMTP client and transmits it, without modification to the message data other than adding trace information (see Section 4.4), to another SMTP server for further relaying or for delivery.

A "gateway" SMTP system (usually referred to just as a "gateway") receives mail from a client system in one transport environment and transmits it to a server system in another transport environment. Differences in protocols or message semantics between the transport environments on either side of a gateway may require that the gateway system perform transformations to the message that are not permitted to SMTP relay systems. For the purposes of this specification, firewalls that rewrite addresses should be considered as gateways, even if SMTP is used on both sides of them (see RFC 2979 [37]).

2.3.11. Mailbox and Address

As used in this specification, an "address" is a character string that identifies a user to whom mail will be sent or a location into which mail will be deposited. The term "mailbox" refers to that depository. The two terms are typically used interchangeably unless the distinction between the location in which mail is placed (the mailbox) and a reference to it (the address) is important. An address normally consists of user and domain specifications. The standard mailbox naming convention is defined to be "local-part@domain"; contemporary usage permits a much broader set of applications than simple "user names". Consequently, and due to a long history of problems when intermediate hosts have attempted to optimize transport by modifying them, the local-part MUST be interpreted and assigned semantics only by the host specified in the domain part of the address.

2.3.12. Sessions and Transactions

This document distinguishes between an "SMTP session" (interchangeable with "mail session") and starting when a connection is made between client and server and a "mail transaction", which is started and terminated by particular commands. For more information and details, see Section 3.1 and Section 3.3.

2.4. General Syntax Principles and Transaction Model

SMTP commands and replies have a rigid syntax. All commands begin with a command verb. All replies begin with a three digit numeric code. In some commands and replies, arguments are required following the verb or reply code. Some commands do not accept arguments (after the verb), and some reply codes are followed, sometimes optionally, by free form text. In both cases, where text appears, it is separated from the verb or reply code by a space character. Complete definitions of commands and replies appear in Section 4.

Verbs and argument values (e.g., "TO:" or "to:" in the RCPT command and extension name keywords) are not case sensitive, with the sole exception in this specification of a mailbox local-part (SMTP Extensions may explicitly specify case-sensitive elements). That is, a command verb, an argument value other than a mailbox local-part, and free form text MAY be encoded in upper case, lower case, or any mixture of upper and lower case with no impact on its meaning. The local-part of a mailbox MUST BE treated as case sensitive. Therefore, SMTP implementations MUST take care to preserve the case of mailbox local-parts. In particular, for some hosts, the user "smith" is different from the user "Smith". However, exploiting the case sensitivity of mailbox local-parts impedes interoperability and is discouraged. Mailbox domains follow normal DNS rules and are hence not case sensitive.

A few SMTP servers, in violation of this specification (and RFC 821) may require that command verbs be encoded by clients in upper case. Implementations MAY wish to employ this encoding to accommodate those servers.

The argument clause consists of a variable-length character string ending with the end of the line, i.e., with the character sequence <CRLF>. The receiver will take no action until this sequence is received.

The syntax for each command is shown with the discussion of that command. Common elements and parameters are shown in Section 4.1.2.

Commands and replies are composed of characters from the ASCII character set [4]. When the transport service provides an 8-bit byte (octet) transmission channel, each 7-bit character is transmitted, right justified, in an octet with the high-order bit cleared to zero. More specifically, the unextended SMTP service provides 7-bit transport only. An originating SMTP client that has not successfully negotiated an appropriate extension with a particular server (see the next paragraph) MUST NOT transmit messages with information in the high-order bit of octets. If such messages are transmitted in violation of this rule, receiving SMTP servers MAY clear the high-order bit or reject the message as invalid. In general, a relay SMTP SHOULD assume that the message content it has received is valid and, assuming that the envelope permits doing so, relay it without inspecting that content. Of course, if the content is mislabeled and the data path cannot accept the actual content, this may result in the ultimate delivery of a severely garbled message to the recipient. Delivery SMTP systems MAY reject such messages, or return them as undeliverable, rather than deliver them. In the absence of a server-offered extension explicitly permitting it, a sending SMTP system is not permitted to send envelope commands in any character set other than US-ASCII. Receiving systems SHOULD reject such commands, normally using "500 syntax error - invalid character" replies.

8-bit message content transmission MAY be requested of the server by a client using extended SMTP facilities, notably the "8BITMIME" extension, RFC 6152 [51]. 8BITMIME SHOULD be supported by SMTP servers. However, it MUST NOT be construed as authorization to transmit unrestricted 8-bit material, nor does 8BITMIME authorize transmission of any envelope material encoded in anything other than US-ASCII. 8BITMIME MUST NOT be requested by senders for material with the high bit on that is not in MIME format with an appropriate content-transfer encoding; servers MAY reject such messages.

The metalinguistic notation used in this document corresponds to the "Augmented BNF" used in other Internet mail system documents. The reader who is not familiar with that syntax should consult the ABNF specification in RFC 5234 [15]. Metalanguage terms used in running text are surrounded by pointed brackets (e.g., <CRLF>) for clarity. The reader is cautioned that the grammar expressed in the metalanguage is not comprehensive. There are many instances in which provisions in the text constrain or otherwise modify the syntax or semantics implied by the grammar.

3. The SMTP Procedures: An Overview

This section contains descriptions of the procedures used in SMTP: session initiation, mail transaction, forwarding mail, verifying mailbox names and expanding mailing lists, and opening and closing exchanges. Comments on relaying, a note on mail domains, and a discussion of changing roles are included at the end of this section. Several complete scenarios are presented in Appendix D.

3.1. Session Initiation

An SMTP session (or "mail session") is initiated when a client opens a connection to a server and the server responds with an opening message.

SMTP server implementations MAY include identification of their software and version information in the connection greeting reply after the 220 code, a practice that permits more efficient isolation and repair of any problems. Implementations MAY make provision for SMTP servers to disable the software and version announcement where it causes security concerns. While some systems also identify their contact point for mail problems, this is not a substitute for maintaining the required "postmaster" address (see Section 4).

The SMTP protocol allows a server to formally reject a mail session while still allowing the initial connection as follows: a 521 response MAY be given in the initial connection opening message instead of the 220. A server taking this approach MUST still wait for the client to send a QUIT (see Section 4.1.1.10) before closing the connection and SHOULD respond to any intervening commands with "503 bad sequence of commands". Since an attempt to make an SMTP connection to such a system is probably in error, a server returning a 521 response on connection opening SHOULD provide enough information in the reply text to facilitate debugging of the sending system. See Section 4.2.4.2.

3.2. Client Initiation

Once the server has sent the greeting (welcoming) message and the client has received it, the client normally sends the EHLO command to the server, indicating the client's identity. In addition to opening the session, use of EHLO indicates that the client is able to process service extensions and requests that the server provide a list of the extensions it supports. Older SMTP systems that are unable to support service extensions, and contemporary clients that do not require service extensions in the mail session being initiated, MAY use HELO instead of EHLO. Servers MUST NOT return the extended EHLO-style response to a HELO command. For a particular connection

attempt, if the server returns a "command not recognized" response to EHLO, the client SHOULD be able to fall back and send HELO.

In the EHLO (or HELO) command, the host sending the command identifies itself; the command may be interpreted as saying "Hello, I am <domain>" (and, in the case of EHLO, "and I support service extension requests").

3.3. Mail Transactions

There are three steps to normal SMTP mail transactions. The transaction starts with a MAIL command that gives the sender identification. (In general, the MAIL command may be sent only when no mail transaction is in progress; see Section 4.1.4.) In a normal session, a series of one or more RCPT commands follows, giving the receiver information. Then, a DATA command initiates transfer of the mail data and is terminated by the "end of mail" data indicator, which also confirms (and terminates) the transaction.

Mail transactions are also terminated by the RSET command (Section 4.1.1.5), the sending of an EHLO command (Section 3.2), or the sending of a QUIT command (Section 3.8). The latter terminates not only any active mail transaction but the SMTP connection itself.

The first step in the procedure is the MAIL command.

```
MAIL FROM:<reverse-path> [SP <mail-parameters> ] <CRLF>
```

This command tells the SMTP-receiver that a new mail transaction is starting and to reset all its state tables and buffers, including any recipients or mail data. The <reverse-path> portion of the first or only argument contains the source mailbox (between "<" and ">" brackets), which can be used to report errors (see Section 4.2 for a discussion of error reporting). If accepted, the SMTP server returns a "250 OK" reply. If the mailbox specification is not acceptable for some reason, the server MUST return a reply indicating whether the failure is permanent (i.e., will occur again if the client tries to send the same address again) or temporary (i.e., the address might be accepted if the client tries again later). Despite the apparent scope of this requirement, there are circumstances in which the acceptability of the reverse-path may not be determined until one or more forward-paths (in RCPT commands) can be examined. In those cases, the server MAY reasonably accept the reverse-path (with a 250 reply) and then report problems after the forward-paths are received and examined. Normally, failures produce 550 or 553 replies.

Historically, the <reverse-path> was permitted to contain more than just a mailbox; however source routing is now deprecated (see Appendix F.2).

The optional <mail-parameters> are associated with negotiated SMTP service extensions (see Section 2.2).

The second step in the procedure is the RCPT command. This step of the procedure can be repeated any number of times.

```
RCPT TO:<forward-path> [ SP <rcpt-parameters> ] <CRLF>
```

The first or only argument to this command includes a forward-path (normally a mailbox local-part and domain, always surrounded by "<" and ">" brackets) identifying one recipient. If accepted, the SMTP server returns a "250 OK" reply and stores the forward-path. If the recipient is known not to be a deliverable address, the SMTP server returns a 550 reply, typically with a string such as "no such user - " and the mailbox name (other circumstances and reply codes are possible).

Historically, the <forward-path> was permitted to contain a source routing list of hosts and the destination mailbox; however, source routes are now deprecated (see Appendix F.2). Clients MUST NOT assume that any SMTP server on the Internet can be used as their mail processing (relaying) site. If a RCPT command appears without a previous MAIL command, the server MUST return a 503 "Bad sequence of commands" response. The optional <rcpt-parameters> are associated with negotiated SMTP service extensions (see Section 2.2).

There are two ways that sender-SMTPs can determine the next-hop system to which to send the message. One is to use the DNS and MX records as described in Section Section 5.1. The other involves a next-hop destination or choice of destinations that are configured into the sender-SMTP to deal with special circumstances such as forwarding all messages to a particular host for further processing.

Since it has been a common source of errors, it is worth noting that spaces are not permitted on either side of the colon following FROM in the MAIL command or TO in the RCPT command. The syntax is exactly as given above.

The third step in the procedure is the DATA command (or some alternative specified in a service extension).

```
DATA <CRLF>
```

If accepted, the SMTP server returns a 354 Intermediate reply and considers all succeeding lines up to but not including the end of mail data indicator to be the message text. When the end of text is successfully received and stored, the SMTP-receiver sends a "250 OK" reply.

Since the mail data is sent on the transmission channel, the end of mail data must be indicated so that the command and reply dialog can be resumed. An SMTP client indicates the end of the mail data by sending a line containing only a "." (period or full stop, hex 2E), that is the character sequence "<CRLF>.<CRLF>". A transparency procedure is used to prevent this from interfering with the user's text (see Section 4.5.2).

The end of mail data indicator also confirms the mail transaction and tells the SMTP server to now process the stored recipients and mail data. If accepted, the SMTP server returns a "250 OK" reply. The DATA command can fail at only two points in the protocol exchange:

If there was no MAIL, or no RCPT, command, or all such commands were rejected, the server MAY return a "command out of sequence" (503) or "no valid recipients" (554) reply in response to the DATA command. If one of those replies (or any other 5yz reply) is received, the client MUST NOT send the message data; more generally, message data MUST NOT be sent unless a 354 reply is received.

If the verb is initially accepted and the 354 reply issued, the DATA command should fail only if the mail transaction was incomplete (for example, no recipients), if resources were unavailable (including, of course, the server unexpectedly becoming unavailable), or if the server determines that the message should be rejected for policy or other reasons.

However, in practice, some servers do not perform recipient verification until after the message text is received. These servers SHOULD treat a failure for one or more recipients as a "subsequent failure" and return a mail message as discussed in Section 6 and, in particular, in Section 6.1. Using a "550 mailbox not found" (or equivalent) reply code after the data are accepted makes it difficult or impossible for the client to determine which recipients failed.

When the RFC 822 format ([17], [16]) is being used, the mail data include the header fields such as those named Date, Subject, To, Cc, and From. Server SMTP systems SHOULD NOT reject messages based on perceived defects in the RFC 822 or MIME (RFC 2045 [30]) message header section or message body. In particular, they MUST NOT reject messages in which the numbers of Resent-header fields do not match or Resent-to appears without Resent-from and/or Resent-date.

Mail transaction commands MUST be used in the order discussed above.

3.4. Address Modification and Expansion

3.4.1. Forwarding for Address Correction or Updating

Forwarding support is most often required to consolidate and simplify addresses within, or relative to, some enterprise and less frequently to establish addresses to link a person's prior address with a current one. Silent forwarding of messages (without server notification to the sender), for security or non-disclosure purposes, is common in the contemporary Internet.

In both the enterprise and the "new address" cases, information hiding (and sometimes security) considerations argue against exposure of the "final" address through the SMTP protocol as a side effect of the forwarding activity. This may be especially important when the final address may not even be reachable by the sender. Consequently, the "forwarding" mechanisms described in Section 3.2 of RFC 821, and especially the 251 (corrected destination) and 551 reply codes from RCPT must be evaluated carefully by implementers and, when they are available, by those configuring systems (see also Section 7.4).

In particular:

* Servers MAY forward messages when they are aware of an address change. When they do so, they MAY either provide address-updating information with a 251 code, or may forward "silently" and return a 250 code. However, if a 251 code is used, they MUST NOT assume that the client will actually update address information or even return that information to the user.

Alternately,

* Servers MAY reject messages or return them as non-deliverable when they cannot be delivered precisely as addressed. When they do so, they MAY either provide address-updating information with a 551 code, or may reject the message as undeliverable with a 550 code and no address-specific information. However, if a 551 code is used, they MUST NOT assume that the client will actually update address information or even return that information to the user.

SMTP server implementations that support the 251 and/or 551 reply codes SHOULD provide configuration mechanisms so that sites that conclude that they would undesirably disclose information can disable or restrict their use. See Section 7.4 for further discussion of that issue.

3.4.2. Aliases and Mailing Lists

Many SMTP-capable hosts support address expansion for multiple delivery via one or both of the alias and the list models. When a message is delivered or forwarded to each address of an expanded list form, the return address in the envelope ("MAIL FROM:") MUST be changed to be the address of a person or other entity who administers the list. This change to the MAIL command does not affect the header section of the message.

An important mail facility is a mechanism for multi-destination delivery of a single message, by transforming (or "expanding" or "exploding") a pseudo-mailbox address into a list of destination mailbox addresses. When a message is sent to such a pseudo-mailbox (sometimes called an "exploder"), copies are forwarded or redistributed to each mailbox in the expanded list. Servers SHOULD simply utilize the addresses on the list; application of heuristics or other matching rules to eliminate some addresses, such as that of the originator, is strongly discouraged. We classify such a pseudo-mailbox as an "alias" or a "list", depending upon the expansion rules.

3.4.2.1. Simple Aliases

To expand an alias, the recipient mailer simply replaces the pseudo-mailbox address in the envelope with each of the expanded addresses in turn; the rest of the envelope and the message body are left unchanged. The message is then delivered or forwarded to each expanded address.

3.4.2.2. Mailing Lists

Processing of a mailing list may be said to operate by "redistribution" rather than by "forwarding" (as in the simple alias case in the subsection above). To expand a list, the recipient mailer replaces the pseudo-mailbox address in the envelope with each of the expanded addresses in turn. The return (backward-pointing) address in the envelope is changed so that all error messages generated by the final deliveries will be returned to a list administrator, not to the message originator, who generally has no control over the contents of the list and will typically find error messages annoying. Note that the key difference between handling simple aliases Section 3.4.2.1 and redistribution (this subsection) is the change to the backward-pointing address. When a system managing a list constrains its processing to the very limited set of modifications and actions described here, it is acting as part of an MTA; such list processing, like alias processing, can be treated as a continuation of email transit.

Mailing list management systems do exist that perform additional, sometimes extensive, modifications to a message and its envelope. Such mailing lists need to be viewed as MUAs that accept a message delivery and then submit a new message for multiple recipients.

3.5. Commands for Debugging Addresses

3.5.1. Overview

SMTP provides commands to verify a user name or obtain the content of a mailing list. This is done with the VRFY and EXPN commands, which have character string arguments. Implementations SHOULD support VRFY and EXPN (however, see Section 3.5.2 and Section 7.3).

For the VRFY command, the string is a user name or a user name and domain (see below). If a normal (i.e., 250) response is returned, the response MAY include the full name of the user and MUST include the mailbox of the user. It MUST be in one of the following forms:

```
User Name <local-part@domain>
<local-part@domain>
local-part@domain
```

When a name that is the argument to VRFY could identify more than one mailbox, the server MAY either note the ambiguity or identify the alternatives. In other words, any of the following are legitimate responses to VRFY:

```
553 User ambiguous
```

or

```
553- Ambiguous; Possibilities are
553-Joe Smith <jsmith@foo.com>
553-Harry Smith <hsmith@foo.com>
553 Melvin Smith <dweep@foo.com>
```

or

```
553-Ambiguous; Possibilities
553- <jsmith@foo.com>
553- <hsmith@foo.com>
553 <dweep@foo.com>
```

Under normal circumstances, a client receiving a 553 reply would be expected to expose the result to the user. Use of exactly the forms given, and the "user ambiguous" or "ambiguous" keywords, possibly supplemented by extended reply codes, such as those described in RFC

3463 [12], will facilitate automated translation into other languages as needed. Of course, a client that was highly automated or that was operating in another language than English might choose to try to translate the response to return some other indication to the user than the literal text of the reply, or to take some automated action such as consulting a directory service for additional information before reporting to the user.

For the EXPN command, the string identifies a mailing list, and the successful (i.e., 250) multiline response MAY include the full name of the users and MUST give the mailboxes on the mailing list.

In some hosts, the distinction between a mailing list and an alias for a single mailbox is a bit fuzzy, since a common data structure may hold both types of entries, and it is possible to have mailing lists containing only one mailbox. If a request is made to apply VRFY to a mailing list, a positive response MAY be given if a message so addressed would be delivered to everyone on the list, otherwise an error SHOULD be reported (e.g., "550 That is a mailing list, not a user" or "252 Unable to verify members of mailing list"). If a request is made to expand a user name, the server MAY return a positive response consisting of a list containing one name, or an error MAY be reported (e.g., "550 That is a user name, not a mailing list").

In the case of a successful multiline reply (normal for EXPN), exactly one mailbox is to be specified on each line of the reply. The case of an ambiguous request is discussed above.

"User name" is a fuzzy term and has been used deliberately. An implementation of the VRFY or EXPN commands MUST include at least recognition of local mailboxes as "user names". However, since current Internet practice often results in a single host handling mail for multiple domains, hosts, especially hosts that provide this functionality, SHOULD accept the "local-part@domain" form as a "user name"; hosts MAY also choose to recognize other strings as "user names".

The case of expanding a mailbox list requires a multiline reply, such as:

```
C: EXPN Example-People
S: 250-Jon Postel <Postel@isi.edu>
S: 250-Fred Fonebone <Fonebone@physics.foo-u.edu>
S: 250 Sam Q. Smith <SQSmith@specific.generic.com>
```

or

```
C: EXPN Executive-Washroom-List
S: 550 Access Denied to You.
```

The character string arguments of the VRFY and EXPN commands cannot be further restricted due to the variety of implementations of the user name and mailbox list concepts. On some systems, it may be appropriate for the argument of the EXPN command to be a file name for a file containing a mailing list, but again there are a variety of file naming conventions on the Internet. Similarly, historical variations in what is returned by these commands are such that the response should be interpreted very carefully, if at all, and SHOULD generally only be used for diagnostic purposes.

3.5.2. VRFY Normal Response

When normal (2yz or 551) responses are returned from a VRFY or EXPN request, the reply MUST include the <Mailbox> name using a "<local-part@domain>" construction, where "domain" is a fully-qualified domain name. In circumstances exceptional enough to justify violating the intent of this specification, free-form text MAY be returned. In order to facilitate parsing by both computers and people, addresses SHOULD appear in pointed brackets. When addresses, rather than free-form debugging information, are returned, EXPN and VRFY MUST return only valid domain addresses that are usable in SMTP RCPT commands. Consequently, if an address implies delivery to a program or other system, the mailbox name used to reach that target MUST be given. Paths (explicit source routes) MUST NOT be returned by VRFY or EXPN.

Server implementations SHOULD support both VRFY and EXPN. For security reasons, implementations MAY provide local installations a way to disable either or both of these commands through configuration options or the equivalent (see Section 7.3). When these commands are supported, they are not required to work across relays when relaying is supported. Since they were both optional in RFC 821, but VRFY was made mandatory in RFC 1123 [10], if EXPN is supported, it MUST be listed as a service extension in an EHLO response. VRFY MAY be listed as a convenience but, since support for it is required, SMTP clients are not required to check for its presence on the extension list before using it.

3.5.3. Meaning of VRFY or EXPN Success Response

A server MUST NOT return a 250 code in response to a VRFY or EXPN command unless it has actually verified the address. In particular, a server MUST NOT return 250 if all it has done is to verify that the syntax given is valid. If only a syntax check is made, 502 (Command not implemented) or 500 (Syntax error, command unrecognized) SHOULD be returned. As stated elsewhere, implementation (in the sense of actually validating addresses and returning information) of VRFY and EXPN are strongly recommended. Hence, implementations that return 500 or 502 for VRFY are not in full compliance with this specification.

There may be circumstances where an address appears to be valid but cannot reasonably be verified in real time, particularly when a server is acting as a mail exchanger for another server or domain. "Apparent validity", in this case, would normally involve at least syntax checking and might involve verification that any domains specified were ones to which the host expected to be able to relay mail. In these situations, reply code 252 SHOULD be returned. These cases parallel the discussion of RCPT verification in Section 2.1. Similarly, the discussion in Section 3.4.1 applies to the use of reply codes 251 and 551 with VRFY (and EXPN) to indicate addresses that are recognized but that would be forwarded or rejected were mail received for them. Implementations generally SHOULD be more aggressive about address verification in the case of VRFY than in the case of RCPT, even if it takes a little longer to do so.

3.5.4. Semantics and Applications of EXPN

EXPN is often very useful in debugging and understanding problems with mailing lists and multiple-target-address aliases. Some systems have attempted to use source expansion of mailing lists as a means of eliminating duplicates. The propagation of aliasing systems with mail on the Internet for hosts (typically with MX and CNAME DNS records), for mailboxes (various types of local host aliases), and in various proxying arrangements has made it nearly impossible for these strategies to work consistently, and mail systems SHOULD NOT attempt them.

3.6. Relaying and Mail Routing

3.6.1. Mail eXchange Records and Relaying

A relay SMTP server is usually the target of a DNS MX record that designates it, rather than the final delivery system. The relay server may accept or reject the task of relaying the mail in the same way it accepts or rejects mail for a local user. If it accepts the task, it then becomes an SMTP client, establishes a transmission channel to the next SMTP server specified in the DNS (according to the rules in Section 5), and sends it the mail. If it declines to relay mail to a particular address for policy reasons, a 550 response SHOULD be returned.

This specification does not deal with the verification of return paths. Server efforts to verify a return path and actions to be taken under various circumstances are outside the scope of this specification.

It is important to note that MX records can point to SMTP servers that act as gateways into other environments, not just SMTP relays and final delivery systems; see Sections 3.7 and 5.

If an SMTP server has accepted the task of relaying the mail and later finds that the destination is incorrect or that the mail cannot be delivered for some other reason, then it MUST construct an "undeliverable mail" notification message and send it to the originator of the undeliverable mail (as indicated by the reverse-path). Formats specified for non-delivery reports by other standards (see, for example, RFC 3461 [39] and RFC 3464 [40]) SHOULD be used if possible.

This notification message must be from the SMTP server at the relay host or the host that first determines that delivery cannot be accomplished. Of course, SMTP servers MUST NOT send notification messages about problems transporting notification messages. One way to prevent loops in error reporting is to specify a null reverse-path in the MAIL command of a notification message. When such a message is transmitted, the reverse-path MUST be set to null (see Section 4.5.5 for additional discussion). A MAIL command with a null reverse-path appears as follows:

```
MAIL FROM:<>
```

As discussed in Section 6.4, a relay SMTP has no need to inspect or act upon the header section or body of the message data and MUST NOT do so except to add its own "Received:" header field (Section 4.4.1 and possibly other trace header fields) and, optionally, to attempt to detect looping in the mail system (see Section 6.3). Of course, this prohibition also applies to any modifications of these header fields or text (see also Section 7.9).

3.6.2. Message Submission Servers as Relays

Many mail-sending clients exist, especially in conjunction with facilities that receive mail via POP3 or IMAP, that have limited capability to support some of the requirements of this specification, such as the ability to queue messages for subsequent delivery attempts. For these clients, it is common practice to make private arrangements to send all messages to a single server for processing and subsequent distribution. SMTP, as specified here, is not ideally suited for this role. A standardized mail submission protocol has been developed that is gradually superseding practices based on SMTP (see RFC 6409 [48]). In any event, because these arrangements are private and fall outside the scope of this specification, they are not described here.

3.7. Mail Gatewaying

While the relay function discussed above operates within the Internet SMTP transport service environment, MX records or various forms of explicit routing may require that an intermediate SMTP server perform a translation function between one transport service and another. As discussed in Section 2.3.10, when such a system is at the boundary between two transport service environments, we refer to it as a "gateway" or "gateway SMTP".

Gatewaying mail between different mail environments, such as different mail formats and protocols, is complex and does not easily yield to standardization. However, some general requirements may be given for a gateway between the Internet and another mail environment.

3.7.1. Header Fields in Gatewaying

Header fields MAY be rewritten when necessary as messages are gatewayed across mail environment boundaries. This may involve inspecting the message body or interpreting the local-part of the destination address in spite of the prohibitions in Section 6.4.

Other mail systems gatewayed to the Internet often use a subset of the RFC 822 header section or provide similar functionality with a different syntax, but some of these mail systems do not have an equivalent to the SMTP envelope. Therefore, when a message leaves the Internet environment, it may be necessary to fold the SMTP envelope information into the message header section. A possible solution would be to create new header fields to carry the envelope information (e.g., "X-SMTP-MAIL:" and "X-SMTP-RCPT:"); however, this would require changes in mail programs in foreign environments and might risk disclosure of private information (see Section 7.2).

3.7.2. Received Lines in Gatewaying

When forwarding a message into or out of the Internet environment, a gateway MUST prepend a Received: line ("header field", see Section 4.4.1), but it MUST NOT alter in any way a Received: line that is already in the header section.

"Received:" header fields of messages originating from other environments may not conform exactly to this specification. However, the most important use of Received: lines is for debugging mail faults, and this debugging can be severely hampered by well-meaning gateways that try to "fix" a Received: line. As another consequence of trace header fields arising in non-SMTP environments, receiving systems MUST NOT reject mail based on the format of a trace header field and SHOULD be extremely robust in the light of unexpected information or formats in those header fields.

The gateway SHOULD indicate the environment and protocol in the "via" clauses of Received header field(s) that it supplies.

3.7.3. Addresses in Gatewaying

From the Internet side, the gateway SHOULD accept all valid address formats in SMTP commands and in the RFC 822 header section, and all valid RFC 822 messages. Addresses and header fields generated by gateways MUST conform to applicable standards (including this one and RFC5322bis [16]). Gateways are, of course, subject to the same rules for handling source routes as those described for other SMTP systems in Section 3.3.

3.7.4. Other Header Fields in Gatewaying

The gateway MUST ensure that all header fields of a message that it forwards into the Internet mail environment meet the requirements for Internet mail. In particular, all addresses in "From:", "To:", "Cc:", etc., header fields MUST be transformed (if necessary) to satisfy the standard header syntax of RFC5322bis [16], MUST reference only fully-qualified domain names, and MUST be effective and useful for sending replies. The translation algorithm used to convert mail from the Internet protocols to another environment's protocol SHOULD ensure that error messages from the foreign mail environment are delivered to the reverse-path from the SMTP envelope, not to an address in the "From:", "Sender:", or similar header fields of the message.

3.7.5. Envelopes in Gatewaying

Similarly, when forwarding a message from another environment into the Internet, the gateway SHOULD set the envelope return path in accordance with an error message return address, if supplied by the foreign environment. If the foreign environment has no equivalent concept, the gateway must select and use a best approximation, with the message originator's address as the default of last resort.

3.8. Terminating Sessions and Connections

An SMTP connection is terminated when the client sends a QUIT command. The server responds with a positive reply code, after which it closes the connection.

An SMTP server MUST NOT intentionally close the connection under normal operational circumstances (see Section 7.8) except:

* After receiving a QUIT command and responding with a 221 reply.

* After detecting the need to shut down the SMTP service and returning a 421 reply code. This reply code can be issued after the server receives any command or, if necessary, asynchronously from command receipt (on the assumption that the client will receive it after the next command is issued).

* After a timeout, as specified in Section 4.5.3.2, occurs waiting for the client to send a command or data.

In particular, a server that closes connections in response to commands that are not understood is in violation of this specification. Servers are expected to be tolerant of unknown commands, issuing a 500 reply and awaiting further instructions from the client.

An SMTP server that is forcibly shut down via external means SHOULD attempt to send a line containing a 421 reply code to the SMTP client before exiting. The SMTP client will normally read the 421 reply code after sending its next command.

SMTP clients that experience a connection close, reset, or other communications failure due to circumstances not under their control (in violation of the intent of this specification but sometimes unavoidable) SHOULD, to maintain the robustness of the mail system, treat the mail transaction as if a 421 response had been received and act accordingly.

There are circumstances, contrary to the intent of this specification, in which an SMTP server may receive an indication that the underlying TCP connection has been closed or reset. To preserve the robustness of the mail system, SMTP servers SHOULD be prepared for this condition and SHOULD treat it as if a QUIT had been received before the connection disappeared.

4. The SMTP Specifications

4.1. SMTP Commands

4.1.1. Command Semantics and Syntax

The SMTP commands define the mail transfer or the mail system function requested by the user. SMTP commands are character strings terminated by <CRLF>. The commands themselves are alphabetic characters terminated by <SP> if parameters follow and <CRLF> otherwise. (In the interest of improved interoperability, SMTP receivers SHOULD tolerate trailing white space before the terminating <CRLF>.) The syntax of the local part of a mailbox MUST conform to receiver site conventions and the syntax specified in Section 4.1.2. The SMTP commands are discussed below. The SMTP replies are discussed in Section 4.2.

A mail transaction involves several data objects that are communicated as arguments to different commands. The reverse-path is the argument of the MAIL command, the forward-path is the argument of the RCPT command, and the mail data is the argument of the DATA command. These arguments or data objects must be transmitted and held, pending the confirmation communicated by the end of mail data

indication that finalizes the transaction. The model for this is that distinct buffers are provided to hold the types of data objects; that is, there is a reverse-path buffer, a forward-path buffer, and a mail data buffer. Specific commands cause information to be appended to a specific buffer, or cause one or more buffers to be cleared.

Several commands (RSET, DATA, QUIT) are specified as not permitting parameters. In the absence of specific extensions offered by the server and accepted by the client, clients MUST NOT send such parameters and servers SHOULD reject commands containing them as having invalid syntax.

4.1.1.1. Extended HELLO (EHLO) or HELLO (HELO)

These commands are used to identify the SMTP client to the SMTP server. The argument clause contains the fully-qualified domain name of the SMTP client, if one is available. In situations in which the SMTP client system does not have a meaningful domain name (e.g., when its address is dynamically allocated and no reverse mapping record is available), the client SHOULD send an address literal (see Section 4.1.3). Additional discussion of domain names in SMTP commands appears in Section 2.3.5.

RFC 2821, and some earlier informal practices, encouraged following the literal by information that would help to identify the client system. That convention was not widely supported, and many SMTP servers considered it an error. In the interest of interoperability, it is probably wise for servers to be prepared for this string to occur, but SMTP clients SHOULD NOT send it.

The SMTP server identifies itself to the SMTP client in the connection greeting reply and in the response to this command.

A client SMTP SHOULD start an SMTP session by issuing the EHLO command. If the SMTP server supports the SMTP service extensions, it will give a successful response, a failure response, or an error response. If the SMTP server, in violation of this specification, does not support any SMTP service extensions, it will generate an error response. Older client SMTP systems MAY, as discussed above, use HELO (as specified in RFC 821) instead of EHLO, and servers MUST support the HELO command and reply properly to it. In any event, a client MUST issue HELO or EHLO before starting a mail transaction.

These commands, and a "250 OK" reply to one of them, confirm that both the SMTP client and the SMTP server are in the initial state, that is, there is no transaction in progress and all state tables and buffers are cleared.

Syntax:

```
ehlo           = "EHLO" SP ( Domain / address-literal ) CRLF

helo           = "HELO" SP Domain CRLF
```

Normally, the response to EHLO will be a multiline reply. Each line of the response contains a keyword and, optionally, one or more parameters. Following the normal syntax for multiline replies, these keywords follow the code (250) and a hyphen for all but the last line, and the code and a space for the last line. The syntax for a positive response, using the ABNF notation and terminal symbols of RFC 5234 [15], is:

```
ehlo-ok-rsp =    ( "250" SP Domain [ SP ehlo-greet ] CRLF )
                 / ( "250-" Domain [ SP ehlo-greet ] CRLF
                 *( "250-" ehlo-line CRLF )
                 "250" SP ehlo-line CRLF )

ehlo-greet =     1*(%d0-9 / %d11-12 / %d14-127)
                 ; string of any characters other than CR or LF

ehlo-line =      ehlo-keyword *( SP ehlo-param )

ehlo-keyword =   (ALPHA / DIGIT) *(ALPHA / DIGIT / "-")
                 ; additional syntax of ehlo-params depends on
                 ; ehlo-keyword

ehlo-param =     1*(%d33-126)
                 ; any CHAR excluding <SP> and all
                 ; control characters (US-ASCII 0-31 and 127
                 ; inclusive)
```

Although EHLO keywords may be specified in upper, lower, or mixed case, they MUST always be recognized and processed in a case-insensitive manner. This is simply an extension of practices specified in RFC 821 and Section 2.4.

The EHLO response MUST contain keywords (and associated parameters if required) for all commands not listed as "required" in Section 4.5.1.

4.1.1.2. MAIL (MAIL)

This command is used to initiate a mail transaction in which the mail data is delivered to an SMTP server that may, in turn, deliver it to one or more mailboxes or pass it on to another system (possibly using SMTP). The argument clause contains a reverse-path and may contain optional parameters. In general, the MAIL command may be sent only when no mail transaction is in progress, see Section 4.1.4.

The reverse-path consists of the sender mailbox. Historically, that mailbox might optionally have been preceded by a list of hosts, but that behavior is now deprecated (see Appendix F.2). In some types of reporting messages for which a reply is likely to cause a mail loop (for example, mail delivery and non-delivery notifications), the reverse-path may be null (see Section 3.6).

This command clears the reverse-path buffer, the forward-path buffer, and the mail data buffer, and it inserts the reverse-path information from its argument clause into the reverse-path buffer.

If service extensions were negotiated, the MAIL command may also carry parameters associated with a particular service extension.

Syntax:

```
mail = "MAIL FROM:" Reverse-path
               [SP Mail-parameters] CRLF
```

4.1.1.3. RECIPIENT (RCPT)

This command is used to identify an individual recipient of the mail data; multiple recipients are specified by multiple uses of this command. The argument clause contains a forward-path and may contain optional parameters.

The forward-path consists of the required destination mailbox. When mail reaches its ultimate destination, the SMTP server inserts it into the destination mailbox in accordance with its host mail conventions.

Prior versions of the SMTP specification included text and examples in this section of use of the deprecated source route construct. If desired, see Appendix F.2 for discussion of that mechanism.

This command appends its forward-path argument to the forward-path buffer; it does not change the reverse-path buffer nor the mail data buffer.

For example, mail received at relay host xyz.com with envelope commands

```
MAIL FROM:<userx@y.foo.org>
RCPT TO:<userc@d.bar.org>
```

will result in a DNS lookup for d.bar.org and transmission to the host specified in the most-preferred MX record that is available (or by the address record if there are no MX records). It will use envelope commands identical to the above, i.e.,

```
MAIL FROM:<userx@y.foo.org>
RCPT TO:<userc@d.bar.org>
```

Since hosts are not required to relay mail at all, xyz.com MAY also reject the message entirely when the RCPT command is received, using a 550 code (since this is a "policy reason").

If the SMTP server determines that a message sent to the mailbox in the forward-path is not deliverable, it MUST either return an appropriate response code (see Section 4.2.2) or generate a non-delivery notification.

If there were multiple failed recipients, either a single notification listing all of the failed recipients or separate notification messages MUST be sent for each failed recipient. For economy of processing by the sender, the former SHOULD be used when possible. All notification messages about undeliverable mail MUST be sent using the MAIL command and MUST use a null return path as discussed in Section 3.6.

If service extensions were negotiated, the RCPT command may also carry parameters associated with a particular service extension offered by the server. The client MUST NOT transmit parameters other than those associated with a service extension offered by the server in its EHLO response.

Syntax:

```
rcpt = "RCPT TO:" ( "<Postmaster@" Domain ">" / "<Postmaster>" /
               Forward-path ) [SP Rcpt-parameters] CRLF
```

Note that, in a departure from the usual rules for local-parts, the "Postmaster" string shown above is treated as case-insensitive.

4.1.1.4. DATA (DATA)

The receiver normally sends a 354 response to DATA, and then treats the lines (strings ending in <CRLF> sequences, as described in Section 2.3.8) following the command as mail data from the sender. This command causes the mail data to be appended to the mail data buffer. Unless some other character or non-character encoding is negotiated with an SMTP extension, the mail data may contain any of the 128 ASCII character codes. Experience has indicated that use of ASCII or ASCII-derived control characters other than SP, HT, CR, and LF may cause problems and SHOULD be avoided when possible.

The mail data are terminated by a line containing only a period, that is, the character sequence "<CRLF>.<CRLF>", where the first <CRLF> is actually the terminator of the previous line (see Section 4.5.2). This is the end of mail data indication. The first <CRLF> of this terminating sequence is also the <CRLF> that ends the final line of the data (message text) or, if there was no mail data, ends the DATA command itself (the "no mail data" case does not conform to this specification since it would require that neither the trace header fields required by this specification nor the message header section required by RFC 5322bis [16] be transmitted). An extra <CRLF> MUST NOT be added, as that would cause an empty line to be added to the message. The only exception to this rule would arise if the message body were passed to the originating SMTP-sender with a final "line" that did not end in <CRLF>; in that case, the originating SMTP system MUST either reject the message as invalid or add <CRLF> in order to have the receiving SMTP server recognize the "end of data" condition.

The custom of accepting lines ending only in <LF>, as a concession to non-conforming behavior on the part of some UNIX systems, has proven to cause more interoperability problems than it solves, and SMTP server systems MUST NOT do this, even in the name of improved robustness. In particular, the sequence "<LF>.<LF>" (bare line feeds, without carriage returns) MUST NOT be treated as equivalent to <CRLF>.<CRLF> as the end of mail data indication.

Receipt of the end of mail data indication requires the server to process the stored mail transaction information. This processing consumes the information in the reverse-path buffer, the forward-path buffer, and the mail data buffer, and on the completion of this command these buffers are cleared. If the processing is successful, the receiver MUST send an OK reply. If the processing fails, the receiver MUST send a failure reply. The SMTP model does not allow for partial failures at this point: either the message is accepted by the server for delivery and a positive response is returned or it is not accepted and a failure reply is returned (see Section 4.4.3 for additional discussion). In sending a positive "250 OK" completion

reply to the end of data indication, the receiver takes full responsibility for the message (see Section 6.1). Errors that are diagnosed subsequently MUST be reported in a mail message.

When the SMTP server accepts a message either for relaying or for final delivery, it inserts a trace record (also referred to interchangeably as a "time stamp line", "Received" line, or "Received:" header field) at the top of the mail data. This trace record indicates the identity of the host that sent the message, the identity of the host that received the message (and is inserting this time stamp), and the date and time the message was received. Relayed messages will have multiple time stamp lines. Details for formation of these lines, including their syntax, is specified in Section 4.4.

Additional discussion about the operation of the DATA command appears in Section 3.3.

Syntax:

```
data = "DATA" CRLF
```

4.1.1.5. RESET (RSET)

This command specifies that the current mail transaction will be aborted. Any stored sender, recipients, and mail data MUST be discarded, and all buffers and state tables cleared. The receiver MUST send a "250 OK" reply to a RSET command with no arguments. A reset command may be issued by the client at any time. It is effectively equivalent to a NOOP (i.e., it has no effect) if issued immediately after EHLO, before EHLO is issued in the session, after an end of data indicator has been sent and acknowledged, or immediately before a QUIT. An SMTP server MUST NOT close the connection as the result of receiving a RSET; that action is reserved for QUIT (see Section 4.1.1.10).

Since EHLO implies some additional processing and response by the server, RSET will normally be more efficient than reissuing that command, even though the formal semantics are the same.

Syntax:

```
rset = "RSET" CRLF
```

4.1.1.6. VERIFY (VRFY)

This command asks the receiver to confirm that the argument identifies a user or mailbox. If it is a user name, information is returned as specified in Section 3.5.

This command has no effect on the reverse-path buffer, the forward-path buffer, or the mail data buffer.

Syntax:

```
vrfy = "VRFY" SP String CRLF
```

4.1.1.7. EXPAND (EXPN)

This command asks the receiver to confirm that the argument identifies a mailing list, and if so, to return the membership of that list. If the command is successful, a reply is returned containing information as described in Section 3.5. This reply will have multiple lines except in the trivial case of a one-member list.

This command has no effect on the reverse-path buffer, the forward-path buffer, or the mail data buffer, and it may be issued at any time.

Syntax:

```
expn = "EXPN" SP String CRLF
```

4.1.1.8. HELP (HELP)

This command causes the server to send helpful information to the client. The command MAY take an argument (e.g., any command name) and return more specific information as a response.

This command has no effect on the reverse-path buffer, the forward-path buffer, or the mail data buffer, and it may be issued at any time.

SMTP servers SHOULD support HELP without arguments and MAY support it with arguments.

Syntax:

```
help = "HELP" [ SP String ] CRLF
```

4.1.1.9. NOOP (NOOP)

This command does not affect any parameters or previously entered commands. It specifies no action other than that the receiver send a "250 OK" reply.

This command has no effect on the reverse-path buffer, the forward-path buffer, or the mail data buffer, and it may be issued at any time. If a parameter string is specified, servers SHOULD ignore it.

Syntax:

```
noop = "NOOP" [ SP String ] CRLF
```

4.1.1.10. QUIT (QUIT)

This command specifies that the receiver MUST send a "221 OK" reply, and then close the transmission channel.

The receiver MUST NOT intentionally close the transmission channel until it receives and replies to a QUIT command (even if there was an error). The sender MUST NOT intentionally close the transmission channel until it sends a QUIT command, and it SHOULD wait until it receives the reply (even if there was an error response to a previous command). If the connection is closed prematurely due to violations of the above or system or network failure, the server MUST cancel any pending transaction, but not undo any previously completed transaction, and generally MUST act as if the command or transaction in progress had received a temporary error (i.e., a 4yz response).

The QUIT command may be issued at any time. Any current uncompleted mail transaction will be aborted.

Syntax:

```
quit = "QUIT" CRLF
```

4.1.1.11. Mail-Parameter and Rcpt-Parameter Error Responses

If the server SMTP does not recognize or cannot implement one or more of the parameters associated with a particular MAIL or RCPT command, it will return code 555.

If, for some reason, the server is temporarily unable to accommodate one or more of the parameters associated with a MAIL or RCPT command, and if the definition of the specific parameter does not mandate the use of another code, it should return code 455.

Errors specific to particular parameters and their values will be specified in the document that defines the parameter.

4.1.2. Command Argument Syntax

The syntax of the argument clauses of the above commands (using the syntax specified in RFC 5234 [15] where applicable) is given below. Some terminals not defined in this document, but are defined elsewhere, specifically:

* In the "core" syntax in Appendix B of RFC 5234 [15]: ALPHA , CRLF , DIGIT , HEXDIG , and SP

* In the message format syntax in RFC5322bis [16]: atext , CFWS , and FWS .

```
Reverse-path   = Path / "<>"

Forward-path   = Path

Path           = "<" Mailbox ">"

Mail-parameters  = esmtp-param *(SP esmtp-param)

Rcpt-parameters  = esmtp-param *(SP esmtp-param)

esmtp-param    = esmtp-keyword ["=" esmtp-value]

esmtp-keyword  = (ALPHA / DIGIT) *(ALPHA / DIGIT / "-")

esmtp-value    = 1*(%d33-60 / %d62-126)
               ; any CHAR excluding "=", SP, and control
               ; characters.  If this string is an email address,
               ; i.e., a Mailbox, then the "xtext" syntax [39]
               ; SHOULD be used.

Keyword        = Ldh-str

Argument       = Atom

Domain         = sub-domain *("." sub-domain)

sub-domain     = Let-dig [Ldh-str]

Let-dig        = ALPHA / DIGIT

Ldh-str        = *( ALPHA / DIGIT / "-" ) Let-dig
```

```
address-literal  = "[" ( IPv4-address-literal /
                   IPv6-address-literal /
                   General-address-literal ) "]"
                   ; See Section 4.1.3

Mailbox        = Local-part "@" ( Domain / address-literal )

Local-part     = Dot-string / Quoted-string
               ; MAY be case-sensitive

Dot-string     = Atom *("."  Atom)

Atom           = 1*atext

Quoted-string  = DQUOTE 1*QcontentSMTP DQUOTE

QcontentSMTP   = qtextSMTP / quoted-pairSMTP

quoted-pairSMTP  = %d92 %d32-126
               ; i.e., backslash followed by any ASCII
               ; graphic (including itself) or SPace

qtextSMTP      = %d32-33 / %d35-91 / %d93-126
               ; i.e., within a quoted string, any
               ; ASCII graphic or space is permitted
               ; without backslash-quoting except
               ; double-quote and the backslash itself.

String         = Atom / Quoted-string
```

Note that the backslash, "\", is a quote character, which is used to indicate that the next character is to be used literally (instead of its normal interpretation). For example, "Joe\,Smith" indicates a single nine-character user name string with the comma being the fourth character of that string.

While the above definition for Local-part is relatively permissive, for maximum interoperability, a mailbox SHOULD NOT be defined with Local-part requiring (or using) the Quoted-string form or with the Local-part being case-sensitive. Further, when comparing a Local-part (e.g., to a specific mailbox name), all quoting MUST be treated as equivalent. A sending system SHOULD transmit the form that uses the minimum quoting possible.

For example, the following 3 local-parts are equivalent and MUST compare equal: "ab cd ef", "ab\ cd ef" and "ab\ \cd ef". Similarly, "fred" and fred must compare equal. White space reduction   MUST NOT be applied to Local-part by intermediate

systems. As particular examples, systems that are not making final delivery MUST NOT make assumptions about the relationships among "ab   cd"@example.com and "ab cd"@example.com or even " "@example.com and ""@example.com.

In the absence of extensions, systems MUST NOT define mailboxes in such a way as to require the use in SMTP of non-ASCII characters (octets with the high order bit set to one) or ASCII "control characters" (decimal value 0-31 and 127) [4][5]. Extensions have been standardized for such use [42][43]. When these extensions are not in use, these characters MUST NOT be used in MAIL or RCPT commands or other commands that require mailbox names.

To promote interoperability and consistent with long-standing guidance about conservative use of the DNS in naming and applications (e.g., see Section 2.3.1 of the base DNS document, RFC 1035 [7]), characters outside the set of alphabetic characters, digits, and hyphen MUST NOT appear in domain name labels for SMTP clients or servers. In particular, the underscore character is not permitted. SMTP servers that receive a command in which invalid character codes have been employed, and for which there are no other reasons for rejection, MUST reject that command with a 501 response (this rule, like others, could be overridden by appropriate SMTP extensions).

4.1.3. Address Literals

Sometimes a host is not known to the domain name system and communication (and, in particular, communication to report and repair the error) is blocked. To bypass this barrier, a special literal form of the address is allowed as an alternative to a domain name. For IPv4 addresses, this form uses four small decimal integers separated by dots and enclosed by brackets such as [123.255.37.2], which indicates an (IPv4) Internet Address in sequence-of-octets form. For IPv6 and other forms of addressing that might eventually be standardized, the form consists of a standardized "tag" that identifies the address syntax, a colon, and the address itself, in a format specified as part of the relevant standards (i.e., RFC 4291 [14] for IPv6).

Specifically:

```
IPv4-address-literal  = Snum 3("."  Snum)

IPv6-address-literal  = "IPv6:" IPv6-addr

General-address-literal  = Standardized-tag ":" 1*dcontent

Standardized-tag  = Ldh-str
```

```
                  ; Standardized-tag MUST be specified in a
                  ; Standards-Track RFC and registered with IANA
                  ; See Section 8.1.2.

   dcontent       = %d33-90 / ; Printable US-ASCII
                  %d94-126 ; excl. "[", "\", "]"

   Snum           = 1*3DIGIT
                  ; representing a decimal integer
                  ; value in the range 0 through 255

   IPv6-addr      = 6( h16 ":" ) ls32
                  / "::" 5( h16 ":" ) ls32
                  / [ h16 ] "::" 4( h16 ":" ) ls32
                  / [ *1( h16 ":" ) h16 ] "::" 3( h16 ":" ) ls32
                  / [ *2( h16 ":" ) h16 ] "::" 2( h16 ":" ) ls32
                  / [ *3( h16 ":" ) h16 ] "::" h16 ":" ls32
                  / [ *4( h16 ":" ) h16 ] "::" ls32
                  / [ *5( h16 ":" ) h16 ] "::" h16
                  / [ *6( h16 ":" ) h16 ] "::"
                  ; This definition is consistent with the one for
                  ; URIs [47].

   ls32           = ( h16 ":" h16 ) / IPv4-address-literal
                  ; least-significant 32 bits of address

   h16            = 1*4HEXDIG
                  ; 16 bits of address represented in hexadecimal
```

4.1.4. Order of Commands

There are restrictions on the order in which these commands may be used.

A session that will contain mail transactions MUST first be initialized by the use of the EHLO command. An SMTP server SHOULD accept commands for non-mail transactions (e.g., VRFY, EXPN, or NOOP) without this initialization.

An EHLO command MAY be issued by a client later in the session. If it is issued after the session begins and the EHLO command is acceptable to the SMTP server, the SMTP server MUST clear all buffers and reset the state exactly as if a RSET command had been issued (specifically, it terminates any mail transaction that was in progress, see Section 3.3). In other words, the sequence of RSET followed immediately by EHLO is redundant, but not harmful other than in the performance cost of executing unnecessary commands. However the response to an additional EHLO command MAY be different from that from prior ones; the client MUST rely only on the responses from the most recent EHLO command.

If the EHLO command is not acceptable to the SMTP server, 501, 500, 502, or 550 failure replies MUST be returned as appropriate. The SMTP server MUST stay in the same state after transmitting these replies that it was in before the EHLO was received.

The SMTP client MUST, if possible, ensure that the domain parameter to the EHLO command is a primary host name as specified for this command in Section 2.3.5. If this is not possible (e.g., when the client's address is dynamically assigned and the client does not have an obvious name), an address literal SHOULD be substituted for the domain name.

An SMTP server MAY verify that the domain name argument in the EHLO command has an address record matching the IP address of the client by looking up the domain name and making the comparison.

The NOOP, HELP, EXPN, VRFY, and RSET commands can be used at any time during a session, or without previously initializing a session. SMTP servers SHOULD process these normally (that is, not return a 503 code) even if no EHLO command has yet been received; clients SHOULD open a session with EHLO before sending these commands.

If these rules are followed, the example in RFC 821 that shows "550 access denied to you" in response to an EXPN command is incorrect unless an EHLO command precedes the EXPN or the denial of access is based on the client's IP address or other authentication or authorization-determining mechanisms.

A mail transaction begins with a MAIL command and then consists of one or more RCPT commands, and a DATA command, in that order. A mail transaction may be aborted by the RSET, a new EHLO, or the QUIT command.

SMTP extensions (see Section 2.2) may create additional commands that initiate, abort, or end the transaction. More generally, any new command MUST clearly document any effect it has on the transaction state.

There may be zero or more transactions in a session. MAIL MUST NOT be sent if a mail transaction is already open, i.e., it should be sent only if no mail transaction had been started in the session, or if the previous one successfully concluded with a successful DATA command, or if the previous one was aborted, e.g., with a RSET or new EHLO.

If the transaction beginning command argument is not acceptable, a 501 failure reply MUST be returned and the SMTP server MUST stay in the same state. If the commands in a transaction are out of order to the degree that they cannot be processed by the server, a 503 failure reply MUST be returned and the SMTP server MUST stay in the same state.

The last command in a session MUST be the QUIT command. The QUIT command SHOULD be used by the client SMTP to request connection closure, even when no session opening command was sent and accepted.

4.2. SMTP Replies

Replies to SMTP commands serve to ensure the synchronization of requests and actions in the process of mail transfer and to guarantee that the SMTP client always knows the state of the SMTP server. Every command MUST generate exactly one reply. Even the command pipelining extension mentioned in Section 2.1 does not change this; it merely allows several commands to be issued before the replies for each are sent together.

The details of the command-reply sequence are described in Section 4.3.

An SMTP reply consists of a three digit number (transmitted as three numeric characters) followed by some text unless specified otherwise in this document. The number is for use by automata to determine what state to enter next; the text is for the human user. The three digits contain enough encoded information that the SMTP client need not examine the text and may either discard it or pass it on to the user, as appropriate. Exceptions are as noted elsewhere in this document. In particular, the 220, 221, 251, 421, and 551 reply codes are associated with message text that must be parsed and interpreted by machines. In the general case, the text may be receiver dependent and context dependent, so there are likely to be varying texts for each reply code. A discussion of the theory of reply codes is given

in Section 4.2.1. Formally, a reply is defined to be the sequence: a three-digit code, <SP>, one line of text, and <CRLF>, or a multiline reply (as defined in the same section). Since, in violation of this specification, the text is sometimes not sent, clients that do not receive it SHOULD be prepared to process the code alone (with or without a trailing space character). Only the EHLO, EXPN, and HELP commands are expected to result in multiline replies in normal circumstances; however, multiline replies are allowed for any command.

In ABNF, server responses are:

```
Greeting       = ( "220 " (Domain / address-literal)
               [ SP textstring ] CRLF ) /
               ( "220-" (Domain / address-literal)
               [ SP textstring ] CRLF
               *( "220-" [ textstring ] CRLF )
               "220" [ SP textstring ] CRLF )

textstring     = 1*(%d09 / %d32-126) ; HT, SP, Printable US-ASCII

Reply-line     = *( Reply-code "-" [ textstring ] CRLF )
               Reply-code [ SP textstring ] CRLF

Reply-code     = %x32-35 %x30-35 %x30-39
```

where "Greeting" appears only in the 220 response that announces that the server is opening its part of the connection. (Other possible server responses upon connection follow the syntax of Reply-line.)

An SMTP server SHOULD send only the reply codes listed in this document or additions to the list as discussed below. An SMTP server SHOULD use the text shown in the examples whenever appropriate.

// See Appendix G.26.

An SMTP client MUST determine its actions only by the reply code, not by the text (except for the "change of address" 251 and 551 and, if necessary, 220, 221, and 421 replies); in the general case, any text, including no text at all (although senders SHOULD NOT send bare codes), MUST be acceptable. The space (blank) following the reply code is considered part of the text. A Sender-SMTP MUST first test the whole 3 digit reply code it receives, as well as any accompanying supplemental codes or information (see RFC 3463 [12] and RFC 5248 [50]). If the full reply code is not recognized, and the additional information is not recognized or missing, the Sender-SMTP MUST use the first digit (severity indication) of a reply code it receives.

The list of codes that appears below MUST NOT be construed as permanent. While the addition of new codes should be a rare and significant activity, with supplemental information in the textual part of the response (including enhanced status codes [12], the successors to that specification, and the associated registry [50]) being preferred, new codes may be added as the result of new Standards or Standards-Track specifications. Consequently, a sender-SMTP MUST be prepared to handle codes not specified in this document and MUST do so by interpreting the first digit only.

In the absence of extensions negotiated with the client, SMTP servers MUST NOT send reply codes whose first digits are other than 2, 3, 4, or 5. Clients that receive such out-of-range codes SHOULD normally treat them as fatal errors and terminate the mail transaction.

4.2.1. Reply Code Severities and Theory

The three digits of the reply each have a special significance. The first digit denotes whether the response is good, bad, or incomplete. An unsophisticated SMTP client, or one that receives an unexpected code, will be able to determine its next action (proceed as planned, redo, retrench, etc.) by examining this first digit. An SMTP client that wants to know approximately what kind of error occurred (e.g., mail system error, command syntax error) may examine the second digit. The third digit and any supplemental information that may be present is reserved for the finest gradation of information.

There are four values for the first digit of the reply code:

2yz Positive Completion reply
The requested action has been successfully completed. A new request may be initiated.

3yz Positive Intermediate reply
The command has been accepted, but the requested action is being held in abeyance, pending receipt of further information. The SMTP client should send another command specifying this information. This reply is used in command sequence groups (i.e., in DATA).

4yz Transient Negative Completion reply
The command was not accepted, and the requested action did not occur. However, the error condition is temporary, and the action may be requested again. The sender should return to the beginning of the command sequence (if any). It is difficult to assign a meaning to "transient" when two different sites (receiver- and sender-SMTP agents) must agree on the interpretation. Each reply in this category might have a different time value, but the SMTP

client SHOULD try again. A rule of thumb to determine whether a reply fits into the 4yz or the 5yz category (see below) is that replies are 4yz if they can be successful if repeated without any change in command form or in properties of the sender or receiver (that is, the command is repeated identically and the receiver does not put up a new implementation).

5yz Permanent Negative Completion reply
The command was not accepted and the requested action did not occur. The SMTP client SHOULD NOT repeat the exact request (in the same sequence). Even some "permanent" error conditions can be corrected, so the human user may want to direct the SMTP client to reinitiate the command sequence by direct action at some point in the future (e.g., after the spelling has been changed, or the user has altered the account status).

It is worth noting that the file transfer protocol (FTP) [19] uses a very similar code architecture and that the SMTP codes are based on the FTP model. However, SMTP uses a one-command, one-response model (while FTP is asynchronous) and FTP's 1yz codes are not part of the SMTP model.

The second digit encodes responses in specific categories:

x0z Syntax: These replies refer to syntax errors, syntactically correct commands that do not fit any functional category, and unimplemented or superfluous commands.

x1z Information: These are replies to requests for information, such as status or help.

x2z Connections: These are replies referring to the transmission channel.

x3z Unspecified.

x4z Unspecified.

x5z Mail system: These replies indicate the status of the receiver mail system vis-a-vis the requested transfer or other mail system action.

The third digit gives a finer gradation of meaning in each category specified by the second digit. The list of replies illustrates this. Each reply text is recommended rather than mandatory, and may even change according to the command with which it is associated. On the other hand, the reply codes must strictly follow the specifications in this section. Receiver implementations should not invent new codes for slightly different situations from the ones described here, but rather adapt codes already defined.

For example, a command such as NOOP, whose successful execution does not offer the SMTP client any new information, will return a 250 reply. The reply is 502 when the command requests an unimplemented non-site-specific action. A refinement of that is the 504 reply for a command that is implemented, but that requests an unimplemented parameter.

The reply text may be longer than a single line; in these cases the complete text must be marked so the SMTP client knows when it can stop reading the reply. This requires a special format to indicate a multiple line reply.

The format for multiline replies requires that every line, except the last, begin with the reply code, followed immediately by a hyphen, "-" (also known as minus), followed by text. The last line will begin with the reply code, followed immediately by <SP>, optionally some text, and <CRLF>. As noted above, servers SHOULD send the <SP> if subsequent text is not sent, but clients MUST be prepared for it to be omitted.

For example:

```
250-First line
250-Second line
250-234 Text beginning with numbers
250 The last line
```

In a multiline reply, the reply code on each of the lines MUST be the same. It is reasonable for the client to rely on this, so it can make processing decisions based on the code in any line, assuming that all others will be the same. In a few cases, there is important data for the client in the reply "text". The client will be able to identify these cases from the current context.

4.2.2. Reply Codes by Function Groups

500 Syntax error, command unrecognized (This may include errors such as command line too long)

501 Syntax error in parameters or arguments

502 Command not implemented (see Section 4.2.4.1)

503 Bad sequence of commands

504 Command parameter not implemented

211 System status, or system help reply

214 Help message (Information on how to use the receiver or the meaning of a particular non-standard command; this reply is useful only to the human user)

220 <domain> Service ready

221 <domain> Service closing transmission channel

421 <domain> Service not available, closing transmission channel (This may be a reply to any command if the service knows it must shut down)

521 <domain> No mail service here.

556 No mail service at this domain.

250 Requested mail action okay, completed

251 User not local; will forward to <forward-path> (See Section 3.4.1)

252 Cannot VRFY user, but will accept message and attempt delivery (See Section 3.5.3)

455 Server unable to accommodate parameters

555 MAIL FROM/RCPT TO parameters not recognized or not implemented

450 Requested mail action not taken: mailbox unavailable (e.g., mailbox busy or temporarily blocked for policy reasons, or server temporarily unavailable if returned before a mail transaction is started)

550 Requested action not taken: mailbox unavailable (e.g., mailbox not found, no access, or command rejected for policy reasons)

451 Requested action aborted: error in processing

551 User not local; please try <forward-path> (See Section 3.4.1)

452 Requested action not taken: insufficient system storage (preferred code for "too many recipients", see Section 4.5.3.1.10)

552 Requested mail action aborted: exceeded storage allocation.

553 Requested action not taken: mailbox name not allowed (e.g., mailbox syntax incorrect). This code is also used as an "ambiguous mailbox" response to VRFY, see Section 3.5.1.

354 Start mail input; end with <CRLF>.<CRLF>

554 Transaction failed (Or, historically in the case of a connection-opening response, "No SMTP service here". 521 is now preferred for that function at connection-opening if the server never accepts mail.)

4.2.3. Reply Codes in Numeric Order

211 System status, or system help reply

214 Help message (Information on how to use the receiver or the meaning of a particular non-standard command; this reply is useful only to the human user)

220 <domain> Service ready

221 <domain> Service closing transmission channel

250 Requested mail action okay, completed

251 User not local; will forward to <forward-path> (See Section 3.4.1)

252 Cannot VRFY user, but will accept message and attempt delivery (See Section 3.5.3)

354 Start mail input; end with <CRLF>.<CRLF>

421 <domain> Service not available, closing transmission channel (This may be a reply to any command if the service knows it must shut down)

450 Requested mail action not taken: mailbox unavailable (e.g., mailbox busy or temporarily blocked for policy reasons, or server temporarily unavailable if returned before a mail transaction is started)

451 Requested action aborted: local error in processing

452 Requested action not taken: insufficient system storage (also preferred code for "too many recipients", see Section 4.5.3.1.10)

455 Server unable to accommodate parameters

500 Syntax error, command unrecognized (This may include errors such as command line too long)

501 Syntax error in parameters or arguments

502 Command not implemented (see Section 4.2.4.1)

503 Bad sequence of commands

504 Command parameter not implemented

521 No mail service (See Section 4.2.4.2.)

550 Requested action not taken: mailbox unavailable (e.g., mailbox not found, no access, or command rejected for policy reasons)

551 User not local; please try <forward-path> (See Section 3.4.1)

552 Requested mail action aborted: exceeded storage allocation.

553 Requested action not taken: mailbox name not allowed (e.g., mailbox syntax incorrect). This code is also used as an "ambiguous mailbox" response to VRFY, see Section 3.5.1.

554 Transaction failed (Or, in the case of a connection-opening response, "No SMTP service here" although 521 is now preferred for the latter. See Section 4.2.4.2.)

555 MAIL FROM/RCPT TO parameters not recognized or not implemented

556 No mail service at this domain. (See Section 4.2.4.2.)

4.2.4. Some specific code situations and relationships

4.2.4.1. Reply Code 502

Questions have been raised as to when reply code 502 (Command not implemented) SHOULD be returned in preference to other codes. 502 SHOULD be used when the command is actually recognized by the SMTP server, but not implemented. If the command is not recognized, code 500 SHOULD be returned. Extended SMTP systems MUST NOT list capabilities in response to EHLO for which they will return 502 (or 500) replies.

4.2.4.2. "No mail accepted" situations and the 521, 554, 556, and 450 codes

Codes 521, 554, and 556 are all used to report different types of permanent "no mail accepted" situations. They differ as follows. 521 is an indication from a system answering on the SMTP port that it does not support SMTP service (a so-called "dummy server" as discussed in RFC 7504 [52] and elsewhere). Obviously, it requires that system exist and that a connection can be made successfully to it. Because a system that does not accept any mail cannot meaningfully accept a RCPT command, any commands (other than QUIT) issued after an SMTP server has issued a 521 reply are client (sender) errors.

When a domain does not intend to accept mail and wishes to publish that fact rather than being subjected to connection attempts, the best way to accomplish that is to use the "Null MX" convention. This is done by advertising a single MX RR (see Section 3.3.9 of RFC 1035 [7]) with an RDATA section consisting of preference number 0 and a zero-length label, written in master files as ".", as the exchange domain, to denote that there exists no mail exchanger for that domain. Reply code 556 is then used by a message submission or intermediate SMTP system (see Section 1.1) to report that it cannot forward the message further because it knows from the DNS entry that the recipient domain does not accept mail. If, despite publishing the DNS entry, the host associated with the server domain chooses to respond on the SMTP port, it SHOULD respond with the 556 code as well. The details of the Null MX convention were first defined in RFC 7505 [53]; see that document for additional discussion of the rationale for that convention.

Reply code 556 would normally be used in response to a RCPT command (or extension command with similar intent) when the SMTP system identifies a domain that it can (or has) determined never accepts mail. Other codes, including 554 and the temporary 450, are used for more transient situations and situations in which an SMTP server cannot or will not deliver to (or accept mail for) a particular system or mailbox for policy reasons rather than ones directly

related to SMTP processing. The 450 code may also be used to reflect a server being temporarily unavailable at connection type or after the EHLO command is issued (i.e., before a mail transaction is initiated).

4.2.4.3. Reply Codes after DATA and the Subsequent <CRLF>.<CRLF>

When an SMTP server returns a positive completion status (2yz code) after the DATA command is completed with <CRLF>.<CRLF>, it accepts responsibility for:

* delivering the message (if the recipient mailbox exists), or

* if attempts to deliver the message fail due to transient conditions, retrying delivery some reasonable number of times at intervals as specified in Section 4.5.4.

* if attempts to deliver the message fail due to permanent conditions, or if repeated attempts to deliver the message fail due to transient conditions, returning appropriate notification to the sender of the original message (using the address in the SMTP MAIL command).

When an SMTP server returns a temporary error status (4yz) code after the DATA command is completed with <CRLF>.<CRLF>, it MUST NOT make a subsequent attempt to deliver that message. The SMTP client retains responsibility for the delivery of that message and may either return it to the user or requeue it for a subsequent attempt (see Section 4.5.4.1).

The user who originated the message SHOULD be able to interpret the return of a transient failure status (by mail message or otherwise) as a non-delivery indication, just as a permanent failure would be interpreted. If the client SMTP successfully handles these conditions, the user will not receive such a reply.

// See Appendix G.27. Proposed text:
// An SMTP response indicating a transient failure status SHOULD
// contain text providing a simple explanation, indicating that
// delivery of the message is delayed.

When an SMTP server returns a permanent error status (5yz) code after the DATA command is completed with <CRLF>.<CRLF>, it MUST NOT make any subsequent attempt to deliver the message. As with temporary error status codes, the SMTP client retains responsibility for the message, but SHOULD NOT again attempt delivery to the same server without user review of the message and response and appropriate intervention.

4.3. Sequencing of Commands and Replies

4.3.1. Sequencing Overview

The communication between the sender and receiver is an alternating dialogue, controlled by the sender. As such, the sender issues a command and the receiver responds with a reply. Unless other arrangements are negotiated through service extensions, the sender MUST wait for this response before sending further commands. One important reply is the connection greeting. Normally, a receiver will send a 220 "Service ready" reply when the connection is completed. The sender SHOULD wait for this greeting message before sending any commands.

Note: all the greeting-type replies have the official name (the fully-qualified primary domain name) of the server host as the first word following the reply code. Sometimes the host will have no meaningful name. See Section 4.1.3 for a discussion of alternatives in these situations.

For example,

```
220 ISIF.USC.EDU Service ready
```

or

```
220 mail.example.com SuperSMTP v 6.1.2 Service ready
```

or

```
220 [10.0.0.1] Clueless host service ready
```

The table below lists alternative success and failure replies for each command. These SHOULD be strictly adhered to. A receiver MAY substitute text in the replies, but the meanings and actions implied by the code numbers and by the specific command reply sequence MUST be preserved. However, in order to provide robustness as SMTP is extended and evolves, the discussion in Section 4.2.1 still applies: all SMTP clients MUST be prepared to accept any code that conforms to the discussion in that section and MUST be prepared to interpret it on the basis of its first digit only.

4.3.2. Command-Reply Sequences

Each command is listed with its usual possible replies. The prefixes used before the possible replies are "I" for intermediate, "S" for success, and "E" for error. Since some servers may generate other replies under special circumstances, and to allow for future extension, SMTP clients SHOULD, when possible, interpret only the first digit of the reply and MUST be prepared to deal with unrecognized reply codes by interpreting the first digit only. Unless extended using the mechanisms described in Section 2.2, SMTP servers MUST NOT transmit reply codes to an SMTP client that are other than three digits or that do not start in a digit between 2 and 5 inclusive.

These sequencing rules and, in principle, the codes themselves, can be extended or modified by SMTP extensions offered by the server and accepted (requested) by the client. However, if the target is more precise granularity in the codes, rather than codes for completely new purposes, the system described in RFC 3463 [12] SHOULD be used in preference to the invention of new codes.

In addition to the codes listed below, any SMTP command can return any of the following codes if the corresponding unusual circumstances are encountered:

500 For the "command line too long" case or if the command name was not recognized. Note that producing a "command not recognized" error in response to the required subset of these commands is a violation of this specification. Similarly, producing a "command too long" message for a command line shorter than 512 characters would violate the provisions of Section 4.5.3.1.4.

501 Syntax error in command or arguments. In order to provide for future extensions, commands that are specified in this document as not accepting arguments (DATA, RSET, QUIT) SHOULD return a 501 message if arguments are supplied in the absence of EHLO-advertised extensions.

421 Service shutting down and closing transmission channel

Specific sequences are:

CONNECTION ESTABLISHMENT

- S: 220
  E: 521, 554, 556, 450 (if the system receiving the connection attempt is able to answer but is temporarily not available to receive email)

EHLO or HELO

- S: 250
  E: 504 (a conforming implementation could return this code only in fairly obscure cases), 550, 502 (permitted only with an old-style server that does not support EHLO), 450 (see note above)

MAIL

- S: 250
  E: 552, 451, 452, 550, 553, 503, 455, 555

RCPT

- S: 250, 251 (but see Section 3.4.1 for discussion of 251 and 551)
  E: 550, 551, 552 (obsolete for "too many recipients; see Section 4.5.3.1.10), 553, 450, 451, 452, 503, 455, 555

DATA

- I: 354 -> data -> S: 250

  o E: 552, 554, 451, 452

  o E: 450, 550 (rejections for policy reasons)

- E: 503, 554

RSET

- S: 250

VRFY

- S: 250, 251, 252
  E: 550, 551, 553, 502, 504

EXPN

- S: 250, 252
  E: 550, 500, 502, 504

HELP

- S: 211, 214
  E: 502, 504

NOOP

- S: 250

QUIT

- S: 221

4.4. Trace Information

When inserted by SMTP, trace information is used to provide an audit trail of message handling. In addition, it indicates a route back to the sender of the message.

4.4.1. Received Header Field (Time Stamp)

When an SMTP server receives a message for delivery or further processing, it MUST insert trace (often referred to as "time stamp" or "Received" information) at the beginning of the message content, as discussed in Section 4.1.1.4.

This line MUST be structured as follows:

* The FROM clause, which MUST be supplied in an SMTP environment, SHOULD contain both (1) the name of the source host as presented in the EHLO command and (2) an address literal containing the IP address of the source, determined from the TCP connection.

* The ID clause MAY contain an "@" as suggested in the Internet Message format specification [16], but this is not required.

* If the FOR clause appears, it MUST contain exactly one <path> entry, even when multiple RCPT commands have been given. Multiple <path>s raise some security issues and have been deprecated, see Section 7.2.

An Internet mail program MUST NOT change or delete a Received: line that was previously added to the message header section. SMTP servers MUST prepend Received lines to messages; they MUST NOT change the order of existing lines or insert Received lines in any other location.

As the Internet grows, comparability of Received header fields is important for detecting problems, especially slow relays. SMTP servers that create Received header fields SHOULD use explicit offsets in the dates (e.g., -0800), rather than time zone names of any type. Local time (with an offset) SHOULD be used rather than UTC when feasible. This formulation allows slightly more information

about local circumstances to be specified. If UTC is needed, the receiver need merely do some simple arithmetic to convert the values. Use of UTC loses information about the time zone-location of the server. If it is desired to supply a time zone name, it SHOULD be included in a comment. If UTC is actually being supplied instead of the local time zone, it should be denoted by a time zone offset of "-0000". Time zones aligned with the prime meridian (e.g., "GMT") are shown as "+0000".

4.4.2. Return-path Header Field

When the delivery SMTP server makes the "final delivery" of a message, it MUST insert a return-path line at the beginning of the mail data. Here, final delivery means the message has left the SMTP environment. Normally, this would mean it had been delivered to the destination user or an associated mail drop, but in some cases it may be further processed and transmitted by another mail system.

It is possible for the mailbox in the return path to be different from the actual sender's mailbox, for example, if error responses are to be delivered to a special error handling mailbox rather than to the message sender. When mailing lists are involved, this arrangement is common and useful as a means of directing errors to the list maintainer rather than the message originator.

A message-originating SMTP system SHOULD NOT send a message that already contains a Return-path header field. SMTP servers performing a relay function MUST NOT inspect the message data, and especially not to the extent needed to determine if Return-path header fields are present. SMTP servers making final delivery MAY remove Return-path header fields before adding their own.

The primary purpose of the Return-path is to designate the address to which messages indicating non-delivery or other mail system failures are to be sent. For this to be unambiguous, exactly one return path SHOULD be present when the message is delivered. Systems using the syntax specified here with non-SMTP transports SHOULD designate an unambiguous address, associated with the transport envelope, to which error reports (e.g., non-delivery messages) should be sent.

It is sometimes difficult for an SMTP server to determine whether or not it is making final delivery since forwarding or other operations may occur after the message is accepted for delivery. Consequently, any further (forwarding, gateway, or relay) systems MAY remove the return path and rebuild the MAIL command as needed to ensure that exactly one such line appears in a delivered message.

4.4.3. Return-path, Non-SMTP Systems, and Gateways

When SMTP systems, especially relay ones that are receiving messages and then processing them for the next hop, special issues arise and care must be taken. In particular:

* a gateway from SMTP -> elsewhere SHOULD insert a return-path header field, unless it is known that the "elsewhere" transport also uses Internet domain addresses and maintains the envelope sender address separately.

* a gateway from elsewhere -> SMTP SHOULD delete any return-path header field present in the message, and either copy that information to the SMTP envelope or combine it with information present in the envelope of the other transport system to construct the reverse-path argument to the MAIL command in the SMTP envelope.

The server must give special treatment to cases in which processing following the end of mail data indication is only partially successful. This could happen if, after accepting several recipients and the mail data, the SMTP server finds that the mail data could be successfully delivered to some, but not all, of the recipients. In such cases, the response to the DATA command MUST be an OK reply. However, the SMTP server MUST compose and send an "undeliverable mail" notification message to the originator of the message.

4.4.4. Additional Trace Fields

"Received" and "Return-path", defined above, are the only two trace fields that are part of SMTP. Additional trace fields, or variations on the definitions here for other mail transports, may be defined and registered as described in [I-D.ietf-emailcore-rfc5322bis].

4.4.5. Trace Information Summary and Analysis

The text above implies that the final mail data will begin with a return path line, followed by one or more time stamp lines. These lines will be followed by the rest of the mail data: first the balance of the mail header section and then the body (RFC 5322bis [16]).

The time stamp line and the return path line are formally defined as follows (the definitions for "FWS" and "CFWS" appear in RFC 5322bis [16]):

```
Return-path-line  = "Return-Path:" FWS Reverse-path <CRLF>
```

```
Time-stamp-line  = "Received:" FWS Stamp <CRLF>

Stamp           = From-domain By-domain Opt-info [CFWS] ";"
                  FWS date-time
                  ; where "date-time" is as defined in RFC5322bis [16]
                  ; but the "obs-" forms, especially two-digit
                  ; years, are prohibited in SMTP and MUST NOT be used.

From-domain     = "FROM" FWS Extended-Domain

By-domain       = CFWS "BY" FWS Extended-Domain

Extended-Domain  = Domain /
                  ( Domain FWS "(" TCP-info ")" ) /
                  ( address-literal FWS "(" TCP-info ")" )

TCP-info        = address-literal / ( Domain FWS address-literal )
                  ; Information derived by server from TCP connection
                  ; not client EHLO.

Opt-info        = [Via] [With] [ID] [For]
                  [Additional-Registered-Clauses]

Via             = CFWS "VIA" FWS Link

With            = CFWS "WITH" FWS Protocol

ID              = CFWS "ID" FWS ( Atom / msg-id )
                  ; msg-id is defined in RFC 5322bis [16]

For             = CFWS "FOR" FWS ( Path / Mailbox )

Additional-Registered-Clauses  = 1*(CFWS Atom FWS String)
                  ; See Section 8.1.4.

Link            = "TCP" / Addtl-Link

Addtl-Link      = Atom
                  ; Additional standard names for links are
                  ; registered with the Internet Assigned Numbers
                  ; Authority (IANA).  "Via" is primarily of value
                  ; with non-Internet transports.  SMTP servers
                  ; SHOULD NOT use unregistered names.

Protocol        = "ESMTP" / "SMTP" / Attdl-Protocol

Addtl-Protocol  = Atom
```

```
                 ; Additional standard names for protocols are
                 ; registered with the Internet Assigned Numbers
                 ; Authority (IANA) in the "mail parameters"
                 ; registry [13].  SMTP servers SHOULD NOT
                 ; use unregistered names.
```

4.5. Additional Implementation Issues

4.5.1. Minimum Implementation

In order to make SMTP workable, the following minimum implementation MUST be provided by all receivers. The following commands MUST be supported to conform to this specification:

```
EHLO
HELO
MAIL
RCPT
DATA
RSET
NOOP
QUIT
VRFY
```

Any system that includes an SMTP server supporting mail relaying or delivery MUST support the reserved mailbox "postmaster" as a case-insensitive local name. This postmaster address is not strictly necessary if the server always returns 554 on connection opening (as described in Section 3.1). The requirement to accept mail for postmaster implies that RCPT commands that specify a mailbox for postmaster at any of the domains for which the SMTP server provides mail service, as well as the special case of "RCPT TO:<Postmaster>" (with no domain specification), MUST be supported.

SMTP systems are expected to make every reasonable effort to accept mail directed to Postmaster from any other system on the Internet. In extreme cases -- such as to contain a denial of service attack or other breach of security -- an SMTP server may block mail directed to Postmaster. However, such arrangements SHOULD be narrowly tailored so as to avoid blocking messages that are not part of such attacks.

4.5.2. Transparency

Without some provision for data transparency, the character sequence "<CRLF>.<CRLF>" ends the mail text and cannot be sent by the user. In general, users are not aware of such "forbidden" sequences. To allow all user composed text to be transmitted transparently, the following procedures are used:

* Before sending a line of mail text, the SMTP client checks the first character of the line. If it is a period, one additional period is inserted at the beginning of the line.

* When a line of mail text is received by the SMTP server, it checks the line. If the line is composed of a single period, it is treated as the end of mail indicator. If the first character is a period and there are other characters on the line, the first character is deleted.

The mail data may contain any of the 128 ASCII characters. All characters are to be delivered to the recipient's mailbox, including spaces, vertical and horizontal tabs, and other control characters. If the transmission channel provides an 8-bit byte (octet) data stream, the 7-bit ASCII codes are transmitted, right justified, in the octets, with the high-order bits cleared to zero.

In some systems, it may be necessary to transform the data as it is received and stored. This may be necessary for hosts that use a different character set than ASCII as their local character set, that store data in records rather than strings, or which use special character sequences as delimiters inside mailboxes. If such transformations are necessary, they MUST be reversible, especially if they are applied to mail being relayed.

4.5.3. Sizes and Timeouts

4.5.3.1. Size Limits and Minimums

There are several objects that have required minimum/maximum sizes. Every implementation MUST be able to receive objects of at least these sizes. Objects larger than these sizes SHOULD be avoided when possible. However, some Internet mail constructs such as encoded X.400 addresses (RFC 2156 [32]) will often require larger objects. Clients MAY attempt to transmit these, but MUST be prepared for a server to reject them if they cannot be handled by it. To the maximum extent possible, implementation techniques that impose no limits on the length of these objects should be used.

Extensions to SMTP may involve the use of characters that occupy more than a single octet each. This section therefore specifies lengths in octets where absolute lengths, rather than character counts, are intended.

4.5.3.1.1. Local-part

The maximum total length of a user name or other local-part is 64 octets.

4.5.3.1.2. Domain

The maximum total length of a domain name or number is 255 octets.

4.5.3.1.3. Path

The maximum total length of a reverse-path or forward-path is 256 octets (including the punctuation and element separators).

4.5.3.1.4. Command Line

The maximum total length of a command line including the command word and the <CRLF> is 512 octets. SMTP extensions may be used to increase this limit.

4.5.3.1.5. Reply Line

The maximum total length of a reply line including the reply code and the <CRLF> is 512 octets. More information may be conveyed through multiple-line replies.

4.5.3.1.6. Text Line

The maximum total length of a text line including the <CRLF> is 1000 octets (not counting the leading dot duplicated for transparency). This number may be increased by the use of SMTP Service Extensions.

4.5.3.1.7. Message Content

The maximum total length of a message content (including any message header section as well as the message body) MUST BE at least 64K octets. Since the introduction of Internet Standards for multimedia mail (RFC 2045 [30]), message lengths on the Internet have grown dramatically, and message size restrictions should be avoided if at all possible. SMTP server systems that must impose restrictions SHOULD implement the "SIZE" service extension of RFC 1870 [11], and SMTP client systems that will send large messages SHOULD utilize it when possible.

4.5.3.1.8. Recipient Buffer

The minimum total number of recipients that MUST be buffered is 100 recipients. Rejection of messages (for excessive recipients) with fewer than 100 RCPT commands is a violation of this specification. The general principle that relaying SMTP server MUST NOT, and delivery SMTP servers SHOULD NOT, perform validation tests on message header fields suggests that messages SHOULD NOT be rejected based on the total number of recipients shown in header fields. A server that imposes a limit on the number of recipients MUST behave in an orderly fashion, such as rejecting additional addresses over its limit rather than silently discarding addresses previously accepted. A client that needs to deliver a message containing over 100 RCPT commands SHOULD be prepared to transmit in 100-recipient "chunks" if the server declines to accept more than 100 recipients in a single message.

4.5.3.1.9. Treatment When Limits Exceeded

Errors due to exceeding these limits may be reported by using the reply codes. Some examples of reply codes are:

500 Line too long.

or

501 Path too long

or

452 Too many recipients (see below)

or

552 Too much mail data (historically also used for too many recipients (see below).

4.5.3.1.10. Too Many Recipients Code

RFC 821 [6] incorrectly listed the error where an SMTP server exhausts its implementation limit on the number of RCPT commands ("too many recipients") as having reply code 552. The correct reply code for this condition is 452. At the time RFC 5321 was written, the use of response code 552 by servers was sufficiently common that client implementation were advised to simply treat it as if 452 had been sent. That advice is no longer necessary or useful.

When a conforming SMTP server encounters this condition, it has at least 100 successful RCPT commands in its recipient buffer. If the server is able to accept the message, then at least these 100 addresses will be removed from the SMTP client's queue. When the client attempts retransmission of those addresses that received 452 responses, at least 100 of these will be able to fit in the SMTP server's recipient buffer. Each retransmission attempt that is able to deliver anything will be able to dispose of at least 100 of these recipients.

If an SMTP server has an implementation limit on the number of RCPT commands and this limit is exhausted, it MUST use a response code of 452. If the server has a configured site-policy limitation on the number of RCPT commands, it MAY instead use a 5yz response code. In particular, if the intent is to prohibit messages with more than a site-specified number of recipients, rather than merely limit the number of recipients in a given mail transaction, it would be reasonable to return a 503 response to any DATA command received subsequent to the 452 code or to simply return the 503 after DATA without returning any previous negative response.

4.5.3.2. Timeouts

An SMTP client MUST provide a timeout mechanism. It MUST use per-command timeouts rather than somehow trying to time the entire mail transaction. Timeouts SHOULD be easily reconfigurable, preferably without recompiling the SMTP code. To implement this, a timer is set for each SMTP command and for each buffer of the data transfer. The latter means that the overall timeout is inherently proportional to the size of the message.

Based on extensive experience with busy mail-relay hosts, the minimum per-command timeout values SHOULD be as follows:

4.5.3.2.1. Initial 220 Message: 5 Minutes

An SMTP client process needs to distinguish between a failed TCP connection and a delay in receiving the initial 220 greeting message. Many SMTP servers accept a TCP connection but delay delivery of the 220 message until their system load permits more mail to be processed.

4.5.3.2.2. MAIL Command: 5 Minutes

4.5.3.2.3. RCPT Command: 5 Minutes

A longer timeout is required if processing of mailing lists and aliases is not deferred until after the message was accepted.

4.5.3.2.4. DATA Initiation: 2 Minutes

This is while awaiting the "354 Start Input" reply to a DATA command.

4.5.3.2.5. Data Block: 3 Minutes

This is while awaiting the completion of each TCP SEND call transmitting a chunk of data.

4.5.3.2.6. DATA Termination: 10 Minutes.

This is while awaiting the "250 OK" reply. When the receiver gets the final period terminating the message data, it typically performs processing to deliver the message to a user mailbox. A spurious timeout at this point would be very wasteful and would typically result in delivery of multiple copies of the message, since it has been successfully sent and the server has accepted responsibility for delivery. See Section 6.1 for additional discussion.

4.5.3.2.7. Server Timeout: 5 Minutes.

An SMTP server SHOULD have a timeout of at least 5 minutes while it is awaiting the next command from the sender.

4.5.4. Retry Strategies

The common structure of a host SMTP implementation includes user mailboxes, one or more areas for queuing messages in transit, and one or more daemon processes for sending and receiving mail. The exact structure will vary depending on the needs of the users on the host and the number and size of mailing lists supported by the host. We describe several optimizations that have proved helpful, particularly for mailers supporting high traffic levels.

Any queuing strategy MUST include timeouts on all activities on a per-command basis. A queuing strategy MUST NOT send error messages in response to error messages under any circumstances.

4.5.4.1. Sending Strategy

The general model for an SMTP client is one or more processes that periodically attempt to transmit outgoing mail. In a typical system, the program that composes a message has some method for requesting immediate attention for a new piece of outgoing mail, while mail that cannot be transmitted immediately MUST be queued and periodically retried by the sender. A mail queue entry will include not only the message itself but also the envelope information.

The sender MUST delay retrying a particular destination after one attempt has failed. In general, the retry interval SHOULD be at least 30 minutes; however, more sophisticated and variable strategies will be beneficial when the SMTP client can determine the reason for non-delivery.

Retries continue until the message is transmitted or the sender gives up; the give-up time generally needs to be at least 4-5 days. It MAY be appropriate to set a shorter maximum number of retries for non-delivery notifications and equivalent error messages than for standard messages. The parameters to the retry algorithm MUST be configurable.

A client SHOULD keep a list of hosts it cannot reach and corresponding connection timeouts, rather than just retrying queued mail items.

Experience suggests that failures are typically transient (the target system or its connection has crashed), favoring a policy of two connection attempts in the first hour the message is in the queue, and then backing off to one every two or three hours.

The SMTP client can shorten the queuing delay in cooperation with the SMTP server. For example, if mail is received from a particular address, it is likely that mail queued for that host can now be sent. Application of this principle may, in many cases, eliminate the requirement for an explicit "send queues now" function such as ETRN, RFC 1985 [29].

The strategy may be further modified as a result of multiple addresses per host (see below) to optimize delivery time versus resource usage.

An SMTP client may have a large queue of messages for each unavailable destination host. If all of these messages were retried in every retry cycle, there would be excessive Internet overhead and the sending system would be blocked for a long period. Note that an SMTP client can generally determine that a delivery attempt has

failed only after a timeout of several minutes, and even a one-minute timeout per connection will result in a very large delay if retries are repeated for dozens, or even hundreds, of queued messages to the same host.

At the same time, SMTP clients SHOULD use great care in caching negative responses from servers. In an extreme case, if EHLO is issued multiple times during the same SMTP connection, different answers may be returned by the server. More significantly, 5yz responses to the MAIL command MUST NOT be cached.

When a mail message is to be delivered to multiple recipients, and the SMTP server to which a copy of the message is to be sent is the same for multiple recipients, then only one copy of the message SHOULD be transmitted. That is, the SMTP client SHOULD use the command sequence: MAIL, RCPT, RCPT, ..., RCPT, DATA instead of the sequence: MAIL, RCPT, DATA, ..., MAIL, RCPT, DATA. However, if there are very many addresses, a limit on the number of RCPT commands per MAIL command MAY be imposed. This efficiency feature SHOULD be implemented.

Similarly, to achieve timely delivery, the SMTP client MAY support multiple concurrent outgoing mail transactions. However, some limit may be appropriate to protect the host from devoting all its resources to mail.

4.5.4.2. Receiving Strategy

The SMTP server SHOULD attempt to keep a pending listen on the SMTP port (specified by IANA as port 25) at all times. This requires the support of multiple incoming TCP connections for SMTP. Some limit MAY be imposed, but servers that cannot handle more than one SMTP transaction at a time are not in conformance with the intent of this specification.

As discussed above, when the SMTP server receives mail from a particular host address, it could activate its own SMTP queuing mechanisms to retry any mail pending for that host address.

4.5.5. Messages with a Null Reverse-Path

There are several types of notification messages that are required by existing and proposed Standards to be sent with a null reverse-path, namely non-delivery notifications as discussed in Section 3.6.1 and Section 3.6.2, other kinds of Delivery Status Notifications (DSNs, RFC 3461 [39]), and Message Disposition Notifications (MDNs, RFC 8098 [44]). All of these kinds of messages are notifications about a previous message, and they are sent to the reverse-path of the

previous mail message. (If the delivery of such a notification message fails, that usually indicates a problem with the mail system of the host to which the notification message is addressed. For this reason, at some hosts the MTA is set up to forward such failed notification messages to someone who is able to fix problems with the mail system, e.g., via the postmaster alias.)

All other types of messages (i.e., any message which is not required by a Standards-Track RFC to have a null reverse-path) SHOULD be sent with a valid, non-null reverse-path.

Implementers of automated email processors should be careful to make sure that the various kinds of messages with a null reverse-path are handled correctly. In particular, such systems SHOULD NOT reply to messages with a null reverse-path, and they SHOULD NOT add a non-null reverse-path, or change a null reverse-path to a non-null one, to such messages when forwarding.

5. Address Resolution and Mail Handling

5.1. Locating the Target Host

Unless special circumstances exist as described in Section 3.3, once an SMTP client lexically identifies a domain to which mail will be delivered for processing (as described in Sections 2.3.5 and 3.6), a DNS lookup MUST be performed to resolve the domain name as specified in RFC 1035 [7] and RFC 1123 Section 5.3.5 [10]). The names are required to be fully-qualified domain names (FQDNs) as discussed in Section 2.3.5.

The lookup first attempts to locate an MX record associated with the name. If a CNAME record is found, the resulting name is processed as if it were the initial name. If a non-existent domain error is returned, this situation MUST be reported as an error. If a temporary error is returned, the message MUST be queued and retried later (see Section 4.5.4.1). If an empty list of MXs is returned, the address is treated as if it was associated with an implicit MX RR with a preference of 0, pointing to that host. If MX records are present, but none of them are usable, or the implicit MX is unusable, this situation MUST be reported as an error.

When the lookup succeeds, the mapping can result in a list of alternative delivery addresses rather than a single address. This can be due to multiple MX records, multihoming, or both. To provide reliable mail transmission, the SMTP client MUST be able to try (and be prepared to retry) each of the relevant addresses in this list in order (see below), until a delivery attempt succeeds. However, as discussed more generally in Section 7.8 there MAY also be a

configurable limit on the number of alternate addresses that can be tried. In any case, the SMTP client SHOULD try at least two addresses.

If one or more MX RRs are found for a given name, SMTP systems MUST NOT utilize any address RRs associated with that name unless they are located using the MX RRs; the "implicit MX" rule above applies only if there are no MX records present. If MX records are present, but none of them are usable, this situation MUST be reported as an error. That domain name also MUST be a primary host name, i.e., it is not allowed to be an alias.

When a domain name associated with an MX RR is looked up and the associated data field obtained, the data field of that response MUST contain a domain name that conforms to the specifications of Section 2.3.5. That domain name, when queried, MUST return at least one address record (e.g., A or AAAA RR) that gives the IP address of the SMTP server to which the message should be directed. Any other response, specifically including a value that will return a CNAME record when queried, lies outside the scope of this Standard. The prohibition on labels in the data that resolve to CNAMEs is discussed in more detail in RFC 2181, Section 10.3 [33].

Two types of information are used to rank the host addresses: multiple MX records, and multihomed hosts.

MX records contain a numerical preference indication that MUST be used in sorting if more than one such record appears. Lower numbers are more preferred than higher ones. The sender-SMTP MUST inspect the list for any of the names or addresses by which it might be known in mail transactions. If a matching record is found, all records at that preference level and higher-numbered ones MUST be discarded from consideration. If there are no records left at that point, it is an error condition, and a 5yz reply code generated (terminating the mail transaction) or the message MUST be returned as undeliverable. If there is a single MX record at the most-preferred preference label, the data field associated with that record is used as the next destination. Otherwise, if there are multiple records with the same preference and there is no clear reason to favor one (e.g., by recognition of an easily reached address), then the sender-SMTP MUST randomize them to spread the load across multiple mail exchangers for a specific organization.

// There are (or may be) some issues with the next paragraph, text
// that dates from RFC 2821 or earlier. See just after it.
The destination host (from either the data field of the preferred MX record or from an address record found in an implicit MX) may be

multihomed. In those cases the domain name resolver will return a list of alternative IP addresses. It is the responsibility of the domain name resolver interface to have ordered this list by decreasing preference if necessary, and the SMTP sender MUST try them in the order presented.

// John Levine has some concerns about the above paragraph. I hope
// this summary is correct, but he suggests that the whole case is
// rare, partially because multihoming in the traditional sense is
// far less common than before connectivity got as complicated
// (firewalls, NATs, etc.) as it is today. Today, the dominant case
// of a host having more than one address facing the public Internet
// involves having one IPv4 address and one IPv6 address. On the
// other hand, multiple addresses for a single domain name (not a
// single host) is common as a way to support load balancing. If we
// were to go back to the last half of the 1980s, I think load
// balancing for mail would have been specified, not by giving
// different physical hosts the same name, but by giving them
// different names and using MX records with the same preference
// level. We don't need to address that distinction and should
// probably avoid trying to do so. John suggests dealing with the
// problem by replacing all (or most) of the paragraph above with the
// following:
When hosts have multiple A and/or AAAA records, DNS servers often return randomized or subset results to aid load balancing, so clients SHOULD use the records in the order returned.

// It seems to me that we can: (i) ignore the problem (or decide
// there really isn't one) and stick with the existing text. (ii)
// Discard the original text and replace it with John's proposed text
// or some variation on it. (iii) Come up with some hybrid of, or
// variation on, both versions, perhaps talking explicitly about
// multihoming and load balancing. Unless we pick (i), we should all
// understand that the MUST of the original is being dropped to a
// SHOULD and see if we are ok with that. What is the WG's pleasure?

Although the capability to try multiple alternative addresses is required, specific installations may want to limit or disable the use of alternative addresses. The question of whether a sender should attempt retries using the different addresses of a multihomed host has been controversial. The main argument for using the multiple addresses is that it maximizes the likelihood of timely delivery, and indeed sometimes the likelihood of any delivery; the counter-argument is that it may result in unnecessary resource use. Note that resource use is also strongly determined by the sending strategy discussed in Section 4.5.4.1.

If an SMTP server receives a message with a destination for which it is a designated Mail eXchanger, it MAY relay the message (potentially after having rewritten the MAIL FROM and/or RCPT TO addresses), make final delivery of the message, or hand it off using some mechanism outside the SMTP-provided transport environment. Of course, neither of the latter require that the list of MX records be examined further.

If it determines that it should relay the message without rewriting the address, it MUST process the MX records as described above to determine candidates for delivery.

5.2. IPv6 and MX Records

In the contemporary Internet, SMTP clients and servers may be hosted on IPv4 systems, IPv6 systems, or dual-stack systems that are compatible with either version of the Internet Protocol. The host domains to which MX records point may, consequently, contain "A RR"s (IPv4), "AAAA RR"s (IPv6), or any combination of them. While RFC 3974 [46] discusses some operational experience in mixed environments, it was not comprehensive enough to justify standardization, and some of its recommendations appear to be inconsistent with this specification. The appropriate actions to be taken either will depend on local circumstances, such as performance of the relevant networks and any conversions that might be necessary, or will be obvious (e.g., an IPv6-only client need not attempt to look up A RRs or attempt to reach IPv4-only servers). Designers of SMTP implementations that might run in IPv6 or dual-stack environments should study the procedures above, especially the comments about multihomed hosts, and, preferably, provide mechanisms to facilitate operational tuning and mail interoperability between IPv4 and IPv6 systems while considering local circumstances.

6. Problem Detection and Handling

6.1. Reliable Delivery and Replies by Email

When the receiver-SMTP accepts a piece of mail (by sending a "250 OK" message in response to DATA), it is accepting responsibility for delivering or relaying the message. It must take this responsibility seriously. It MUST NOT lose the message for frivolous reasons, such as because the host later crashes or because of a predictable resource shortage. Some reasons that are not considered frivolous are discussed in the next subsection and in Section 7.8.

If there is a delivery failure after acceptance of a message, the receiver-SMTP MUST formulate and mail a notification message. This notification MUST be sent using a null ("<>") reverse-path in the

envelope. The recipient of this notification MUST be the address from the envelope return path (or the Return-Path: line). However, if this address is null ("<>"), the receiver-SMTP MUST NOT send a notification. Obviously, nothing in this section can or should prohibit local decisions (i.e., as part of the same system environment as the receiver-SMTP) to log or otherwise transmit information about null address events locally if that is desired.

Some delivery failures after the message is accepted by SMTP will be unavoidable. For example, it may be impossible for the receiving SMTP server to validate all the delivery addresses in RCPT command(s) due to a "soft" domain system error, because the target is a mailing list (see earlier discussion of RCPT), or because the server is acting as a relay and has no immediate access to the delivering system.

To avoid receiving duplicate messages as the result of timeouts, a receiver-SMTP MUST seek to minimize the time required to respond to the final <CRLF>.<CRLF> end of data indicator. See RFC 1047 [21] for a discussion of this problem.

6.2. Unwanted, Unsolicited, and "Attack" Messages

Utility and predictability of the Internet mail system requires that messages that can be delivered should be delivered, regardless of any syntax or other faults associated with those messages and regardless of their content. If they cannot be delivered, and cannot be rejected by the SMTP server during the SMTP transaction, they should be "bounced" (returned with non-delivery notification messages) as described above. In today's world, in which many SMTP server operators have discovered that the quantity of undesirable bulk email vastly exceeds the quantity of desired mail and in which accepting a message may trigger additional undesirable traffic by providing verification of the address, those principles may not be practical.

As discussed in Section 7.8 and Section 7.9 below, dropping mail without notification of the sender is permitted in practice. However, it is extremely dangerous and violates a long tradition and community expectations that mail is either delivered or returned. If silent message-dropping is misused, it could easily undermine confidence in the reliability of the Internet's mail systems. So silent dropping of messages should be considered only in those cases where there is very high confidence that the messages are seriously fraudulent, pose a significant risk, or are otherwise inappropriate.

To stretch the principle of delivery if possible even further, it may be a rational policy to not deliver mail that has an invalid return address, although the history of the network is that users are

typically better served by delivering any message that can be delivered. Reliably determining that a return address is invalid can be a difficult and time-consuming process, especially if the putative sending system is not directly accessible or does not fully and accurately support VRFY and, even if a "drop messages with invalid return addresses" policy is adopted, it SHOULD be applied only when there is near-certainty that the return addresses are, in fact, invalid.

Conversely, if a message is rejected because it is found to contain hostile content (a decision that is outside the scope of an SMTP server as defined in this document), rejection ("bounce") messages SHOULD NOT be sent unless the receiving site is confident that those messages will be usefully delivered. The preference and default in these cases is to avoid sending non-delivery messages when the incoming message is determined to contain hostile content.

6.3. Loop Detection

Simple counting of the number of "Received:" header fields in a message has proven to be an effective, although rarely optimal, method of detecting loops in mail systems. SMTP servers using this technique SHOULD use a large rejection threshold, normally at least 100 Received entries. Whatever mechanisms are used, servers MUST contain provisions for detecting and stopping trivial loops.

6.4. Compensating for Irregularities

Unfortunately, variations, creative interpretations, and outright violations of Internet mail protocols do occur; some would suggest that they occur quite frequently. The debate as to whether a well-behaved SMTP receiver or relay should reject a malformed message, attempt to pass it on unchanged, or attempt to repair it to increase the odds of successful delivery (or subsequent reply) began almost with the dawn of structured network mail and shows no signs of abating. Advocates of rejection claim that attempted repairs are rarely completely adequate and that rejection of bad messages is the only way to get the offending software repaired. Advocates of "repair" or "deliver no matter what" argue that users prefer that mail go through it if at all possible and that there are significant market pressures in that direction. In practice, these market pressures may be more important to particular vendors than strict conformance to the standards, regardless of the preference of the actual developers.

The problems associated with ill-formed messages were exacerbated by the introduction of the split-UA mail reading protocols (Post Office Protocol (POP) version 2 [18], Post Office Protocol (POP) version 3

[28], IMAP version 2 [23], and PCMAIL [22]). These protocols encouraged the use of SMTP as a posting (message submission) protocol, and SMTP servers as relay systems for these client hosts (which are often only intermittently connected to the Internet). Historically, many of those client machines lacked some of the mechanisms and information assumed by SMTP (and indeed, by the mail format protocol, RFC 822 [17]). Some could not keep adequate track of time; others had no concept of time zones; still others could not identify their own names or addresses; and, of course, none could satisfy the assumptions that underlay RFC 822's conception of authenticated addresses.

In response to these weak SMTP clients, many SMTP systems now complete messages that are delivered to them in incomplete or incorrect form. This strategy is generally considered appropriate when the server can identify or authenticate the client, and there are prior agreements between them. By contrast, there is at best great concern about fixes applied by a relay or delivery SMTP server that has little or no knowledge of the user or client machine. Many of these issues are addressed by using a separate protocol, such as that defined in RFC 6409 [48], for message submission, rather than using originating SMTP servers for that purpose.

The following changes to a message being processed MAY be applied when necessary by an originating SMTP server, or one used as the target of SMTP as an initial posting (message submission) protocol:

* Addition of a message-id field when none appears

* Addition of a date, time, or time zone when none appears

* Correction of addresses to proper FQDN format

The less information the server has about the client, the less likely these changes are to be correct and the more caution and conservatism should be applied when considering whether or not to perform fixes and how. These changes MUST NOT be applied by an SMTP server that provides an intermediate relay function.

In all cases, properly operating clients supplying correct information are preferred to corrections by the SMTP server. In all cases, documentation SHOULD be provided in trace header fields and/or header field comments for actions performed by the servers.

7. Security Considerations

7.1. Mail Security and Spoofing

SMTP mail is inherently insecure in that it is feasible for even fairly casual users to negotiate directly with receiving and relaying SMTP servers and create messages that will trick a naive recipient into believing that they came from somewhere else. Constructing such a message so that the "spoofed" behavior cannot be detected by an expert is somewhat more difficult, but not sufficiently so as to be a deterrent to someone who is determined and knowledgeable. Consequently, as knowledge of Internet mail increases, so does the knowledge that SMTP mail inherently cannot be authenticated, or integrity checks provided, at the transport level. Real mail security lies only in end-to-end methods involving the message bodies, such as those that use digital signatures (see RFC 1847 [26] and, e.g., Pretty Good Privacy (PGP) in RFC 4880 [49] or Secure/Multipurpose Internet Mail Extensions (S/MIME) in RFC 8551 [45]).

Various protocol extensions and configuration options that provide authentication at the transport level (e.g., from an SMTP client to an SMTP server) improve somewhat on the traditional situation described above. However, in general, they only authenticate one server to another rather than a chain of relays and servers, much less authenticating users or user machines. Consequently, unless they are accompanied by careful handoffs of responsibility in a carefully designed trust environment, they remain inherently weaker than end-to-end mechanisms that use digitally signed messages rather than depending on the integrity of the transport system.

Efforts to make it more difficult for users to set envelope return path and header "From" fields to point to valid addresses other than their own are largely misguided: they frustrate legitimate applications in which mail is sent by one user on behalf of another, in which error (or normal) replies should be directed to a special address, or in which a single message is sent to multiple recipients on different hosts. (Systems that provide convenient ways for users to alter these header fields on a per-message basis should attempt to establish a primary and permanent mailbox address for the user so that Sender header fields within the message data can be generated sensibly.)

This specification does not further address the authentication issues associated with SMTP other than to advocate that useful functionality not be disabled in the hope of providing some small margin of protection against a user who is trying to fake mail.

7.2. Hiding Addresses from Trace

Addresses that do not appear in the message header section may appear in the RCPT commands to an SMTP server for a number of reasons. The two most common involve the use of a mailing address as a "list exploder" (a single address that resolves into multiple addresses) and the appearance of "blind copies". When more than one RCPT command is present, and in order to avoid defeating some of the purpose of these mechanisms, SMTP clients and servers SHOULD NOT copy the RCPT command arguments into the header section, either as part of trace header fields or as informational or private-extension header fields. See Section 7.6 for discussion of some related issues.

There is no inherent relationship between either "reverse" (from the MAIL command) or "forward" (RCPT) addresses in the SMTP transaction ("envelope") and the addresses in the header section. Receiving systems SHOULD NOT attempt to deduce such relationships and use them to alter the header section of the message for delivery. The popular "Apparently-to" header field is a violation of this principle as well as a common source of unintended information disclosure and SHOULD NOT be used.

7.3. VRFY, EXPN, and Security

As discussed in Section 3.5, individual sites may want to disable either or both of VRFY or EXPN for security reasons (see below). As a corollary to the above, implementations that permit this MUST NOT appear to have verified addresses that are not, in fact, verified. If a site disables these commands for security reasons, the SMTP server MUST return a 252 response, rather than a code that could be confused with successful or unsuccessful verification.

Returning a 250 reply code with the address listed in the VRFY command after having checked it only for syntax violates this rule. Of course, an implementation that "supports" VRFY by always returning 550 whether or not the address is valid is equally not in conformance.

On the public Internet, the contents of mailing lists have become popular as an address information source for so-called "spammers." The use of EXPN to "harvest" addresses has increased as list administrators have installed protections against inappropriate uses of the lists themselves. However, VRFY and EXPN are still useful for authenticated users and within an administrative domain. For example, VRFY and EXPN are useful for performing internal audits of how email gets routed to check and to make sure no one is automatically forwarding sensitive mail outside the organization. Sites implementing SMTP authentication may choose to make VRFY and

EXPN available only to authenticated requestors. Implementations SHOULD still provide support for EXPN, but sites SHOULD carefully evaluate the tradeoffs.

Whether disabling VRFY provides any real marginal security depends on a series of other conditions. In many cases, RCPT commands can be used to obtain the same information about address validity. On the other hand, especially in situations where determination of address validity for RCPT commands is deferred until after the DATA command is received, RCPT may return no information at all, while VRFY is expected to make a serious attempt to determine validity before generating a response code (see discussion above).

7.4. Mail Rerouting Based on the 251 and 551 Response Codes

Before a client uses the 251 or 551 reply codes from a RCPT command to automatically update its future behavior (e.g., updating the user's address book), it should be certain of the server's authenticity. If it does not, it may be subject to a man in the middle attack.

7.5. Information Disclosure in Announcements

There has been an ongoing debate about the tradeoffs between the debugging advantages of announcing server type and version (and, sometimes, even server domain name) in the greeting response or in response to the HELP command and the disadvantages of exposing information that might be useful in a potential hostile attack. The utility of the debugging information is beyond doubt. Those who argue for making it available point out that it is far better to actually secure an SMTP server rather than hope that trying to conceal known vulnerabilities by hiding the server's precise identity will provide more protection. Sites are encouraged to evaluate the tradeoff with that issue in mind; implementations SHOULD minimally provide for making type and version information available in some way to other network hosts.

7.6. Information Disclosure in Trace Fields

In some circumstances, such as when mail originates from within a LAN whose hosts are not directly on the public Internet, trace (e.g., "Received") header fields produced in conformance with this specification may disclose host names and similar information that would not normally be available. This ordinarily does not pose a problem, but sites with special concerns about name disclosure should be aware of it. Also, the optional FOR clause should not be supplied when the same message is sent to multiple recipients in the same mail transaction in order not to inadvertently disclose the identities of

"blind copy" recipients to others.

7.7. Information Disclosure in Message Forwarding

As discussed in Section 3.4.1, use of the 251 or 551 reply codes to identify the replacement address associated with a mailbox may inadvertently disclose sensitive information. Sites that are concerned about those issues should ensure that they select and configure servers appropriately.

7.8. Local Operational Requirements and Resistance to Attacks

In recent years, there has been an increase of attacks on SMTP servers, either in conjunction with attempts to discover addresses for sending unsolicited messages or simply to make the servers inaccessible to others (i.e., as an application-level denial of service attack). There may also be important local circumstances that justify departures from some of the limits specified in this documents especially ones involving maximums or minimums. While the means of doing so are beyond the scope of this Standard, rational operational behavior requires that servers be permitted to detect such attacks and take action to defend themselves. For example, if a server determines that a large number of RCPT commands are being sent, most or all with invalid addresses, as part of such an attack, it would be reasonable for the server to close the connection after generating an appropriate number of 5yz (normally 550) replies.

7.9. Scope of Operation of SMTP Servers

It is a well-established principle that an SMTP server may refuse to accept mail for any operational or technical reason that makes sense to the site providing the server. However, cooperation among sites and installations makes the Internet possible. If sites take excessive advantage of the right to reject traffic, the ubiquity of email availability (one of the strengths of the Internet) will be threatened; considerable care should be taken and balance maintained if a site decides to be selective about the traffic it will accept and process.

Relay function through arbitrary sites, as part of hostile efforts to hide the actual origins of mail, has become so common that most sites limit the use of the relay function to known or identifiable sources.

Implementations SHOULD provide the capability to perform this type of analysis of message sources and potential message rejection as a result. When mail is rejected for these or other policy reasons, a 550 code SHOULD be used in response to EHLO (or HELO), MAIL, or RCPT as appropriate.

8. IANA Considerations

8.1. SMTP-related Registries

IANA maintains five registries in support of this specification, the first two of which were created for RFC 2821 or earlier. The additional ones were defined or expanded by RFC 5321 or added subsequently. The registry references listed are as of the time of publication; IANA does not guarantee the locations associated with the URLs in those references. The subsections that follow describe the general purpose and intent for those registries and substantive modifications to existing ones. Information about actual registry structure requirements appears in Section 8.3.

8.1.1. Simple Mail Transfer Protocol (SMTP) Service Extensions

8.1.1.1. Registration Models

In order to accommodate both a significant review of proposed extensions for those who find that useful and a minimally restrictive registration procedure for those who simply want to avoid name conflicts and similar problems, this registry supports two different registration models.

The first, "Simple Mail Transfer Protocol (SMTP) Service Extensions" [57], often referred to in this document as [the] "Service Extension Registry", consists of SMTP service extensions with the associated keywords, and, as needed, parameters, verbs, and related information.

As noted in Section 2.2.2, SMTP Extensions MUST be registered.

```
// Note in draft: It is assumed that much of the following will be
// replaced by either a normative reference to draft-klensin-iana-
// consid-hybrid or by one to RFC8126bis if either is approved or
// sufficiently close to approval before this document is ready for
// publication.
```

The would-be registrant shall pick between the two models described below although, if the first is attempted and proves unsuccessful, the second may then be chosen:

1. IETF Review and Approval

   The document goes through the normal IETF review and approval process, culminating in a published Standards Track, BCP, Experimental, or, in rare cases specifically approved by the IESG, an IETF Stream Informational RFC. The intent is that the

extension and its specification will represent careful IETF community review and consensus on its technical merits, utility, and clarity of explanation. The change controller for all such extensions will be the IETF.

This model is approximately equivalent to "IETF Review" as described in RFC 8126/ BCP 26 [3], but involves a stronger preference for a Standards Track or Experimental publication as a result.

// NOTE IN DRAFT: Section numbers from RFC 8126 have been
// deliberately omitted here and below in the expectation (or at
// least hope) that document will be replaced or updated to
contain
// an explicit "two alternatives" model like this one before or
soon
// after this document is published.

2. Simple Registration

The sole purpose of this option is to get the extension name and contact information registered in order to minimize the risk of the same extension name being used for different purposes. The intent is that there be no barrier to such registrations other than the time and effort required to submit the request itself. Registrants are encouraged to provide documentation of the extension, its interactions with other specifications, etc., and to consult individuals or groups with SMTP experience for advice, but none of that is required. The change controller for all such extensions will be the registrant unless otherwise specified in the registration request.

Even if this model is chosen, it is expected that registrants will supply all of the information in the list below and as described above and in Section 2.2.2 as either part of the registration or in supplemental documents that will be referenced from the registry. However, the primary goals of getting extensions registered according to this model are to avoid conflicts about naming (e.g., two different deployed extensions with the same name or keyword) and to either identify a stable and generally available specification or to establish contact information for additional information. Consequently, if no information is available for some of the listed items, notably Section 8.1.1.2, Paragraph 2, Item 9 and Section 8.1.1.2, Paragraph 2, Item 10, the registry entry should be made with the absence of such data noted in the registry as "Not supplied".

This model is approximately equivalent to "First Come First Served" as described in RFC 8126/ BCP 26. [3]

IANA will, based on their own judgment, identify in the registry which model was chosen by a column entry or will create a pair of subregistries, one for each model.

8.1.1.2. Registry Information

The following information shall be supplied as part of a registration application and will be incorporated into the registry.

(1) the textual name of the SMTP service extension;

(2) a summary of the purpose of the extension and what it is expected to accomplish.

(3) the EHLO keyword value associated with the extension;

(4) the syntax and possible values of parameters associated with the EHLO keyword value;

(5) any additional SMTP verbs associated with the extension (additional verbs will usually be, but are not required to be, the same as, or begin with, the EHLO keyword value);

(6) any new parameters the extension associates with the MAIL or RCPT verbs;

(7) Level of approval. Values: Standards Track, IETF Experimental, Individual (or reasonable abbreviations).

(8) Message submission use. Values: Keyword indicating relevance for use in message submission as described in Section 7 of RFC 6409 [48]. For any registration prior to the publication of <<This document>> for which this information was not specified in RFC 6409 or the registration request, this entry in the registry will be set to "MUST NOT".

(9) a description of how support for the extension affects the behavior of a server and client SMTP;

(10) the increment by which the extension is increasing the maximum length of the commands MAIL and/or RCPT over that specified in this Standard or other registrations that might reasonably be expected to interact with it; and

(11) Contact information for the submitter or other responsible party and identification of the change controller. The change controller value MUST be "IETF" for all SMTP extensions specified in IETF Stream RFCs and MUST provide appropriate authority and contact information for all other extensions.

8.1.2. Address Literal Tags

The second registry, "Address Literal Tags" [62], consists of "tags" that identify forms of domain literals other than those for IPv4 addresses (specified in RFC 821 and in this document). The initial entry in that registry is for IPv6 addresses (specified in this document). Additional literal types require standardization (IETF "Standards Action") before being used; none are anticipated at this time.

8.1.3. Mail Transmission Types

The third, "Mail Transmission Types" [58], established by RFC 821 and renewed by this specification, is a registry of link and protocol identifiers to be used with the "via" and "with" subclauses of the time stamp ("Received:" header field) described in Section 4.4. Link and protocol identifiers in addition to those specified in this document may be registered only according to the "RFC Required" procedure described in RFC 8126/ BCP 26 [3].

An additional subregistry has been added to the "VIA link types" and "WITH protocol types" subregistries of this registry to contain registrations of "Additional-registered-clauses" as described above and in the subsection that follows.

8.1.4. Additional Registered Clauses

As mentioned in Section 4.4.4 above, additional clauses for the "Received:" header field may be added by future specifications (details below). IANA has created a registry for such clauses [60]. The registry will contain the Clause Name; the name of, and pointer to, any associated enabling Service Extension; a short description; a syntax summary; and a reference to a more complete specification.

As new clauses are defined, they may, in principle, specify creation of their own registries if the Strings consist of reserved terms or keywords rather than less restricted strings. Additional clauses, may be registered only by the "IETF Review" procedure described in RFC 8126/ BCP 26 [3].

8.2. New Registry Actions with <<This Document>>

(1) [IANA is requested to] To improve clarity VRFY is added to the Service Extensions Registry [57] immediately before the entry for EXPN. The Note should read: "Implementation support for VRFY is required in servers but the listing in the EHLO response is optional". See Section 3.5.2 for details on this subject.

(2) For <<This Document>>, several changes have been made in Section 2.2.2 and above, particularly in Section 8.1, most as consequences from the change from requiring IETF approval for registration of new extensions.

8.3. Specification of Registry Group and Registry Structure

The Mail Parameters Registry Group should be reorganized as follows.

(1) The registry for Address Literals [62] should be consolidated into the Mail Parameters Registry Group. Those literals are specified only for email purposes and have no established meaning elsewhere.

(2) Please insert a category for "Associated registries located elsewhere" after the "Registries included below" group at the top of the MAIL Parameters Group page. There should be one entry in the new category: the name "Message Headers" and a link to the appropriate registry [61]. Also, consider renaming that registry group: there is no obvious reason to capitalize "MAIL", all the the registries and subregistries in the group are SMTP-related, and the most important of them is not about parameters at all. So, perhaps, "SMTP-related Registries" or some such thing. If that is done, "MAIL Parameters" should probably be retained as a comment or parenthetical note because it is almost certainly referenced from other documents.

(3) The registry referred to in this document as "Simple Mail Transfer Protocol (SMTP) Service Extensions" or "Service Extensions Registry" is currently called "SMTP Service Extensions" [57]. That name is acceptable, but should be reviewed by IANA to see if a different name would be more appropriate.

(4) In the header of the Service Extensions Registry, the reference to RFC 5321 Section 2.2 should be changed to point to Section 8.1.1 and Section 2.2.2 of this document in that order. If "SMTP Service Extension Parameters" is maintained as a subregistry (see Section 8.3, Paragraph 2, Item 8 below), the same fix should be applied there and supplemented by a pointer to the Service Extension Registry itself.

(5) The description in the Service Extension Registry should make it explicit that any of the descriptions called for in the bullet items listed in Section 8.1.1 except for the first six and Section 8.1.1.2, Paragraph 2, Item 11 may be supplied by reference to a stable and easily available external document.

(6) Those bullet point fields should be separated; i.e., none of them should be combined into "Description". Information that is not supplied with the registration or supplemental documents should be explicitly identified as "Not supplied by submitter" or with an explicit note pointing to that phrase.

(7) The "Registration Procedure(s)" for the Service Extensions Registry and, if it is retained, the value for "SMTP Service Extension Parameters" subregistry should be changed to "Specification Required".

(8) The fields in the Service Extension Registry for keyword parameter values (Section 8.1.1.2, Paragraph 2, Item 4) and MAIL or RCPT parameter values (Section 8.1.1.2, Paragraph 2, Item 6) should either contain those values or, if IANA prefers to retain separate subregistries, explicit pointers to the associated entries in those subregistries (not just a pointer to the subregistry itself). In the latter cases, the header for the subregistries should point back to the associated numbered item in the Service Extension Registry.

(9) Another subregistry question arises with the MT-PRIORITY extension. The authors of RFC 6710 (one of whom is obviously quite handy) should review how that extension is handled, including whether the use of "MT-PRIORITY" in some places as "PRIORITY" in others is confusing. At present, it is listed as a Service Extension, in the Extension Parameters list, and with its own separate registry in the Mail Parameters registry group [59] [56]. At least that registry should be shown as a sub-subregistry of the Service Extensions Registry and/or carefully cross-referenced.

(10) In the "Mail Transmission Types" registry [58], the first part of the "Note" should read "The Simple Mail Transfer Protocol <<This Document>> and the Standard for Internet Message Formats [16] specify that...". The Reference in that registry should be to Section 8.1.3 and [16].

9. Acknowledgments

Many people contributed to the development of RFCs 2821 and 5321. Those documents should be consulted for those acknowledgments.

Neither this document nor RFCs 2821 or 5321 would have been possible without the many contribution and insights of the late Jon Postel and Ned Freed. Jon Postel's contributions of course include the original specification of SMTP in RFC 821. A considerable quantity of text from RFC 821 still appears in this document as do several of Jon's original examples that have been updated only as needed to reflect other changes in the specification. Ned Freed's many contributions from multiple perspectives, as author or co-author of several of the documents that were folded into this one, and an extremely careful reader who identified and proposed corrections to problems that others missed, were similarly invaluable.

The following filed errata against RFC 5321 that were not rejected at the time of submission: Jasen Betts, Adrien de Croy, Guillaume Fortin-Debigare, Roberto Javier Godoy, David Romerstein, Dominic Sayers, Rodrigo Speller, Alessandro Vesely, and Brett Watson. Some of those individuals made additional suggestions after the EMAILCORE WG was initiated. In addition to the above, several of whom continued to make other suggestions, specific suggestions that led to corrections and improvements in early versions of the current specification were received from Dave Crocker, Ned Freed, Arnt Gulbrandsen, Tony Hansen, Barry Leiba, Ivar Lumi, Pete Resnick, Hector Santos, Paul Smith and others.

chetti contributed an analysis that clarified the ABNF productions that implicitly reference other documents.

The EMAILCORE Working Group was chartered in September 2020 with Alexey Melnikov and Seth Blank as co-chairs. Todd Herr replaced Seth Blank early in 2021. Without their leadership and technical contributions, and the efforts of WG participants under their guidance, this document would never have been completed. Many participants in the WG reviewed the document or portions of it and made comments that resulted in improvement. During the last stages of working group consideration of this document, careful reviews of the specification in its entirety by Alexey Melnikov, John Levine, Rob Sayre, and Tim Wicinski contributed significantly to the clarity of the final version and its relationship to other IETF work.

10. References

10.1. Normative References

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

[2] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[3] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[4] ANSI, "USA Code for Information Interchange", ANSI X3.4-1968, 1968. ANSI X3.4-1968 has been replaced by newer versions with slight modifications, but the 1968 version remains definitive for the Internet. The 1968 version is also described for Internet purposes in RFC 20 [5].

[5] Cerf, V., "ASCII format for network interchange", STD 80, RFC 20, DOI 10.17487/RFC0020, October 1969, <https://www.rfc-editor.org/info/rfc20>.

[6] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, DOI 10.17487/RFC0821, August 1982, <https://www.rfc-editor.org/info/rfc821>.

[7] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[8] Mockapetris, P., "Domain names - implementation and specification", Section 2.3.1, RFC 1035, November 1987, <https://www.rfc-editor.org/rfc/rfc1035>. // RFC Editor: If this explanation is not gone before you get this // document, please remove it. After a rather detailed analysis, // there is no standards track document that explicitly updates RFC // 1035 (or 1034) to allow leading digits in "preferred syntax" // domain names. What we have is the "situation described in the I-D // (reference [8] of draft- ietf-emailcore-rfc5321bis-24) in which // RFC 1123 allows leading digits in "host names" but does not do so // in the section that updates the DNS specs. As pointed out in RFC // 8499, "host name" has been used and misused for several different // things, including as only the first (leftmost) label in a fully- // qualified domain name. The discussion of the "host name" // terminology in that RFC has been read to settle the issue by // equating "host name" syntax to "preferred name syntax", but it // does not update 1123 and, equally important, uses "often meant" to // describe the

relationship between the two terms and, in the // subsequent paragraph and as mentioned above, talks about the other // things that "host name" might mean (including only the first label // "of a fully-qualified domain name). The text in the reference in // the document is an attempt to explain this situation without // making claims about, e.g., document updates that are not supported // by RFC metadata or the structure of RFC 1123. The adventure of // trying to figure the above out turned up another possible issue or // two. This document references RFC 1035 while rfc5322bis uses a // reference group and references STD13. We cannot use STD13 in // several of the places 1035 is referenced from rfc5321bis because // we call out section numbers and, while the "preferred name syntax" // is defined in both RFC 1034 and RFC 1035 (please don't ask me why // -- I don't have a clue), the section numbers are different. If // anyone has useful thoughts about what we should do about that, do // say something on-list. This section of RFC 1035 defined the "preferred name syntax" as excluding leading digits in those names. Whether the restriction was accidental or deliberate, at least one second-level domain name starting with a digit had appeared in the DNS by the end of 1986, almost a year before RFC 1035 appeared. The restriction was removed for "host names" in RFC 1123 [10]. The terminology description for the DNS [9] clarified that "host name" was "often meant to be a domain name that follows the rules... of 'preferred name syntax'". So the syntax for "domain name" in Section 2.3.5 above is consistent with the DNS specifications as generally interpreted.

[9] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <https://www.rfc-editor.org/info/rfc8499>.

[10] Braden, R., Ed., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, DOI 10.17487/RFC1123, October 1989, <https://www.rfc-editor.org/info/rfc1123>.

[11] Klensin, J., Freed, N., and K. Moore, "SMTP Service Extension for Message Size Declaration", STD 10, RFC 1870, DOI 10.17487/RFC1870, November 1995, <https://www.rfc-editor.org/info/rfc1870>.

[12] Vaudreuil, G., "Enhanced Mail System Status Codes", RFC 3463, DOI 10.17487/RFC3463, January 2003, <https://www.rfc-editor.org/info/rfc3463>.

[13] Newman, C., "ESMTP and LMTP Transmission Types Registration", RFC 3848, DOI 10.17487/RFC3848, July 2004, <https://www.rfc-editor.org/info/rfc3848>.

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

[15] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[16] Resnick, P., "Internet Message Format", 2024, <https://datatracker.ietf.org/doc/draft-ietf-emailcore-rfc5322bis/>. Note to RFC Editor and WG: This reference, and citations to it (including mentions of "RFC 5322bis", "RFC5322bis", and other variations) should be updated to the correct RFC number and other information when work on rfc5321bis (this document) and rfc5322bis is complete. All references to the original RFC 5322 has been removed from this specification. Once that update has been accomplished, this note should be removed.

10.2. Informative References

[17] Crocker, D., "STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES", STD 11, RFC 822, DOI 10.17487/RFC0822, August 1982, <https://www.rfc-editor.org/info/rfc822>.

[18] Butler, M., Postel, J., Chase, D., Goldberger, J., and J. Reynolds, "Post Office Protocol: Version 2", RFC 937, DOI 10.17487/RFC0937, February 1985, <https://www.rfc-editor.org/info/rfc937>.

[19] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, DOI 10.17487/RFC0959, October 1985, <https://www.rfc-editor.org/info/rfc959>.

[20] Partridge, C., "Mail routing and the domain system", STD 10, RFC 974, DOI 10.17487/RFC0974, January 1986, <https://www.rfc-editor.org/info/rfc974>.

[21] Partridge, C., "Duplicate messages and SMTP", RFC 1047, DOI 10.17487/RFC1047, February 1988, <https://www.rfc-editor.org/info/rfc1047>.

[22] Lambert, M., "PCMAIL: A distributed mail system for personal computers", RFC 1056, DOI 10.17487/RFC1056, June 1988, <https://www.rfc-editor.org/info/rfc1056>.

[23] Crispin, M., "Interactive Mail Access Protocol: Version 2", RFC 1176, DOI 10.17487/RFC1176, August 1990, <https://www.rfc-editor.org/info/rfc1176>.

[24] Klensin, J., Freed, N., Ed., Rose, M., Stefferud, E., and D. Crocker, "SMTP Service Extensions", RFC 1425, DOI 10.17487/RFC1425, February 1993, <https://www.rfc-editor.org/info/rfc1425>.

[25] Durand, A. and F. Dupont, "SMTP 521 Reply Code", RFC 1846, DOI 10.17487/RFC1846, September 1995, <https://www.rfc-editor.org/info/rfc1846>.

[26] Galvin, J., Murphy, S., Crocker, S., and N. Freed, "Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted", RFC 1847, DOI 10.17487/RFC1847, October 1995, <https://www.rfc-editor.org/info/rfc1847>.

[27] Klensin, J., Freed, N., Rose, M., Stefferud, E., and D. Crocker, "SMTP Service Extensions", STD 10, RFC 1869, DOI 10.17487/RFC1869, November 1995, <https://www.rfc-editor.org/info/rfc1869>.

[28] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, DOI 10.17487/RFC1939, May 1996, <https://www.rfc-editor.org/info/rfc1939>.

[29] De Winter, J., "SMTP Service Extension for Remote Message Queue Starting", RFC 1985, DOI 10.17487/RFC1985, August 1996, <https://www.rfc-editor.org/info/rfc1985>.

[30] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, DOI 10.17487/RFC2045, November 1996, <https://www.rfc-editor.org/info/rfc2045>.

[31] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, DOI 10.17487/RFC2047, November 1996, <https://www.rfc-editor.org/info/rfc2047>.

[32] Kille, S., "MIXER (Mime Internet X.400 Enhanced Relay): Mapping between X.400 and RFC 822/MIME", RFC 2156, DOI 10.17487/RFC2156, January 1998, <https://www.rfc-editor.org/info/rfc2156>.

[33] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997, <https://www.rfc-editor.org/info/rfc2181>.

[34] Freed, N. and K. Moore, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", RFC 2231, DOI 10.17487/RFC2231, November 1997, <https://www.rfc-editor.org/info/rfc2231>.

[35] Klensin, J., Ed., "Simple Mail Transfer Protocol", RFC 2821, DOI 10.17487/RFC2821, April 2001, <https://www.rfc-editor.org/info/rfc2821>.

[36] Freed, N., "SMTP Service Extension for Command Pipelining", STD 60, RFC 2920, DOI 10.17487/RFC2920, September 2000, <https://www.rfc-editor.org/info/rfc2920>.

[37] Freed, N., "Behavior of and Requirements for Internet Firewalls", RFC 2979, DOI 10.17487/RFC2979, October 2000, <https://www.rfc-editor.org/info/rfc2979>.

[38] Vaudreuil, G., "SMTP Service Extensions for Transmission of Large and Binary MIME Messages", RFC 3030, DOI 10.17487/RFC3030, December 2000, <https://www.rfc-editor.org/info/rfc3030>.

[39] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", RFC 3461, DOI 10.17487/RFC3461, January 2003, <https://www.rfc-editor.org/info/rfc3461>.

[40] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 3464, DOI 10.17487/RFC3464, January 2003, <https://www.rfc-editor.org/info/rfc3464>.

[41] Melnikov, A., Ed. and B. Leiba, Ed., "Internet Message Access Protocol (IMAP) - Version 4rev2", RFC 9051, DOI 10.17487/RFC9051, August 2021, <https://www.rfc-editor.org/info/rfc9051>.

[42] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, DOI 10.17487/RFC6530, February 2012, <https://www.rfc-editor.org/info/rfc6530>.

[43] Yao, J. and W. Mao, "SMTP Extension for Internationalized Email", RFC 6531, DOI 10.17487/RFC6531, February 2012, <https://www.rfc-editor.org/info/rfc6531>.

[44] Hansen, T., Ed. and A. Melnikov, Ed., "Message Disposition Notification", STD 85, RFC 8098, DOI 10.17487/RFC8098, February 2017, <https://www.rfc-editor.org/info/rfc8098>.

[45] Schaad, J., Ramsdell, B., and S. Turner, "Secure/ Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification", RFC 8551, DOI 10.17487/RFC8551, April 2019, <https://www.rfc-editor.org/info/rfc8551>.

[46] Nakamura, M. and J. Hagino, "SMTP Operational Experience in Mixed IPv4/v6 Environments", RFC 3974, DOI 10.17487/RFC3974, January 2005, <https://www.rfc-editor.org/info/rfc3974>.

[47] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[48] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, DOI 10.17487/RFC6409, November 2011, <https://www.rfc-editor.org/info/rfc6409>.

[49] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, DOI 10.17487/RFC4880, November 2007, <https://www.rfc-editor.org/info/rfc4880>.

[50] Hansen, T. and J. Klensin, "A Registry for SMTP Enhanced Mail System Status Codes", BCP 138, RFC 5248, DOI 10.17487/RFC5248, June 2008, <https://www.rfc-editor.org/info/rfc5248>.

[51] Klensin, J., Freed, N., Rose, M., and D. Crocker, Ed., "SMTP Service Extension for 8-bit MIME Transport", STD 71, RFC 6152, DOI 10.17487/RFC6152, March 2011, <https://www.rfc-editor.org/info/rfc6152>.

[52] Klensin, J., "SMTP 521 and 556 Reply Codes", RFC 7504, DOI 10.17487/RFC7504, June 2015, <https://www.rfc-editor.org/info/rfc7504>.

[53] Levine, J. and M. Delany, "A "Null MX" No Service Resource Record for Domains That Accept No Mail", RFC 7505, DOI 10.17487/RFC7505, June 2015, <https://www.rfc-editor.org/info/rfc7505>.

[54] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <https://www.rfc-editor.org/info/rfc5321>.

[55] Klensin, J.C., Ed., Murchison, K., Ed., and E. Sam, Ed., "Applicability Statement for IETF Core Email Protocols", 6 August 2021, <https://datatracker.ietf.org/doc/draft-ietf-emailcore-as/>.

[56] Internet Assigned Number Authority (IANA), "Mail Parameters", 2022, <https://www.iana.org/assignments/mail-parameters/mail-parameters.xhtml>.

[57] Internet Assigned Number Authority (IANA), "IANA Mail Parameters: SMTP Service Extensions", 2022, <https://www.iana.org/assignments/mail-parameters/mail-parameters.xhtml#mail-parameters-2>.

[58] Internet Assigned Number Authority (IANA), "IANA Mail Parameters: Mail Transmission Types", 2022, <https://www.iana.org/assignments/mail-parameters/mail-parameters.xhtml#mail-parameters-5>.

[59] Internet Assigned Number Authority (IANA), "IANA Mail Parameters: SMTP PRIORITY extension Priority Assignment Policy", 2022, <https://www.iana.org/assignments/mail-parameters/mail-parameters.xhtml#mail-parameters-9>.

[60] Internet Assigned Number Authority (IANA), "IANA Mail Parameters: Additional-registered-clauses", 2022, <https://www.iana.org/assignments/mail-parameters/mail-parameters.xhtml#mail-parameters-8>.

[61] Internet Assigned Number Authority (IANA), "Message Headers", 2022, <https://www.iana.org/assignments/message-headers/message-headers.xhtml>.

[62] Internet Assigned Number Authority (IANA), "Address Literal Tags", 2007, <http://www.iana.org/assignments/address-literal-tags>.

[63] RFC Editor, "RFC Errata - RFC 5321", 2019, <https://www.rfc-editor.org/errata/rfc5321>. Captured 2019-11-19

[64] IANA, "SMTP Service Extensions", 2021, <https://www.iana.org/assignments/mail-parameters/mail-parameters.xhtml#mail-parameters-2>. Notes in draft: RFC Editor: Please adjust date field to reflect whatever you want for a registry that is updated periodically. IANA: Please determine if the above URL is a sufficiently stable reference and adjust as appropriate if it is not.

[65] RFC Editor, "RFC Errata: RFC 5321", 2022, <https://www.rfc-editor.org/errata/rfc5321>. Captured 2022-11-23

Appendix A. TCP Transport Service

The TCP connection supports the transmission of 8-bit bytes. The SMTP data is 7-bit ASCII characters. Each character is transmitted as an 8-bit byte with the high-order bit cleared to zero. Service extensions may modify this rule to permit transmission of full 8-bit data bytes as part of the message body, or, if specifically designed to do so, in SMTP commands or responses.

Appendix B. Generating SMTP Commands from Internet Message Format Header Fields

Under ideal circumstances, SMTP servers as specified in this document would receive complete information, including proper envelope information, either from prior SMTP clients or from Message Submission systems that conform to RFC 6409. SMTP servers that receive complete information would interact with the message body only by prepending trace information as discussed in Section 4.4 above.

On the other hand, there are systems in use that do not provide the complete information in the expected form. Some are "gateways" as described in Section 2.3.10 (possibly including the special case of firewalls) and Section 3.7. Some of those systems use an Internet Message Format [16] header section (or something similar without other information) as a substitute for a mail submission protocol that conforms to RFC 6409 [48] or otherwise require that SMTP-receivers make up commands from information in what SMTP considers

the message body before such a message is transmitted to the next system by the corresponding SMTP-sender. This Appendix discusses some of the issues and appropriate actions when those situations are encountered.

Nothing in this appendix, or elsewhere in this specification, encourages or allows an SMTP-receiver to alter message headers that are compliant with Internet Message Format specifications before the message is passed on to another system by the corresponding SMTP-sender. When such messages are, or appear to be, compliant in that way, the message content is to be altered only by the addition, at the beginning of the content, of trace fields as specified in Section 4.4.

While direct communication between a MUA and MTA (rather than through a Submission Server [48]) is a private matter, not covered by any Internet Standard, there are problems with this approach. For example, there have been repeated problems with proper handling of "bcc" copies and redistribution lists when information that conceptually belongs to the mail envelope is not separated early in processing from header field information (and kept separate).

It is recommended that an MUA provide its initial ("submission client") MTA with an envelope separate from the message itself. However, if the envelope is not supplied, the envelope SHOULD be generated as follows:

1. Each recipient address from a TO, CC, or BCC header field SHOULD be copied to a RCPT command (generating multiple message copies if that is required for queuing or delivery). This includes any addresses listed in a RFC 822 "group". Any BCC header fields SHOULD then be removed from the header section.

2. The return address in the MAIL command SHOULD, if possible, be derived from the system's identity for the submitting (local) user, and the "From:" header field otherwise. If there is a system identity available, it SHOULD also be copied to the Sender header field if it is different from the address in the From header field. (Any Sender header field that was already there SHOULD be removed.) Systems may provide a way for submitters to override the envelope return address, but may want to restrict its use to privileged users. This will not prevent mail forgery, but may lessen its incidence; see Section 7.1.

When an MTA is being used in this way, it bears responsibility for ensuring that the message being transmitted is valid. The mechanisms for checking that validity, and for handling (or returning) messages that are not valid at the time of arrival, are part of the MUA-MTA interface and not covered by this specification.

A submission protocol based on Standard RFC 822 information alone MUST NOT be used to gateway a message from a foreign (non-SMTP) mail system into an SMTP environment. Additional information to construct an envelope must come from some source in the other environment, whether supplemental header fields or the foreign system's envelope.

Attempts to gateway messages using only their header "To" and "Cc" fields have repeatedly caused mail loops and other behavior adverse to the proper functioning of the Internet mail environment. These problems have been especially common when the message originates from an Internet mailing list and is distributed into the foreign environment using envelope information. When these messages are then processed by a header-section-only remailer, loops back to the Internet environment (and the mailing list) are almost inevitable.

Appendix C. Placeholder (formerly Source Routes)

// This entire section has been removed, with some material moved
// into Appendix F.2. This comment is retained as a temporary
// placeholder because the WG, the Ticket list, and various email
// threads refer to Appendix letters and it would not be good to
// create confusion about that while rfc5321bis is under development.

Appendix D. Scenarios

This section presents complete scenarios of several types of SMTP sessions. In the examples, "C:" indicates what is said by the SMTP client, and "S:" indicates what is said by the SMTP server.

D.1. A Typical SMTP Transaction Scenario

This SMTP example shows mail sent by Smith at host bar.com, and to Jones, Green, and Brown at host foo.com. Here we assume that host bar.com contacts host foo.com directly. The mail is accepted for Jones and Brown. Green does not have a mailbox at host foo.com.

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
```

```
S: 250-DSN
S: 250 HELP
C: MAIL FROM:<Smith@bar.com>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: RCPT TO:<Green@foo.com>
S: 550 No such user here
C: RCPT TO:<Brown@foo.com>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Blah blah blah...
C: ...etc. etc. etc.
C: .
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

D.2. Aborted SMTP Transaction Scenario

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
S: 250-DSN
S: 250 HELP
C: MAIL FROM:<Smith@bar.com>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: RCPT TO:<Green@foo.com>
S: 550 No such user here
C: RSET
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

D.3. Relayed Mail Scenario

Step 1 -- Source Host to Relay Host
The source host performs a DNS lookup on XYZ.COM (the destination address) and finds DNS MX records specifying xyz.com as the best preference and foo.com as a lower preference. It attempts to open a connection to xyz.com and fails. It then opens a connection to foo.com, with the following dialogue:

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
S: 250-DSN
S: 250 HELP
C: MAIL FROM:<JQP@bar.com>
S: 250 OK
C: RCPT TO:<Jones@XYZ.COM>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Date: Thu, 21 May 1998 05:33:29 -0700
C: From: John Q. Public <JQP@bar.com>
C: Subject:  The Next Meeting of the Board
C: To: Jones@xyz.com
C:
C: Bill:
C: The next meeting of the board of directors will be
C: on Tuesday.
C:                         John.
C: .
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

Step 2 -- Relay Host to Destination Host

foo.com, having received the message, now does a DNS lookup on xyz.com. It finds the same set of MX records, but cannot use the one that points to itself (or to any other host as a worse preference). It tries to open a connection to xyz.com itself and succeeds. Then we have:

```
S: 220 xyz.com Simple Mail Transfer Service Ready
C: EHLO foo.com
S: 250 xyz.com is on the air
C: MAIL FROM:<JQP@bar.com>
S: 250 OK
C: RCPT TO:<Jones@XYZ.COM>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Received: from bar.com by foo.com ; Thu, 21 May 1998
C:     05:33:29 -0700
C: Date: Thu, 21 May 1998 05:33:29 -0700
C: From: John Q. Public <JQP@bar.com>
C: Subject:  The Next Meeting of the Board
C: To: Jones@xyz.com
C:
C: Bill:
C: The next meeting of the board of directors will be
C: on Tuesday.
C:                         John.
C: .
S: 250 OK
C: QUIT
S: 221 xyz.com Service closing transmission channel
```

D.4. Verifying and Sending Scenario

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
S: 250-DSN
S: 250-VRFY
S: 250 HELP
C: VRFY Crispin
S: 250 Mark Crispin <Admin.MRC@foo.com>
C: MAIL FROM:<EAK@bar.com>
S: 250 OK
C: RCPT TO:<Admin.MRC@foo.com>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Blah blah blah...
C: ...etc. etc. etc.
C: .
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

Appendix E. Other Gateway Issues

In general, gateways between the Internet and other mail systems SHOULD attempt to preserve any layering semantics across the boundaries between the two mail systems involved. Gateway-translation approaches that attempt to take shortcuts by mapping (such as mapping envelope information from one system to the message header section or body of another) have generally proven to be inadequate in important ways. Systems translating between environments that do not support both envelopes and a header section and Internet mail must be written with the understanding that some information loss is almost inevitable.

Appendix F. Deprecated Features of RFC 821

A few features of RFC 821 have proven to be problematic and SHOULD NOT be used in Internet mail. Some of these features were deprecated in RFC 2821 in 2001; source routing and two-digit years in dates were deprecated even earlier, by RFC 1123 in 1989. Of the domain literal forms, RFC 1123 required support only for the dotted decimal form. With the possible exception of old, hardware-embedded, applications, there is no longer any excuse for these features to appear on the contemporary Internet.

F.1. TURN

This command, described in RFC 821, raises important security issues since, in the absence of strong authentication of the host requesting that the client and server switch roles, it can easily be used to divert mail from its correct destination. Its use is deprecated; SMTP systems SHOULD NOT use it unless the server can authenticate the client.

F.2. Source Routing

RFC 821 utilized the concept of explicit source routing to get mail from one host to another via a series of relays. Source routes could appear in either the <forward-path> or <reverse-path> to show the hosts through which mail would be routed to reach the destination. The requirement to utilize source routes in regular mail traffic was eliminated by the introduction of the domain name system "MX" record by RFC 974 in early 1986 and the last significant justification for them was eliminated by the introduction, in RFC 1123, of a clear requirement that addresses following an "@" must all be fully-qualified domain names. Issues involving local aliases for mailboxes were addressed by the introduction of a separate specification for mail submission [48]. Consequently, there are no remaining justifications for the use of source routes other than support for very old SMTP clients. Even use in mail system debugging is unlikely to work because almost all contemporary systems either ignore or reject them.

Historically, for relay purposes, the forward-path may have been a source route of the form "@ONE,@TWO:JOE@THREE", where ONE, TWO, and THREE MUST be fully-qualified domain names. This form was used to emphasize the distinction between an address and a route. The mailbox (here, JOE@THREE) is an absolute address, and the route is information about how to get there. The two concepts should not be confused.

SMTP servers MAY continue to accept source route syntax as specified in this appendix. If they do so, they SHOULD ignore the routes and utilize only the target domain in the address. If they do utilize the source route, the message MUST be sent to the first domain shown in the address.

In particular, a server MUST NOT guess at shortcuts within the source route.

SMTP clients MUST NOT attempt to utilize explicit source routing.

If source routes appear in mail received by an SMTP server contrary to the requirements and recommendations in this specification, RFC 821 and the text below should be consulted for the mechanisms for constructing and updating the forward-path. A server that is reached by means of a source route (e.g., its domain name appears first in the list in the forward-path) MUST remove its domain name from any forward-paths in which that domain name appears before forwarding the message and MAY remove all other source routing information. Any source route information in the reverse-path SHOULD be removed by servers conforming to this specification.

The following information is provided for historical information so that the source route syntax and application can be understood if needed.

Syntax:
The original form of the <Path> production in Section 4.1.2 was:

```
Path  = "<" [ A-d-l ":" ] Mailbox ">"

A-d-l  = At-domain *( "," At-domain )

At-domain  = "@" Domain
```

For example, suppose that a delivery service notification must be sent for a message that arrived with:
MAIL FROM:<@a.example,@b.example:user@d.example>
The notification message MUST be sent using:
RCPT TO:<user@d.example>

F.3. HELO

As discussed in Sections 3.1 and 4.1.1, EHLO SHOULD be used rather than HELO when the server will accept the former. Servers MUST continue to accept and process HELO in order to support older clients.

F.4. #-literals

RFC 821 provided for specifying an Internet address as a decimal integer host number prefixed by a pound sign, "#". In practice, that form has been obsolete since the introduction of TCP/IP. It is deprecated and MUST NOT be used.

F.5. Dates and Years

When dates are inserted into messages by SMTP clients or servers (e.g., in trace header fields), four-digit years MUST BE used. Two-digit years are deprecated; three-digit years were never permitted in the Internet mail system.

F.6. Sending versus Mailing

In addition to specifying a mechanism for delivering messages to user mailboxes, RFC 821 provided additional, optional, commands to deliver messages directly to the user's terminal screen. These commands (SEND, SAML, SOML) were rarely implemented, and changes in workstation technology and the introduction of other protocols may have rendered them obsolete even where they are implemented.

Clients SHOULD NOT use SEND, SAML, or SOML commands. If a server implements them, the implementation model specified in RFC 821 [6] MUST be used and the command names MUST be published in the response to the EHLO command.

Appendix G. Other Outstanding Issues

[[RFC Editor: Please remove this section before publication.]]

In December 2019, an issue was raised on the ietf-smtp@ietf.org list that led to a broad discussion of ways in which existing practice had diverged from the specifications and recommendations of RFC 5321 in the more than eleven years since it was published (some of those issues probably affect the boundary between RFC 5321 and 5322 and hence the latter as well). In most cases, those divergences call for revision of the Technical Specification to match the practice, clarification of the specification text in other ways, or a more comprehensive explanation of why the practices recommended by the specification should really be followed.

Those discussions raised two other issues, which were that

* The publication of the Submission Server specification of RFC 6409 in November 2011 may not have been fully reflected in RFC 5321 (despite the even earlier publication of RFC 4409) and

* There may be inconsistencies between the July 2009 Internet Mail Architecture description of RFC 5598 and the model described in RFC 5321. The issue called out in Appendix H.3 below may be an example of one of those inconsistencies.

Those discrepancies should be identified and discussed and decisions made to fix them (and where) or to ignore them and let them continue.

There has also been discussion on the mailing list, perhaps amounting to very rough consensus, that any revision of RFC 5321 and/or 5322 should be accompanied by a separate Applicability Statement document that would make recommendations about applicability or best practices in particular areas rather than trying to get everything into the two technical specifications. This appendix does not attempt to identify which issues should get which treatment.

This work is now (starting in the last half of 2020) being considered in the EMAILCORE WG. This appendix will act as a temporary record of issues that should be discussed and decided upon before a revised SMTP specification (or a related Applicability Statement) is published, issues that have not been reflected in errata (see Appendix I.1 below for those covered by errata).

Ticket numbers listed below and in the appendix that follows reference the list in <https://github.com/ietf-wg-emailcore/emailcore/issues/> (formerly <https://trac.ietf.org/trac/emailcore/report/1)>).

G.1. IP Address literals (closed)

Closed for RFC5321bis; issue for A/S. See Appendix H.1

G.2. Repeated Use of EHLO (closed)

Closed, see Appendix H.2

G.3. Meaning of "MTA" and Related Terminology (closed)

Done, note added -11, closed. Appendix H.3.

G.4. Originator, or Originating System, Authentication (closed)

This topic should be addressed in the A/S. See Appendix H.4.

G.5. Remove or deprecate the work-around from code 552 to 452 (closed)

Closed, see Appendix H.6.

G.6. Clarify where the protocol stands with respect to submission and TLS issues (Closed)

Reclassified to A/S issue, no ticket assigned, see Appendix H.28.

G.7. Probably-substantive Discussion Topics Identified in Other Ways (closed or OBE)

The following issues were identified as a group in the opening Note but called out specifically only in embedded CREF comments in versions of this draft prior to the first EMAILCORE version. In many cases, those CREF comments were removed after issues were closed. All issues in this category have either been closed, sometimes because they have been reassigned to other documents and/or or replaced with other topics and, where appropriate, tickets.

G.7.1. Issues with 521, 554, and 556 codes (closed)

Closed, see Appendix H.8. Note additional discussion started 2022-05-18 may require reopening or a new ticket, see Appendix H.30. There is also a loose end noted with a CREF comment in Section 3.1

G.7.2. SMTP Model, terminology, and relationship to RFC 5598 (closed)

Closed, see Appendix H.29.

G.7.3. Resolvable FQDNs and private domain names (closed)

Several tickets listed, all appear to be closed. See Appendix H.7.

G.7.4. Possible clarification about mail transactions and transaction state (closed)

Closed. See Appendix H.5).

G.7.5. Issues with mailing lists, aliases, and forwarding (closed)

Closed. See Appendix H.36.

G.7.6. Requirements for domain name and/or IP address in EHLO (closed)

RFC5321bis parts closed (see Appendix H.9). Some work still to be done in A/S.

G.7.7. Does the 'first digit only' and/or non-listed reply (c code text need clarification)? (closed)

Closed, see Appendix H.10.

G.7.8. Size limits (closed)

Closed, see Appendix H.11

G.7.9. Discussion of 'blind' copies and RCPT (closed)

Closed, see Appendix H.26.

G.7.10. Further clarifications needed to source routes? (closed)

Closed, see Appendix H.12

G.7.11. Should 1yz Be Revisited? (closed)

Closed, see Appendix H.13

G.7.12. Review Timeout Specifications (closed)

To be discussed in A/S, see Appendix H.14.

G.7.13. Possible SEND, SAML, SOML Loose End (closed)

Closed, see Appendix H.15

G.7.14. Abstract Update (closed)

Closed, see Appendix H.16.

G.7.15. Informative References to MIME and/or Message Submission (closed)

Closed, see Appendix H.17

G.7.16. Mail Transaction Discussion (closed)

Probably duplicate of Ticket #11, see Appendix H.5

G.7.17. Hop by hop Authentication and/or Encryption (closed)

Closed (no action required in rfc5321bis), see Appendix H.18

G.7.18. More Text About 554 Given 521, etc. (closed)

Closed, see Appendix H.30.

G.7.19. Minimum Lengths and Quantities (closed)

All work that needs to be done in this area in this document appears to have been done.
Closed, no ticket assigned, see Appendix H.31.

G.8. Enhanced Reply Codes and DSNs (closed)

This is part of Ticket #40 and most of the topic should be covered in the A/S. Treated as closed, see Appendix H.19.

G.9. Revisiting Quoted Strings (closed)

Closed. See Appendix H.27.

G.10. Internationalization (closed)

Text in the document has been changed to make use of non-ASCII characters with extension more explicit. Further discussion of these topics probably belongs in the A/S.
Closed, see Appendix H.32.

G.11. SMTP Clients, Servers, Senders, and Receivers (closed)

Text has been adjusted. Closed, see Appendix H.20.

G.12. Extension Keywords Starting in 'X-' (closed)

Closed, see Appendix H.21.

G.13. Deprecating HELO (closed)

Closed, see Appendix H.22.

G.14. The FOR Clause in Trace Fields: Semantics, Security Considerations, and Other Issues (closed)

Ticket #55, marked closed in issue tracker, 2023-02-13. See Appendix H.39.

G.15. Resistance to Attacks and Operational Necessity (closed)

Closed, see Appendix H.23

G.16. Mandatory 8BITMIME (closed)

Closed (despite Ticket #40 interation), See Appendix H.25.

G.17. New tickets created between 2022-01-21 and 2022-03-01 (closed (all of them))

To keep issues synchronized between this document and the tracker (now at <https://github.com/ietf-wg-emailcore/emailcore/issues?q=is%3Aopen+is%3Aissue>) a list of new issues added between the January 2022 interim and the end of the week before the cutoff for completing and posting draft-ietf-emailcore-rfc5321bis-10 are listed below.

// Editor's note: there are redundancies below in listed section
// numbers and references. They are to be sure pointers remain
// accurate if section numbers change. They look silly, but are not
// bugs. Because all of Appendixes G and H will disappear before the
// RFC Editor gets or starts processing the document, there is no
// possibility of them turning into long-term problems.

* #58 Clarification of what is a domain name alias and who can substitute them (Closed 2022-05-19, see Appendix H.24)
* #59 Case sensitive commands?
  Closed. See Appendix H.24)
* #60 Restricted-capability clients?
  Closed. See Appendix H.24.
* #61 Explaining mailing lists. Identified in tracker as duplicate of #12. See Appendix H.24 and Appendix H.36 for more information.
* #62 null mx vs server domain in 4.2.4.2 (See Section 4.2.4.2)
  Closed. See Appendix H.24.
* #63 VRFY in required commands in 4.5.1 (See Section 4.5.1).
  Changing this would also impact Section 3.5.1, Section 3.5.2, Section 3.5.3, and Section 7.3.
  Document adjusted per WG discussion through 2022-05-17.
  Closed. See Appendix H.24

G.18. Approval Required to Register an SMTP Service Extension with IANA (closed)

Closed. See Appendix H.34.

G.19. Inconsistencies between rfc5321bis and IANA registry and related issues (closed)

Closed with the rewrites to Section 8.1.1 in drafts -14 through -17.
See Appendix H.33.

G.20. Side-effects of approval change for Service Extension from "Standards Track or IESG Approved Experimental" to "Specification Required" (closed)

Fixed and/or overtaken by events with the rewrites to Section 8.1.1 in -17. See Appendix H.35.

G.21. Appendix B and Message Submission (closed)

No evidence of consensus in the WG to make further changes. Was Ticket #75. Closed; see Appendix H.38

G.22. IANA Registration Model for Registries Other Than Service Extensions (Closed)

Closed per note from Alexey on mailing list, 2023-02-13. Was Ticket #76. See Appendix H.37.

G.23. Headers Inserted in Mail Transport (closed)

Tentatively closed as part of the trace field cleanup. The tickets have been closed so, presumably, so is this. Tickets #81 and #7. See Appendix H.40.

G.24. Describing the "Operational Requirements" Loopholes (tentatively closed)

Alexey proposed closing this on 2023-03-21. A suggestion was received on 2023-04-23 that has been largely incorporated, but, as of 2023-12-10, the ticket has not been officially closed. See Appendix H.41.

G.25. Relocate paragraphs after the first out of Section 3.6.2 and to 3.6.1 (tentatively closed)

All but the first paragraph of Section 3.6.2 appear to belong at the end of Section 3.6.1 instead.
Ticket #88. See Appendix H.42

G.26. Remove SHOULD requirement for use of reply text

A requirement has been present since at least RFC 2821 that SMTP servers sending reply codes SHOULD use the text specified with them (Cf. Section 4.2). Alexey suggests "This requirement is hard to test compliance with" and should be dropped as a requirement. The counter argument is that a principal goal of this specification is to provide guidance to implementers -- with compliance testing being secondary -- and the SHOULD is justified on that basis. Of course,

if it is the case that "no one" follows that recommendation anyway, that would justify taking it out.

See the message thread starting with <https://mailarchive.ietf.org/arch/msg/emailcore/xUCSrarl_-QDn4XRgTUolha_TSE> and the proposed text in Section 4.2.
Ticket #89.

G.27. Section 4.2.4.3 appears to impose a BCP 14 requirement on users

Section 4.2.4.3 says "The user who originated the message SHOULD be able...". It seems undesirable impose BCP 14-level requirements on users. This should either be dropped to a "should" or restated as the requirement on servers as has been intended since at least RFC 2821 where that statement appeared.

See the message thread starting with <https://mailarchive.ietf.org/arch/msg/emailcore/YClG-kpxLjs9Ni3Ic1SIM67olC4> and the proposed text in Section 4.2.4.3.
Ticket #90.

G.28. Misplaced paragraph about special treatment in Section 4.4.3

The last paragraph of Section 4.4.3 discusses special treatment cases after DATA if that command is not completely successful. It does not seem to belong there. Perhaps it should be moved to Section 3.3, making that section even longer relative to RFC stylistic conventions adopted when we started forbidding references to page numbers.

See the message thread starting with <https://mailarchive.ietf.org/arch/msg/emailcore/F9omW9UodTAikTKLjBrWmmt1xA8>
Ticket #91.

Appendix H. Completed Items Moved from Appendix G

[[RFC Editor: Please remove this section before publication.]]

This appendix contains items identified as "closed" and moved from Appendix G (but referenced from there) to allow easier identification and tracking of open issues. Note that the subsection names deliberately duplicate those of the earlier appendix. It does not include the earlier errata items listed in Appendix I.

H.1. IP Address literals

The specification is unclear about whether IP address literals, particularly IP address literals used as arguments to the EHLO command, are required to be accepted or whether they are allowed to be rejected as part of the general "operational necessity" exception. Some have suggested that rejection of them is so common as an anti-spam measure that the use of such literals should be deprecated entirely in the specification, others that the are still useful and used and/or that, whatever is said about IP address literals within an SMTP session (e.g., in MAIL or RCPT commands), they should continue to be allowed (and required) in EHLO.
Ticket #1 (closed for rfc5321bis; issue for A/S).

H.2. Repeated Use of EHLO

While the specification says that an SMTP client's sending EHLO again after it has been issued (starting an SMTP session and treats it as if RSET had been sent (closing the session) followed by EHLO, there are apparently applications, at least some of them involving setting up of secure connections, in which the second EHLO is required and does not imply RSET. Does the specification need to be adjusted to reflect or call out those cases?

After extended discussion in October 2020, it appears that the easiest fix to these problems is to clarify the conditions for termination of a mail transaction in Section 3.3 and to clearly specify the effect of a second (or subsequent) EHLO command in Section 4.1.4.
See also Appendix H.5.
Ticket #2. (closed - Both changes have been made in draft-ietf-emailcore-rfc5321bis-01).

H.3. Meaning of "MTA" and Related Terminology

A terminology issue has come up about what the term "MTA" actually refers to, a question that became at least slightly more complicated when we formalized RFC 6409 Submission Servers. Does the document need to be adjusted to be more clear about this topic? Note that the answer may interact with the question asked in Section 2 above. Possibly along the same lines, RFC 2821 changed the RFC 821 terminology from "sender-SMTP" and "receiver-SMTP" to "SMTP client" and "SMTP server" respectively. As things have evolved, it is possible that newer terminology is a source of confusion and that the terminology should be changed back, something that also needs discussion.
Ticket #3 (Closed, note added rfc5321bis-11)

H.4. Originator, or Originating System, Authentication (to A/S)

Should RFC 5321bis address authentication and related issues or should Section 3.4.2 or other text be reshaped (in addition to or instead of the comment on that section) to lay a better foundation for such work, either in the context of mailing lists or more generally?
This may interact with Erratum 4055 and Ticket #30 below.

H.5. Possible clarification about mail transactions and transaction state

Original CREF from Section 3.3 was
[5321bis]: This section would be improved by being more specific about where mail transactions begin and end and then talking about "transaction state" here, rather than specific prior commands. --JcK and there should probably be a reference in Section 4.1.4.
Ticket #11. Closed 2022-05-26.

H.6. Remove or deprecate the work-around from code 552 to 452

The suggestion in Section 4.5.3.1.10 may have outlived its usefulness and/or be inconsistent with current practice. Should it be removed and/or explicitly deprecated?
SHOULD requirement removed.
Ticket #5 (fixed and closed).

H.7. Resolvable FQDNs and private domain names

Several CREF comments on this subject were removed from Section 2.3.5 prior to rfc5321bis-10.
Tickets #9 (definition of domain name; not identified in tracker), #10 (meaning of "resolvable domain name"; closed 2021-06-12), and #41 (closed -- no change 2021-04-05).

H.8. Issues with 521, 554, and 556 codes

See new Section 4.2.4.2. More text may be needed, there or elsewhere, about choices of codes in response to initial opening and to EHLO, especially to deal with selective policy rejections. In particular, should we more strongly discourage the use of 554 on initial opening. And should we make up a 421 code (or a new 4yz code, perhaps 454) code for situations where the server is temporarily out of service?
Ticket #6 (closed).

H.9. Requirements for domain name and/or IP address in EHLO (mostly closed, some to A/S)

Text in Section 4.1.4; change made in -05.
Ticket #19 (was ticket #47 -- done in rfc5321bis, more in A/S).

H.10. Does the 'first digit only' and/or non-listed reply code text need clarification?

Resolved. Text in Section 4.2 changed 2021-02-08 and CREF comment in Section 4.3.1 removed.

Perhaps unresolved -- ongoing discussion on mailing list after IETF 110.
Ticket #13 (fixed and closed).

H.11. Size limits

Once a decision is made about line length rules for RFC 5322bis, review the size limit discussions in this document, particularly the CREF comment (Note in Draft) at the end of the introductory material to Section 4.5.3 to be sure this document says what we want it to say. (See the additional question about minimum quantities, etc., in Appendix H.31.)
Ticket #14 (closed - no action) and maybe Ticket #38 (to A/S).

H.12. Further clarifications needed to source routes?

The current text largely deprecates the use of source routes but suggests that servers continue to support them.
Ticket #17 (Closed 20220125).

H.13. Should 1yz Be Revisited?

RFC 5321 depreciated the "positive preliminary reply" response code category with first digit "1", so that the first digit of valid SMTP response codes must be 2, 3, 4, or 5. It has been suggested (see mail from Hector Santos with Subject "SMTP Reply code 1yz Positive Preliminary reply", March 5, 2020 12:56 -0500, on the SMTP list) that these codes should be reinstated to deal with some situations that became more plausible after 5321 was published. Do we need to take this up again?
Ticket #18 (no, closed).

H.14. Review Timeout Specifications

RFC 5321 (and its predecessors going back to 821) specify minimum periods for client and server to wait before timing out. Are those intervals still appropriate in a world of faster processors and faster networks? Should they be updated and revised? Or should more qualifying language be added?
Ticket #16.

H.15. Possible SEND, SAML, SOML Loose End

Per discussion (and Ticket #20), the text about SEND, SAML, and SOML has been removed from the main body of the document so that the only discussion of them now appears in Appendix F.6. Per the editor's note in that appendix, is any further discussion needed? WG conclusion was "no".
Ticket #20 (closed)

H.16. Abstract Update

Does the Abstract need to be modified in the light of RFC 6409 or other changes?
Ticket #52 (changes made; closed)

H.17. Informative References to MIME and/or Message Submission

Should RFC 2045 (MIME) and/or RFC 6409 (Message Submission) be referenced at the end of Section 1.2? There is now a reference and brief discussion in that section.
Ticket #53 (more general reference to the A/S, closed).

H.18. Hop by hop Authentication and/or Encryption

Should this document discuss hop-by-hop authentication or, for that matter, encryption? (See CREF in Section 2.)
Propose "No, it shouldn't" (20211101 conversation with Todd, reaffirmed 20220121 plenary)
Ticket #50 (work with in A/S. Closed).

H.19. Enhanced Reply Codes and DSNs

Enhanced Mail System Status Codes (RFC 3463) [12] were added to SMTP before RFC 5321 was published and are now, together with a corresponding registry [50], widely deployed and in extensive use in the network. Similar, the structure and extensions options for Delivery Status Notifications [40] is implemented, deployed, and in wide use. Is it time to fold all or part of those mature specifications into the SMTP spec or at least to mention and

normatively reference them? And, as an aside, do those specs need work or, if they are kept separate, is it time to move them to Internet Standard?

At least one of the current references to RFC 3463 indicates that it SHOULD be used. That presumably makes the reference normative because one needs that specification to know what the present document requires. It has been moved in the -03 version of this draft, but, unless it is moved to Internet Standard, it will require downref treatment.
Part of Ticket #40.

H.20. SMTP Clients, Servers, Senders, and Receivers

RFC 821 used the terms "SMTP-sender" and "SMTP-receiver". In RFC 2821 (and hence in 5321), we switched that to "client" and "server" (See the discussion in Section 1.2). In part because a relay is a server and then a client (in some recent practice, even interleaving the two functions by opening the connection to the next host in line and sending commands before the incoming transaction is complete), RFC 5321 continues to use the original terminology in some places. Should we revisit that usage, possibly even returning to consistent use of the original terminology?
After discussion at IETF 113, small change made to Section 1.2.
Ticket #3 (closed).

H.21. Extension Keywords Starting in 'X-'

Section 2.2.2 contains a discussion of SMTP keywords starting in "X". Given general experience with such things and RFC 6648, is there any reason to not deprecate that practice entirely and remove that text? If we do so, should the former Section 4.1.5 be dropped or rewritten to make clear this is an obsolete practice?
Material formerly in 4.1.5 eliminated in rfc5321bis-06.
Ticket #42 (resolved with -06 and closed).

H.22. Deprecating HELO

RFC 5321 (and 2821 before it) very carefully circle around the status of HELO, even recommending its use as a fallback when EHLO is sent and a "command not recognized" response is received. We are just a few months short of 20 years; is it time to deprecate the thing and clean out some or all of that text? And, given a recent (4Q2020) discussion on the EMAILCORE list, should EHLO be explicitly bound to SMTP over TCP with the older transports allowed only with HELO? While those questions may seem independent, separating them is fairly hard given the way the text is now constructed.

Resolved 2021-01-19: No change
Ticket #43 (closed).

H.23. Resistance to Attacks and Operational Necessity

Section 7.8 is often cited as allowing an exception to the rules of the specification for reasons of operational necessity, not just attack resistance. I (JcK) believe the broader interpretation was intended by YAM (the section was new in RFC 5321). Recommendation: change the title to explicitly include "Local Operational Requirements" and add text to indicate that attack resistance is not the only possible source of such requirements.
Ticket #48 (done, closed)

H.24. New tickets created between 2022-01-21 and 2022-03-01 and subsequently closed.

* #58 Clarification of what is a domain name alias and who can substitute them (see Section 2.3.5). Closed 2022-05-19.

* #59 Case sensitive commands? (See Section 2.4). Closed 2022-05-26.

* #60 Restricted-capability clients? (See Section 3.3). Text in rfc5321bis-11 has been adjusted slightly in the hope of clarifying what is going on. Mailing list discussion thread: <https://mailarchive.ietf.org/arch/msg/emailcore/XtKLDF7MezR4FUd22x8OFWT-f5k/> Closed 2022-07-15.

* #61 Explaining mailing lists (See Section 3.4.2). Note that Section also interacts with Tickets #4, #30, #34, Appendix H.4, #12, and Appendix H.36. Tracker indicates this is a duplicate of Ticket #12
  Closed 2022-08-22.

* #62 null mx vs server domain in 4.2.4.2 (See Section 4.2.4.2) Per IETF113, no change, but was reopened 2022 May with additional comments. Closed again 2022-08-22.

H.25. Mandatory 8BITMIME

There was extensive discussion on the mailing list in October 2021 about messages with and without 8-bit (i.e., octets with the leading on) content and a tentative conclusion that support for 8BITMIME should be required. If that is the WG's conclusion, we need to figure out what to say and where to say it. SHOULD added to Section 2.4 in rfc5321bis-10.
Anything more probably goes to A/S.

Ticket #64. Marked as duplicate of part of Ticket #40 in tracker on 2022-07-04 and therefore closed.

H.26. Discussion of 'blind' copies and RCPT

CREF comment that originally appeared in Section 7.2 (left over from 2007) suggested deleting "especially" and "the full set of" -- copying the first one can be as harmful as copying all of them, at least without verifying that the addresses do appear in the headers. That fix has been applied. The WG also discussed the appropriateness of continuing to use "blind" and concluded that the damage caused by changing terminology would outweigh any advantages.
Ticket #15.

H.27. Revisiting Quoted Strings (reclassified)

Recent discussions both in and out of the IETF have highlighted instances of non-compliance with the specification of a Local-part consisting of a Quoted-string, whether any content of QcontentSMTP that actually requires special treatment consists of qtextSMTP, quoted-pairSMTP, or both. Section 4.1.2 (of RFC 5321, repeated above) ends with a few paragraphs of warnings (essentially a partial applicability statement), the first of which cautions against cleverness with either Quoted-string or case sensitivity as a threat to interoperability.

The Quoted-string portion of that discussion has apparently been widely not read or ignored. Do we need to do something else? If we do an Applicability Statement, would it be useful to either reference the discussion in this document from there or to move the discussion there and reference it (normatively?) from here?

There has been a separate discussion of empty quoted strings in addresses, i.e., whether the <qtextSMTP> production should be required to included at least one non-whitespace character. It is separate from this issue but would be further impacted or distorted from the considerations identified in this Section.

Text modified in -07 and further modified in -11.
Ticket #21. Closed in rfc5321bis-12 as noted in rfc5321bis-11. May also interact with Ticket #35 (which is an A/S issue).

H.28. Clarify where the protocol stands with respect to submission and TLS issues (reclassified)

1. submission on port 587

2. submission on port 465

3. TLS relay on a port different from 25 (whenever)

4. Recommendations about general use of transport layer (hop by hop) security, particularly encryption including consideration of RFC 8314.

Reclassified to A/S issue.
Apparently no ticket originally assigned although related to Ticket #53 and Appendix H.17.
Ticket #80 for A/S issue.

H.29. SMTP Model, terminology, and relationship to RFC 5598

CREF comments addressing this subject that appeared in (much?) earlier versions of this document removed from Section 2, Section 2.3.10. The second bullet item in the introductory portion of Appendix G may also be relevant.
No specific tickets were assigned but most of all of these issues have been discussed and changes made as needed. Closed.

H.30. More Text About 554 Given 521, etc.

Does reply code 554 need additional or different explanation in the light of the addition of the new 521 code and/or the new text in 5321bis Section 4.2.4.2)? (Former CREF after RCPT in Section 4.2.3 removed in earlier version.)
All needed text has been adjusted. Closed, no ticket assigned.

H.31. Minimum Lengths and Quantities

Are the minimum lengths and quantities specified in Section 4.5.3 still appropriate or do they need adjusting? CREF discussing that issue appeared in earlier drafts. There might also be a potential interaction with the proposed LIMITS SMTP extension (draft-freed-smtp-limits) should that ever be adopted.
Closed after discussion, no ticket assigned.

H.32. Internationalization

RFC 5321 was completed long before work on internationalization of email addresses and headers (other than by use of encoded words in MINE). In particular, it came before the work of the EAI WG that led to the SMTPUTF8 specifications, specifically RFCs 6530ff. The second explanatory paragraph at the end of Section 4.1.2 ("Systems MUST NOT define mailboxes ...") is an extremely strong prohibition against the use of non-ASCII characters in SMTP commands and the requirements about message content in Section 2.3.1 an equally strong one for content. Would it be appropriate to add something like "in the

absence of relevant extensions" there? Also, given [mis]behavior seen in the wild, does that paragraph (or an A/S) need an explicit caution about SMTP servers or clients assuming they can apply the popular web convention of using %NN sequences as a way to encode non-ASCII characters (<pct-encoded> in RFC 3986) and assuming some later system will interpret it as they expect? Would it be appropriate to add an Internationalization Considerations section to the body of this document if only for the purpose of pointing people elsewhere?

More broadly, while the EAI WG's extensions for non-ASCII headers and addresses are explicitly out of scope for the EMAILCORE WG (at least for 5321bis (and 5322bis), those documents make assumptions and interpretations of the core documents. Are there areas in which 5321bis could and should be clarified to lay a more solid foundation for the EAI/SMTPUTF8 work and, if so, what are they?

Text added in rfc5321bis-11, Section 4.1.2 and in rfc5321bis-14. Additional discussion on this subject belongs in the A/S. Ticket #78 for cautions about encodings -- assigned to A/S, not rfc5321bis. Ticket #79 for I18n Considerations Section, also in A/S. Also mentioned in Ticket #40.

H.33. **Inconsistencies between rfc5321bis and IANA registry and related** issues

The descriptions and specifications for the IANA SMTP-related Mail Registries (<https://www.iana.org/assignments/mail-parameters/mail-parameters.xhtml>), and the reference from the Service Extension registry portion of that collection back to RFC 5321 are incorrect and/or confusing. Also a registry must be specified for Additional Registered Clauses.

* Ticket #67 (Additional Clauses)
* Ticket #69 (inconsistencies)
* Ticket #70 (section reference)
* Ticket #71 (SMTP Registration template should have a field specifying level of approval)
* Ticket #72 (Add a field to SMTP registration template about whether the extension is suitable for SMTP Submission)

Closed with the rewrites to Section 8.1.1 in drafts -14 through -17.

H.34. Approval Required to Register an SMTP Service Extension with IANA

RFC 5321 and its predecessors required that SMTP Service Extensions be registered with IANA and that they be documented in "Standards-Track or Experimental RFCs specifically approved by the IESG for this purpose". Sentiment in the IETF has shifted away from that level of approval and the work it requires toward trying to be sure the relevant keywords are documented and registered to prevent ambiguity and naming conflicts. Consequently the requirement is being changed to Specification Required.
See also Appendix H.33 and Appendix H.35.

Closed with adoption of two-option model in -16 and -17.
Ticket #56.

H.35. Side-effects of approval change for Service Extension from "Standards Track or IESG Approved Experimental" to "Specification Required"

There are more of these but, at least, the change (and text in RFC 8126) requires specifying a Change Controller because it is no longer obviously the IETF/IESG.

(Some text tentatively added to Section 2.2.2 in rfc5321bis-13. Then that was revised and the important parts of it incorporated into the IANA registration instructions Section 8.1.1 in rfc5321bis-14.)
Ticket #68. Fixed and/or overtaken by events with the rewrites to Section 8.1.1 in -17.

H.36. Issues with mailing lists, aliases, and forwarding

From the former CREF comment in Section 3.4.2: May also want to note forwarding as an email address portability issue. Note that, if changes are made in this area, they should be kept consistent with the description and discussion of the 251 and 551 codes in Section 4.2 and Section 3.5 as well as Section 3.4.1 to avoid introducing inconsistencies. In addition, there are some terminology issues about the use of the term "lists", identified in erratum 1820, that should be reviewed after any more substantive changes are made to the relevant sections.
Ticket #12 and Ticket #34 (Ticket #34/ erratum 1820 resolved in -06 and closed). (Ticket #61 identified in tracker as duplicate of Ticket #12.)
Closed.

H.37. IANA Registration Model for Registries Other Than Service Extensions

The WG decided to shift the registration model for Service Extensions from "Standards Track or IESG-approved Experimental" to "Specification Required". No decisions have been made yet about other mail-related registries established by this document in Section 8.1, specifically the "VIA link types" and "WITH protocol types" discussed in Section 8.1.3, the additional clauses for "Received:" headers discussed in Section 8.1.4, and the IP address literal indicators discussed in Section 8.1.2. All three sets will be left unchanged, and this item closed unless other decisions are made, presumably before IETF 115.

Post-IETF 115 update (for rfc5321bis-16): Slide for that meeting proposed changing link and protocol types to IETF Review. No conclusion on mailing list yet; see "Editor's Analysis" (<https://mailarchive.ietf.org/arch/msg/emailcore/NVd2BdnLkURNjLaSG3S6rqXJY0s>). CREF comment added to Section 8.1.3 above. Additional clauses and IP address literals were not discussed.

Further update after 2022-12-07 Interim meeting (partially reflected on rfc5321bis-17): No change to IP address literal registrations. Additional clauses still under discussion.
Ticket #76. Closed per email from Alexey 2023-02-13.

H.38. Appendix B and Message Submission

Appendix B was written long ago and does not distinguish very carefully from the case where an MSA is involved from direct MUA-MTA communications. On the other hand, the situation it describes cannot arise with a conforming message submission system. Should it be rewritten and, if so, how much?

Change made to explicitly reference the Submission Server specification.
Ticket #75. Closed 2023-02-20 after there was no clear consensus in the WG (or even significant discussion) about the need for additional text or what it should be.

H.39. The FOR Clause in Trace Fields: Semantics, Security Considerations, and Other Issues

Marked closed in issue tracker, 2023-02-13.

The FOR clause in time-stamp ("Received:") fields is seriously under-defined. It is optional, the syntax is clear, but its semantics and use, while perhaps obvious from content and the application of common sense, have never been defined ("never" going back to 821). Do we want to better define it? Is there any chance that a definition would invalid existing, conforming and sensible, implementations? If we do want to define semantics, draft text and advice as to where it should go are invited.

(Paragraph added 2021-08-18, draft -04)
In particular, recent discussions point strongly to the need for a statement to the effect that the value of the for clause must contain one of the addresses that caused the message to be routed to the recipient of this message copy (thanks Ned), that no more than one address can appear, and that showing one address when there are multiple RCPT commands may be a security and/or privacy issue (thanks Ned and Viktor and see <https://mailarchive.ietf.org/arch/msg/ietf-smtp/hMkwHT-6bi_AwYIxbFJBX5pqjiA>). More detailed or specific guidance, including case analysis, are probably material for the A/s, but that is obviously up to the WG.

Note the existing discussions in Section 7.2 and Section 7.6 as they may need adjustment, or at least cross-references, especially if FOR is more precisely defined.

There is probably an error in Section 7.6. Its last sentence implies a possible interaction between messages with multiple recipients and the FOR clause of trace fields. However, because the syntax of the FOR clause only allows one Mailbox (or Path), it isn't clear if that statement is meaningful. Should it be revised to discuss other situations in which including FOR might not be desirable from a security or privacy standpoint? (See above -- this paragraph deliberately not changed in -04 through -11).
Ticket #55. Marked closed in issue tracker, 2023-02-13.

H.40. Headers Inserted in Mail Transport

The process of revising Section 4.4 for rfc5321bis-14 and explicitly treating "Return-path:" as a trace field in this spec raised a possibly-interesting problem. To date at least, the only mail headers that are inserted by SMTP clients or servers and specified by this document are trace fields, i.e., "Received:" and "Return-path." Is that a principle? Should we say that SMTP systems have no business inserting headers other than those that provide trace information? Or are there other types of header fields we can see SMTP systems inserting and, if so, what are they?" See also the former Section 8.1.5 and Section 4.4.4.
Ticket #81 about how additional trace fields are registered.

Also Ticket #7 Better definition for trace header fields. Tentatively closed as part of the trace field cleanup, July 2023, draft -19.

H.41. Describing the "Operational Requirements" Loopholes

The discussion in Section 7.8 and Section 7.9, and the pointers to them from Section 6.2 and elsewhere essentially provide the basis for implementations to deviate, in multiple ways, from the intent of RFC 5321 and 5321bis while claiming conformance. Is the present text what we want? Should it be made more explicit about what is allowed and what isn't (whether either is possible is obviously part of the question)? If we intend to address those issues in the A/S (not just the limits question which is already there), do we want explicit forward pointers to it in the existing sections?

(This topic mentioned on the mailing list for the first time in <https://mailarchive.ietf.org/arch/msg/emailcore/ Zv7XtCUPavpacrLslgDiK7M-FJ8 >, 2023-01-15). After a call for closing the ticket was made 2023-03-07 if no more input was received, a suggestion was received on 2023-04-23 that has been largely incorporated. No more comments received.
Ticket #82.

H.42. Relocate paragraphs after the first out of Section 3.6.2 and to 3.6.1 (closed)

All but the first paragraph of Section 3.6.2 appear to belong at the end of Section 3.6.1 instead.

Despite dating from RFC 5321 or earlier, this appears to be a simple editorial improvement. Changed.
Ticket #88.

Appendix I. Detailed Change Descriptions and Logs

[[RFC Editor: Please remove the first three subsections of this section before publication. The last subsection, including its title, will then become the entire final form of the appendix]]

I.1. RFC 5321 Errata Summary

This document addresses the following errata filed against RFC 5321 since its publication in October 2008 [63]. More details on each of these can be found at <https://www.rfc-editor.org/errata/rfc5321>. As with the previous appendix, ticket numbers included below reference <https://github.com/ietf-wg-emailcore/emailcore/issues>

1683 ABNF error. (closed) Section 4.4
Ticket #23 (fixed, closed).

4198 Description error. (closed) Section 4.2.
RESOLVED 2020-12-14, ticket #24 (closed).

2578 Syntax description error. (closed) Section 4.1.2
Ticket #25 (fixed, closed)

1543 Wrong code in description (closed) Section 3.8
Ticket #26 (fixed, closed)

4315 ABNF - IPv6 Section 4.1.3 (closed). Former description in the document body was: [5321bis]The IPv6 syntax has been adjusted since 5321 was published (the erratum mentions RFC 5952, but RFC 6874 and draft-carpenter-6man-rfc6874bis should also be considered). See the rewritten form and the comment in the section cited in the previous sentence, at least for the RFC 5952 issues. See https://www.rfc-editor.org/errata/eid4315
Ticket #27 (closed 2021-01-19).

5414 ABNF for Quoted-string (closed) Section 4.1.2
Ticket #22 (fixed, closed).

1851 Location of text on unexpected close Section 4.1.1.5 (closed).
Text moved per email 2020-12-31.
Ticket #28 (fixed, closed).

3447 Use of normative language (e.g., more "MUST"s), possible confusion in some sections Section 4.4.
Ticket #7

5711 (closed) Missing leading spaces in example Appendix D.3.
As of 2021-03-15, both the txt and html-ized versions of draft-ietf-emailcore-rfc5321bis-02 were showing identical output for both parts of the example, so the problem appears to be OBE at worst.
Ticket #29 (closed 2021-03-16)

4055 (closed) Erratum claims the the description of SPF and DKIM is wrong. It is not clear what 5321bis should really say about them, but the current text probably needs work (or dropping, which is what the proposed erratum suggests).
Text changed; ticket should probably be closed after WG reviews -04.
Ticket #30 (resolved and closed).

I.2. Changes from RFC 5321 (published October 2008) to the initial (-00) version of this draft

1. Acknowledgments section (Section 9) trimmed back and then revised for new document.

2. Introductory paragraph to Appendix F extended to make it clear that these features were deprecated a long time ago and really should not be in use any more.

3. Adjusted some language to clarify that source routes really, really, should not be used or depended upon.

4. IPv6 address syntax replaced by a copy of the IPv6 URI syntax and a note added.

5. Production index added as a first step in tying all productions to their sources. As part of the effort to make the document more easily navigable, table of contents entries have been created for the individual command descriptions.

6. Clarified the relationship between the SMTP "letters, digits, and hyphens" and DNS "preferred name syntax" (Section 2.3.5).

7. Revised the reply code sections to add new 521 and 556 codes, clarify relationships, and be explicit about the requirement for clients to rely on first digits rather than the sequences in Section 4.3.2.

8. In conjunction with the above, explicitly obsolete RFCs 1846 and 7504

9. Incorporated a correction reflecting Errata ID 2578.

10. Some small editorial changes made to eliminate redundant statements that were very close together. Other, equally small, editorial changes have been made to improve grammar or clarity.

11. A few questions, marked "[[5321bis Editor's Note:", or "[[Note in Draft" have been added for the group to resolve. Other questions, especially those in the errata summary, are simply included in narrative comments in CREFs.

12. Checked and rationalized "response" (to a command) and "reply code" terminology. One can talk about a "999 response" but only a "999 reply code". There is no such thing as a "response code".

13. Added note about length limit on mailbox names ("email addresses").

14. Added an "errata summary" subsection to this change log/ comparison to 5321 in this Appendix. The entire Appendix will, of course, disappear at the time of RFC publication unless someone wants to make a strong case for retaining it.

15. Rationalized CREFs to 2821, 5321, 5321bis etc.; added note to readers below the Abstract.

16. Temporarily added a "Note on Reading This Working Draft" after the Abstract.

I.3. Changes Among Versions of rfc5321bis

I.3.1. Changes from draft-klensin-rfc5321bis-00 (posted 2012-12-02) to -01

Substantively, these two versions differ only by suppression of the CREF and other discussion associated with the evolution from RFC 2821 to RFC 5321. That change includes an update to the document's Note to Readers, the date, the file name, and the addition of this change log subsection.

I.3.2. Changes from draft-klensin-rfc5321bis-01 (20191203) to -02

* Minor clarifications to improve text, e.g., addition of NOOP to the list of non-mail transaction examples in Section 4.1.4.

* Added topics exposed in the ietf-smtp list and the IETF list "dogfood" discussion during December 2019 and an index listing of substantive issues identified only in CREFs in the prior draft as a new Appendix G.

I.3.3. Changes from draft-klensin-rfc5321bis-02 (2019-12-27) to -03

* Added more text to Appendix H.8 to specifically call out the session-opening policy issues surrounding these codes.

* Added discussion of "1yz" reinstatement in Appendix H.13.

* Added discussion of timeouts in Appendix H.14.

* Added subsection on Enhanced Status Codes and DSNs to the outstanding issues list Appendix H.19.

* Replaced reference to RFC 1652 (8BITMIME) with the Internet Standard version, RFC 6152.

* With help from cketti, clarified the ABNF productions whose terminals appear in other documents.

* Added discussions of Quoted-string, Internationalization, and client-server versus sender-receiver terminology to Appendix G.

* Added note to the Abstract.

I.3.4. Changes from draft-klensin-rfc5321bis-03 (2020-07-02) to draft-ietf-emailcore-rfc5321bis-00

* Added a paragraph about non-null quoted strings to Appendix H.27.

* Added an explicit pointer to email insecurity and TLS to Appendix H.28. Inspired by Ben Kaduk's comment on the WG Charter, 2020-09-09.

* Converted document from individual to emailcore WG effort.

I.3.5. Changes from draft-ietf-emailcore-rfc5321bis-00 (2020-10-06) to -01

* Editorial: Corrected "blackslash" to "backslash"

* Rewrote the introduction to Appendix G slightly to reflect the creation of the EMAILCORE WG.

* Applied fixes for repeated use of EHLO. See Appendix H.2.

* Added two new questions, one about "X" extensions (Appendix H.21) and one about the status of HELO (Appendix H.22).

* Removed mention of SEND, SAML, SOML from the main body of the text (Ticket #20).

* Added a warning about side effects to Appendix H.36.

* Added ticket numbers to descriptions of issues and changes, adjusted some text so relationships would be more clear, and added subsections to the Appendix G and H lists to pick up on tickets that were not easily identified in those sections of with the text.

* Made several additions to the Index, including one to deal with SEND et al., as above.

I.3.6. Changes from draft-ietf-emailcore-rfc5321bis-01 (2020-12-25) to -02

* Corrected discussion mailing list to point to emailcore@ietf.org in the introductory note.

* Added new subsection(s) to Appendix G to reflect newly discovered issues.

* Changed "as discussed in" references in Section 4.5.5 per ticket #45.

* Corrected a misleading use of the term "mailbox" in Section 3.3.

* Changed descriptions of use of first digit in replies per ticket #13. See Appendix H.10.

* Moved paragraph per ticket #28, erratum 1851.

* Added more clarifying cross-references, clarified some CREFs, and cleaned out some of those that no longer seemed relevant.

* Removed "updates 1123" is unnecessary and obsolete.

* Updated several references.

I.3.7. Changes from draft-ietf-emailcore-rfc5321bis-02 (2021-02-21) to -03

* Editorial: Fixed some instances of constructions like "RCPT TO command". The name of the command is RCPT. Sloppy editing in 2008.

* Added text and cross-references to clarify the role of 452 and 552 in "too many recipients" situations.

* Added Appendix H.23 to discuss changes to better reflect "operational necessity" issue.

* Added detail for erratum 5711, ticket #29.

* Added new subsections of Appendix G.7 to keep some previously-unnoted CREF notes from getting lost. Also removed some CREFs that were notes on changes made before the WG was created or appeared to no longer have value and trimmed or rewrote some of the remaining ones.

* More discussion of Ticket #13, See Appendix H.10.

* Identified Ticket #41 as closed. See Appendix Appendix H.7; notes removed from Section 2.3.5.

* "SHOULD" requirement for interpreting 552 "too many recipients" removed from Section 4.5.3.1.10, explanation added, and text cleaned up. Also removed the parenthetical historical notes on the return code definitions in Section 4.2. See Appendix H.6. (Ticket #5)

* Modified Appendix H.19 to add a note about the normative status of RFC 3463 and moved that reference.

* Several clarifications to initiation and termination of mail transactions in Section 4.1.4.

* Several additional minor editorial improvements.

* Note for drafts -03 and -04 only, modified somewhat for -05 but outdated from -06 forward: Some issues are still outstanding: Notes were posted to the list on 2021-07-09 about tickets #7 (5322bis issue?), #10 , #14 (closed), #20 (closed), #30 (closed), and #42 (closed). Even though some comments about them appeared in the subsequent day or so, there appears to have been insufficient time for discussions to stabilize sufficiently for changes to be included in this version of the I-D.

I.3.8. Changes from draft-ietf-emailcore-rfc5321bis-03 (2021-07-10) to -04

* Clarified that the "period" in <CRLF>.<CRLF> is really the ASCII one in Section 3.3.

* Several other small editorial corrections.

* Added several notes about the possible need to add text to reflect the presence of MSAs and to clarify whether MUAs send messages directly to MTAs or whether, in that case, the MUAs are just incorporating MSA functions.

* Added new text to Appendix H.39 reflecting discussions of the Received...FOR issue.

* Adjusted discussion of erratum 4315 (Ticket #27) to reflect more recent IPv6 syntax developments.

* Adjusted discussion of the various "mail not accepted" codes, rewrote Section 4.2.4.2, annotated and inserted cross-references in relevant response code descriptions and (tentatively)

identified this document as obsoleting RFC 7505. Editor's guess, reinforced by a brief conversation with John Levine (lead author of 7505), is that we should incorporate text as needed and obsolete it. The changes include replacing the reference to the "nullMX" I-D with RFC 7505, which I am appalled that neither I nor anyone else noticed earlier. Cf. Appendix H.8, Section 4.2.4.2, and Ticket #6.

I.3.9. Changes from draft-ietf-emailcore-rfc5321bis-04 (2021-10-03) to -05

* Took a first step toward rewriting and updating the introductory material. It is only a first step; suggestions welcome.

* Minor editorial fixes.

* Correct text about domain name checking in Section 4.1.4, probably fixing ticket #19. See CREF added there.

* Added Appendix H.25 a placeholder for the 8BITMIME discussion and possible action.

* Additional changes to the description and organization of trace field materials. Intended to resolve the 5321bis part of Ticket #7.

* Remaining patch to SEND, etc., discussion in Appendix F.6 applied and CREF removed.

* Removed discussion of "X-" and edited associated text. The fix may or may not be sufficient to resolve Ticket #42 (later closed).

* Verified that the problems of getting four-level sections (e.g., "4.1.1.1" and other command-specific ones) into the table of contents and the index reflecting page numbers still exist and updated the introductory note.

I.3.10. Changes from draft-ietf-emailcore-rfc5321bis-05 (2021-10-24) to -06

* Finished making changes for "X-" and commands starting in "X". Changes made in -05 were incomplete. This should allow closing Ticket #42.

* Removed spurious "for use in delivery notifications" from 3.6.2. Was just a pasting-type error.

* Changed "In other words" to "In particular" in Section 2.3.5 per Ticket #10 and July 2021 mailing list discussion. Removed associated CREF.

* Converted to xml2rfc v3 (thanks to John Levine for doing the hard parts) and then modified the introductory note accordingly.

* Started reworking the Abstract -- see revised CREF there.

* Rewrote Section 2.3.3 slightly to note the existence of submission servers and removed the CREF.

* Updated Appendix H.18 and slightly modified CREF note in Section 2 -- proposed to not get 5321bis involved with this (Ticket #50).

* Rewrote parts of Section 3.4.2 to clarify text and respond to Ticket #34.

* Inserted suggested text info CREF at end of Section 1.2. Comments welcome. Soon.

I.3.11. Changes from draft-ietf-emailcore-rfc5321bis-06 (2021-11-07) to -07

* Reviewed closed tickets and discussion with co-chairs after IETF 112 and updated text. Sections or items that are, according to the ticket list, completely closed have been identified by "(closed)" in or near their titles.

* Changed the suggestion for references to other documents mentioned in G.7.14 and Section 1.2 to actual text. Cleaned things up and, per note from Alexey 2021-11-17, have marked Ticket #53 as closed.

* New text added and old text replaced about quotes in Section 4.1.2, text rearranged and edited a bit per Appendix H.27, and CREF added about alternatives. Changes reflect mailing list comments through

* Last sentence (about source routing) removed from Section 2.1. Also adjusted text in Section 3.3, Section 4.1.1.3 but work is still needed there (see new CREFs in that section) and Section 6.1. The former Appendix C and references to it have been removed, leaving a placeholder to avoid changing subsequent appendix numbering before IETF Last Call (and maybe its completion) No changes have yet been made to Appendix F.2 but it is likely to require some work in the next version of the document. This is entirely about Ticket #17, which should not be closed until that appendix is updated.

I.3.12. Changes from draft-ietf-emailcore-rfc5321bis-07 (2021-12-04) to -08

Other than the partial cleanup for "forwarding" and "aliasing" and miscellaneous editorial fixes and corrections (including cleaning out unused references), changes in this version reflect the conclusions of the EMAILCORE interim meeting held 2021-12-09. References to "slides" are to the deck at https://datatracker.ietf.org/doc/slides-interim-2021-emailcore-01-sessa-chairs-slides/ and the minutes at https://notes.ietf.org/notes-emailcore-interim-dec-2021

* (Slides 9 through 12): Removed source route examples from Section 4.1.1.3 and added a new paragraph explaining what happened to them. For slides 11 and 12, see below for more general Appendix F.2 discussion.
  (Cf Appendix H.12 and Ticket #17.)

* (Slides 13 through 14): Domain names, Section 2.3.5. Removed "resolvable". Changed "alias" to "host alias" (although, after looking at the actual text, the intent seems clear from the CNAME label comment and, of course, the term "host" has been controversial in DNS circles and the minutes are not clear on the desirability of this change). Inserted "MUST" for the FQDN. A cross-reference to the domain name discussion in this section has been added to Section 4.1.1.1 in an attempt to resolve that discussion.
  In going carefully through this material, it became obvious that the discussions in Section 2.3.5 and Section 5 were confusing and somewhat redundant. Those sections have been rewritten to clarify intent, hint that extensions may modify (or have modified) a few of the rules, improve cross-references, and remove redundant text. Domain name issues are still under discussion on the WG mailing list as of 2021-12-18 and it is possible that the above changes may have introduced new issues, so additional changes are possible.
  (Cf target="G-domain"/> and Tickets #9 and maybe #10.)

* Aliasing and forwarding:
  Consolidated former sections 3.4 and 3.9 into a new Section 3.4, making them subsections. The new subsection probably still needs work and maybe an introductory paragraph, but even bringing the two subsections together may reduce some sources of confusion identified on the mailing list. Added cross-reference to security considerations from the new Section 3.4.1.

All other issues discussed during the interim appear to be unresolved and were deferred to the mailing list.

As what should be the third and final step in deprecation of source routes and removal of them from the main text, the appendix that discusses them (Appendix F.2) has been rewritten, adjusting language and incorporating some materials from the former Appendix C.

I.3.13. Changes from draft-ietf-emailcore-rfc5321bis-08 (2021-12-31) to -09

* Multiple small editorial changes.

* Started tuning Appendix I.2 preparatory to an actual "Changes from" section.

* Moved and rewrote a paragraph that seemed to be out of place from Section 4.4.1 to Section 4.1.1.3 per November discussion. See the note in the latter section for discussion.

* Removed "for initial submission of messages" from Section 2.3.5 and changed "may" to "MAY" in the last bullet point there, per Interim. Removed comment/ Editor's Note from that section: further instructions and evidence of consensus needed to do anything additional with it.
  Ticket #9

* In Section 3.4.2, rewrote the first sentence to make it descriptive rather than normative. Also removed the last sentence of that paragraph. Both per the editor's understanding of the Interim's conclusions, but the latter was put in because of problems with people thinking changing the argument to the MAIL command also required changing "From:" in the headers, so this should be carefully reviewed on list. Comment removed from that section -- the dead horse has been kicked past recognition.
  Ticket #4.

* In Appendix F.2, changed the requirement for server support to MAY, and prohibited client support, for source routing. Also made a small wording change. Per Interim.
  Ticket #17

With this draft, comments in the running text ("//" at the beginning of lines in the plaintext version) that seem to no longer be relevant either generally or after the discussions during the 2022-01-21 Interim are being removed. The "Notes on Reading..." at the beginning of the document (just below the Abstract) have been revised accordingly. Sections from which comments were removed this time include:

* Abstract, comment introduced in -06 (No comments on it through -08 are interpreted as consent;

* Section 2 (any discussion needed will be in A/S);

* Section 2.3.10 (discussion seems to have ended);

* Section 4.1.1.3 (no further discussion during Interim, so assume comment is no longer needed);

* Section 4.1.2 (no further discussion since -08 appeared or during Interim, assumed to not require further work);

* Section 4.5.3.1 (further discussion will be in A/S);

* Section 4.2.2 (this comment obsolete since revision -04 of this document).

* Cross-checked ticket notes and annotations in this document against the ticket system. Consistent for closed tickets as of 2022-01-31.

I.3.14. Changes from draft-ietf-emailcore-rfc5321bis-09 (2022-02-01 to -10

* Small editorial fixes, including a lingering typographical error or two.

* Captured some additional sections into the TOC.

* Added an additional index subsection for terminology, including the terms of Section 2.3 and a few others. More entries may be needed.

* Modified Section 3.4.2 to flag continuing uncertainty about decisions in the January Interim and subsequent list discussion about some text in that Section.

* Modified Section 4.2.4.2 to correct confusing phrasing and make a placeholder for a possible addition raised on email 2022-03-05.

* Added Appendix H.25 and Appendix G.17 so all open (and most closed) applicable ticket numbers are identified in this document.

I.3.15. Changes from draft-ietf-emailcore-rfc5321bis-10 (2022-03-07) to -11

* In order to improve tracking, removed all closed items for Appendix G to a new Appendix H, making this section/ appendix into Appendix I. Stubs are retained in Appendix G so as to not mess up external references to the relevant numbers. As part of the same process and to lower the odds of things falling through the cracks, a new "Ticket Index" has been added to permit accessing text in those appendices by ticket number. As with those appendices G and H, this will disappear before the document becomes an RFC.

* Small editorial fixes, e.g., "this document" in the text referred to RFC 5321 and would not be accurate for the current draft and to clarify Section 3.3 slightly. Also better identified closed items in Appendices G and H.1 and removed more now-obsolete CREF comments.

* Upgraded the "ASCII" reference and removed "US-ASCII" usage. Under the principle of minimum change, more citations of X.4-1968 have been left intact with RFC 20 also cited where it seemed important and a pointer has been inserted from the former reference to the latter.

* Editorial improvement to the "end of mail data" discussion in Section 3.3 per a discussion that appeared on tool-discuss in July 2021. No substantive change, just making the requirement painfully obvious.

* Editorial improvements and additional examples in the discussion of local-part quoted strings in Section 4.1.2 as discussed in Appendix H.27 and Ticket #21.

* Modified Section 3.4.2 with edited version of text specified at IETF 113 to avoid confusion about header field changes when MAIL command parameter is changed.

* Addressed Ticket #3 about "sender-SMTP" versus "SMTP client" by adding a note indicating that they are equivalent (per IETF 113 and Alexey's follow-up message 2022-05-5).

* Addressed the VRFY issues per mailing list discussions. Ticket #63 [should be] closed.

* The portion of Section 1.2 that discussed submission had gotten a bit hard to follow. It has been cleaned up a bit. I strongly recommend checking it and making suggestions if it is not to your taste. The CREF comment that was there before has been removed.

I.3.16. Changes from draft-ietf-emailcore-rfc5321bis-11 (2022-05-24) to -12

* Usual small editorial corrections, including fixing several places where "^nbsp;" appeared in -11.

* Adjusted acknowledgments to pick up Ned Freed.

* Corrected the requirements terminology paragraph to include RFC 8174.

* Updated Section 3.3 and Section 5.1 to remove the "restricted capability" handwaving, identify forwarders as an example of where MX processing might not be needed, and create cross-references between the two sections (Ticket #60).

* Added some additional cross-references, including in Section 7.2.

* Updated Appendices G and H to reflect tracker actions that affect this version of the draft.

* Per discussion at IETF 114 and in draft-klensin-email-for-clause, problem text in Section 7.6 has been tentatively replaced with text Alexey suggested.

I.3.17. Changes from draft-ietf-emailcore-rfc5321bis-12 (2022-07-09) to -13

This is a snapshot version of the evolving document, intended to allow the WG to see what things look like before, e.g., additional changes to clarify and specify IANA actions and the form and content of the various registries.

* Closed ticket #62 and adjusted information for several other tickets that were closed without notification to the mailing list).

* Adjusted registration requirements for SMTP Service Extensions by removing requirement text from Section 2.2.2 and adding Appendix H.34. Also started cleaning up some related issues (see Appendix H.33. The text in Section 8.1 will ultimately be changed to whatever is worked out with the WG and IANA.

* Started to work through changes consequent of the loosened IANA registration requirements for Service Extensions.

* Small editorial fixes, including correcting a section title and moving text incorrectly added to Appendix G to Appendix H.

I.3.18. Changes from draft-ietf-emailcore-rfc5321bis-13 (2022-09-13) to -14

* Altered Appendix B to get "RFC 822" out of the title and align the text. Also added Appendix H.38 to identify related issues and started to distinguish references to RFC 5832 (which should remain) from those that should point to 5322bis when it gets a number. There may not be any of the former, but they all need checking.

* Extensively modified IANA Considerations (Section 8), completely rewrote Section 2.2.2, and made a small change to Section 2.2.1 to reflect WG preferences/ instructions, both generally and to better reflect the implications of the "just get it registered to avoid name conflicts" change. Also addressed tickets #68, #69, #70, #71, and #72.

* Added Appendix H.37 about the procedures IANA is to follow about registries other than Service Extensions and Appendix H.40 about loose ends with trace and other extension fields.

* Reorganized Section 4.4 to separate timestamp, Return-path, and other information into separate subsections. Text might still need more work.

I.3.19. Changes from draft-ietf-emailcore-rfc5321bis-14 (2022-10-23) to -15

* Small editorial fixes

* Updated Appendix G and Appendix H to close issues for which text was supplied in -14 or earlier and for which there have been no comments and to synchronize with Alexey's lists.

* Dropped several more CREF comments/questions. Most of them have received on comments in over a year.

* Began work on structure of document for publication, including inserting a placeholder for the final Appendix I, now in Appendix I.4 and starting to rewrite the introduction to Section 1.2.

* Added a new terminology subsection, Section 2.3.12, to make the "session" versus "transaction" more prominent. This should really be the first entry under "SMTP Terminology" but I am trying to preserve the 5321 section numbering, at least until we decide we are nearly finished.

* Several small changes to allow the use of reply code 450 to indicate that the server is temporarily not available when used outside a mail transaction.

I.3.20. Changes from draft-ietf-emailcore-rfc5321bis-15 (2022-11-06) to -16

* Usual small editorial corrections and changes.

* Modified Appendix H.37 to more explicitly describe the problem, adding notes as indicated there.

* Forked Section 8.1.1, temporarily inserting new text and retaining the older version.

* Small modification to Section 8.1.4 to correct a bad crossreference.

* Consistent with discussion at IETF 115, removed the suggested new registry for additional trace header fields.

* Updated the note and set up Appendix I.4 for new text starting in the next draft.

I.3.21. Changes from draft-ietf-emailcore-rfc5321bis-16 (2022-12-06) to -17

* Editorial patches.

* Per discussion at 2022-12-07 Interim and Alexey's note to the list on 2022-12-08 of Section 8.1.3, changed the registration procedure for "with" and "via" (both all in "Received:") to "RFC Required" (after a brief detour through "IETF Review"). Address literal tag registration procedure remains unchanged.

* During the Interim, there was a discussion about changing the registration procedure for Additional Registered Clauses (Section 8.1.4) from "Standards Action/IETF approved Experimental" to something else, like "IETF Review". The conclusion from that discussion is that, while those at the meeting seemed to favor making that change (by a VOTE of 3-1), it should be taken to the mailing list. That has not happened but the author has, in a rare moment of succumbing to consensus by exhaustion, made the change and added a note. This can be undone if needed after/if the WG ever gets around to discussing it and reaching a conclusion.

* Per discussion during the Interim and on the mailing list, settled on the two-step model for Service Extensions registrations that was described as "Option 2" in Section 8.1.1 of 5321bis-16, got rid of "Option 1" and started reworking the surrounding text.

* Rewrote the introductory "Note on Reading..." (just below the Abstract) to more accurately reflect the state of reviews and document maturity.

* First version of the rewrite to Section 2.2.2 to adjust for the change to the two-option Service Registration model.

* Closed several Appendix G items that seem to be settled, most of them about IANA registries, to Appendix H.

I.3.22. Changes from draft-ietf-emailcore-rfc5321bis-17 (2022-12-31) to -18

* Corrected the change log in Appendix I.3.21 to identify the changes to Section 2.2.2, which was accidentally omitted.

* Corrected more typos (most introduced in -17).

* Usual minor editorial fixes, some related rearrangement of text within sections, and adjustments to the index.

* Added Appendix H.41 about exceptions to the spec that are considered reasonable and whether they are properly described.

* Further tuned the text in Section 8.1.1 to improve clarity.

* Added a temporary CREF note to the beginning of Section 4.4 to indicate that further changes may be needed depending on the WG's conclusions about documentation of trace information.

* Pushed back work on Appendix I.4 to the next revision. At least.

I.3.23. Changes from draft-ietf-emailcore-rfc5321bis-18 (2023-02-07) to -19

* Editorial corrections

* More closed issue/ticket cleanups and consequent moves from Appendix G to Appendix H. Note removed from end of Section 8.1.4.

* Recorded ticket information for Appendix H.41.

* Removed several comments that called out changes in -17 or earlier and for which there has been no response or further discussion. I have not removed the comments from Appendix I.1 and a few other places, and have not removed ones that seem current.

* Made the "trace" field-related changes as discussed on the mailing list. See specifically https://mailarchive.ietf.org/arch/msg/emailcore/ITYEtb_OhE_IfijfRfbqCNL_QDM and https://mailarchive.ietf.org/arch/msg/emailcore/H6Zqwm0KErXpINc6SqjL4FsotdU this version of the present document is intended to parallel draft-ietf-emailcore-rfc5322bis-06 as discussed in those notes.

I.3.24. Changes from draft-ietf-emailcore-rfc5321bis-19 (2023-07-23) to -20

* Corrected a few typographical or presentation errors including a missing word.

* Restored Section 4.4.2, which had been accidentally dropped in -10.

* Inserted a note in Section 8.1.1 to reflect the possibility of removing the "two models" discussion and, instead, referencing draft-klensin-iana-consid-hybrid or a RFC8126bis that incorporates that policy.

I.3.25. Changes from draft-ietf-emailcore-rfc5321bis-20 (2023-11-16) to -21

* Fixed more minor typographical errors (one so far).

* Divided Section 8.1.1 into two subsections and tuned text slightly. That should make it a bit easier to follow and, if IANA wants the document to specify exactly how the registry is laid out, would make that requirement easier to satisfy.

* Modified Section 8.1.1.2, Paragraph 2, Item 8 to incorporate RFC 6409 Table 1 values in the SMTP Extensions Registry.

* Added index entries for message submission.

* Small adjustments to the introductory material (xref target="HistoryContext"/>) to reflect the IANA Considerations restructuring.

I.3.26. Changes from draft-ietf-emailcore-rfc5321bis-21 (2023-11-27) to -22

* Edited Appendix B including new introductory text, removing a header-checking requirement that has been inappropriate (even in RFC 2821) given changes to the IMF spec, and minor editorial tweaks.

* Modified Section 7.9 to use new text suggested in response to Ticket #82 with no comments received after that.

* Additional <cref> comments that no longer seem useful have been dropped.

* Miscellaneous other small changes including an update to the Acknowledgments.

I.3.27. Changes from draft-ietf-emailcore-rfc5321bis-22 (2023-12-05) to -23

* Corrected use of 556 code in Section 4.2.4.2

* Moved more references to RFC 5322 to point to rfc5322bis.

* Added additional index entries for limits (such as lengths) and timeouts.

* Moved another item from Appendix G to Appendix H. Appending G is now empty substantively (i.e., no un-closed items).

* Modified the list of updates for rfc5321bis-22 above to incorporate one that was missed.

* Tweaked the Abstract to make the nits checker happier.

I.3.28. Changes from draft-ietf-emailcore-rfc5321bis-23 (2023-12-10) to -24

* Corrected many small editorial errors and poorly-written statements, especially those identified by Tim Wicinski.

* Found and fixed more "nbsp" problems.

I.3.29. Changes from draft-ietf-emailcore-rfc5321bis-24 (2024-01-11) to -25

* As usual, small editorial errors that have been noticed have been corrected.

* The discrepancy between the RFC 1035 preferred syntax and practices today (and since circa late 1986) has been flagged, with text and references adjusted.

* The proposed final change summary (currently Appendix I.4) has been folded in after correcting numerous small errors. Substantially identical text was posted to the EMAILCORE list on 2024-01-15 and no comments have been received. While suggestions will still be welcomed, the bar for signification changes rises and time grows short.

* Because no one has identified a case where it is still needed, the reference to RFC5322 (as distinct from 5322bis) has been dropped.

* Added a sentence to Section 4.2 to clarify that the PIPELINING extension does not change the principle that each command generates exactly one reply.

I.3.30. Changes from draft-ietf-emailcore-rfc5321bis-25 (2024-01-24) to -26

* Many small errors and glitches caught and fixed. Many thanks to John Levine, Alexey Melnikov, and Rob Sayre for the careful reading and aggressive and constructive nit-picking.

* Several small modifications to Section 1.2 to make the relationship to other work more clear.

* More adjustments to "ASCII" versus "US-ASCII".

* Corrected the descriptions of code 553 in Section 4.2.2 and Section 4.2.3 to include the VRFY case.

* Notes added to Section 4.2 and Section 4.2.4.3 about "SHOULD" versus "should" choices. WG feedback needed.

* Added a note for WG consideration and decision to Section 5 about multihoming and load balancing. The WG needs to decide what to do there. See the note for more information.

I.3.31. Changes from draft-ietf-emailcore-rfc5321bis-26 (2024-02-09) to -27

* More small editorial improvements.

* Reopened Appendices G and H to record new tickets, tentatively resolved ticket #88, and, in some cases, added proposed text to address others. Note that "proposed" in the context is the editor's best effort to summarize referenced list discussion prior to posting of this version, not a statement of preferences.

* These changes are not believed to affect Appendix I.4.

I.4. Summary of changes from RFC 5321 (published in October 2008) to <<This Document>>

As discussed in Section 1.2, this specification combines material from several earlier ones. The most numerous changes from RFC 5321 have been editorial in nature. Those changes have included correcting long-standing errors, improving terminology and its consistent use, updating references to documents that have been replaced, adding additional cross-references within the document, and reorganizing material to make it easier to follow. In general, those changes are not called out in the list below. The order of changes in the list below is not significant.

// EMAILCORE WG participants: I've made choices below about
// references to particular sections when those seemed appropriate
// and omitted others when that did not appear to add much value. If
// you disagree, speak up. Soon.

// RFC Editor: The list that follows was numbered in order to make
// review discussion convenient. Unless you prefer it, I'd rather
// have a bullet list in the final version to reinforce the "order
// not significant" message.

1. All of the outstanding errata [65] filed against RFC 5321 have been addressed. That list does include some editorial issues.

2. The discussion of SMTP Service Extensions and how they are registered with IANA has been extensively revised and a new registration model defined. The reasons for this are discussed in Section 2.2.2.

3. Corrected, updated, or clarified a few ABNF syntax errors.

4. Improved the descriptions of the applicability of several reply codes. Also included descriptions of codes added since RFC 5321 was published.

5. An index was added to make it easier for readers to find specific terminology, ABNF productions, command arguments, and so on. Several additional cross-references have been added for the same reasons.

6. Clarified the relationship between mail transactions, repeated uses of EHLO within an SMTP session, and command arguments and responses between transactions.

7. Improved the discussion of the distinction of Message Submission Agents (MSAs), particularly those described in RFC 6409 [48], and Mail Transfer Agents (MTAs) as exemplified by this specification. This document does not alter RFC 6409 in any way.

8. The discussion of "trace information" has been reworked to make it more clear and more consistent with the discussion in the Message Format specification [16]. While the textual changes are extensive, it is not believed that any of them make substantive changes to the SMTP definition.

Index

A C M S T

A

Argument Syntax
ALPHA Section 4.1.2, Paragraph 2, Item 1
Additional-Registered-Clauses Section 4.4.5, Paragraph 3.26.1
Addtl-Link Section 4.4.5
Addtl-Protocol Section 4.4.5
Argument Section 4.1.2
Atom Section 4.1.2
By-domain Section 4.4.5, Paragraph 3.10.1
Domain Section 4.1.2
Dot-string Section 4.1.2
Extended-Domain Section 4.4.5
For Section 4.4.5
Forward-Path Section 4.1.2
From-domain Section 4.4.5, Paragraph 3.8.1
General-address-literal Section 4.1.3
Greeting Section 4.2
ID Section 4.4.5
IPv4-address-literal Section 4.1.3
IPv6-addr Section 4.1.3
IPv6-address-literal Section 4.1.3
Keyword Section 4.1.2

Ldh-str Section 4.1.2
Let-dig Section 4.1.2
Link Section 4.4.5
Local-part Section 4.1.2
Mail-parameters Section 4.1.2
Mailbox Section 4.1.2
Opt-info Section 4.4.5
Path Section 4.1.2
Protocol Section 4.4.5
QcontentSMTP Section 4.1.2
Quoted-string Section 4.1.2
Rcpt-parameters Section 4.1.2
Reply-code Section 4.2
Reply-line Section 4.2
Return-path-line Section 4.4.5, Paragraph 3.2.1
Reverse-Path Section 4.1.2
Snum Section 4.1.3
Stamp Section 4.4.5, Paragraph 3.6.1
Standardized-tag Section 4.1.3
String Section 4.1.2
TCP-info Section 4.4.5
Time-stamp-line Section 4.4.5, Paragraph 3.4.1
Via Section 4.4.5
With Section 4.4.5
address-literal Section 4.1.2
atext Section 4.1.2, Paragraph 2, Item 2
dcontent Section 4.1.3
esmtp-keyword Section 4.1.2
esmtp-param Section 4.1.2
esmtp-value Section 4.1.2
h16 Section 4.1.3
ls32 Section 4.1.3
qtextSMTP Section 4.1.2
quoted-pairSMTP Section 4.1.2
sub-domain Section 4.1.2
textstring Section 4.2

C

Commands and Syntax
data Section 4.1.1.4, Paragraph 8, Item 1
ehlo Section 3.2, Paragraph 1; Section 4.1.1.1, Paragraph 1
expn Section 4.1.1.7, Paragraph 4, Item 1
help Section 4.1.1.8, Paragraph 5, Item 1
mail Section 4.1.1.2
noop Section 4.1.1.9, Paragraph 4, Item 1
quit Section 4.1.1.10, Paragraph 5, Item 1
rcpt Section 4.1.1.3, Paragraph 15

rset Section 4.1.1.5, Paragraph 4, Item 1
send, saml, soml Appendix F.6; Appendix H.15, Paragraph 1
vrfy Section 4.1.1.6, Paragraph 4, Item 1

M

Message Submission
As relays Section 3.6.2
Correcting messages Section 6.4, Paragraph 4
Domain names Section 2.3.5, Paragraph 2
Pointer to RFC 6409 Section 1.2, Paragraph 5; Section 2.1, Paragraph 4
Reply codes Section 4.2.4.2, Paragraph 2
SMTP Extension Registration Section 8.1.1.2, Paragraph 2, Item 8
With generated commands Appendix B

S

Sizes, Lengths, and Timeouts *_Section 4.5.3_*
Command Line length Section 4.5.3.1.4
DATA Termination Timeout Section 4.5.3.2.6
Data Block/ TCP Wait Timeout Section 4.5.3.2.5
Data Initiation Timeout Timeout Section 4.5.3.2.4
Domain name or number length Section 4.5.3.1.2
Exceeding Limits Section 4.5.3.1.9
Local part length Section 4.5.3.1.1
Mail Command Timeout Section 4.5.3.2.2
Message Content Size Section 4.5.3.1.7
Minimum Number of Recipients Section 4.5.3.1.8
Path lengths Section 4.5.3.1.3
RCPT Command Timeout Section 4.5.3.2.3
Reply Line length Section 4.5.3.1.5
Server Wait Timeout Section 4.5.3.2.7
Text Line length Section 4.5.3.1.6
Source Routes *_Appendix F.2_*
A-d-l Appendix F.2
At-domain Appendix F.2
Path Appendix F.2

T

Terminology
Address Section 2.3.11, Paragraph 1
Buffer Section 2.3.6, Paragraph 1
Commands and Replies Section 2.3.7, Paragraph 1
Domain Names Section 2.3.5, Paragraph 1
Gateway Section 2.3.10, Paragraph 2

Host Section 2.3.4, Paragraph 1
Lines Section 2.3.8, Paragraph 1
Mail Agent Section 2.3.3, Paragraph 1
Mail object Section 2.3.1, Paragraph 1
Message Content Section 2.3.9, Paragraph 1
Originator Section 2.3.10, Paragraph 1
Senders and Receivers Section 2.3.2, Paragraph 1
address RR Section 2.3.5, Paragraph 3
primary host name Section 2.3.5, Paragraph 4, Item 1
Ticket Index
1 Appendix H.1, Paragraph 1
10 Appendix H.32, Paragraph 3
11 Appendix G.7.16, Paragraph 1; Appendix H.5, Paragraph 1
12 Appendix H.36, Paragraph 1
13 Appendix H.10, Paragraph 2
14 Appendix H.11, Paragraph 1
15 Appendix H.26, Paragraph 1
16 Appendix H.14, Paragraph 1
17 Appendix H.12, Paragraph 1
18 Appendix H.13, Paragraph 1
19 Appendix H.9, Paragraph 1
2 Appendix H.2, Paragraph 2
20 Appendix H.15, Paragraph 1
21 Appendix H.27, Paragraph 4
22 Appendix I.1
23 Appendix I.1
24 Appendix I.1
25 Appendix I.1
26 Appendix I.1
27 Appendix I.1
28 Appendix I.1
29 Appendix I.1
3 Appendix H.3, Paragraph 1
30 Appendix H.4, Paragraph 1; Appendix I.1
40 Appendix G.8, Paragraph 1; Appendix H.19, Paragraph 2
42 Appendix H.21, Paragraph 1
43 Appendix H.20, Paragraph 1; Appendix H.22, Paragraph 2
48 Appendix H.23, Paragraph 1
5 Appendix H.6, Paragraph 1
50 Appendix H.18, Paragraph 1
52 Appendix H.16, Paragraph 1
53 Appendix H.17, Paragraph 1; Appendix H.28, Paragraph 2
55 Appendix H.39, Paragraph 5
56 Appendix H.34, Paragraph 2
58 Appendix H.24, Paragraph 1.1.1
59 Appendix H.24, Paragraph 1.2.1
6 Appendix H.8, Paragraph 1
60 Appendix H.24, Paragraph 1.3.1

61 Appendix H.24, Paragraph 1.4.1
62 Appendix G.17, Paragraph 3.5.1; Appendix H.24, Paragraph 1.5.1
63 Appendix G.17, Paragraph 3.6.1
64 Appendix H.25, Paragraph 1
67 Appendix H.33, Paragraph 3
68 Appendix H.35, Paragraph 2
7 Appendix I.1
75 Appendix H.38, Paragraph 2
81 Appendix H.40, Paragraph 1
82 Appendix H.41, Paragraph 2; Appendix H.42, Paragraph 2
89 Appendix G.26, Paragraph 2
9 Appendix H.7, Paragraph 1
90 Appendix G.27, Paragraph 2
91 Appendix G.28, Paragraph 2

Author's Address

John C. Klensin
1770 Massachusetts Ave, Suite 322
Cambridge, MA 02140
United States of America
Email: john-ietf@jck.com