정보통신단체표준 제정일 : 2003년 10월 6일
TTAS.IF-RFC2821

단순 우편 전송 규약 표준
(Simple Mail Transfer Protocol)

TTA 한국정보통신기술협회
Telecommunications Technology Association

서 문

1. 표준의 목적

본 표준은 인터넷 전자 우편 전송의 기본 프로토콜에 대하여 필요한 모든 사항을 포함한 규격이다. 이 문서는 기능을 통합하고 갱신하며 분명하게 하지만 다음과 같은 기존 기능을 추가하거나 변경하지는 않는다.

- RFC 821에 대한 원래의 SMTP(Simple Mail Transfer Protocol) 규격
- RFC 1035 및 RFC 97의 메일전송을 위한 도메인 이름의 시스템 요구사항 및 관련사항
- RFC 1123에서의 설명 및 적용 정보
- SMTP 확장 메커니즘에서 발췌한 자료

2. 참조 권고 및 표준

2.1 국제표준(권고) :

- IETF RFC 2821 Simple Mail Transfer Protocol

2.2 국내 표준 :

- KICS.IF-RFC821 단순우편전송규약(SMTP) 표준
- KICS.IF-RFC822 전자우편 메시지 형식(MAIL) 표준
- KICS.IF-RFC1123 인터네트 호스트 요구사항 - 응용 및 지원 표준

2.3 기타 : 해당사항 없음

3. 국제표준(권고)과의 비교

3.1 국제표준(권고)과의 관련성

본 표준은 IETF(Internet Engineering Task Force)에서 제정한 RFC 2821 사실표준을 수용하였다.

3.2 상기 국제 표준(권고) 등에 대한 추가 사항 등 :

이 표준은 RFC821 및 RFC974를 사용하지 않고, RFC 1123의 메일 전송 자료를 갱신한다. 그러나, RFC 821에는 1990년대 중반까지는 인터넷에서 자주 사용되지 않았던 일부 기능들을 명시하고 있으며, (부록에서) 몇 개의 전송 모델이 추가적으로 제시되어 있다.

이러한 내용을 분명하게 하고 분량을 줄이기 위해서 관련 내용을 여기에서는 생략하였으므로 이러한 내용이 필요할 경우, RFC 821을 참조한다. 또한 이 표준에는 부연 설명을 필요로 하는 RFC 1123에서 발췌한 몇 가지 추가 자료가 포함되어 있다. 이 자료는 다양한 목록 및 뉴스그룹에 초점을 둔 추적과 SMTP 확장이 배포될 때 표시되는 특별한 판독이나 해석 문제에 의한 다양한 방식이 확인되어 있다. 본 표준이 통합 차원을 벗어나서 사실상 이전 문서와 다른 경우, 이 표준이 기술적으로나 내용적으로 그러한 문서들을 대체한다.

SMTP가 메일 전송 및 배달 프로토콜로서 설계되었을 지라도, 이 표준에는 또한 POP [3, 26]과 IMAP [6]에 권장되어 있는 바와 같이 메일 의뢰(submission) 프로토콜에서처럼 사용 용도에 대한 중요한 정보가 들어 있다. 추가적인 메일 의뢰에 대한 설명은 RFC 2476 [15]에 적혀있다. 관련문서 RFC2822[32]는 메시지 헤더, 메시지 본문, 그에 대한 형식과 구조 및 관계를 설명한다. RFC 2821은 다음 문헌을 참조하여 작성되었다.

[1] American National Standards Institute (formerly United States of America Standards Institute), X3.4, 1968, "USA Code for Information Interchange". ANSI X3.4-1968 has been replaced by newer versions with slight modifications, but the 1968 version remains definitive for the Internet.

[2] Braden, R., "Requirements for Internet hosts - application and support", STD 3, RFC 1123, October 1989.

[3] Butler, M., Chase, D., Goldberger, J., Postel, J. and J. Reynolds, "Post Office Protocol - version 2", RFC 937, February 1985.

[4] Callas, J., Donnerhacke, L., Finney, H. and R. Thayer, "OpenPGP Message Format", RFC 2440, November 1998.

[5] Crispin, M., "Interactive Mail Access Protocol - Version 2", RFC 1176, August 1990.

[6] Crispin, M., "Internet Message Access Protocol - Version 4", RFC 2060, December 1996.

[7] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", RFC 822, August 1982.

[8] Crocker, D. and P. Overell, Eds., "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[9] De Winter, J., "SMTP Service Extension for Remote Message Queue Starting", RFC 1985, August 1996.

[10] Fajman, R., "An Extensible Message Format for Message Disposition Notifications", RFC 2298, March 1998.

[11] Freed, N, "Behavior of and Requirements for Internet Firewalls", RFC 2979, October 2000.

[12] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, December 1996.

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

[14] Galvin, J., Murphy, S., Crocker, S. and N. Freed, "Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted", RFC 1847, October 1995.

[15] Gellens, R. and J. Klensin, "Message Submission", RFC 2476, December 1998.

[16] Kille, S., "Mapping between X.400 and RFC822/MIME", RFC 2156, January 1998.

[17] Hinden, R and S. Deering, Eds. "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

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

[19] Klensin, J., Freed, N., Rose, M., Stefferud, E. and D. Crocker, "SMTP Service Extensions", STD 10, RFC 1869, November 1995.

[20] Klensin, J., Freed, N., Rose, M., Stefferud, E. and D. Crocker, "SMTP Service Extension for 8bit-MIMEtransport", RFC 1652, July 1994.

[21] Lambert, M., "PCMAIL: A distributed mail system for personal computers", RFC 1056, July 1988.

[22] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[23] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, December 1996.

[24] Moore, K., "SMTP Service Extension for Delivery Status Notifications", RFC 1891, January 1996.

[25] Moore, K., and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 1894, January 1996.

[26] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.
[27] Partridge, C., "Mail routing and the domain system", RFC 974, January 1986.
[28] Partridge, C., "Duplicate messages and SMTP", RFC 1047, February 1988.
[29] Postel, J., ed., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", STD 7, RFC 793, September 1981.
[30] Postel, J., "Simple Mail Transfer Protocol", RFC 821, August 1982.
[31] Ramsdell, B., Ed., "S/MIME Version 3 Message Specification", RFC 2633, June 1999.
[32] Resnick, P., Ed., "Internet Message Format", RFC 2822, April 2001.
[33] Vaudreuil, G., "SMTP Service Extensions for Transmission of Large and Binary MIME Messages", RFC 1830, August 1995.
[34] Vaudreuil, G., "Enhanced Mail System Status Codes", RFC 1893, January 1996.

3.2.1 선택 항목
3.2.2 National Matter 항목
3.2.3 기타 항목

3.3 참조한 국제표준(권고)과 본 표준의 장 구성 비교표

RFC 1652	본 표준
Abstract	서문과 2. 표준의 구성 및 범위
1. Introduction	1.개요
2. SMTP Model	3. SMTP 모델
3. The SMTP Procedures : An Overview	4. SMTP 절차 : 개요
4. The SMTP Specifications	5. SMTP 규격
5. Address Resolution and Mail handling	6. 주소결정과 메일처리
6. Problem Detection and Handling	7. 문제 발견 및 처리
7. Security Consideration	8. 보안고려사항
8. IANA Considerations	9. IANA 고려사항
9. References	서문
10. Editor's Address	생략
11. Acknowledgements	생략
Appendix A. TCP Transfer Service	부록 I. TCP 전송 서비스
Appendix B. Generating SMTP Commands from RFC 322 Headers	부록 II. RFC822 헤더에서 SMTP 명령 생성
Appendix C. Source Routes	부록 III. 소스 라우트
Appendix D. Scenarios	부록 IV. 시나리오
Appendix E. Other Gateway Issues	부록 V. 다른 게이트웨이 문제
Appendix F. Deprecated Features of RFC 821	부록 VI. RFC821에서 사용이 금지된 기능
Full Copyright Statement	서문

4. 지적 재산권 관련 사항 :

는 것을 전제로 한다. 그러나, 인터넷 표준 작업에 정의된 저작권 절차를 준수해야 하는 경우에 있어 인터넷 표준 개발 목적에 필요한 경우 또는 영어 이외의 언어로 번역해야 하는 경우를 제외하고는, 인터넷 협회(Internet Society) 또는 그 밖의 인터넷 회사에 저작권 정보 또는 참고 자료를 제거하는 등 어떠한 방법으로든 이 문서 자체를 변조할 수 없다.

위에 제공된 제한된 권한은 영구적이며 인터넷 협회(Internet Society) 또는 그 후속 기관이나 양수인에 의해 폐지되지 않을 것이다.

이 문서와 여기에 포함된 정보는 “있는 그대로”로 제공되며 인터넷 협회(Internet Society)와 인터넷 엔지니어링 특별 조사단은 여기에서의 정보 사용이 모든 권한을 위배하지 않는다는 보증이나 시장성 또는 특수한 목적을 위한 적합성에 대한 암시적 보증으로만 제한하지 않고, 명시적 또는 암시적 모든 보증을 부인한다.

5. 적합 인증 관련 사항 : 해당사항 없음

6. 표준의 이력

판수	제・개정일	제정판 내용
제1판	2003 년 10월 6일	제정

Preface

1. Objective

This document is a self-contained specification of the basic protocol for the Internet electronic mail transport. It consolidates, updates and clarifies, but doesn't add new or change existing functionality of the following:

- the original SMTP (Simple Mail Transfer Protocol) specification of RFC 821 [30],
- domain name system requirements and implications for mail transport from RFC 1035 [22] and RFC 974 [27],
- the clarifications and applicability statements in RFC 1123 [2],

and

- material drawn from the SMTP Extension mechanisms [19].

2. Normative standards and recommendations

2.1 International Standards

- IETF RFC 2821 Simple Mail Transfer Protocol

2.2 Domestic Standards

- KICS.IF-RFC821 단순우편전송규약(SMTP) 표준
- KICS.IF-RFC822 전자우편 메시지 형식(MAIL) 표준
- KICS.IF-RFC1123 인터네트 호스트 요구사항 - 응용 및 지원 표준

2.3 Others : N/A

3. The Comparison of international standards

3.1 Relation to International Standards

본 표준은 IETF(Internet Engineering Task Force)에서 제정한 RFC 2821 사실표준을 수용하였다.

3.2 Addition of International Standards

It obsoletes RFC 821, RFC 974, and updates RFC 1123 (replaces the mail transport

materials of RFC 1123). However, RFC 821 specifies some features that were not in significant use in the Internet by the mid-1990s and (in appendices) some additional transport models. Those sections are omitted here in the interest of clarity and brevity; readers needing them should refer to RFC 821.

It also includes some additional material from RFC 1123 that required amplification. This material has been identified in multiple ways, mostly by tracking flaming on various lists and newsgroups and problems of unusual readings or interpretations that have appeared as the SMTP extensions have been deployed. Where this specification moves beyond consolidation and actually differs from earlier documents, it supersedes them technically as well as textually.

Although SMTP was designed as a mail transport and delivery protocol, this specification also contains information that is important to its use as a 'mail submission' protocol, as recommended for POP [3, 26] and IMAP [6]. Additional submission issues are discussed in RFC 2476 [15].

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 the identify the sending and receiving SMTP processes, respectively.

A companion document [32] discusses message headers, message bodies and formats and structures for them, and their relationship.

References

[1] American National Standards Institute (formerly United States of America Standards Institute), X3.4, 1968, "USA Code for Information Interchange". ANSI X3.4-1968 has been replaced by newer versions with slight modifications, but the 1968 version remains definitive for the Internet.

[2] Braden, R., "Requirements for Internet hosts - application and support", STD 3, RFC 1123, October 1989.

[3] Butler, M., Chase, D., Goldberger, J., Postel, J. and J. Reynolds, "Post Office Protocol - version 2", RFC 937, February 1985.

[4] Callas, J., Donnerhacke, L., Finney, H. and R. Thayer, "OpenPGP Message Format", RFC 2440, November 1998.

[5] Crispin, M., "Interactive Mail Access Protocol - Version 2", RFC 1176, August 1990.

[6] Crispin, M., "Internet Message Access Protocol - Version 4", RFC 2060, December

1996.

[7] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", RFC 822, August 1982.

[8] Crocker, D. and P. Overell, Eds., "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[9] De Winter, J., "SMTP Service Extension for Remote Message Queue Starting", RFC 1985, August 1996.

[10] Fajman, R., "An Extensible Message Format for Message Disposition Notifications", RFC 2298, March 1998.

[11] Freed, N, "Behavior of and Requirements for Internet Firewalls", RFC 2979, October 2000.

[12] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, December 1996.

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

[14] Galvin, J., Murphy, S., Crocker, S. and N. Freed, "Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted", RFC 1847, October 1995.

[15] Gellens, R. and J. Klensin, "Message Submission", RFC 2476, December 1998.

[16] Kille, S., "Mapping between X.400 and RFC822/MIME", RFC 2156, January 1998.

[17] Hinden, R and S. Deering, Eds. "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

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

[19] Klensin, J., Freed, N., Rose, M., Stefferud, E. and D. Crocker, "SMTP Service Extensions", STD 10, RFC 1869, November 1995.

[20] Klensin, J., Freed, N., Rose, M., Stefferud, E. and D. Crocker, "SMTP Service Extension for 8bit-MIMEtransport", RFC 1652, July 1994.

[21] Lambert, M., "PCMAIL: A distributed mail system for personal computers", RFC 1056, July 1988.

[22] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034,

November 1987.

[23] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, December 1996.

[24] Moore, K., "SMTP Service Extension for Delivery Status Notifications", RFC 1891, January 1996.

[25] Moore, K., and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 1894, January 1996.

[26] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, May 1996.

[27] Partridge, C., "Mail routing and the domain system", RFC 974, January 1986.

[28] Partridge, C., "Duplicate messages and SMTP", RFC 1047, February 1988.

[29] Postel, J., ed., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", STD 7, RFC 793, September 1981.

[30] Postel, J., "Simple Mail Transfer Protocol", RFC 821, August 1982.

[31] Ramsdell, B., Ed., "S/MIME Version 3 Message Specification", RFC 2633, June 1999.

[32] Resnick, P., Ed., "Internet Message Format", RFC 2822, April 2001.

[33] Vaudreuil, G., "SMTP Service Extensions for Transmission of Large and Binary MIME Messages", RFC 1830, August 1995.

[34] Vaudreuil, G., "Enhanced Mail System Status Codes", RFC 1893, January 1996.

3.2.1 Optional parts

3.2.2 National Matters

3.2.3 Others

3.3 Chapter sequence comparison with referenced International Standards

RFC 1652	본 표준
Abstract	서문과 2. 표준의 구성 및 범위
1. Introduction	1.개요
2. SMTP Model	3. SMTP 모델
3. The SMTP Procedures : An Overview	4. SMTP 절차 : 개요
4. The SMTP Specifications	5. SMTP 규격
5. Address Resolution and Mail handling	6. 주소결정과 메일처리
6. Problem Detection and Handling	7. 문제 발견 및 처리
7. Security Consideration	8. 보안고려사항
8. IANA Considerations	9. IANA 고려사항
9. References	서문
10. Editor's Address	생략
11. Acknowledgements	생략
Appendix A. TCP Transfer Service	부록 I. TCP 전송 서비스
Appendix B. Generating SMTP Commands from RFC 322 Headers	부록 II. RFC822 헤더에서 SMTP 명령 생성
Appendix C. Source Routes	부록 III. 소스 라우트
Appendix D. Scenarois	부록 IV. 시나리오
Appendix E. Other Gateway Issues	부록 V. 다른 게이트웨이 문제
Appendix F. Deprecated Features of RFC 821	부록 VI. RFC821에서 사용이 금지된 기능
Full Copyright Statement	서문

4. Intellectual property rights

notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.
The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

5. Requirements for conformance and certification : N/A

6. History

Edition	Issued Date	Contents
The 1st edition	Dec. 6. 2002	Established

목 차

Index

2003. 10. 6.

단순 우편 전송 규약 표준
Simple Mail Transfer Protocol

1. 개 요

SMTP(Simple Mail Transfer Protocol)의 목적은 신뢰할 수 있고 효율적으로 메일을 전송하는 것이다. SMTP는 전송을 담당하는 하위 시스템과는 독립되어 있으며 신뢰할 수 있는 순서로 배열된 데이터 스트림 채널만을 필요로 한다. 본 표준은 특별히 TCP 상에서의 전송을 설명하지만, 다른 전송방법도 사용할 수 있다. RFC 821에 있는 부록에서 다른 전송방법 중 몇 가지를 설명하고 있다.

SMTP의 중요한 기능 중 하나는 흔히 "SMTP 메일 릴레이"(4.8절 참조)라고 하는 여러 네트워크에 걸쳐서 메일을 전송하는 기능이다. 네트워크는 공용 인터넷 상에서 TCP를 이용하여 접근할 수 있는 호스트, 방화벽으로 분리된 TCP/IP 인트라넷 상에서의 TCP를 이용하여 접근할 수 있는 호스트 또는 TCP가 아닌 다른 전송방법을 사용하는 그 밖의 LAN 또는 WAN 환경에서의 호스트로 구성된다. SMTP를 사용하면, 프로세스는 동일한 네트워크에 있는 다른 프로세스로 메일을 전송할 수 있으며, 또한 양쪽 네트워크에 접근할 수 있는 릴레이나 게이트웨이 프로세스를 경유하게 되면 다른 네트워크에 있는 프로세스에게도 메일을 전달할 수도 있다.

2. 표준의 구성 및 범위

본 표준의 3장에서는 SMTP 모델을 정의하고 본 표준에서 사용하고 있는 용어를 정의하고 있다. 모호함을 없애기 위해 이전에 사용되었던 용어가 필요한 경우가 아니라면, 이 문서는 현재의 '클라이언트'와 '서버'라는 용어를 사용하여 각각 전송 및 수신 SMTP프로세스를 구별한다. 4장에서는 SMTP의 절차를 설명하고 5장에서는 명령과, 답장 그리고 상호간의 순서 등을 지정하고 있다. 6장에서는 주소를 결정(Resolution)하고 메일을 처리하는 내용을 7장에서는 문제점의 처리를 설명하고 있다. 8,9장에서는 고려사항을 설명하고 있다. 본 표준은 기능을 통합하고 갱신하며 분명하게 하지만 다음과 같은 기존 기능을 추가하거나 변경하지는 않는다.

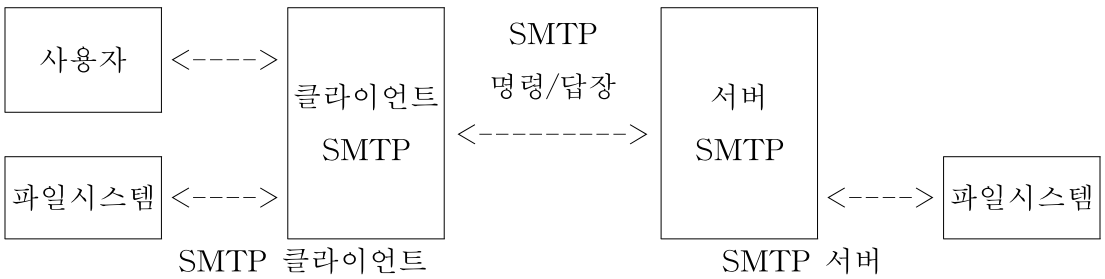
- RFC 821에 대한 원래의 SMTP(Simple Mail Transfer Protocol) 규격
- RFC 1035 및 RFC 97의 메일전송을 위한 도메인 이름의 시스템 요구사항 및 관련사항
- RFC 1123에서의 설명 및 적용 정보
- SMTP 확장 메커니즘에서 발췌한 자료 badge promissory balance

본 표준은 RFC821 및 RFC974를 사용하지 않고, RFC 1123의 메일 전송 자료를 갱신한다. 그러나, RFC 821에는 1990년대 중반까지는 인터넷에서 자주 사용되지 않았던 일부 기능들을 명시하고 있으며, (부록에서) 몇 개의 전송 모델이 추가적으로 제시되어 있다. 이러한 내용을 분명하게 하고 분량을 줄이기 위해서 관련 내용을 여기에서는 생략하였으므로 이러한 내용이 필요할 경우, RFC 821을 참조한다. 또한 이 표준에는 부연 설명을 필요로 하는 RFC 1123에서 발췌한 몇 가지 추가 자료가 포함되어 있다. 이 자료는 다양한 목록 및 뉴스그룹에 초점을 둔 추적과 SMTP 확장이 배포될 때 표시되는 특별한 판독이나 해석 문제에 의한 다양한 방식이 확인되어 있다. 본 표준이 통합 차원을 벗어나서 사실상 이전 문서와 다른 경우, 본 표준이 기술적으로나 내용적으로 그러한 문서들을 대체한다. SMTP가 메일 전송 및 배달 프로토콜로서 설계되었을 지라도, 이 표준에는 또한 POP [3, 26]과 IMAP [6]에 권장되어 있는 바와 같이 메일 의뢰(submission) 프로토콜에서처럼 사용 용도에 대한 중요한 정보가 들어 있다. 추가적인 메일 의뢰에 대한 설명은 RFC 2476 [15]에 적혀있다. 관련문서 RFC2822[32]는 메시지 헤더, 메시지 본문, 그에 대한 형식과 구조 및 관계를 설명한다.

3. SMTP 모델

3.1 기본 구조

SMTP는 다음 그림과 같이 설계할 수 있다.



SMTP 클라이언트에 전송할 메시지가 있는 경우, 클라이언트는 SMTP 서버에 대하여 양방향 전송 채널을 설정한다. SMTP 클라이언트의 임무는 하나 또는 그 이상의 SMTP 서버에 메일 메시지를 전송하거나 전송이 실패하였다면 이들에게 메일의 메시지 전송 실패를 보고하는 것이다.

메일 메시지가 SMTP 클라이언트에 제공되는 방법과 클라이언트에서 메일 메시지가 전송될 도메인 이름을 결정하는 방법은 한 지역에 국한된 문제이므로 이 문서에서는 설명하지 않는다. 어떤 경우에는, SMTP 클라이언트로 전송되거나 SMTP 클라이언트에 의해 결정된 도메인 이름(들)이 메일 메시지의 최종 목적지(들)를 식별할 것이다. 그밖에 SMTP 클라이언트가 POP [3, 26] 또는 IMAP [6] 프로토콜과 함께 구현된 경우 또는 SMTP 클라이언트가 고립된 전송 서비스 환경의 내부에 있는 경우, 결정된 도메인 이름은 모든 메일 메시지가 릴레이 되어야 하는 중간 목적지를 식별할 것이다. 개별 메시지에 적힌 대상 도메인 이름이 무엇이던지 간에 모든 트래픽을 전송하는 SMTP 클라이언트와, 한번에 완료할 수 없는 메시지 전송을 재 시도하기 위한 대기열을 유지하지 않는 SMTP 클라이언트는 이 표준을 따른다고 할 수는 있지만 완전한 기능을 갖추었다고 간주되지는 않는다. 약간은 부족한 기능을 가지고 있는 구현에 의해 사용되는 중간릴레이와 목적지를 포함하여 완전한 기능을 갖춘 SMTP의 구현에는 이 표준에 설명된 모든 대기열 저장, 재시도 및 다른 주소 사용 기능이 지원된다고 가정한다.

본 표준은 먼저 대상 도메인 이름을 결정하고 나서, SMTP 클라이언트에서 메시지의 사본이 전송될 SMTP 서버의 ID를 판단한 다음, 해당하는 전송을 수행하는 방법을 설명한다. SMTP 서버로 메일을 전송하기 위해 SMTP 클라이언트는 해당 SMTP 서버로 양방향 전송 채널을 설정한다. SMTP 클라이언트는 대상 도메인 이름을 중간 메일 익스체인저(eXchanger) 호스트 또는 최종 대상 호스트로 변환하여 SMTP 서버의 역할을 수행하는 호스트의 주소를 결정한다.

SMTP 서버는 최종 대상이나 중간 “릴레이”(즉, 메시지를 수신한 이후 SMTP 클라이언트의 역할을 가정할 수도 있음) 또는 “게이트웨이”(즉, SMTP 이외의 특정 프로토콜을 사용하여 추후에 메시지를 전송할 수도 있음)가 될 수 있다. SMTP 명령은 SMTP 클라이언트에 의해 생성되어 SMTP 서버로 전송된다. SMTP 답장은 SMTP 서버로부터 명령에 대한 응답으로 SMTP 클라이언트에게 전송된다.

다른 말로 하면, 메시지 전송이 원래의 SMTP 발신자와 최종 SMTP 수신자간에 단일 연결에서 발생하거나 중간 시스템을 경유하는 일련의 홉(hop)에서 발생할 수 있다. 어느 경우든, 메시지에 대한 공식적인 책임 이전이 발생한다. 프로토콜은 서버가 메시지를 전달하거나 메시지 전달 실패를 올바르게 보고할 책임을 수락할 것을 요구한다.

일반적으로 전송 채널이 설정되고 초기 핸드셰이킹이 완료되고 나면, SMTP 클라이언트는 메일 트랜잭션을 초기화한다. 이러한 트랜잭션은 발기인(originator), 메일의 대상 및 메시지 내용(헤더 또는 다른 구조 포함) 자체의 전송을 지정하는 여러 명령들로 이루어진다. 동일한 메시지가 다수의 수신자에게 전송되는 경우 이 프로토콜은 동일한 대상(또는 중간 릴레이) 호스에 존재하는 모든 수신자를 위해서 하나의 데이터 사본만을 전송할 것을 장려한다.

서버는 응답으로 각 명령에 답장한다. 응답은 명령이 수락되었거나, 추가 명령이 예상되거나, 아니면 임시 또는 영구적인 오류 조건이 존재함을 나타낼 수 있다. 발신자 또는 수신자를 지정하는 명령에는 3.2에서 설명되는 것처럼 서버에서 허용하는 SMTP 서비스 확장이 포함될 수 있다. 이 다이얼로그는 명령 파이프라인 [13]과 같은 상호 합의된 확장 요청으로 인해 변경할 수 있을 지라도 의도적으로 한 번에 하나씩만 주고받을 수 있는 경직된 형태이다.

해당 메일 메시지가 전송되고 나면, 클라이언트는 연결 종료를 요청하거나 다른 메일 트랜잭션의 시작을 요청할 수 있다. 또한, SMTP 클라이언트는 전자 메일 주소의 검증이나 가입자 주소의 메일링 목록 검색과 같은 보조 서비스를 이용하기 위해서 SMTP 서버와의 연결을 사용할 수 있다.

위에서 제안했던 바와 같이, 이 프로토콜은 메일 전송 메커니즘을 제공한다. 대개 매일 전송은 두 개 호스트가 동일한 전송 서비스를 사용하고 있을 때 보내는 사용자의 호스트에서 받는 사용자의 호스트로 직접 발생한다. 동일한 전송 서비스를 사용하고 있지 않은 경우에 하나 이상의 릴레이 SMTP 서버를 거쳐서 메일이 전송된다. SMTP 릴레이 또는 다른 전송 환경으로의 게이트웨이로서 작동하는 중간 호스트는 일반적으로 도메인 이름 서비스(DNS)의 메일 익스체인저(Mail eXchanger) 메커니즘을 사용하여 선택한다.

일반적으로 중간 호스트는 DNS의 MX 레코드를 통해서 결정되며, 명시적인 방법인 "소스 라우팅"(6장과 부록 III과 VI.2 참조)을 통해서 결정되지는 않는다.

3.2 확장 모델

3.2.1 배경

RFC 821이 완성된 이후 1990년에 시작되어 거의 10년에 걸친 노력 끝에, 이 프로토콜은 클라이언트와 서버가 원래의 SMTP 요구사항 이외의 공유 기능을 이용하는 데 합의할 수 있도록 해주는 “서비스 확장” 모델로 변경되었다. SMTP 확장 메커니즘은 확장된 SMTP 클라이언트 및 서버가 서로를 인식할 수 있는 방법을 정의하고 있으며, 서버는 자신이 지원할 수 있는 확장된 서비스가 무엇인지를 클라이언트에게 알려줄 수 있다.

최신 SMTP 구현 제품은 기본적인 확장 메커니즘을 지원해야 한다. 예를 들면, 서버는 특별한 서비스 확장을 구현하지 않는 경우라도 EHLO 명령을 지원해야 하며 클라이언트는 HELO가 아닌 EHLO를 우선적으로 사용해야 한다. (그러나, 이전 표준에 부합되는 구현 제품과의 호환성을 위해, SMTP 클라이언트와 서버는 대체 시스템으로서 원래의 HELO 메커니즘 또한 지원해야 한다.) 상호 작동성의 목적으로 HELO의 다른 특성들을 식별해야 하는 경우가 아니라면 이 문서는 EHLO만을 설명한다.

SMTP는 널리 배포되어 있으며 잘 구현된 제품에서 그 견고함이 잘 입증되었다. 그러나, 인터넷 업종에 종사하는 사람들은 프로토콜이 처음 설계되었을 때 예상하지 못했던 일부 서비스들이 중요하다고 생각하게 되었다. 이러한 서비스들을 추가하는 경우에 추가한 작업이 이전에 구현되어 사용되고 있는 제품을 계속해서 만족스럽게 작동하게 해주어야 한다. 확장 구조는 다음과 같이 구성된다.

- 이전 HELO를 대신하는 SMTP 명령 EHLO
- SMTP 서비스 확장의 등록
- SMTP MAIL과 RCPT 명령에 대하여 추가되는 매개변수
- 비ASCII도 전송되는 [33] DATA와 같이 이 프로토콜에서 정의된 명령의 선택적 요구사항

SMTP의 장점은 기본적으로는 단순함에서 비롯된다. 다양한 프로토콜을 사용해보면 옵션을 거의 제공하지 않는 프로토콜이 대부분이며 옵션을 많이 제공하는 프로토콜은 명료하지 못한 경우가 많다.

확장의 이점이 무엇인가를 불문하여 모든 확장은 구현, 배포 및 상호 작동성 비용 면에서 신중하고 세밀하게 살펴보아야 한다. 많은 경우에 SMTP 서비스를 확장하는 데 드는 비용이 그 확장이 제공하는 이점을 능가한다.

3.2.2 확장의 정의 및 등록

IANA는 SMTP 서비스 확장에 관한 등록사항을 담당하고 있다. 해당 EHLO 키워드는 각 확장에 대해서 값이 매겨져 있다. IANA를 통해 등록된 각 서비스 확장은 공식적인 표준(Standards Track)또는 IESG에서 승인한 실험적 프로토콜 문서에 정의되어야 한다. 정의에는 다음 정보가 포함되어야 한다.

- SMTP 서비스 확장의 원문 이름
- 확장과 관련된 EHLO 키워드 값
- EHLO 키워드 값과 관련된 매개변수로서 사용할 수 있는 값과 구문
- 확장과 관련되어 추가되는 SMTP 동사(verb)
  (대개의 경우 추가되는 동사는 EHLO 키워드 값과 동일하겠지만 반드시 동일할 필요는 없다.)
- MAIL 또는 RCPT 동사에 사용되는 것으로 확장에 의해 새로 만들어진 모든 매개변수
- 확장을 이루기 위한 지원책들이 서버 및 클라이언트 SMTP의 작동에 어떠한 영향을 미치는 지에 대한 설명
- 이 표준에서 서술한 표준에 따라서, 확장이 MAIL 및 RCPT 명령의 최대 길이를 증가시키는 증가분

또한, 대문자 또는 소문자 “X″로 시작하는 모든 EHLO 키워드 값은 로컬 SMTP 서비스를 확장하는 과정에서 상호 약속을 통해 독점적으로 사용된다. ”X″로 시작하는 키워드는 등록된 서비스 확장에 사용되어서는 안 된다. 반대로, “X″로 시작하지 않은 EHLO 응답에 제공된 키워드 값은 표준, 최종표준에 중간적 형태의 표준 또는 IANA를 통해 등록된 IESG에서 승인한 실험적 SMTP 서비스 확장에 부합되어야 한다. 이러한 규칙에 부합되는 서버는 서비스 확장으로 등록하지 않은 ”X″ 접두사가 붙지 않은 키워드 값을 제공해서는 안 된다.

추가 동사 및 매개변수 이름에는 EHLO 키워드와 동일한 규칙이 적용된다. 특히, “X"로 시작하는 동사는 등록이나 표준화되지 않아도 사용할 수 있는 로컬 확장을 의미한다. 반대로, "X"로 시작하지 않은 동사는 항상 등록한 후 사용해야 한다.

3.3 용어

이 문서의 영어 원문에서 "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY" 및 "OPTIONAL" 키워드는 아래 설명된 바와 같이 해석되었다.

1) MUST : 이 단어 또는 "REQUIRED"나 "SHALL" 용어는 정의가 규격의 필수 요구사항임을 의미한다.

2) MUST NOT : 이 어구 또는 "SHALL NOT" 어구는 정의가 규격의 필수 금지 사항임을 의미한다.

3) SHOULD : 이 단어 또는 "RECOMMENDED" 형용사는 특수한 상황에서 특정 항목을 무시해야 할 합당한 이유가 있을 때에는 사용하지 않아도 무방하다. 그러나 이 단어가 포함된 문장을 선택하지 않으려면 전체 의미를 이해하고 우선 상황을 신중히 검토해야 한다.

4) SHOULD NOT : 이 어구 또는 "NOT RECOMMENDED" 어구는 특수한 상황에서 특수한 작동을 수용해야 할 합당한 이유가 있을 때는 사용하여도 무방하다. 그러나 이 어구가 포함된 문장을 선택하려면 우선 상황을 신중히 검토해야 한다.

5) MAY : 이 단어 또는 "OPTIONAL" 형용사는 실제로 항목을 임의로 선택할 수 있음을 의미한다. 어떤 벤더는 이 항목이 특정 시장에서 필요로 하거나 제품을 향상시킨다고 판단하여 추가할 수도 있지만, 다른 벤더는 동일한 항목을 포함시키지 않을 수도 있다. 특수한 옵션을 포함하지 않은 채로 구현한 제품은 해당 옵션을 포함하는 다른 구현 제품과의 상호 작동성을 지원해야 한다. 마찬가지로, 특수한 옵션을 포함하는 구현 제품은 해당 옵션을 포함하지 않는 다른 구현 제품과의 상호 작동성을 지원해야 한다.(물론 해당 옵션이 제공하는 기능에 대해서는 제외됨)

3.3.1 메일 객체

SMTP는 메일 객체를 전송한다. 메일 객체에는 엔벨로프(envelope)와 컨텐츠가 들어 있다.

SMTP 엔벨로프는 일련의 SMTP 프로토콜 단위(4장에서 설명함)로서 전송된다. SMTP 엔벨로프는 발기인 주소(오류 보고서가 전달될 곳), 하나 이상의 수신자 주소 및 프로토콜 확장 요소(선택 사항)로 구성된다. 예전에는 수신자 주소 규격 명령(RCPT TO)에 대하여 직접적인 표시와 같은 별도의 배달 모드를 지정할 수 있다. 그러나 요즈음 이러한 변경은 신통치 않은 방법으로 평가절하 되고 있다(부록 VI의 VI. 6절 참조).

SMTP 컨텐츠는 SMTP DATA 명령 프로토콜 단위로 전송되며 헤더와 본문이라는 두 가지 부분이 있다. 컨텐츠가 본 표준 외의 최신 표준에 부합되는 경우 헤더는 메시지 형식 규격 [32]에서처럼 일련의 구조화된 필드/값을 가진다. 구조화된 경우, 본문은 MIME [12]에 따라 정의된다. 컨텐츠는 원래대로 텍스트 형태로 되어 있으며 US-ASCII 레퍼토리[1]를 사용하여 표현된다. SMTP 확장(“8BITMIME″ [20] 등)이 컨텐츠 본문에 관한 이 제한을 완화시켰지만, 컨텐츠 헤더는 항상 US-ASCII 레퍼토리를 사용하여 인코드 된다. MIME 확장 [23]은 US-ASCII 레퍼토리 외부의 헤더 값을 표시하는 알고리즘을 정의하고 있기는 하지만, 아직까지는 실제적으로 US-ASCII 레퍼토리만을 사용하고 있다.

3.3.2 발신자와 수신자

RFC 821에서는 SMTP 트랜잭션에 참여하는 두 개 호스트를 “SMTP 발신자”와 “SMTP 수신자”로 명명했다. 이 문서는 현재의 산업 기술의 변화를 반영하여 이 두 개 요소를 각각 ″SMTP 클라이언트“(또는 간단히 ”클라이언트“)와 “SMTP 서버”(또는 간단히 “서버”)로 바꿔서 지칭한다. 제공된 호스트가 중계소에서는 서버와 클라이언트 모두로 사용될 수 있으므로 경우에 따라서는 “수신자”와 “발신자” 용어가 여전히 사용된다.

3.3.3 메일 에이전트와 메시지 저장소

추가된 메일 시스템 용어는 RFC 821이 공포된 이후에 일반화되었으며, 필요한 경우 이 표준에서도 종종 사용된다. 특히, SMTP 서버와 클라이언트는 메일 전송 서비스를 제공하므로 “메일 전송 에이전트(Mail Transfer Agent : MTA)″로 칭한다. ″메일 사용자 에이전트(Mail User Agent(MUA 또는 UA)″는 일반적으로 메일의 소스 및 대상으로 생각하면 된다. 소스에서는 MUA가 사용자로부터 전송될 메일을 수집하여 MTA에게 전달할 것이다. 최종(″배달″) MTA는 메일을 MUA에게 전달하는 역할(아니면 최소한 전송할 책임이 있음. 예를 들면, ”메시지 저장소“에 메시지를 저장함으로써)을 수행한다고 생각하면 될 것이다. 그러나, 이러한 용어들이 다른 환경에서 최소한 높은 정밀도를 나타내며,

사용되는 경우 MUA와 MTA 사이에 내포된 경계는 흔히 정확하게 일치하지 않으므로 인터넷 메일 관행을 따른다. 그러므로, 독자는 이러한 용어가 다른 곳에 사용될 경우에 내포될 책임성 및 밀접한 관계를 추측함에 있어 신중해야 한다.

3.3.4 호스트

이 규격의 목적 상, 호스트는 인터넷(어떤 경우, 사설 TCP/IP 네트워크)에 연결되고 SMTP 프로토콜을 지원하는 컴퓨터 시스템을 의미한다. 호스트는 이름에 의해 식별된다 (“도메인” 참조). 가급적이면 숫자 주소로는 호스트를 식별하지 않는 것이 바람직하다.

3.3.5 도메인

도메인(또는 도메인 이름)은 하나 이상의 점으로 구분된 요소들로 이루어진다. SMTP 목적 상, 이러한 구성 요소(DNS 용어 [22]에서의 ″라벨″)는 ASCII 문자 세트 [1]에서 발췌한 글자, 숫자 및 하이픈으로만 구성될 수 있다. 도메인 이름은 도메인 이름의 계층 구조에서 호스트의 이름이나 다른 객체의 이름으로 사용된다. 예를 들면, 호스트 이름을 나타내는 대신에 도메인은 메일을 배달할 때 사용되는 별칭(CNAME RR의 라벨)이나 메일 익스체인저(Mail eXchanger) 레코드의 라벨을 지칭할 수 있다. [22]와 이 규격의 5장을 참조하기 바란다.

이 문서와 [22]에 설명된 바와 같이 도메인 이름은 완전한 정식 이름(흔히 “FQDN″이라고 함)이다. FQDN 형태가 아닌 도메인 이름은 지역에서만 사용되는 별칭에 불과하다. 이러한 별칭은 어떠한 SMTP 트랜잭션에도 사용되어서는 안 된다.

3.3.6 버퍼와 상태 테이블

양측에서 현재 상태에 대하여 동일한 견해를 신중히 관리하고 있으므로 SMTP 세션은 안정적이다. 본 표준은 서버에 존재하는 가상적인 “버퍼”와 “상태테이블”을 사용하여 세션의 상태를 모델링한다. 이 정보는 예를 들어 버퍼 정보를 버리거나 현재상태가 이전의 어떤 상태로 이전되도록 “버터를 소거하거나” “상태테이블을 초기화”하는 것으로 클라이언트에 의해 사용될 수 있다.

3.3.7 줄(line)

서비스 확장에 의해 변경되지 않은 경우 SMTP 명령과 메시지 데이터는 “줄(line)” 단위로 전송된다. 줄은 시퀀스 ASCII 문자 “CR”(6진값 0D)와 그 바로 뒤에 이어지는 ASCII 문자 ”LF”(6진값 0A)로 끝나는 0개 이상의 데이터 문자로 구성된다. 이 문서에서는 그런 종결 시퀀스를 <CRLF>로 표기한다. 표준에 부합되는 구현 제품은 다른 문자 또는 문자 시퀀스를 줄 종결자로 인식하거나 생성해서는 안 된다. 서버에서는 줄 길이를 제한할 수도 있다.(5.5.3절 참조)

또한, 텍스트에 나타나는 “bare” ”CR” 또는 ”LF” 문자(즉, 두 개의 문자가 상대방 없이 홀로 사용된 경우)는 메일 구현 제품 및 메일 시스템을 도구로 사용하는 응용 프로그램에서 오랫동안 문제의 원인이 되었다. SMTP 클라이언트 구현 제품은 줄 종결자로 사용되어야 하는 경우를 제외하고는 이러한 문자를 전송해서는 안되며, 위에서 말한 것처럼 오직 <CRLF> 시퀀스로만 전송해야 한다.

3.3.8 발기인, 배달, 릴레이 및 게이트웨이 시스템

이 규격은 전자 메일을 전송하는 데 있어 시스템의 역할에 기초하여 네 가지 종류의 SMTP 시스템에 대한 차이를 명시한다. ”발기(originating)” 시스템(SMTP 발기인이라고도 함)은 메일을 인터넷으로 가져오거나, 보다 일반적인 경우, 전송 서비스 환경으로 가져온다. ”배달” SMTP 시스템은 전송 서비스 환경에서 메일을 수신하여 메일 사용자 에이전트로 전달하거나 메일 사용자 에이전트가 순차적으로 액세스하게 될 메시지 저장소에 저장하는 시스템이다. “릴레이(relay)” SMTP 시스템(보통은 간단히 ”릴레이”라고 함)은 SMTP 클라이언트에서 메일을 받은 다음, 추적 정보를 추가할 뿐 메시지 데이터를 변경하지 않고 다음의 릴레이 또는 배달을 위해 그 메일을 다른 SMTP 서버로 전송한다.

”게이트웨이” SMTP 시스템(일반적으로는 “게이트웨이”라고 함)은 하나의 전송 환경에 있는 클라이언트 시스템으로부터 메일을 받아 다른 전송 환경에 있는 서버 시스템으로 그 메일을 전송한다. 게이트웨이에 연결된 양쪽에서의 전송 환경이 다르므로 메시지 의미론이나 프로토콜에 차이가 발생한다. 이러한 차이를 없애기 위해서 게이트웨이 시스템은 SMTP 릴레이 시스템에서 허용하는 메시지로 변환해야 하는 경우도 있다. 이 규격의 목적 상, SMTP가 방화벽의 양쪽에서 동일하게 SMTP가 사용된다 할 지라도, 주소를 다시 기록하는 방화벽을 게이트웨이로 간주해야 한다([11] 참조).

3.3.9 메시지 컨텐츠와 메일 데이터

이 문서에서 "메시지 컨텐츠" 및 "메일 데이터"라는 두 가지 용어는 DATA 명령이 받아들여진 후부터 데이터의 끝 표시가 전송되기 이전에 전송된 정보를 설명할 때 같은 의미로 사용된다. 메시지 컨텐츠에는 메시지 헤더와 구조화가 가능한 메시지 본문이 포함되어 있다. MIME 규격[12]은 구조화된 메시지 본문에 대한 표준 메커니즘을 제공한다.

3.3.10 우편함과 주소

이 규격에서 사용된 바와 같이, "주소"는 메일이 전송될 사용자 또는 전송된 메일이 저장될 위치를 식별하는 문자열이다. "우편함"이란 용어는 그러한 메일 보관소를 지칭한다. 메일이 저장되는 위치와 메일 참조(주소)를 반드시 구분해야 하는 경우가 아니라면, 대개 이들 두 개 용어는 같은 의미로 사용된다. 주소는 일반적으로 사용자와 도메인 규격으로 이루어진다. 표준화된 우편함 명명 규정은 “로컬 부분@domain"으로 정의되는데, 최신 사용에서는 단순한 ”사용자 이름“보다 훨씬 더 광범위한 응용 프로그램들을 허용하고 있다. 그 결과와 중간 호스트가 변경을 통해 전송을 최적화하려 시도했을 경우에 발생하는 많은 문제들로 인해, 주소의 도메인 부분에 지정된 호스트에 의해서만 로컬 부분이 해석되고 의미가 부여되어야 한다.

3.3.11 응답

SMTP 응답 명령에 대한 답장으로 전송 채널을 통해 수신자가 발신자에게 전송하는 승인(찬성 또는 반대)이다. 응답의 일반적인 형태는 숫자 완료 코드(성공 또는 실패를 나타냄)로서 일반적으로 그 뒤에는 텍스트 문자열이 이어진다. 이 코드는 프로그램에서 사용하기 위한 것이며 텍스트는 일반적으로 사람을 위한 것이다. 최신 규격[34]은 보충 및 보다 특정적인 완료 코드의 사용을 비롯하여 응답 문자열의 세부적인 구조화를 명시하고 있다.

3.4 일반 구문 원리와 트랜잭션 모델

SMTP의 명령과 응답은 엄격한 구문을 따른다. 모든 명령은 명령 동사로 시작한다. 모든 응답은 세 자리 숫자 코드로 시작한다. 일부 명령 및 응답에서는 동사 또는 응답 코드 뒤에 인수가 지정되어야 한다. 일부 명령은 (동사 뒤에서) 인수를 받지 않지만, 일부 답장

코드는 자유로운 형태의 텍스트를 통해 선택적으로 인수를 받기도 한다. 두 가지 경우 모두에서, 텍스트가 나타나는 경우 공백 문자를 삽입하여 동사 또는 응답 코드로부터 텍스트가 분리된다. 명령 및 응답의 전체 정의는 5장에 명시되어 있다.

동사와 인수 값(예를 들면, RCPT 명령에서 "TO:" 또는 "to:"와 확장 이름 키워드)은 대소문자를 구분하지 않는다. 단 이 규격에서 유일한 예외 사항인 우편함 로컬 부분은 제외된다.(SMTP 확장이 명시적으로 대소문자를 구분하는 요소를 명시할 수 있다) 즉, 명령 동사, 우편함 로컬 부분 이외의 인수 값 및 자유로운 형태의 텍스트는 대문자, 소문자 또는 의미에 영향을 주지 않으면서 대/소문자의 결합으로 사용될 수 있다. 그러나, 사실상 우편함 로컬 부분에서는 그렇지 않다. 우편함의 로컬 부분은 대/소문자를 구분하는 것으로 다루어야 한다. 그러므로, SMTP 구현 제품은 우편함 로컬 부분의 대/소문자를 보존해야 한다. 우편함 도메인은 대/소문자를 구분하지 않는다. 특히, 어떤 호스트의 경우 사용자 “smith"는 사용자 "Smith"와는 다르다. 그러나, 우편함 로컬 부분의 대/소문자를 구분하는 특성을 잘못 사용하게 되면 상호운용성을 저해할 수 있으므로 사용하지 않는 것이 좋다.

이 규격(과 RFC 821)에 위배되는 일부 SMTP 서버에서는 클라이언트에 의해 명령 동사가 대문자로 인코드되어야 한다. 이런 서버의 요건을 충족시키기 위해 구현 제품에서 이러한 인코딩 방식을 채택할 수도 있다.

인수 필드는 줄 끝(즉, 문자 시퀀스 <CRLF>)으로 끝나는 가변 길이의 문자열로 이루어진다. 수신자는 이 시퀀스가 수신될 때까지 아무런 동작도 취하지 않을 것이다.

각 명령에 대한 구문이 설명과 함께 제공된다. 일반적인 구성 요소와 매개변수가 5.1.2절에 나와 있다.

명령과 응답은 ASCII 문자 세트 [1]에 포함된 문자로 구성된다. 전송 서비스가 8비트 바이트(옥텟) 전송 채널을 제공하는 경우, 각 7비트 문자는 최상위 비트가 0으로 소거된 옥텟에서 오른쪽으로 정렬된 상태로 전송된다. 특별히 확장되지 않은 SMTP 서비스는 7비트 전송만을 제공한다. 이런 확장에 대해 특정 서버와의 협정을 성공적으로 수행하지 못한 발기 SMTP 클라이언트는 옥텟의 상위 비트에 정보를 담은 메시지를 전송해서는 안 된다. 이 규칙에 위배되는 메시지가 전송되는 경우, 수신 SMTP 서버는 상위 비트를 소거하거나 해당 메시지를 잘못된 메시지로서 거부할 수 있다. 일반적으로, 릴레이

SMTP에서는 수신한 메시지 컨텐츠가 유효하다고 가정해야 한다. 엔벨로프에서 그러한 작업을 허용한다고 가정하면서 해당 컨텐츠를 검사하지 않고 릴레이 해야 한다. 물론, 컨텐츠의 라벨이 잘못 붙여지거나 데이터 경로가 실제 컨텐츠를 받을 수 없는 경우라면, 최종적으로 크게 왜곡된 메시지가 수신자에게 배달될 수 있다. 배달 SMTP 시스템은 그러한 메시지를 배달하지 않고 거부(거절)할 수 있다. 어떠한 송신 SMTP 시스템도 US-ASCII 이외의 문자 세트로 된 엔벨로프 명령을 전송할 수 없다. 수신 시스템은 일반적으로 "500 구문 오류 - 잘못된 문자(500 syntax error - invalid character)" 응답을 사용하여 그러한 명령을 거부해야 한다.

클라이언트는 확장된 SMTP 기능, 특히 "8BITMIME" 확장 [20]을 사용하여 서버에게 8비트 메시지 컨텐츠를 전송할 수 있다. SMTP 서버가 8BITMIME을 지원해야 한다. 그러나, 제한되지 않은 8비트 정보를 전송할 수 있는 권한으로 해석해서는 안 된다. 발신자는 적절한 컨텐츠 전송 인코딩을 사용한 MIME 형식이 아니라 상위 비트에 정보가 포함된 자료를 위해 8BITMIME을 요청해서는 안 된다. 서버는 그러한 메시지를 거부할 수 있다.

이 문서에 사용되는 메타언어 식 표기는 다른 인터넷 메일 시스템 문서에서 사용되는 "증가된 BNF"에 해당한다. 이 구문을 잘 모르는 독자는 ABNF 규격 [8]을 참조해야 할 것이다. 명확한 설명을 위해 이하 텍스트에 사용되는 메타언어 용어에는 각 괄호(예를 들면, <CLRLF>)를 사용하였다.

4. SMTP 절차 : 개요

이 절은 SMTP에 사용되는 절차, 즉, 세션 시작, 메일 트랜잭션, 메일 전달, 메일 이름 확인 및 메일링 목록 확장, 열기 및 닫기 교환(exchange)에 대해 설명한다. 이 절의 끝 부분에서는 릴레이, 메일 도메인 및 역할 변경에 대해 설명한다. 부록 IV는 몇 개의 완전한 시나리오를 제공하고 있다.

4.1 세션 시작

SMTP 세션은 클라이언트가 서버와의 연결을 열고, 서버가 열기 메시지에 응답할 때 시작된다.

SMTP 서버 구현 제품은 220 코드 이후에 연결 인사 응답 안에 소프트웨어 ID 및 버전 정보를 포함할 수 있다. 220 코드는 모든 문제를 보다 효율적으로 분리하고 복구할 수 있도록 하는 관례이다. 구현 제품은 소프트웨어 및 버전 정보를 제공하는 것이 보안 문제를 일으키는 경우, SMTP 서버가 그러한 정보를 제공하지 못하게 할 수 있다. 일부 시스템의 경우 메일 문제의 접점을 식별하기도 하지만, 그러한 접점이 필요한 "포스트마스터" 주소를 유지하는 것을 대신하여 사용되지는 않는다(5.5.1절 참조).

SMTP 프로토콜에서는 서버가 전기 연결을 허용하면서 공식적으로 트랜잭션을 거부할 수 있다. 초기의 연결 열기 메시지에 대한 응답으로 220대신에 554응답을 제공하는 것이 예이다. 이 방법을 사용하는 서버는 연결을 닫기 전에 클라이언트가 QUIT를 보낼 때까지(5.1.1.10절 참조) 기다려야 하며, "503 잘못된 명령 순서(503 bad sequence of commands)"로 중간에 발생하는 모든 명령에 응답해야 한다. 그러한 시스템에 SMTP 연결을 설정하려는 시도는 아마도 오류를 일으킬 것이므로, 연결설정에 대해 554 응답을 반환하는 서버는 송신 시스템의 디버깅 작업을 쉽게 할 수 있을 만큼 충분한 정보를 응답 텍스트에 제공해야 한다.

4.2 클라이언트 시작

일단 서버가 환영 메시지를 전송하고 클라이언트가 그 메시지를 수신하고 나면, 일반적으로 클라이언트는 클라이언트가 누군지를 식별할 수 있게 하는 EHLO 명령을 서버에게 전송한다. EHLO를 사용한다는 것은 세션을 여는 것 외에도, 클라이언트가 서비스 확장을 처리할 수 있다는 것을 나타내며, 서버에게는 서버에서 지원되는 확장 목록을 제공할 것을 요청하는 것이다. 서비스 확장을 지원할 수 없는 이전 SMTP 시스템과 시작되는 메일 세션에서 서비스 확장을 필요로 하지 않는 같은 시대의 클라이언트는 EHLO 대신에 HELO를 사용할 수 있다. 서버는 HELO 명령에 대해 확장된 EHLO 방식의 응답을 반환해서는 안 된다. 서버가 EHLO에 대해 "인식되지 않는 명령"이라는 응답을 돌려보내면 클라이언트는 EHLO 대신에 HELO를 전송할 수 있을 것이다.

EHLO 명령에서 명령을 전송하는 호스트는 자신이 누군 지를 확인하여 준다. 이 명령은 "Hello, I am <도메인>"(EHLO의 경우, 서비스 확장 요청을 지원함)으로 해석할 수 있다.

4.3 메일 트랜잭션

SMTP 메일 트랜잭션에는 세 가지 단계가 있다. 트랜잭션은 발신자 ID를 제공하는 MAIL 명령에서 시작한다. (일반적으로, MAIL 명령은 진행 중인 트랜잭션이 없을 때만 전송될 수 있다. 5.1.4절 참조) 수신자 정보를 제공하는 하나 이상의 연속된 RCPT 명령들이 이어진다. 그리고 나면 DATA 명령이 메일 데이터의 전송을 시작하며 "메일의 끝"을 나타내는 데이터 표시기에 의해 종료된다. 또한 이 표시기는 트랜잭션이 완료되었음을 승인하여 준다.

이 절차에서 첫 번째 단계는 MAIL 명령이다.

MAIL FROM:<역경로> [SP <메일 매개변수>] <CRLF>

이 명령은 SMTP 수신자에게 새로운 메일 트랜잭션이 시작될 것이므로 수신자 정보 또는 메일 데이터를 비롯하여 모든 상태 테이블 및 버퍼를 초기화할 것을 지시한다. 유일한 인수이기도 한 첫 번째 <역경로> 부분은 소스 우편함("<"과 ">" 사이에 지정됨)을 제공하는데, 이 우편함은 오류를 보고하는 데 사용할 수 있다(오류 보고에 대한 자세한 설명은 5.2절을 참조한다). 지시 내용을 수락하는 경우 SMTP 서버는 250 OK 답장을 보낸다. 어떠한 이유에서인지 우편함 규격이 받아들여질 수 없는 경우라면 서버는 오류가 영구적인지(즉, 클라이언트가 동일한 주소를 다시 전송하려 할 때 오류가 다시 발생하는 경우) 임시적인지(즉, 클라이언트가 나중에 다시 시도했을 때 주소가 받아들여질 경우)를 나타내는 답장을 반환해야 한다. 이 요구사항이 사용되는 범위가 분명할 지라도, 하나 이상의 순경로(forward-path)(RCPT 명령에서)를 검토할 때까지 역경로의 허용 여부를 결정할 수 없는 상황들이 있다. 그러한 경우에 있어 서버는 역경로(250 답장으로)를 합리적으로 받아들인 다음 순경로를 수신하여 검토한 후, 역 경로에 관련된 문제를 보고할 수 있다. 일반적으로 오류가 발생되면 550 또는 553 답장이 반환된다.

예전에는 <역경로>에 우편함 이외의 것들을 포함할 수 있었지만, 최신 시스템에서는 소스 라우팅을 사용해서는 안 된다(부록 III 참조).

선택사항인 <메일 매개변수>의 사용은 협정된 SMTP 서비스 확장과 관련이 있다(3.2절 참조).

이 절차에서 두 번째 단계는 RCPT 명령이다.

RCPT TO:<순경로> [SP <rcpt 매개변수>] <CRLF>

이 명령의 첫 번째 또는 유일한 인수는 한 명의 수신자를 나타내는 순경로(forward-path)(일반적으로 "<"와 ">" 사이에 지정되는 우편함과 도메인)이다. 이 값이 받아들여질 경우 SMTP 서버는 250 OK 답장을 보내고 순경로를 저장한다. 수신자의 주소가 배달 가능한 주소가 아닌 것으로 확인되면 SMTP 서버는 "no such user -"와 메일 박스 이름과 같은(다른 상황 및 답장코드로 가능함) 문자열과 함께 550 답장을 보낸다. 이 단계는 몇 번이고 반복될 수 있다.

<순경로>에는 우편함 이외의 것들도 포함될 수 있다. 예전에는 <순경로>가 목적지 우편함과 소스 라우팅에 포함되는 호스트 목록이 될 수 있었지만, 최신 SMTP 클라이언트에서는 소스 라우트를 이용할 수 없다(부록 C 참조). 서버는 순경로에서 소스 라우팅 목록을 접하게 되는 상황을 대비해야 하지만, 라우트를 무시하거나 수반되는 릴레이 서비스의 지원을 거절할 수 있다. 마찬가지로, 서버는 다른 호스트 또는 시스템을 목적지로 하는 메일을 받아드리지 않고 거절할 수 있다. 이러한 제한은 모든 SMTP 기능을 지원하지 않은 클라이언트에 대해서는 서버가 릴레이로서 사용되지 않게 한다. 따라서, 제한된 기능을 제공하는 클라이언트는 인터넷 상의 모든 SMTP 서버를 메일 처리 (릴레이) 사이트로 사용할 수 있다고 가정해서는 안 된다. RCPT 명령이 이전 MAIL 명령 없이 나타나는 경우 서버는 503 잘못된 명령 순서 응답을 보낸다. 선택사항인 <rcpt 매개변수>의 사용은 협정된 SMTP 서비스 확장과 관련이 있다(3.2절 참조).

이 절차에서 세 번째 단계는 DATA명령(또는 서비스 확장에서 지정된 다른 명령)이다.

DATA <CRLF>

이 명령이 받아들여지는 경우 SMTP 서버는 354 중간 답장을 전송하며 메일 데이터 종결자 바로 전까지의 모든 연속된 줄을 메시지 텍스트로 간주한다. 텍스트의 끝이 성공적으로 수신되어 저장되는 경우 SMTP 수신자는 250 OK 답장을 보낸다.

메일 데이터가 전송 채널 상에서 전송되므로 명령과 답장 다이얼로그가 계속하여 사용될 수 있도록 메일 데이터의 끝이 표시되어야 한다. SMTP는 "."(구두점 또는 마침표)만을 포함하는 줄을 전송하여 메일 데이터의 끝을 나타낸다. 사용자의 텍스트와의 충돌로부터 보호하기 위해 투명 절차가 사용된다(5.5.2절 참조).

또한 메일 데이터의 끝 표시기는 메일 트랜잭션의 완료를 승인하여 SMTP 서버에게 저장된 수신자와 메일 데이터를 처리하라고 지시한다. 이 지시가 받아들여질 경우 SMTP 서버는 250 OK 답장을 보낸다. 프로토콜 교환에 사용되는 DATA 명령은 다음과 같은 두 가지 경우에 실패할 수 있다.

- MAIL 또는 RCPT 명령이 없거나 그러한 명령이 거부되는 경우 서버는 DATA 명령에 대해 순서를 벗어난 명령 (503)이나 유효한 수신자 없음 (554) 응답을 반환할 수 있다. 이러한 응답(또는 다른 5로 시작되는 답장) 중 하나가 수신되면 클라이언트는 메시지 데이터를 전송해서는 안 된다. 대체로 354 답장이 수신되는 경우가 아니라면 메시지 데이터는 전송해서는 안 된다.
- 동사가 처음에 받아들여지고 354 답장이 반환되었다면, 메일 트랜잭션이 불완전하거나 (예를 들면, 수신자 없음) 리소스를 사용할 수 없거나(물론 서버에서 갑작스럽게 사용할 수 없게 된 경우 포함), 아니면 서버에서 해당 메시지가 정책이나 다른 이유로 인해 거부되어야 한다고 판단한 경우에만 DATA 명령이 실패할 것이다.

그러나, 실제로 일부 서버의 경우에는 메시지 텍스트가 수신될 때까지 수신자 검증을 수행하지 않는다. 이러한 서버는 하나 이상의 수신자에 대해 발생된 오류를 “후속 오류”로 취급하여 7장에서 설명하는 것과 같이 메일 메시지를 반환해야 한다. 데이터 이후에 "550우편함을 찾을 수 없음 (또는 이에 상응하는 다른) 응답 코드를 사용하면 클라이언트는 실패한 수신자를 판단하기가 어렵거나 또는 판단이 불가능하다.

RFC 822 형식[7, 32]을 사용하면, 메일 데이터에 Date(날짜), Subject(제목), To(받는 사람), Cc(참조), From(보내는 곳)과 같은 메모 헤더 항목이 포함된다. 서버 SMTP 시스템은 RFC 822 또는 MIME [12]의 메시지 헤더나 메시지 본문에서 파악한 결점에 기초하여 메시지를 거부해서는 안 된다. 특히, 재전송 필드 개수가 일치하지 않거나 재전송 보내는 곳 또는 재전송 날짜 없이 재전송 받는 사람 필드만 사용된 메시지를 거부해서는 안 된다.

위에서 설명한 순서대로 메일 트랜잭션 명령을 사용해야 한다.

4.4 주소 교정을 위한 전달 또는 갱신

전달 지원은 대개 임의의 기업 안에서 또는 그 기업과 관련하여 주소를 통합하고 간소

화하는 데 필요하며, 드문 경우이기는 하지만 사용자의 이전 주소를 현재의 주소와 연결하기 위해 주소를 설정할 때도 필요하다. 보안 또는 비공개의 목적 상, 알리지 않는(silent) 메시지 전달(서버에서 발신자에게 통보하지 않음)이 최신 인터넷에서 일반적으로 사용되고 있다.

기업과 "새 주소"의 경우 모두에 있어, 정보 은닉에 대한(과 때로는 보안) 고려 사항은 전달 작업의 역효과로서의 SMTP 프로토콜을 통한 "최종" 주소의 노출에 대하여 반대 의견을 주장하고 있다. 이 주장은 최종 주소가 발신자에 의해서 도달할 수 없는 경우에 특히 중요할 수 있다. 결과적으로, 구현자는 RFC 821의 3.2절에 설명된 "전달" 메커니즘과 특히 RCPT의 251(교정된 대상)과 551 답장 코드를, 환경구성 시스템을 사용할 수 있는 경우, 신중히 평가해야 한다.

특히:

* 서버는 주소 변경 사실을 알았을 때 메시지를 전달할 수 있다. 전달 시, 251 코드와 함께 주소 업데이트 정보를 제공하거나 "변경 사실을 알리지 않고" 250 코드를 전달 및 반환할 수 있다. 그러나, 251 코드가 사용되면 클라이언트가 실제로 변경된 주소 정보를 갱신하거나 사용자에게 해당 정보를 반환할 것이라고 가정해서는 안 된다.

혹은,

* 서버는 주소가 지정되었을 때 배달할 수 없는 경우 메시지를 거절하거나 돌려보낼 수 있다. 이런 경우 서버는 551 코드와 함께 주소 업데이트 정보를 제공하거나, 550 코드와 함께 어떠한 주소와 관련된 정보도 없이 메시지를 배달 불가능한 것으로서 거절할 수 있다. 그러나, 551 코드가 사용되면 클라이언트가 실제로 변경된 주소 정보를 갱신하거나 사용자에게 해당 정보를 반환할 것이라고 가정해서는 안 된다.

251 및 551 답장 코드를 지원하는 SMTP 서버 구현 제품은 고의는 아니라 할지라도 정보를 공개할 수도 있는 사이트에 대해서는 사용할 수 없게 하거나 사용을 제한할 수 있도록 가급적이면 환경구성 메커니즘을 제공하는 것이 바람직하다.

4.5 주소 디버깅 명령

4.5.1 개요

SMTP는 사용자 이름을 검증하거나 메일링 목록의 내용을 획득할 수 있는 명령을 제공한다. 즉, VRFY와 EXPN 명령을 사용하면 되는데, 이 명령들은 문자열 인수를 사용한다. 구현 제품은 VRFY 및 EXPN을 지원해야 한다(그러나, 4.5.2절과 8.3절을 참조한다).

VRFY 명령의 경우, 문자열은 사용자 이름 또는 도메인(아래 참조)이다. 일반(즉, 250) 응답이 반환되면 그 응답에는 사용자의 전체 이름이 포함될 수 있으며 사용자의 우편함은 반드시 포함되어야 한다. 다음과 같은 형태 중 하나이어야 한다.

사용자 이름 <로컬 부분@도메인>
로컬 부분@도메인

VRFY의 인수로 사용된 이름이 두 개 이상의 우편함을 나타낼 수 있는 경우 서버는 모호하다고 할 수도 있고 다른 가능한 우편함 목록을 제공할 수도 있다. 다시 말하면 다음 중 하나를 VRFY에 대한 적당한 응답으로서 사용할 수 있다는 뜻이다.

553 User ambiguous
또는
553- Ambiguous; 가능한 우편함 주소는
553-Joe Smith <jsmith@foo.com>
553-Harry Smith <hsmith@foo.com>
553 Melvin Smith <dweep@foo.com>
또는
553-Ambiguous; 가능한 우편함 주소는
553- <jsmith@foo.com>
553- <hsmith@foo.com>
553 <dweep@foo.com>

「일반적인 상황에서 553 답장을 받는 클라이언트는 사용자에게 해당 결과를 보여주려 할 것이다.」 정확히 제공된 형태를 사용하고 [34]에 설명된 것처럼 확장된 답장 코드에

의해 추가된 “user ambiguous” 이나 ”ambiguous”같은 키워드를 사용하면 필요할 때마다 다른 언어로 쉽고 편리하게 자동으로 번역될 수 있을 것이다. 물론, 고도로 자동화되었거나 영어 이외의 언어에서 작동하는 클라이언트는 사용자에게 응답의 리터럴 텍스트가 아닌 다른 지시 사항을 반환하기 위해 응답을 번역하려 하거나 사용자에게 보고하기 이전에 추가 정보를 제공하기 위해 디렉토리 서비스를 참고하는 등의 자동화된 어떤 행위를 수행하려 할 것이다.

EXPN 명령의 경우, 문자열이 메일링 목록을 나타내므로, 성공적인(즉, 250) 두 줄 이상으로 구성된 응답은 사용자의 전체 이름을 포함할 수 있으며, 메일링 목록에 속하는 각각의 우편함 정보를 제공해야 한다.

일부 호스트에서는 공통 데이터 구조 안에 메일링 목록과 별칭을 모두 유지할 수 있으며 하나의 우편함만을 포함하는 메일링 목록을 가질 수 있기 때문에, 메일링 목록과 단일 우편함의 별칭을 구분하기가 쉽지 않다. 메일링 목록에 VRFY를 적용하라는 요청이 있을 경우 주소가 잘 지정된 메시지가 목록에 있는 모든 사람에게 전달된다면 긍정 응답이 제공될 수도 있으나 그렇지 않을 경우 오류(예를 들면, “550 사용자가 아닌 메일링 목록임” 또는 ”252메일링 목록의 멤버를 확인할 수 없음” 가 보고될 것이다. 사용자 이름을 확장하라는 요청이 있을 경우 서버는 하나의 이름을 포함하는 목록으로 구성된 긍정 응답을 반환할 수도 있으나, 그렇지 않을 경우에는 오류(예를 들면,”550 메일링 이름이 아닌 사용자 이름임”) 가 보고될 수도 있다.

성공적인 두 줄 이상으로 된 응답(EXPN의 경우 일반적임)에서는 답장의 각 줄에는 대해 정확히 한 개의 우편함이 지정되어야 한다. 모호한 요청의 경우는 위에 설명되어 있다.

”사용자 이름”은 구분이 불명확한 용어로서 신중히 사용되어야 한다. VRFY 또는 EXPN 명령을 구현한 제품은 로컬 우편함을 “사용자 이름”으로서 인식하는 기능을 포함해야 한다. 그러나, 현재의 인터넷 관례에서는 흔히 다중 도메인을 위한 메일을 단일 호스트에서 처리하므로, 호스트, 특히 이 기능을 제공하는 호스트는 “로컬 부분@도메인” 형태를 “사용자 이름”으로 받아들여야 하며, 또한 호스트는 다른 문자열을 “사용자 이름”으로 인식할 수도 있다.

우편함 목록을 확장하려면 다음과 같은 두 줄 이상으로 된 답장이 필요하다.

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>

또는

C: EXPN Executive-Washroom-List

S: 550 Access Denied to You.

VRFY와 EXPN 명령의 문자열 인수는 사용자 이름 및 우편함 목록 개념을 구현하는 제품이 다양하므로 더 이상 세부적으로 제한될 수는 없다. 일부 시스템에서는 EXPN 명령의 인수로 메일링 목록을 포함하는 파일의 파일 이름을 사용할 수 있지만, 인터넷에는 다양한 파일 명명 규정이 있다. 마찬가지로, 이러한 명령들이 반환하는 결과물의 변화를 살펴볼 때 응답은 매우 신중히 해석되어야 하며 일반적으로는 진단 목적으로만 사용되어야 한다.

4.5.2 VRFY 일반 응답

VRFY와 EXPN요청에 대하여 정상(2로 시작하는 코드 또는 551) 응답이 반환될 때, 일반적으로 이 응답에는 "<로컬 부분@도메인>"과 같은 우편함 이름이 포함된다. 여기서 "도메인"은 구분에 표현된 대로 정식 도메인 이름이어야 한다. 이 규격의 의도와는 위배되지만 충분한 근거가 있어 허용될 수 있는 예외적인 경우에서는 자유 형태의 텍스트가 반환될 수 있다. 컴퓨터와 사람 모두에 의한 구문 분석을 쉽게 하기 위해서, 주소는 각괄호를 사용한 형태로 표시되어야 한다. 자유 형태의 디버깅 정보가 아닌 주소를 반환할 때는 EXPN 및 VRFY는 SMTP RCPT 명령에서 유용한 유효 도메인 주소만을 반환해야 한다. 결과적으로, 주소가 프로그램 또는 다른 시스템으로의 배달을 위한 것이라면 해당 대상에 도달하는 데 사용될 우편함 이름이 제공되어야 한다. 경로(명시적 소스 라우트)가 VRFY 또는 EXPN에 의해 반환되어서는 안 된다.

서버 구현 제품은 VRFY 및 EXPN 모두를 지원해야 한다. 보안상의 이유를 위해 구현 제품은 구성 옵션 또는 그에 해당하는 기능을 통해 이러한 명령 모두 또는 그 중 하나를 사용할 수 없게 하는 로컬 설치 방법을 제공할 수 있다. 이러한 명령이 지원되는 경우 릴레이 기능이 지원될 때 릴레이를 가로질러 동작하도록 요구되지는 않는다. 이러한 명령이 RFC 821에서는 모두 선택 사항으로 제공되었으므로, 이 명령들을 지원하는 경우에는

EHLO 응답에서 서비스 확장 목록으로 기재해야 한다.

4.5.3 VRFY 또는 EXPN 정상(success) 응답의 의미

실제로 주소를 검증한 경우가 아니라면 서버는 VRFY 또는 EXPN 명령에 대한 응답으로 250 코드를 반환해서는 안 된다. 특히, 서버가 한 일이 오로지 제공된 구문이 유효하다는 것을 검증할 것이라면 서버는 250 코드를 반환해서는 안 된다. 그 경우, 502 (구현되지 않은 명령) 또는 500 (구문 오류, 인식할 수 없는 명령) 코드를 반환해야 한다. 별도로 언급된 바와 같이, VRFN 및 EXPN의 구현(실제로 주소를 확인하고 정보를 반환함에 있어)을 적극적으로 권장하고 있다. 그러므로, VRFY에 대해 500 또는 502를 반환하는 구현은 이 규격을 완벽히 따른다고 할 수는 없다.

주소가 유효한 것으로 보이지만 실시간으로, 특히 서버가 다른 서버 또는 도메인의 메일 익스체인저(exchanger)로 작동하는 경우에는 적절히 확인될 수 없는 상황이 있을 수 있다. 이러한 경우 “명백한 유효성”은 일반적으로 최소한의 구문 검사는 포함해야하고, 지정된 도메인이 호스트가 메일을 릴레이 할 수 있는 도메인임을 확인하는 작업도 포함할 수 있다. 이러한 상황에서는 252 응답 코드를 반환해야 한다. 이러한 경우는 3.1절에서 설명한 RCPT 검증과 유사하다. 마찬가지로, 주소가 인지는 되었지만 그 주소를 대상으로 하여 수신된 메일이 전달될 수도 있고 또는 반송될 수도 있는 것을 설명하기 위해, VRFY(및 EXPN)와 함께 251과 551 응답코드를 사용하는 곳에 4.4절의 설명이 응용될 수 있다. 일반적으로 구현 제품은 작업을 수행하는 데 보다 더 많은 시간이 소요될 지라도 RCPT의 경우에서보다 VRFY의 경우에서 주소 검증에 대해 더 많은 노력을 기울인다.

4.5.4 EXPN의 의미론과 이용

EXPN은 흔히 메일링 목록과 다수의 대상-주소・별칭에 관련된 문제를 디버깅하고 파악하는 데 매우 유용하다. 어떤 시스템에서는 메일링 목록의 소스 확장을 중복 제거 수단으로 사용하려 할 것이다. 호스트(대개 MX와 CNMAE DNS 레코드 사용), 우편함(다양한 유형의 로컬 호스트 별칭) 및 다양한 프록시 장치를 위해 인터넷 상에서 메일을 장착한 앨리어싱 시스템(aliasing system)이 증가함에 따라 이러한 전략이 일관적으로 작동하기는 거의 불가능하므로, 메일 시스템은 그러한 시도를 하지 말아야 한다.

4.6 도메인

도메인 이름이 SMTP에 사용될 경우 오직 결정할 수 있는(확인할 수 있는) 정식 도메인 이름(FQDN)만 허용된다. 다른 말로 하면, MX RR 또는 A RR(5장에서 설명함)로 결정될 수 있는 이름이 허용되며, 대상이 MX 또는 A RR로 결정될 수 있는 이름은 CNAME RR이다. 로컬 별칭이나 인증되지 않은(unqualified) 이름은 사용하지 말아야 한다. 그러나, 다음과 같은 두 가지 예외적인 경우에서는 FQDN을 사용하지 않을 수 있다.

- EHLO 명령에 제공된 도메인 이름이 주 호스트 이름(A RR로 결정될 수 있는 도메인 이름)이거가 호스트에 이름이 없다면 5.1.1.1절에 설명된 바와 같이 주소 리터럴인 경우.

- 예약된 우편함 이름 “postmaster"는 도메인 자격(qualification)(5.1.1.3절 참조) 없이 RCPT 명령에 사용될 수 있으며 그렇게 사용될 경우에는 허용해야 한다.

4.7 릴레이

일반적으로, 도메인 시스템에 있는 메일 익스체인저(eXchanger) 레코드를 사용하게 되면, 인터넷 메일 시스템에서 명시적인 소스 라우트를 사용할 필요가 없게 된다. 이전에 발생하였던 해석상의 많은 문제들로 인해 명시적인 소스 라우트의 사용이 바람직하지 않게 되었다. SMTP 클라이언트는 특별한 경우를 제외하고는 명시적인 소스 라우트를 생성해서는 안 된다. SMTP 서버는 메일 릴레이로서의 작동이나 소스 라우트를 지정하는 주소를 거절할 수 있다. 라우트 정보를 접할 경우, SMTP 서버는 라우트 정보를 무시하고 단순히 라우트의 마지막 요소로 지정된 최종 목적지로 전송할 수 있으며, 또한 그렇게 해야 한다. 「어떤 문제를 해결하기 위해 소스 라우팅에 지정된 중간 호스트에 의지하는 발신자가 DNS에 나타나지 않은 이름을 목적지로 사용하는 잘못된 관행이 있었다.」 소스 라우트가 제거되면 이러한 실행은 오류를 일으킬 것이다. 이 점이 바로 SMTP 클라이언트가 잘못된 소스 라우트를 생성해서는 안되거나 일련의 이름 결정에 의존하는 몇 가지 이유 중 하나이다.

소스 라우트가 사용되지 않는 경우, 순경로에서 역경로를 만들기 위한 RFC 821에 설명된 프로세스는 적용되지 않으며 배달 시에 역경로는 단순히 MAIL 명령에 나타난 주소일 것이다.

릴레이 SMTP 서버는 대개 최종 배달 시스템이라기보다는 해당 서버를 가리키는 DNS MX 레코드에서의 대상이다. 릴레이 서버는 로컬 사용자를 위해 메일을 받아드리거나 거절하는 것과 동일한 방법으로 메일의 릴레이 작업을 받아들이거나 거절할 수 있다. 작업을 받아들이는 경우 SMTP 클라이언트가 되어서 (6장의 규칙에 따라) DNS에 지정된 다음 번 SMTP 서버로 설정한 다음, 그 서버로 메일을 보낸다. 정책상의 이유로 특정 주소로 메일을 릴레이 하는 작업을 거절하는 경우 550 응답이 반환되어야 한다.

많은 메일 전송 클라이언트가 존재하는데, 특히 POP3 또는 IMAP을 통해 메일을 수신하는 장치도 갖추고 있다. 이러한 클라이언트는 후속 배달 시도를 위해 메시지를 대기열에 저장할 수 있는 기능과 같이 이 규격의 요구사항 중 일부만을 지원할 수 있는 제한된 기능을 제공한다. 이러한 클라이언트의 경우에는 처리와 그 이후의 배포를 위해 하나의 서버로 모든 메시지를 전송하는 사적인 구성을 가지는 것이 일반적이다. 여기에 명시된 바와 같이 SMTP는 이러한 역할에 매우 적합하지는 않으며 궁극적으로 현재의 관행을 대신하게 될 표준화된 메일 의뢰 프로토콜의 제정 작업이 진행 중이다. 어떻든, 이러한 준비는 사적이며 이 규격의 범위를 벗어나므로 여기서는 설명하지 않는다.

MX 레코드가 SMTP 릴레이와 최종 배달 시스템이 아닌 다른 환경으로의 게이트웨이로서 작동하는 SMTP 서버를 가리킬 수 있다는 사실이 중요하다. 4.8절과 6장을 참조한다.

SMTP 서버가 메일 릴레이 작업을 받아들인 다음, 나중에 대상이 올바르지 않거나 메일이 어떠한 이유로 인하여 배달될 수 없다는 사실을 알게되면 SMTP 서버는 “배달할 수 없는 메일”이라는 알림 메시지를 만들어서 (역경로에 지정된 대로) 배달할 수 없는 메일의 발기인에게 전송해야 한다. 가능하다면 다른 표준(예를 들면 [24, 25] 참조)에 나와 있는 배달되지 않은 메일의 보고서에서 지정한 형식을 사용해야 한다.

이 알림 메시지는 릴레이 호스트 또는 배달이 수행될 수 없음을 처음으로 결정하는 호스트의 SMTP 서버에서 얻어져야 한다. 물론, SMTP 서버는 알림 메시지 전송 문제에 대해서 또다시 알림 메시지를 보내서는 안 된다. 오류 보고에서의 루프를 막기 위한 한 가지 방법은 알림 메시지의 MAIL 명령에서 역경로를 널(Null)로 지정하는 것이다. 그러한 메시지가 전송될 때 역경로는 널로 설정되어야 한다(자세한 설명은 5.5.5절 참조). 널 역경로를 사용하는 MAIL 명령은 다음과 같다.

MAIL FROM:<>

3.4.1절에서 설명한 바와 같이, 릴레이 SMTP는 메시지 데이터의 헤더 또는 본문에 대해 검사하거나 작동할 필요가 없으며, 고유의 “Received:” 헤더를 추가하고 선택적으로 메일 시스템에서의 루프를 검색하려는 시도를 수행하는 것(7.2절 참조)을 제외하고는 그러한 작업을 수행해서는 안 된다.

4.8 메일 게이트웨이 연결(gatewaying)

위에서 설명한 릴레이 기능은 인터넷의 SMTP 전송 서비스 환경 안에서 작동하지만, MX 레코드 또는 다양한 형태의 명시적 라우팅의 경우에는 중간 SMTP 서버가 하나의 전송 서비스와 다른 전송 서비스 간 변환 기능을 수행해야 할 것이다. 3.3.8절에 설명된 바와 같이, 시스템이 두 가지 전송 서비스 환경 간 경계에 있는 경우 그러한 시스템을 “게이트웨이” 또는 “게이트웨이 SMTP”라고 지칭한다.

다양한 메일 형식 및 프로토콜과 같이 서로 다른 메일 환경간에 메일을 연결(gatewaying)하는 작업은 복잡하므로 표준화하기가 쉽지 않다. 그러나, 인터넷과 다른 메일 환경 간 게이트웨이에 대한 몇 가지 일반적인 요구사항이 있을 수 있다.

4.8.1 게이트웨이 연결에서 헤더 필드

메시지가 메일 환경 경계를 거쳐서 연결될 때(gatewaying) 헤더 필드가 다시 작성될 수 있다. 3.4.1절에서는 금지하고 있지만, 이 작업에는 메시지 본문 검사 또는 목적지 주소의 로컬 부분 해석 작업이 필요할 수 있다.

게이트웨이를 통해서 인터넷에 연결된 다른 메일 시스템은 흔히 일부 RFC 822 헤더를 사용하거나 다른 구문으로 비슷한 기능을 제공하지만, 이러한 메일 시스템 중에는 SMTP 엔벨로프에 해당하는 요소가 없는 시스템도 있다. 그러므로, 메시지가 인터넷 환경을 떠날 때는 메시지 헤더 안에 SMTP 엔벨로프 정보를 넣어두어야 할 것이다. 사용할 수 있는 방법으로서 엔벨로프 정보(예를 들면, “X_SMTP-MAIL:”과 ”X-SMTP-RCPT:”)를 전달하기 위해 새 헤더 필드를 만드는 방법이 있을 것이다. 그러나, 이 방법을 사용하려면 외부 환경에서 사용하는 메일 프로그램을 변경해야 하며 개인 정보가 노출될 위험이 따를 것이다(8.2절 참조).

4.8.2 게이트웨이 연결에서의 수신된 줄

인터넷 환경으로 또는 인터넷 환경 외부로 메시지를 전달하는 경우 게이트웨이는 앞에 Received: 줄을 삽입해야 하지만 이미 헤더 안에 있는 Received: 줄을 변경해서는 안 된다.

다양한 환경에서 만들어지는 메시지의 "Received:" 필드는 이 규격과 정확히 일치하지 않을 수도 있다. 그러나, Received: 줄의 가장 중요한 사용은 메일 오류를 디버깅하는 것이다. 그러나 이 디버깅은 Received: 줄을 "교정"하려는 의도를 가진 호의적인 게이트웨이에 의해 크게 변경될 수 있다. 비 SMTP 환경에서 발생하는 추적 필드의 또 하나의 결과로서, 수신 시스템은 추적 필드의 형식에 기초하여 메일을 거부해서는 안되며 그러한 필드에서의 사소하게 잘못된 정보나 형식에 대해서 매우 견고해야 한다.

게이트웨이는 자체(게이트웨이)에서 제공하는 Received 필드(들)의 "via" 절에 환경과 프로토콜을 나타내야 한다.

4.8.3 게이트웨이 연결에서의 주소

인터넷 안에서 게이트웨이는 SMTP 명령과 RFC 822 헤더, 그리고 모든 유효한 RFC 822 메시지에서 사용된 모든 유효한 주소를 받아들여야 한다. 게이트웨이에서 생성한 주소와 헤더는 해당 (이 표준과 RFC 822를 포함하여) 인터넷 표준에 적용될 수 있어야 한다. 물론, 4.3절에서 다양한 SMTP 시스템에 대해 설명한 바와 같이 게이트웨이는 소스 라우트를 처리하기 위한 동일한 규칙을 따라야 한다.

4.8.4 게이트웨이 연결에서의 다양한 헤더 필드

게이트웨이는 인터넷 메일 환경으로 전달되는 메시지의 모든 헤더 필드가 인터넷 메일의 요구사항을 충족할 수 있도록 해야 한다. 특히, "From:", "To:", "Cc:" 필드 등에서의 모든 주소는 RFC 822 구문을 충족하도록 (필요할 경우) 변환되어야 하며, 정식 도메인 이름만을 참조해야 하고 응답을 보내는데 효과적이고 유용해야 한다. 인터넷 프로토콜에서 다른 환경의 프로토콜로 메일을 변환하는 데 사용되는 변환 알고리즘은, 외부 메일 환경에서의 오류 메시지가 RFC 822 메시지의 "FROM:" 필드(또는 다른 필드)에 열거된 발신자가 아닌 SMTP 엔벨로프의 반환 경로로 전달되도록 해야 할 것이다.

4.8.5 게이트웨이 연결에서의 엔벨로프

마찬가지로, 다른 환경에서 인터넷으로 메시지를 전달할 때, 게이트웨이는 외부환경에서 제공된 오류메시지 반환 주소에 따라 엔벨로프(envlope) 반환 주소를 설정해야 한다. 외부 환경에 상응하는 개념이 없다면 게이트웨이는 메시지 발기인의 주소를 마지막 수단의 기본값으로 사용하여 최상의 근사값을 선택한 후 사용해야 한다.

4.9 세션 및 연결 종료

클라이언트가 QUIT 명령을 보낼 때 SMTP 연결이 종료된다. 서버는 긍정적인 답장 코드로 응답한 다음, 연결을 종료한다.

SMTP 서버는 다음의 경우를 제외하고는 의도적으로 연결을 닫아서는 안 된다.

- QUIT 명령을 수신하고 나서 221 응답으로 회답한 후.

- SMTP 서비스를 종료해야 하는 필요성을 감지하고 나서 421 응답 코드를 반환한 후. 이 응답 코드는 서버가 명령을 수신한 이후 또는 필요할 경우 (다음 명령이 발생된 후 클라이언트가 이 코드를 수신한다고 가정할 때) 명령 수신과는 별도로 발생할 수 있다.

특히, 이해되지 않는 명령에 대해 연결을 종료하는 서버는 이 규격에 위배된다. 서버는 알 수 없는 명령에 대해서는 550응답을 보내고 인내하면서 클라이언트의 다음 명령을 기다린다.

외부적인 수단을 통해 강제적으로 종료되는 SMTP 서버는 종료하기 전에 SMTP 클라이언트에게 421 응답 코드를 포함하는 줄을 전송하려는 시도를 해야 할 것이다. SMTP 클라이언트는 일반적으로 다음 명령을 전송한 후 421 응답 코드를 읽을 것이다.

제어 하에 있지 않은 환경(이 규격의 의도에는 어긋나지만 피할 수 없는)으로 인해 연결 종료, 초기화 또는 다양한 통신상의 오류를 겪게 되는 SMTP 클라이언트는 메일 시스템의 견고성을 유지하기 위해 451 응답이 수신된 것처럼 메일 트랜잭션을 다루며 그에 맞게 작동해야 할 것이다.

4.10 메일링 목록과 별칭

SMTP 기능을 제공하는 호스트는 다중 배달을 위한 주소 확장에 사용된 별칭과 목록 모델 두 가지 모두를 지원해야 한다. 메시지가 확장된 목록 형태의 각 주소로 배달되거나 전달되는 경우 엔벨로프의 반환 주소(“MAIL FROM:″)는 사용자 또는 목록을 관리하는 다른 객체의 주소로 변경되어야 한다. 그러나, 이 경우 메시지 헤더 [32]는 변경되지 않은 상태로 유지되어야 한다. 특히, 메시지 헤더의 ”From″ 필드는 절대로 변경되지 않아야 한다.

중요한 메일 기능 중 하나로서 의사 우편함 주소를 목적지 우편함 주소의 목록으로 변환하여(또는 “확장”하거나 “증가”시켜) 하나의 메시지를 다중 목적지에 배달하는 방법이 있다. 메시지가 그러한 의사 우편함(때로는 “익스플로더(exploder)″라고도 함)으로 전송되는 경우, 사본은 확장된 목록의 각 우편함으로 전달되거나 재 배포된다. 서버는 단순히 목록 상의 주소를 이용해야 한다. 발기인에서처럼 특정 주소를 제거하기 위해 발견적인 방법을 적용하거나 다른 일치 규칙을 적용하는 것이 바람직하다. 여기서는 확장 규칙에 따라 그러한 의사 우편함(pseudo-mailbox)을 ”별칭“ 또는 ”목록“으로 분류한다.

4.10.1 별칭

별칭을 확장하려면 수신자 메일러는 단순히 엔벨로프에 있는 의사 우편함 주소를 각각의 확장된 주소로 교체한다. 엔벨로프의 나머지와 메시지 본문은 변경되지 않은 상태로 유지된다. 그리고 나면 메시지가 각각의 확장된 주소로 배달되거나 전달된다.

4.10.2 목록

메일링 목록은 “전달(forwarding)″에 의해서 보다는 ”재배포(redistribution)″에 의해 작동된다고 할 수 있다. 목록을 확장하려면 수신자 메일러는 엔벨로프에 있는 의사 우편함 주소를 확장된 모든 주소로 교체한다. 최종 배달에 의해 생성된 모든 오류 메시지가 메시지 발기인이 아닌 목록 관리자로 반환되도록 엔벨로프의 반환 주소가 변경된다. 메시지 발기인은 일반적으로 목록의 내용에 대한 제어권을 가지고 있지 않으며 일반적으로 성가신 오류 메시지를 찾아낼 것이다.

5. SMTP 규격

5.1 SMTP 명령

5.1.1 명령 의미론과 구문

SMTP 명령은 사용자가 요청한 메일 시스템의 기능이나 메일 전송을 정의한다. SMTP 명령은 <CRLF>에 의해 종료되는 문자열이다. 명령 자체는 뒤에 매개변수가 따라오는 경우<SP>에 의해, 그 외의 경우<CRLF>에 의해 종료되는 알파벳 문자이다. (상호운용성 향상을 위해 SMTP 수신기에서 가급적이면 종료 표시인<CRLF>이전에 나오는 여백을 허용하는 것이 바람직하다. 우편함의 로컬 부분에 대한 구문은 수신기 사이트 규정과 5.1.2절에 지정된 구문을 따라야 한다. 아래에서는 SMTP 명령을 설명하고 있다. SMTP 응답은 5.2절에서 설명하고 있다.

메일 트랜잭션은 다양한 명령의 인수로서 전달되는 몇 개의 데이터 객체를 포함한다. 역경로는 MAIL 명령의 인수이며, 순경로는 RCPT 명령의 인수이고 메일 데이터는 DATA 명령의 인수이다. 이러한 인수 또는 데이터 객체가 전송되어야 하며 트랜잭션을 마무리하는 메일 데이터의 끝 표시에 의한 확인을 기다리며 보관되고 있어야 한다. 이 작업 모델에서는 여러 가지 유형의 데이터 객체들을 유지하기 위해 개별 버퍼가 제공된다. 즉, 역경로 버퍼, 순경로 버퍼 및 메일 데이터 버퍼가 제공된다. 특정 명령들이 실행되어 특정 버퍼에 정보를 추가하거나 한 개 이상의 버퍼를 소거한다.

몇 개의 명령(RSET, DATA, QUIT)은 매개변수를 허용하지 않는 것으로 지정된다. 서버에서 제공하고 클라이언트에서 허용하는 특정 확장이 없는 경우, 클라이언트는 그러한 매개변수를 전송해서는 안되며 서버는 잘못된 구문을 가지는 것처럼 그러한 매개변수를 포함하는 명령을 거절해야 한다.

5.1.1.1 확장된 HELLO (EHLO) 또는 HELLO (HELO)

이러한 명령들은 SMTP 클라이언트를 SMTP 서버에게 식별하는 데 사용된다. 인수 필드에는 사용할 수 있는 경우 SMTP 클라이언트의 정식 도메인 이름이 지정되어 있다. SMTP 클라이언트 시스템에 의미 있는 도메인 이름이 없는 상황(예를 들면, 주소가 동적으로 할당되고 역 맵핑 레코드를 사용할 수 없는 경우)에서 클라이언트는 주소 리터럴

(5.1.3절 참조)을 전송해야 하며 원한다면 그 뒤에 클라이언트를 식별하는 데 도움이 될 정보를 지정할 수도 있다. SMTP 서버는 연결인사응답 및 이 명령에 대한 응답에서 서버 자신을 SMTP 클라이언트에게 식별하여 준다.

클라이언트 SMTP는 EHLO 명령을 실행하여 SMTP 세션을 시작한다. SMTP 서버가 SMTP 서비스 확장을 지원하는 경우 이 서버는 성공적인 응답, 실패 응답 또는 오류 응답을 제공할 것이다. 이 규격에 위배되는 SMTP 서버로서 SMTP 서비스 확장을 지원하지 않는 경우, 이 서버는 오류 응답을 생성할 것이다. 위에서 설명한 바와 같이 이전의 클라이언트 SMTP 시스템은 EHLO의 대신에 HELO(RFC 821에서 지정한 바와 같이)를 사용하므로, 서버는 HELO 명령과 그에 대한 적절한 응답을 지원해야 한다. 어떤 경우든 클라이언트는 메일 트랜잭션을 시작하기 전에 HELO 또는 EHLO를 실행해야 한다.

이러한 명령 중 하나에 대한 "250 OK" 답장은 SMTP 클라이언트와 SMTP 서버 모두가 초기 상태에 있음을 확인시켜준다. 즉, 진행 중인 트랜잭션이 없고 모든 상태 테이블과 버퍼가 소거되었음을 확인시켜준다.

구문:

ehlo = "EHLO" SP 도메인 CRLF
helo = "HELO" SP 도메인 CRLF

일반적으로 EHLO에 대한 응답은 2줄 이상이 될 것이다. 응답의 각 줄에는 키워드와 한 개 또는 그 이상의 매개변수(선택 사항)가 포함된다. 2줄 이상으로 구성되는 답장에 대한 일반 구문에 따라 이러한 키워드는 마지막 줄을 제외한 모든 줄의 경우에는 코드(250)와 하이픈 뒤에 오며, 마지막 줄의 경우에는 코드와 공백 뒤에 온다. [8]의 ABNF 표기와 터미널 기호를 사용하는 긍정적인 응답에 대한 구문은 다음과 같다.

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

ehlo-greet  = 1*(%d0-9 / %d11-12 / %d14-127)
        ; CR 또는 LF 이외의 모든 문자로 된 문자열
```

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

ehlo-keyword = (ALPHA / DIGIT) *(ALPHA / DIGIT / "-")
; ehlo-params의 추가 구문은 ehlo-keyword에 좌우된다.

ehlo-param = 1*(%d33-127)
; <SP>와 모든 제어 문자(US-ASCII 0-31 포함)를
제외한 모든 CHAR

EHLO 키워드가 대문자, 소문자 또는 대문자와 소문자의 결합으로 사용될 수 있을 지라도, 이 키워드는 항상 대/소문자를 구분하지 않는 방식으로 인식되고 및 처리되어야 한다. 이 방법은 RFC 821과 3.4.1절에 명시된 관행을 단순히 확장한 것이다.

5.1.1.2 MAIL (MAIL)

이 명령은 SMTP 서버로 메일 데이터가 배달되는 메일 트랜잭션을 초기화하는 데 사용된다. 이 SMTP 서버는 메일 데이터를 하나 이상의 우편함 또는 (아마도 SMTP를 사용하여) 다른 시스템으로 전달할 수 있다. 인수 필드에는 역경로가 포함되며 선택 사항인 매개변수 포함될 수 있다. 일반적으로, MAIL 명령은 진행 중인 메일 트랜잭션이 없을 때만 보내질 수 있으며, 자세한 내용은 5.1.4절을 참조하기 바란다.

역경로는 발신자 우편함으로 구성된다. 예전에는 선택적으로 호스트 목록을 해당 우편함 앞에 지정할 수도 있지만 이제는 이러한 방식이 무시된다(부록 III 참조). 답장이 메일 루프를 일으킬 가능성이 있는 특정한 유형의 보고 메시지(예를 들면, 메일 배달 및 배달 불가 알림 메시지)에서 역경로는 널이 될 수도 있다(4.7절 참조).

이 명령은 역경로 버퍼, 순경로 버퍼 및 메일 데이터 버퍼를 소거한다. 또한 이 명령의 역경로 정보를 역경로 버퍼에 삽입한다.

서비스 확장이 협정된 후 MAIL 명령은 특정 서비스 확장과 관련된 매개변수를 전달할 수도 있다.

구문:

"MAIL FROM:" ("<>" / 역경로)
[SP 메일 매개변수] CRLF

5.1.1.3 RECIPIENT (RCPT)

이 명령은 메일 데이터의 개별 수신자를 식별하는 데 사용된다. 이 명령의 반복적인 사용을 통해 다수의 수신자가 지정된다. 인수 필드에는 순경로가 포함되며 선택 매개변수도 포함될 수 있다.

일반적으로 순경로는 요청된 목적지 우편함으로 구성된다. 송신 시스템은 소스 라우트로 알려진 선택 사항인 호스트 목록을 생성해서는 안 된다. 수신 시스템은 소스 라우트 구문을 인식할 수 있어야 한다. 하지만 소스 라우트 기술을 제거하고 소스 라우트가 제공되지 않은 것처럼 우편함과 관련된 도메인 이름을 이용해야 한다.

마찬가지로, 릴레이 호스트는 소스 라우트를 제거하거나 무시해야 하며, 소스 라우트에 명기된 이름이 역경로로 복사되어서는 안 된다. 메일이 해당 최종 목적지에 도달한 경우 (순경로에는 목적지 우편함만 포함됨) SMTP 서버는 호스트 메일 규정에 따라 그 메일을 목적지 우편함에 추가한다.

예를 들면, 다음과 같은 엔벨로프 명령을 사용하여 릴레이 호스트 xyz.com에 수신되는 메일은

MAIL FROM:<userx@y.foo.org>
RCPT TO:<@hosta.int,@jkl.org:userc@d.bar.org>

일반적으로 다음과 같은 엔벨로프 명령을 사용하여 호스트 d.bar.org에 직접 전송될 것이다.

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

부록 III에서 명시한 바와 같이, xyz.com도 엔벨로프 명령을 사용하여 메일을 hosta.int로 넘겨주거나

MAIL FROM:<userx@y.foo.org>
RCPT TO:<@hosta.int,@jkl.org:userc@d.bar.org>

또는 엔벨로프 명령을 사용하여 jkl.org로 넘겨줄 수 있다.

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

물론, 호스트는 전혀 메일을 릴레이 할 필요가 없기 때문에, ("정책상의 이유로") 550 코드를 사용하여 RCPT 명령이 수신될 때 xyz.com이 메시지를 완전히 거부할 수도 있다.

서비스 확장이 협정된 경우, RCPT 명령은 서버가 제공한 특정 서비스 확장과 관련된 매개변수를 전달할 수도 있다. 클라이언트는 EHLO 응답에서 서버가 제공한 서비스 확장과 관련된 매개변수 이외의 매개변수를 전송해서는 안 된다.

구문:
"RCPT TO:" ("<Postmaster@" domain ">" / "<Postmaster>" / 순경로)
[SP Rcpt-parameters] CRLF

5.1.1.4 DATA (DATA)

일반적으로 수신기는 DATA에 대해 354 응답을 전송하고 난 다음 명령 뒤에 이어지는 줄(3.3.7절에 설명된 바와 같이 <CRLF> 시퀀스로 끝나는 문자열)을 발신자로부터의 메일 데이터로 다룬다. 이 명령이 메일 데이터를 메일 데이터 버퍼에 추가시킨다. SP. HT, CR 및 LF 이외의 제어 문자를 사용할 경우 문제가 발생할 수 있으므로 가능한 한 피해야 하지만, 메일 데이터에 128 ASCII 문자 코드 중 하나가 포함될 수도 있다.

메일 데이터는 단 하나의 구두점을 포함하는 줄, 즉, 문자 시퀀스 "<CRLF>.<CRLF>"로 종료된다(5.5.2절 참조). 이 줄은 메일 데이터의 끝을 표시한다. 이 종료 시퀀스의 첫 번째 <CRLF>는 데이터(메시지 텍스트)의 최종 줄을 끝내는 것이거나 데이터가 없을 경우에는 DATA 명령 자체를 종료하는 것이다. 메시지에 공백 줄을 삽입하려는 의도를 가진 또 하나의<CRLF>는 추가되지 않는다. 이 규칙의 한가지 예외는 메시지 본문이 <CRLF>로 끝나지 않는 최종 "줄"을 가진 채로 발기 SMTP-발신자에게 전달된 경우이다. 이 경우, 발기 SMTP 시스템은 메시지를 잘못된 메시지로서 거부하거나 수신 SMTP 서버가 "데이터의 끝" 상황을 인식하도록 <CRLF>를 추가해야 한다.

일부 UNIX 시스템에서 규정에 따르지 않는 작동에 대한 묵인으로서 <LF>로만 끝나는 줄을 허용하는 관례는 그것이 해결하는 것보다 더 많은 상호 문제를 야기하는 것으로 판명되었으므로, 견고성 향상이라는 이유에서조차 SMTP 서버 시스템은 그러한 방식으로 수행되어서는 안 된다. 특히, "<LF>.<LF>"(캐리지 리턴 없이 홀로 사용된 라인 피드) 시퀀스를 메일 데이터의 끝 표시로서의 <CRLF>.<CRLF>에 해당하는 문자열로 취급해서는 안 된다.

메일 데이터 끝 표시를 수신한 서버는 저장된 메일 트랜잭션 정보를 처리해야 한다. 이 처리는 역경로 버퍼, 순경로 버퍼 및 메일 데이터 버퍼의 정보를 사용하므로, 이 명령이 완료될 때 이러한 버퍼의 내용이 소거된다. 성공적으로 처리되는 경우 수신자는 OK 답장을 보내야 한다. 처리가 실패하면 수신자는 실패 응답을 보내야 한다. 이 때, SMTP 모델은 부분 실패를 허용하지 않는다. 즉, 서버가 배달하기 위해 메시지를 받아들이고 긍정적인 응답을 반환하거나, 받아들이지 않고 실패 응답을 반환해야 한다. 데이터 끝 표시에 대해 긍정적인 완료 답장을 보낼 경우, 수신자는 메시지에 대한 모든 책임을 맡게 된다(7.1절 참조). 5.4절에 설명된 바와 같이 그 이후에 진단되는 오류는 메일 메시지 형식으로 보고되어야 한다.

SMTP 서버가 릴레이 또는 최종 배달을 위해 메시지를 받아들이는 경우 서버는 메일 데이터의 시작 부분에 추적 레코드("시간 스탬프 줄" 또는 "Received" 줄이라고도 함)를 삽입한다. 이 추적 레코드는 메시지를 전송한 호스트의 ID, 메시지를 수신한 (이 시간 스탬프를 삽입하고 있는) 호스트의 ID 및 메시지가 수신된 날짜와 시간을 나타낸다. 릴레이된 메시지에는 여러 개의 시간 스탬프 줄이 있을 것이다. 구문을 포함하여 이러한 줄의 구성에 대한 세부 내용은 5.4절에 명시되어 있다.

DATA 명령의 작동에 대한 추가 설명은 4.3절에 명시되어 있다.

구문:

"DATA" CRLF

5.1.1.5 RESET (RSET)

이 명령은 현재의 메일 트랜잭션이 중지되도록 한다. 저장된 모든 발신자, 수신자 및 메일 데이터가 지워져야 하므로, 모든 버퍼와 상태 테이블이 소거된다. 수신자는 인수를

사용하지 않은 RSET 명령에 대해 “250 OK″ 답장을 전송해야 한다. 클라이언트는 언제라도 초기화 명령을 실행할 수 있다. 이 명령은 EHLO 직후, EHLO가 세션에서 실행되기 전, 데이터 끝 표시가 전송되고 승인된 후 또는 QUIT 직전에 실행될 경우 등은 사실상 NOOP(즉, 효과가 없는 경우)를 사용한 것과 같다. SMTP 서버는 RSET를 수신한 결과로서 연결을 종료해서는 안 된다. 연결 종료는 QUIT 명령용으로 예약되어 있다(5.1.1.10절 참조).

EHLO가 서버에 의한 추가 처리 및 응답을 포함하므로, 형식적인 의미는 동일할 지라도 일반적으로 EHLO 명령을 다시 실행하는 것보다 RSET이 훨씬 효율적일 것이다.

이 규격의 의도와는 반대되는 상황이 있다. 그러한 상황에서는 SMTP 서버가 기본 TCP 연결이 종료거나 초기화되었음을 나타내는 표시를 수신할 수도 있다. 메일 시스템의 견고성을 유지하기 위해 SMTP 서버는 이 조건에 대비해야 하며 연결이 사라지기 전에 QUIT가 수신된 것처럼 이 조건을 다루어야 한다.

구문:

″RSET″ CRLF

5.1.1.6 VERIFY (VRFY)

이 명령은 수신자에게 전달된 인수가 사용자를 나타내는지 또는 우편함을 나타내는 지 확인하여줄 것을 요청한다. 인수가 사용자 이름일 경우 정보는 4.5절에 명시된 대로 반환된다.

이 명령은 역경로 버퍼, 순경로 버퍼 또는 메일 데이터 버퍼에 아무런 영향도 미치지 않는다.

구문:

″VRFY″ SP String CRLF

5.1.1.7 EXPAND (EXPN)

이 명령은 수신자에게 전달된 인수가 메일링 목록을 나타내는 지 확인할 것을 요청하

며, 메일링 목록을 나타내는 경우 해당 목록의 멤버들을 반환할 것을 요청한다. 명령이 성공적으로 실행된 경우 4.5절에 설명된 바와 같이 정보를 포함한 답장이 반환된다. 멤버가 하나인 목록의 경우를 제외하고는 이 답장은 여러 줄을 포함한다.

이 명령은 역경로 버퍼, 순경로 버퍼 또는 메일 데이터 버퍼에 아무런 영향도 미치지 않으며 언제라도 실행될 수 있다.

구문:

"EXPN" SP 문자열 CRLF

5.1.1.8 HELP (HELP)

이 명령을 실행하면 서버는 클라이언트에게 유용한 정보를 전송하여 준다. 이 명령은 인수(예를 들면, 명령 이름)를 사용하고 응답으로서 보다 특정적인 정보를 반환한다.

이 명령은 역경로 버퍼, 순경로 버퍼 또는 메일 데이터 버퍼에 아무런 영향도 미치지 않으며 언제라도 실행될 수 있다.

SMTP 서버는 인수를 사용하지 않는 HELP를 지원해야 하며 인수를 사용한 HELP 명령을 지원할 수도 있다.

구문:

"HELP" [SP 문자열] CRLF

5.1.1.9 NOOP (NOOP)

이 명령은 어떠한 매개변수 또는 이전에 입력된 명령에 아무런 영향을 미치지 않는다. 수신자에게 OK 답장을 전송하라고 요청하는 것 외에는 어떠한 작업도 요구하지 않는다.

이 명령은 역경로 버퍼, 순경로 버퍼 또는 메일 데이터 버퍼에 아무런 영향도 미치지 않으며 언제라도 실행될 수 있다. 매개변수 문자열이 지정되는 경우 서버는 그 문자열을 무시해야 한다.

구문:

"NOOP" [SP 문자열] CRLF

5.1.1.10 QUIT (QUIT)

이 명령은 수신자가 OK 답장을 전송하고 나서 전송 채널을 닫아야 한다고 지정한다.

수신자는 (오류가 있었을 지라도) QUIT 명령을 수신한 후 그 명령에 응답할 때까지 의도적으로 전송 채널을 닫아서는 안 된다. 발신자는 QUIT 명령을 보낼 때까지 의도적으로 전송 채널을 닫아서는 안되며 (이전 명령에 대해 오류 응답이 있었을 지라도) 답장을 수신할 때까지 기다려야 한다. 위의 규정에 위배되거나 시스템 또는 네트워크 실패로 인해 연결이 영구적으로 닫히는 경우 서버는 진행중인 트랜잭션을 취소해야 하지만 이전에 완료된 트랜잭션의 실행을 취소해서는 안되며, 일반적으로 진행 중인 명령이나 트랜잭션이 임시 오류(이를테면, 4로 시작하는 응답)를 수신한 것처럼 작동해야 한다.

QUIT 명령은 언제라도 실행될 수 있다.

구문:

"QUIT" CRLF

5.1.2 명령 인수 구문

(해당되는 경우 [8]에 지정된 구문을 사용하는) 위의 명령의 인수 필드에 대한 구문이 아래에 명시되어 있다. 아래 제공된 결과 중 일부는 부록 III에 설명된 대로 반드시 소스 라우트와 함께 사용되어야만 한다. ALPHA, DIGIT, SP, CR, LF, CRLF와 같은 이 문서에 정의되지 않은 터미널은 “핵심” 구문[8(6장)] 또는 메시지 형식 구문 [32]로 정의되어 있다.

```
역경로 = 경로
순경로 = 경로
경로 = "<" [ A-d-l ":" ] 우편함 ">"
A-d-l = At-domain *( "," A-d-l )
    ; “소스 라우트”라고도 하는 이 형태를 허용해야 하며
    ; 생성해서는 안되고, 무시해야 한다는 것에 주목하라.
At-domain = "@" 도메인
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 / %d62-127)
; "=", SP 및 제어 문자를 제외한 모든 CHAR
키워드 = Ldh-str
인수 = 원자
도메인 = (하위 도메인 1*("." 하위 도메인)) / 주소 리터럴
하위 도메인 = Let-dig [Ldh-str]

주소 리터럴 = "[" IPv4-주소 리터럴 /
IPv6-주소 리터럴 /
일반 주소 리터럴 "]"
; 5.1.3절 참조

우편함 = 로컬 부분 "@" 도메인

로컬 부분 = 구두점 문자열 / 인용 문자열
; 대/소문자를 구분할 것임

구두점 문자열 = 원자 *("." 원자)

원자 = 1*atext

인용 문자열 = DQUOTE *qcontent DQUOTE

문자열 = 원자 / 인용 문자열

로컬-부분에 대한 위의 정의는 상호운용성을 최대화하기 위해 매우 관대한 편이지만, 메일을 수신할 호스트는 로컬-부분에서 인용 문자열을 필요로 하거나(또는 사용하거나) 로컬 부분이 대/소문자를 구분하는 경우 우편함 정의를 피해야 한다. 로컬 부분을 생성해야 하거나 비교(예를 들면, 특정 우편함 이름과 비교)해야 하는 목적을 가진 경우 모든 인용 형태는 동일하게 다뤄져야 하며 송신 시스템은 가능한 한 최소한의 인용을 사용하는 형태를 전송해야 한다.

시스템은 비ASCII 문자(1로 설정된 상위 비트 세트를 가진 옥텟) 또는 ASCII "제어 문자"(십진수 0-31 및 127)를 SMTP에서 사용하기 위한 방식으로 우편함을 정의해서는 안 된다. 이러한 문자는 MAIL 또는 RCPT 명령이나 우편함 이름을 필요로 하는 다른 명령에서 사용해서는 안 된다.

역슬래시("\")는 인용 문자로서 다음 문자가 (일반 해석이 아닌) 글자 그대로 사용되어야 함을 나타낼 때 사용된다. 예를 들면, "Joe\,Smith"는 필드의 네 번째 문자가 쉼표인 하나의 아홉 개 문가로 구성된 사용자 필드를 나타낸다.

상호운용성을 촉진시키고 명명규칙과 응용 부분에서 장기간 유지되어왔던 DNS의 전통적인 사용에 대한 지침과의 일관성을 유지하기 위해(예를 들면, 기본 DNS 문서의 3.3.1 절, RFC1035 [22]를 참조), 알파벳, 숫자 및 하이픈 이외의 문자가 SMTP 클라이언트 또는 서버의 도메인 이름의 라벨에 나타나서는 안 된다. 특히, 밑줄 문자는 허용되지 않는다. 잘못된 문자 코드가 사용되었거나 다른 거절 이유가 없는 경우 명령을 수신하는 SMTP 서버는 501 응답으로 해당 명령을 거절해야 한다.

5.1.3 주소 리터럴

때로는 호스트가 도메인 이름 시스템에 알려지지 않으며 통신(특히, 오류를 복구하거나 보고하기 위한 통신)이 차단되는 경우도 있다. 이 장벽을 통과하기 위해 특수한 주소 리터럴 형태가 도메인 이름의 대안으로서 허용된다. IPv4 주소의 경우, 이 형태는 점으로 구분되고 [123.255.37.2]와 같이 각 괄호가 지정하는 네 개의 작은 십진 정수를 사용한다. 이 형태는 옥텟 시퀀스 형태로 (IPv4) 인터넷 주소를 나타낸다. IPv6 및 표준화될 다른 형태의 주소 지정의 경우, 그 형태는 IPv6 표준 [17]의 일부로 지정되는 형식에서 주소 구문, 콜론 및 주소 자체를 나타내는 표준화된 "태그"로 구성된다.

특히:

IPv4-주소 리터럴 = Snum 3("." Snum)
IPv6-주소 리터럴 = "IPv6:" IPv6-addr
일반 주소 리터럴 = 표준화된 태그 ":" 1*dcontent
표준화된 태그 = Ldh-str
; 표준 RFC에 지정되고 IANA로 등록되어야 한다.

Snum = 1*3DIGIT ; 0 ~ 255 범위에 속하는 십진 정수 값을 나타냄.

Let-dig = ALPHA / DIGIT

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

IPv6-addr = IPv6-full / IPv6-comp / IPv6v4-full / IPv6v4-comp

IPv6-hex = 1*4HEXDIG

IPv6-full = IPv6-hex 7(":" IPv6-hex)

IPv6-comp = [IPv6-hex *5(":" IPv6-hex)] "::" [IPv6-hex *5(":" IPv6-hex)]

; "::"는 최소 2개의 16비트 제로(0) 그룹을 나타냄.

; "::" 외에 6개의 그룹이 존재할 수 있다.

IPv6v4-full = IPv6-hex 5(":" IPv6-hex) ":" IPv4-주소 리터럴

IPv6v4-comp = [IPv6-hex *3(":" IPv6-hex)] "::" [IPv6-hex *3(":" IPv6-hex) ":"] IPv4-주소 리터럴

; "::"는 최소 2개의 16비트 제로(0) 그룹을 나타냄.

; "::" 외에 4개의 그룹이 존재할 수 있으며

; IPv4-주소 리터럴이 존재할 수 있다.

5.1.4 명령의 순서

이러한 명령이 사용되는 순서에는 제한이 없다.

메일 트랜잭션이 포함될 세션은 EHLO 명령을 사용하여 처음으로 초기화되어야 한다. SMTP 서버는 이러한 초기화 없이도 비메일 트랜잭션(예를 들면, VRFY 또는 EXPN)을 위한 명령은 허용해야 한다.

EHLO 명령은 세션에서 나중에 클라이언트에 의해 실행될 수 있다. 세션이 시작된 후 EHLO가 실행되면 SMTP 서버는 모든 버퍼를 지우고 RSET 명령이 실행된 것처럼 정확하게 상태를 초기화해야 한다. 다른 말로 하면, RSET 시퀀스 바로 직후에 오는 EHLO는 중복된 것이다. 하지만 불필요한 명령을 실행함으로 인해 성능 비용이 많이 들뿐 아무런 해를 끼치지는 않는다.

SMTP 서버에서 EHLO 명령을 받아들일 수 없다면, 해당되는 경우 501, 500 또는 502 실패 응답이 반환해야 한다. SMTP 서버는 이러한 답장을 전송한 후 EHLO가 수신되기

전에 유지하고 있었던 동일한 상태로 유지되어야 한다.

가능하다면 SMTP 클라이언트는 EHLO 명령에 추가되는 도메인 매개변수가 해당 호스트를 위한 유효한 호스트 이름(CNAME 또는 MX 이름이 아님)이 되도록 해야 한다. 이렇게 하는 것이 불가능할 경우(예를 들면, 클라이언트의 주소가 동적으로 지정되고 클라이언트가 명확한 이름을 가지고 있지 않은 경우) 주소 리터럴은 도메인 이름과 클라이언트를 식별하는 데 도움을 주기 위해 제공된 보충 정보를 대신해야 한다.

SMTP 서버는 EHLO 명령에 추가되는 도메인 이름 매개변수가 실제로 클라이언트의 IP 주소에 해당함을 확인할 수 있다. 그러나, 서버는 검증이 실패되는 경우 이러한 이유로 메시지의 수신을 거절해서는 안 된다. 검증 실패에 대한 정보는 로그 기록 및 추적용일 뿐이다.

NOOP, HELP, EXPN, VRFY 및 RSET 명령은 세션진행 중에 또는 그전에 세션을 초기화하지 않고도 언제라도 사용될 수 있다. SMTP 서버는 아직 EHLO가 수신되지 않았을 지라도 일반적으로 이러한 명령을 처리해야 한다(즉, 503 코드를 반환해서는 안 된다). 클라이언트는 이러한 명령을 전송하기 전에 EHLO를 사용하여 세션을 열어야 한다.
이러한 규칙에 따르자면, EHLO 명령이 EXPN보다 선행하지 않거나 액세스의 거부가 클라이언트의 IP 주소 또는 다른 인증이나 권한 결정 방법에 기초하는 것이 아니라면 EXPN 명령에 대한 응답으로 “550 access denied to you(액세스가 거부됨)"를 표시하는 RFC 821의 예는 올바르지 않다.

MAIL 명령(또는 구식 SEND, SOML 또는 SAML 명령)은 메일 트랜잭션을 시작한다. 시작하고 나면, 메일 트랜잭션은 명령을 시작하는 트랜잭션, 하나 이상의 RCPT 명령 및 DATA 명령의 순서대로 구성된다. 메일 트랜잭션은 RSET(또는 새 EHLO) 명령에 의해 중지될 수 있다. 세션에는 0개 이상의 트랜잭션이 있을 수 있다. MAIL(또는 SEND, SOML, SAML)은 메일 트랜잭션이 이미 열려 있는 경우에는 전송되지 않아야 한다, 즉, 메일 트랜잭션이 세션에서 시작되지 않았거나 이전 메일 트랜잭션이 성공적인 DATA 명령으로 끝났거나, 아니면 이전 메일 트랜잭션이 RSET을 통해 중지된 경우에만 전송되어야 할 것이다.

명령 인수를 가지고 있는 트랜잭션을 받아드릴 수 없는 경우 501 실패 응답이 반환되어야 하며 SMTP 서버가 동일한 상태로 유지되어야 한다. 트랜잭션의 명령이 서버에서

처리할 수 없을 정도가 되면 503 실패 응답이 반환되어야 하며 SMTP 서버가 동일한 상태로 유지되어야 한다.

세션의 마지막 명령은 QUIT 명령이어야 한다. QUIT 명령은 세션 중에 언제든지 사용할 수는 없지만, 전송되어 받아들여진 세션 열기 명령이 없을 지라도 클라이언트 SMTP에서 연결 닫기를 요청할 때는 이 명령을 사용해야 한다.

5.1.5 개인용 명령

3.2.2절에 명시된 바와 같이 "X"로 시작하는 명령은 클라이언트(송신)와 서버(수신) SMTP 에이전트 사이에 상호 약속을 통해 사용될 수 있다. 그러한 명령을 인식하지 않는 SMTP 서버는 "500 Command not recognized(명령이 인식되지 않음)" 응답을 보내게 될 것이다. 확장된 SMTP 서버는 EHLO 명령에 대한 응답으로 이러한 개인용 명령들과 관련된 특징 이름을 나열할 수 있다.

"X"로 시작하지 않는 것으로서 SMTP 시스템에서 전송하거나 받아들인 명령은 3.2.2절의 요구사항을 따라야 한다.

5.2 SMTP 답장

SMTP 명령에 대한 답장은 메일 전송 과정에서의 요청과 작업의 동기화를 유지하고 SMTP 클라이언트가 항상 SMTP 서버의 상태를 알고 있음을 보장하는 데 사용된다. 모든 명령은 정확히 한 개의 답장을 생성해야 한다.

명령-답장 시퀀스에 대한 자세한 내용은 5.3절에서 설명한다.

SMTP 답장은 세 자리 숫자(세 자리 숫자 문자로 전송됨)와 이 문서에서 별도의 언급이 없을 경우 그 뒤에 이어지는 텍스트로 이루어진다. 숫자는 오토마타(automata)에서 다음 상태를 결정하는 데 사용하기 위한 것이다. 텍스트는 사람을 위한 용도이다. 세 자리 숫자에는 SMTP 클라이언트가 텍스트를 검토할 필요가 없도록 충분히 인코딩 된 정보가 들어 있으며 이 정보를 버리거나 해당되는 경우 사용자에게 넘겨줄 수 있다. 이 문서의 다른 곳에 예외 경우가 언급되어 있다. 특히, 220, 221, 251, 421 및 551 응답 코드는 기기에 의해 구문이 분석되고 해석되어야 하는 메시지 내용과 관련이 있다. 일반적인 경우,

텍스트는 수신기와 내용에 의존하므로 아마도 각 응답 코드에 대해 다양한 텍스트가 있을 것이다. 응답 코드 이론에 대한 설명은 4.52.1절에 명시되어 있다. 형식적으로, 답장은 (5.2.1절에 정의된 바와 같이) 세 자리 숫자 코드, <SP>, 한 줄 텍스트 및 <CRLF> 또는 두 줄 이상의 답장 순서로 정의된다. 경우에 따라 이 규격에 위배되는 텍스트는 전송되지 않으므로, 그러한 텍스트를 수신하지 않는 클라이언트는 (후위 공백 문자를 사용하든 안 하든 상관없이) 코드만을 처리할 수 있어야 할 것이다. 일반적인 상황에서 실행했을 때 오직 EHLO, EXPN 및 HELP 명령의 경우에만 두 줄 이상의 답장을 만들겠지만, 두 줄 이상의 답장은 모든 명령에 허용된다.

ABNF에서 서버 응답은 다음과 같다.

Greeting = "220 " 도메인 [SP 텍스트] CRLF
Reply-line = 응답 코드 [SP 텍스트] CRLF

여기서 "Greeting"은 서버가 연결 부분을 열고 있음을 알리는 220 응답에만 나타난다.

SMTP 서버는 이 문서에 열거된 응답 코드만 보내야 한다. SMTP 서버는 해당되는 경우 예에 표시된 텍스트를 사용해야 한다.

SMTP 클라이언트는 텍스트에 의해서가 아닌 응답 코드에 의해서만 작업을 결정해야 한다("주소 변경" 251과 551 및 필요할 경우, 220, 221 및 421 답장의 경우 제외). 일반적인 경우, (발신자가 빈 코드를 보내지 말아야 하겠지만) 텍스트가 없는 경우를 포함하여 모든 텍스트가 받아들여져야 한다. 응답 코드 뒤에 이어지는 공백(비어있음)은 텍스트의 일부로 간주된다. 가능할 경우 수신자 SMTP는 응답 코드의 첫 번째 숫자(심각도 표시)를 테스트해야 한다.

아래 나와 있는 코드 목록이 영구적인 것으로 해석되어서는 안 된다. 선호하는 응답의 텍스트 부분에 제공되는 보충 정보와 함께 새 코드를 추가하는 일은 드물게 발생하는 중요한 작업이겠지만, 새로운 표준이나 표준(Standards Track) 규격의 결과로 새 코드를 추가할 수도 있다. 따라서, 발신자 SMTP는 이 문서에 지정되지 않은 코드를 처리할 수 있도록 준비해야 하며 첫 번째 자리 숫자만을 해석하여 그러한 추가 작업을 수행해야 한다.

5.2.1 응답 코드의 심각도와 이론

응답에서 세 자리 숫자에서 각 숫자는 특수한 의미를 지니고 있다. 첫 번째 자리는 응답이 양호한지 불량한지 아니면 불완전한지를 나타낸다. 단순한 SMTP 클라이언트 또는 예상치 못한 코드를 수신한 SMTP 클라이언트는 이 첫 번째 자리를 확인하여 다음 작업(계획된 대로 실행, 재실행, 생략 등)을 결정할 수 있을 것이다. 발생한 오류의 종류(예를 들면 메일 시스템 오류, 명령 구문 오류)가 무엇인지를 대략적으로 알고자 하는 SMTP 클라이언트는 두 번째 자리를 확인할 수 있다. 세 번째 자리와 존재할 수 있는 모든 보충 정보는 정보의 세부 등급을 지정하기 위해 남겨둔다.

답장 코드의 첫 번째 자리에는 5개의 값을 사용할 수 있다.

1yz 긍정 예비 답장

명령이 받아들여졌지만 요청된 작업이 미결된 상태로 있으며 이 응답의 정보 확인이 보류되고 있다. SMTP 클라이언트는 작업을 계속할 것인지 중지할 것인지를 지정하는 또 하나의 명령을 보내려 할 것이다. (참고: 확장되지 않은 SMTP는 이러한 유형의 답장을 허용하는 어떠한 명령도 제공하지 않으므로 계속 명령이나 중지 명령이 없다.)

2yz 긍정 완료 답장

요청된 작업이 성공적으로 완료되었다. 새로운 요청이 시작될 수 있다.

3yz 긍정 중간 답장

명령이 받아들여졌지만, 요청된 작업이 미결된 상태로 있으며 추가 정보의 수신을 기다리고 있다. SMTP 클라이언트는 이 정보를 지정하는 또 하나의 명령을 전송해야 한다. 이 답장은 명령 순서 그룹에서(예를 들면, DATA에서) 사용된다.

4yz 임시 부정 완료 답장

명령이 받아들여지지 않았으며 요청된 작업이 실행되지 않았다. 그러나, 오류 조건이 임시 조건이므로 작업이 다시 요청될 수 있다. 발신자는 (있다면) 명령 순서의 시작 부분으로 돌아가야 한다. 두 개의 서로 다른 사이트(수신자 SMTP 에이전트와 발신자 SMTP 에이전트)가 해석에 합의해야 하는 경우 “임시(transient)”라는 단어에 의미를 할당하기가 어렵다. 이 범주에 속하는 각 응답은 다른 시간 값을 가질 수 있지만 SMTP 클라이언트는 다시 시도하려 할 것이다. 답장이 4로 시작하는 범주에 속하는지 5로 시작하는 범주에

속하는지 결정할 수 있는 간단한 방법(아래 참조)은 명령 형태 또는 발신자나 수신자의 등록정보(속성)가 변경되지 않고 반복될 경우(즉, 명령이 동일하게 반복되고 수신자가 새 구현을 제시하지 않는 경우) 답장이 성공적으로 전송될 수 있다면 해당 답장은 4로 시작하는 범주에 속한다고 할 수 있다.

5yz 영구 부정 완료 답장

명령이 받아들여지지 않았고 요청한 작업이 발생하지 않았다. SMTP 클라이언트가 (동일한 시퀀스에서) 동일한 요청을 반복하지 않아야 한다. "영구" 오류 조건이 교정될지라도 인간 사용자가 SMTP 클라이언트에게 차후 어느 시점(예를 들면, 철자가 변경되거나 사용자가 계정 상태를 변경한 후)에서의 직접 작업에 의해 명령 순서를 다시 시작하게 할 수 있다.

두 번째 자리 숫자는 특정 범위에 속하는 응답을 인코드한다.

x0z 구문: 이러한 답장은 구문 오류를 나타내는데, 어떠한 기능 범주에도 속하지 않은 명령, 구현되지 않았거나 불필요한 명령을 교정한다.

x1z 정보: 상태 또는 도움말과 같은 정보 요청에 대한 응답이다.

x2z 연결: 전송 채널을 가리키는 응답이다.

x3z : 지정되지 않았음.

x4z : 지정되지 않았음.

x5z 메일 시스템 : 이 응답은 요청한 전송이나 다른 메일 시스템 작업에 대한 수신자 메일 시스템의 상태를 나타낸다.

세 번째 자리 숫자는 두 번째 자리 숫자에서 지정한 각 범주에 대해서 의미의 세부 등급을 제공한다. 응답 목록이 이를 반영한다.

각 응답 텍스트는 필수 요소가 아닌 권장 요소이며, 심지어는 관련된 명령에 따라 변경될 수도 있다. 한편, 응답 코드는 이 절에 명시된 규격을 엄격히 준수해야 한다. 수신자

구현 제품은 여기에 설명된 상황과 조금 다른 상황에 대해 새 코드를 생성해서는 안되며, 그보다는 이미 정의된 코드 중에서 적절한 것을 사용해야 한다.

예를 들어, 성공적으로 실행될 경우에도 SMTP 클라이언트에게 어떠한 새로운 정보도 제공하지 않는 NOOP와 같은 명령은 250 답장을 반환할 것이다. 명령에서 특정 사이트에 국한되지 않고 구현되지 않은 작업을 요청하는 경우에는 502 답장이 반환된다. 명령은 구현되었지만 구현되지 않은 매개변수를 가진 경우에 이 답장은 504 답장으로 변경된다.

답장 텍스트가 두 줄 이상이 될 수도 있다. 이러한 경우 답장의 판독을 멈출 수 있는 시점을 SMTP 클라이언트가 알도록 전체 텍스트에 마크 표시를 해야 한다. 그러기 위해서는 두 줄 이상으로 된 답장을 나타내기 위한 특수한 형식이 필요하다.

두 줄 이상으로 된 답장의 형식은 마지막 줄을 제외한 모든 줄에서 답장 코드로 시작하여 그 바로 뒤에 하이픈('-")(빼기 기호라고도 함)이 이어지고 그 뒤에 텍스트가 이어져야 한다. 마지막 줄은 답장 코드로 시작하고 그 바로 뒤에 <SP>가 이어지며 원할 경우 텍스트가 올 수 있으며 그 뒤에 <CRLF>가 이어질 것이다. 위에서 언급한 바와 같이, 서버는 후속 텍스트가 전송되지 않는 경우 <SP>를 전송해야 하지만 클라이언트는 <SP>가 생략된 경우도 대비해야 한다.

예를 들면:

123-첫 번째 줄
123-두 번째 줄
123-234 숫자로 시작하는 텍스트
123 마지막 줄

대부분의 경우 SMTP 클라이언트는 단순히 응답 코드로 시작되어 <SP> 또는 <CRLF>가 이어지는 줄을 검색하고 이전의 모든 줄을 무시해야 한다. 몇 가지 경우에 있어 응답 "텍스트"에는 클라이언트에 중요한 데이터가 있다. 클라이언트는 현재 컨텍스트에서 이러한 경우를 식별할 수 있을 것이다.

5.2.2 기능 그룹에 의한 응답 코드

500 구문 오류, 명령을 인식할 수 없음

(여기에는 명령줄과 같은 매우 긴 오류가 포함될 수도 있음)

501 매개변수 또는 인수의 구문 오류

502 명령이 구현되지 않았음 (4.2.4절 참조)

503 잘못된 명령 순서

504 명령 매개변수가 구현되지 않았음

211 시스템 상태 또는 시스템 도움 응답

214 도움말 메시지

(수신자 사용법과 특정 비표준 명령의 의미에 대한 정보. 이 답장은 인간 사용자에게만 유용함)

220 <domain> 서비스 준비

221 <domain> 서비스가 전송 채널을 닫음.

421 <domain> 서비스를 사용할 수 없어 전송 채널을 닫음.

(이 답장은 서비스가 종료해야 한다는 사실을 알고 있을 경우 모든 명령에 대한 답장이 될 수도 있음)

250 요청한 메일 작업 정상(OK), 완료

251 로컬 사용자 아님. <순경로>로 전달될 것임.

(4.4절 참조)

252 사용자를 확인(VRFY)할 수 없지만, 메시지를 받고 배달을 시도할 것임.

(4.5.3절 참조)

450 요청된 메일 작업이 수행되지 않았음: 우편함을 사용할 수 없음

(예를 들면, 우편함 사용 중)

550 요청된 작업을 수행하지 않았음: 우편함을 사용할 수 없음

(예를 들면, 우편함을 찾을 수 없거나 액세스할 수 없거나 아니면 정책상의 이유로 명령이 거부됨)

451 요청된 작업이 중지됨: 처리 시 오류

551 로컬 사용자 아님; <순경로>로 시도하기 바람.

(4.4절 참조)

452 요청된 작업이 수행되지 않음: 시스템 저장 공간이 부족함.

552 요청된 메일 작업이 중지됨: 저장 공간 할당을 초과함

553 요청된 작업이 수행되지 않음: 우편함 이름이 허용되지 않음

(예를 들면, 우편함 구문이 올바르지 않음)

354 메일 입력 시작. <CRLF>.<CRLF>로 끝남

554 트랜잭션 실패(또는 연결 열기 응답 여기에 SMTP 서비스 없음에서)

5.2.3 수치 순서로 된 답장 코드

211 시스템 상태 또는 시스템 도움 응답

214 도움말 메시지

(수신자 사용법과 특정 비표준 명령의 의미에 대한 정보. 이 답장은 인간 사용자에게만 유용함)

220 <도메인> 서비스 준비

221 <도메인> 서비스에서 전송 채널을 닫음

250 요청된 메일 작업 정상(OK), 완료

251 로컬 사용자 아님. <순경로>로 전달될 것임.

(4.4절 참조)

252 사용자를 확인(VRFY)할 수 없지만, 메시지를 받고 배달을 시도할 것임.

(4.5.3절 참조)

354 메일 입력 시작. <CRLF>.<CRLF>로 끝남

421 <domain> 서비스를 사용할 수 없어 전송 채널을 닫음.

(이 답장은 서비스가 종료해야 한다는 사실을 알고 있을 경우 모든 명령에 대한 답장이 될 수도 있음)

450 요청된 메일 작업이 수행되지 않았음: 우편함을 사용할 수 없음

(예를 들면, 우편함 사용 중)

451 요청된 작업이 중지됨: 처리 시 오류

452 요청된 작업이 수행되지 않음: 시스템 저장 공간이 부족함.

500 구문 오류, 인식되지 않은 명령

(여기에는 명령줄과 같은 매우 긴 오류가 포함될 수도 있음)

501 매개변수 또는 인수에서의 구문 오류

502 명령이 구현되지 않음 (5.2.4절 참조)

503 잘못된 명령 순서

504 명령 매개변수가 구현되지 않았음

550 요청된 작업을 수행하지 않았음: 우편함을 사용할 수 없음

(예를 들면, 우편함을 찾을 수 없거나 액세스할 수 없거나 아니면 정책상의 이유로 명령이 거부됨)

551 로컬 사용자 아님; <순경로>로 시도하기 바람.

(4.4절 참조)

552 요청된 메일 작업이 중지됨: 저장 공간 할당을 초과함
553 요청된 작업이 수행되지 않음: 우편함 이름이 허용되지 않음
(예를 들면, 우편함 구문이 올바르지 않음)
554 트랜잭션 실패(또는 연결 열기 응답 여기에 SMTP 서비스 없음에서)

5.2.4 응답 코드 502

응답 코드 502(명령이 구현되지 않음)가 다른 코드보다 우선적으로 반환되어야 하는 경우 문제가 발생되어 왔다. 502는 SMTP 서버에서 실제로 명령을 인식하지만 구현하지는 않는 경우에 사용되어야 한다. 이 명령이 인식되지 않는 경우 500이 반환되어야 한다. 확장된 SMTP 시스템은 EHLO에 대한 응답으로 502(또는 500) 응답을 반환할 기능들의 목록을 표시해서는 안 된다.

5.2.5 DATA와 후속 <CRLF>.<CRLF> 이후의 답장 코드

DATA 명령이 <CRLF>.<CRLF>로 완료된 후에 SMTP 서버가 긍정 완료 상태(2yz 코드)를 반환하는 경우 다음에 대한 책임을 받아들여야 한다.

- 메시지 배달(수신자 우편함이 존재하는 경우) 또는

- 임시 조건으로 인해 배달하지 못한 메시지를 배달하려 하는 경우 4.5.4절에 명시된 바와 같이 시간 간격을 두고 적당한 횟수만큼 배달 재시도.

- 영구 조건으로 인해 배달하지 못한 메시지를 배달하려 하거나 임시 조건으로 인해 배달하지 못한 메시지를 반복하여 배달하려 하는 경우, (SMTP MAIL 명령에서 주소를 사용하여) 원래의 메시지의 발신자에게 적절한 알림 메시지를 전송함

DATA 명령이 <CRLF>.<CRLF>로 완료된 후 SMTP 서버가 영구적인 오류 상태(5yz) 코드를 반환하는 경우 SMTP 서버는 해당 메시지를 배달하려는 후속 시도를 해서는 안 된다. SMTP 클라이언트는 해당 메시지에 대한 배달 책임을 지니고 있으며 사용자에게 해당 메시지를 반송하거나 추후에 다시 시도하기 위해 대기열에 넣어둘 수 있다.(5.5.4.1절 참조)

메시지를 발기한 사용자는 영구 오류를 해석할 때처럼 임시 오류 상태(메일 메시지 또는 다른 수단을 통해)의 반환을 배달 불가능 표시로 해석할 수 있어야 한다. 예를 들어, 클라이언트 SMTP에서 이러한 상태(임시오류는 재전송을 시도하고 영구오류는 사용자에게 반환하는 처리)를 성공적으로 수행하는 경우 사용자는 임시오류에 대한 답장을 받지 않을 것이다.

DATA 명령이 <CRLF>.<CRLF>와 함께 사용된 후 SMTP 서버가 영구 오류 상태(5yz) 코드를 반환하는 경우 서버는 메시지를 배달하려는 후속 시도를 해서는 안 된다. 임시 오류 상태 코드와 마찬가지로, SMTP 클라이언트는 메시지에 대한 책임을 지니고 있지만, 메시지의 사용자 검토 및 중재 없이 동일한 서버로 배달을 다시 시도해서는 안 된다.

5.3 명령과 응답의 순서 지정(sequencing)

5.3.1 순서 지정(sequencing) 개요

발신자와 수신자 사이의 통신은 발신자에 의해 제어되는 한번씩 주고받는 통신이다. 그러한 통신에서는 발신자가 명령을 실행하고 수신자는 그에 대한 응답으로 답신한다. 서비스 확장을 통해 다른 요소들이 협정되지 않는 경우 발신자는 추가 명령을 보내기 전에 이 응답을 기다려야 한다.

중요한 응답 중 하나로서 연결에 대한 인사 답장이 있다. 일반적으로, 수신자는 연결이 완료될 때 220 “서비스 준비” 응답을 보낼 것이다. 발신자는 어떤 명령을 보내기 전에 이 인사 메시지를 기다려야 한다.

참고: 모든 유형의 인사 응답에서 응답 코드 이후에 오는 첫 번째 단어는 서버 호스트의 공식적인 이름(정식 주 도메인 이름)이다. 때로는 호스트가 의미 없는 이름을 가질 수도 있다. 이러한 경우의 대안 방법에 대한 설명은 5.1.3절을 참조하기 바란다.

예를 들면,

220 ISIF.USC.EDU 서비스 준비

또는

220 mail.foo.com SuperSMTP v 6.1.2 서비스 준비

또는

220 [10.0.0.1] Clueless 호스트 서비스 준비

아래 표는 각 명령에서 사용할 수 있는 선택적인 성공 및 실패 응답을 열거한 것이다. 이 표에 나타난 내용, 즉 수신자가 응답에 사용된 텍스트 내용을 변경할 수는 있지만 코드 번호 및 특정 명령-응답 시퀀스에 내포된 의미와 작업은 변경할 수 없다는 규정은 엄격히 준수되어야 한다.

5.3.2 명령-응답 시퀀스

각 명령은 통상적으로 가능한 응답과 함께 표시된다. 응답 앞에 사용되는 접두사로서는 중간 답장을 나타내는 “I”, 전송 성공을 나타내는 ”S”, 오류를 나타내는 “E”가 있다. 일부 서버의 경우 특수한 상황에서 다른 답장들을 생성할 수 있고 미래의 확장을 허용해야 하므로 SMTP 클라이언트는 가능하다면 답장의 첫 번째 자리만을 해석하거나 인식되지 않는 답장 코드에 대해서는 첫 번째 자리만을 해석하여 처리할 수 있어야 한다. 5.2절에 설명된 메커니즘을 사용하여 확장된 경우가 아니라면 SMTP 서버는 세 자리 이외 또는 2 ~5의 숫자로 시작하지 않는 답장코드를 SMTP 클라이언트에 전송해서는 안 된다.

이러한 순서 지정 규칙과 원칙적으로 코드 자체는 서버에서 제공하고 클라이언트가 받아들인(요청한) SMTP 확장에 의해 확장 또는 수정될 수 있다.

아래 열거된 코드 외에도, 모든 SMTP 명령은 해당되는 특정 상황을 만나게 되면 다음 코드 중 하나를 반환할 수 있다.

500 ”명령줄이 너무 긴“ 경우 또는 명령 이름이 인식되지 않은 경우. 이러한 명령 중 일부 명령에 대한 응답으로 명령이 인식되지 않음 오류를 표시하는 것은 이 규격에 위배된다는 것을 참고하라.

501 명령이나 인수의 구문 오류. 차후에 확장을 제공하기 위해 인수를 받지 않는 명령으로서 이 문서에 명시된 명령(DATA, RSET, QUIT)은 인수가 EHLO 알림 확장 없이 제공되는 경우 501 메시지를 반환해야 한다.

421 서비스가 종료되며 전송 채널을 닫는 경우.

특정 시퀀스(순서)는 다음과 같다.

CONNECTION ESTABLISHMENT
  S: 220
  E: 554
EHLO or HELO
  S: 250
  E: 504, 550
MAIL
  S: 250
  E: 552, 451, 452, 550, 553, 503
RCPT
  S: 250, 251 (그러나 251과 551의 설명은 3.4절 참조)
  E: 550, 551, 552, 553, 450, 451, 452, 503, 550
DATA
  I: 354 -> 데이터 -> S: 250
            E: 552, 554, 451, 452
  E: 451, 554, 503
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

5.4 추적 정보

SMTP 서버가 배달이나 추후 처리를 위해 메시지를 받는 경우 5.1.1.4절에 설명된 바와 같이 메시지 내용의 시작 부분에 추적("시간 스탬프" 또는 "Received") 정보를 삽입해야 한다.

이 줄은 다음과 같이 구성되어야 한다.

- SMTP 환경에 제공되어야 하는 FROM 필드는 TCP 연결에서 결정되어지는 것으로, (1) EHLO 명령에 제공된 소스 호스트의 이름과 (2) 소스 IP 주소를 포함하는 주소 리터럴 두 가지 모두를 포함해야 한다.

- ID 필드는 RFC 822에 제안된 바와 같이 "@"를 포함할 수도 있지만 필수 사항은 아니다.
- FOR 필드는 다수의 RCPT 명령이 제공된 경우 <경로> 엔트리 목록을 포함할 수 있다. 이 경우 보안 문제가 발생할 수 있으므로 일반적으로 사용하지 않는 것이 좋다. (8.2절 참조)

인터넷 메일 프로그램은 이전에 메시지 헤더에 추가된 Received: 줄을 변경해서는 안 된다. SMTP 서버는 메시지 앞에 Received 줄을 삽입해야 한다. SMTP 서버는 기존 줄의 순서를 변경하거나 다른 위치에 Received 줄을 삽입해서는 안 된다.

인터넷이 성장함에 따라, 특히 저속 릴레이에서는 문제를 발견함에 있어, Received 필드의 호환성이 중요하다. Received 필드를 만드는 SMTP 서버는 어떤 유형의 시간대 이름보다는 날짜에 있어 명시적 간격(예를 들면, -0800)을 사용해야 한다. 해당되는 경우 UT보다 로컬 시간(간격 사용)을 사용하는 것이 더 좋다. 이 공식화를 통해 로컬 상황에 대한 좀더 많은 정보가 지정될 수 있다. UT가 필요하다면 수신자는 단순한 산술 연산만을 수행하여 값을 변환하면 된다. UT를 사용하면 서버의 시간대 위치에 대한 정보를 잃게 된다. 시간대 이름을 제공하려는 경우에는 설명 안에 포함시켜야 한다.

배달 SMTP 서버가 메시지를 "최종적으로 배달"하는 경우 서버는 메일 데이터의 시작 부분에 반환-경로 줄을 삽입한다. 반환-경로는 반드시 사용해야 한다. 메일 시스템은 반환-경로를 지원해야 한다. 반환-경로 줄은 MAIL 명령에 매개변수로 사용된 <역경로>의 정보를 보존한다. 여기서 최종 배달이란 메시지가 SMTP 환경을 떠났음을 의미한다. 일

반적으로는 목적지 사용자 또는 관련 메일 우편함으로 배달되었음을 의미하지만, 어떤 경우 다른 메일 시스템에서 세부적으로 처리하거나 전송할 수도 있다.

반환-경로의 우편함이 실제 발신자의 우편함과 다를 수도 있다. 예를 들면, 오류 응답을 메시지 발신자가 아닌 특별한 오류 처리 우편함으로 배달해야 하는 경우이다. 메일링 목록이 포함되면 이러한 경우는 흔히 있는 일이며 메시지 발기인이 아닌 목록 관리자(maintainer)에게 오류를 전달하는 수단으로서 유용하게 사용할 수 있다.

위의 내용은 최종 메일 데이터가 반환-경로 줄로 시작되어, 그 뒤에 하나 이상의 시간 스탬프 줄이 이어짐을 암시한다. 이러한 줄 뒤에는 메일 데이터 헤더와 본문 [32]이 이어질 것이다.

때로는 전달이나 다른 작업이 메시지가 배달을 위해 받아들여진 후에 발생할 수 있어 SMTP 서버가 최종적으로 배달하는 지 여부를 결정하기가 어렵다. 결과적으로, 그 이후의(전달, 게이트웨이 또는 릴레이) 시스템이 반환-경로를 제거하고 필요할 경우 배달된 메시지에 그런 줄이 정확히 한 줄만 나타나도록 MAIL 명령을 다시 작성할 수도 있다.

메시지를 발기하는 SMTP 시스템은 이미 반환-경로 헤더가 포함되어 있는 메시지를 전송해서는 안 된다. 릴레이 기능을 수행하는 SMTP 서버는 메시지 데이터를 검사해서는 안되며, 특히 반환-경로 헤더가 존재하는 지를 결정하는 데 필요한 수준까지도 검사하지 말아야 한다. 최종 배달을 수행하는 SMTP 서버는 반환-경로 헤더를 없애고 나서 고유의 반환 경로를 추가할 수도 있다.

반환-경로의 기본 목적은 배달되지 않음 또는 기타 다른 메일 시스템 오류를 나타내는 메시지가 전송되어야 할 주소를 지정하는 것이다. 이 목적을 명확히 하기 위해서는 메시지가 전달될 때 정확히 한 개의 반환-경로가 존재해야 한다. 비 SMTP 전송 시 RFC 822 구문을 사용하는 시스템은 오류 보고서(예를 들면 배달되지 않은 메시지)가 전송될 전송 엔벨로프와 관련된 명확한 주소를 지정해야 한다.

사적(historical) 참고: 반환-경로 헤더(또는 MAIL 명령의 엔벨로프 역경로 주소)를 오류 메시지의 대상으로 사용하는 것에 모순된다고 여겨지는 RFC 822의 내용은 인터넷에는 적합하지 않는다. 역경로 주소(반환 경로에 복사될 때)는 배달 오류 메시지를 포함하는 모든 메일의 대상으로 사용되어야 한다.

특히:

- SMTP에서 다른 곳으로의 게이트웨이는 “다른 곳” 전송이 인터넷 도메인 주소를 사용하고 엔벨로프 발신자 주소를 개별적으로 유지함을 알 수 없다면 반환 경로 헤더를 삽입해야 한다.

- 다른 곳에서 SMTP로의 게이트웨이는 메시지에 존재하는 모든 반환 경로를 삭제한 다음 해당 정보를 SMTP 엔벨로프로 복사하거나 SMTP 엔벨로프에 MAIL 명령의 역경로 인수를 생성하기 위해 해당 정보와 다른 전송 시스템의 엔벨로프에 존재하는 정보를 결합해야 한다.

서버는 메일 데이터 끝 표시에 의해 진행되는 처리가 부분적으로만 성공적으로 수행되는 경우를 특수하게 다루어야 한다. 이러한 일은 몇 명의 수신자와 메일 데이터를 받아들인 후 메일 데이터가 수신자 모두가 아닌 일부에게 성공적으로 배달될 수 있음을 SMTP 서버가 알게 되는 경우에는 발생할 수 있다. 그러한 경우, DATA 명령에 대한 응답은 OK 응답이어야 한다. 그러나, SMTP 서버는 메시지의 발기인에게 “배달할 수 없는 메일” 알림 메시지를 작성하여 보내야 한다.

메시지를 받지 못한 수신자 모두를 열거한 하나의 알림 메시지 또는 메시지를 받지 못한 각각의 수신자를 위해 개별 알림 메시지가 전송되어야 한다. 발신자의 처리를 간소화하기 위해 가능하다면 전자의 방법을 사용하는 것이 좋다. 모든 배달할 수 없는 메일에 대한 알림 메시지는 (구식 SEND, SOML 또는 SAML 명령을 처리한 결과로 얻게되는 경우라도) MAIL 명령을 사용하여 전송되며 4.7절에서 설명한 바와 같이 널 반환 경로를 사용한다.

시간 스탬프 줄과 반환 경로 줄은 형식적으로 다음과 같이 정의된다.

반환 경로 줄 = "Return-Path:" FWS 역경로 <CRLF>

시간 스탬프 줄 = "Received:" FWS 스탬프 <CRLF>

스탬프 = 시작 도메인 끝 도메인 선택 정보 ";" FWS 날짜와 시간

; 여기서 "날짜와 시간"은 [32]의 명시된 대로 정의되지만

; "구식(obs-)" 형태이다. 특히 두 자리 연도는
; SMTP에서 금지되므로 사용해서는 안 된다.

시작 도메인 = "FROM" FWS 확장된 도메인 CFWS

끝 도메인 = "BY" FWS 확장된 도메인 CFWS

확장된 도메인 = 도메인 /
(도메인 FWS "(" TCP 정보 ")") /
(주소 리터럴 FWS "(" TCP 정보 ")")

TCP 정보 = 주소 리터럴 / (도메인 FWS 주소 리터럴)
; 클라이언트 EHLO명령이 아닌 TCP 연결에서
; 서버가 파생시키는 정보

선택 정보 = [Via] [With] [ID] [For]

Via = "VIA" FWS 링크 CFWS

With = "WITH" FWS 프로토콜 CFWS

ID = "ID" FWS String / 메시지 IDCFWS

For = "FOR" FWS 1*(경로 / 우편함) CFWS

링크 = "TCP" / 추가 링크
추가 링크 = 원자
; 링크의 추가 표준 이름은 IANA(Internet Assigned Numbers Authority)를
; 통해 등록되며 "Via"는 비인터넷 전송을 사용할 때의 기본 값이다.
; SMTP 서버는 등록된 이름만을 사용해야 한다.

프로토콜 = "ESMTP" / "SMTP" / 추가 프로토콜
추가 프로토콜 = 원자

; 프로토콜의 추가 표준 이름은 IANA(Internet Assigned Numbers Authority)를
; 통해 등록되며 SMTP 서버는 등록된 이름만을 사용해야 한다.

5.5 추가 구현 문제

5.5.1 최소 구현

SMTP를 사용할 수 있으려면 모든 수신자에 대해 다음의 최소 사항이 구현되어야 한다. 다음 명령은 이 규격을 준수하기 위해 지원되어야 한다.

EHLO
HELO
MAIL
RCPT
DATA
RSET
NOOP
QUIT
VRFY

메일 릴레이 기능 및 배달 기능을 지원하는 SMTP 서버가 포함된 모든 시스템은 예약된 우편함 “postmaster"를 대/소문자를 구분하지 않은 로컬 이름으로 지원해야 한다. 이 포스트마스터 주소는 (4.1절에서 설명한 바와 같이) 연결을 열 때 서버가 항상 554를 반환하는 경우, 반드시 필요하지 않는다. 포스트마스터를 위한 메일을 받아들일 수 있다는 것이 의미하는 바는 "RCPT TO:<Postmaster>" (도메인이 지정되지 않음)의 특별한 경우를 포함하여 메일 서비스를 제공하는 SMTP서버가 있는 어떤 도메인에서도 포스트마스터를 위한 우편함을 기술하는 RCPT명령이 지원되어야 한다는 것이다.

SMTP 시스템은 인터넷 상의 다른 시스템에서 포스트마스터로 전달되는 메일을 받아드리는 데 상당한 노력을 다해야 한다. 서비스 공격의 거부 또는 다른 보안 문제를 포함하기 위한 경우 등과 같은 최악의 경우에서는 SMTP 서버가 포스트마스터로 전달되는 메일을 차단할 수 있다. 그러나, 그러한 공격에 포함되지 않는 메시지까지 차단하는 일이 없도록 하는 방법들은 조정하기가 어려울 것이다.

5.5.2 투명성

데이터 투명성에 대한 규정은 없지만, "<CRLF>.<CRLF>" 문자 시퀀스는 메일 텍스트의 끝을 나타내는 것으로 사용자는 이 시퀀스를 전송할 수 없다. 일반적으로, 사용자는 그러한 "금지된" 시퀀스를 인식하지 못한다. 모든 사용자가 작성한 텍스트가 투명하게 전송되도록 하기 위해 다음 절차가 사용된다.

- 한 줄의 메일 텍스트를 전송하기 전에 SMTP 클라이언트는 줄의 첫 번째 문자를 검사한다. 그 문자가 구두점이면 추가로 한 개의 구두점을 줄의 시작 부분에 삽입한다.

- SMTP 서버에서 메일 텍스트 줄을 수신하는 경우 SMTP 서버는 텍스트 줄을 검사한다. 줄이 구두점 하나로만 이루어진 경우 그 구두점은 메일 끝 표시기로 다뤄진다. 첫 번째 문자가 구두점이고 줄에 다른 문자가 있는 경우 첫 번째 문자는 삭제된다.

메일 데이터에는 128개의 ASCII 문자 중 하나가 포함될 수 있다. 모든 문자들은 공백, 가로 및 세로 탭, 제어 문자를 포함하여 수신자의 우편함에 배달되어야 한다. 전송 채널이 8비트 바이트(옥텟) 데이터 스트림을 제공하는 경우 7비트 ASCII 코드는 옥텟에서 오른쪽으로 정렬되어 전송된다. 이 때 최상위 비트는 0으로 소거된다. 릴레이 기능을 제공하는 SMTP 시스템에서 이러한 조건들의 특수한 처리에 대한 설명은 4.7절을 참조하기 바란다.

일부 시스템에서는 수신되고 및 저장될 때 데이터를 변환해야 하는 경우도 있다. 이러한 변환 작업은 로컬 문자 세트로서 ASCII와는 다른 문자 세트를 사용하거나, 문자열이 아닌 레코드로 문자를 저장하거나, 아니면 특수한 문자 시퀀스를 우편함 내부의 구분 기호로 사용하는 호스트에 필요할 것이다. 그러한 변환이 필요할 경우, 특히 릴레이 되고 있는 메일에 적용되는 경우에는 변환을 역으로 바꿀 수도 있어야 한다.

5.5.3 크기와 시간 제한

5.5.3.1 크기 제한값 및 최소값

최소/최대 크기를 필요로 하는 몇 개의 객체가 있다. 각 구현 제품은 적어도 이런 크기의 객체를 수신할 수 있어야 한다. 이러한 제한보다 큰 객체는 가능한 한 피해야 한다.

그러나, 인코드된 X.400 주소[16]와 같은 일부 인터넷 메일 생성물에서는 흔히 보다 큰 객체를 필요로 한다.(클라이언트에서 이러한 객체를 전송하려 할 수도 있지만 서버가 처리할 수 없는 경우라면 서버는 그러한 객체를 거부할 수 있어야 한다.) 가능한 최대 범위까지 확장하려면 이러한 객체의 길이에 대해 제한을 두지 않는 구현 기술을 사용해야 한다.

로컬 부분

사용자 이름 또는 다른 로컬 부분의 최대 총 길이는 64개 문자이다.

도메인

도메인 이름이나 숫자의 최대 총 길이는 255개 문자이다.

경로

역경로 또는 순경로의 최대 총 길이는 256개 문자(구두점과 요소 분리 기호 포함)

명령줄

명령 단어와 <CRLF>를 포함하는 명령줄의 최대 총 길이는 512개 문자이다. SMTP 확장을 사용하면 이 제한값을 증가시킬 수 있다.

응답 줄

응답 코드와 <CRLF>를 포함하여 응답 줄의 최대 총 길이는 512개 문자이다. 그 이상의 정보는 여러 줄의 응답을 통해 전달될 수 있다.

텍스트 줄

<CRLF>를 포함하여 텍스트 줄의 최대 총 길이는 1000개(투명성을 위해 추가된 중복된 구두점은 개수에 넣지 않음) 문자이다. SMTP 서비스 확장을 사용하면 이 값을 증가시킬 수 있다.

메시지 컨텐츠

메시지 컨텐츠(메시지 헤더 및 메시지 본문 포함)의 최대 총 길이는 최소한 64K 옥텟이어야 한다. 멀티미디어 메일에 대한 인터넷 표준 [12]의 도입으로 인해 인터넷 상에의 메시지 길이가 크게 증가했으므로 모든 가능한 경우에 있어 메시지 크기 제한을 피해야 한다. 제한을 설정해야 하는 SMTP 서버는 “크기” 서비스 확장 [18]을 구현해야 하

며, 대형 메시지를 전송할 SMTP 클라이언트 시스템은 가능하다면 그 확장을 사용해야 한다.

수신자 버퍼

버퍼에 저장되어야 하는 최대 총 수신자 수는 100명이다. 100개의 RCPT 명령보다 적은 수의 RCPT 명령을 사용한 메시지를 거부(수신자 수를 초과하는 경우)하는 것은 이 규격에 위배된다. 릴레이 SMTP 서버와 배달 SMTP 서버가 메시지 헤더에 대해 확인 테스트를 수행해서는 안 된다는 일반적인 원칙은 헤더 필드에 표시된 총 수신자 수에 기초하여 메시지를 거부하는 것은 바람직하지 않다고 제안한다. 수신자의 수에 제한을 두는 서버는 이전에 받았던 주소를 소리 없이 버리기보다는 제한을 초과하는 추가 주소를 거부하는 등 규칙적인 방식으로 작동해야 한다. 100개 이상의 RCPT 명령을 포함하는 메시지를 전달해야 하는 클라이언트는 서버가 하나의 메시지에서 100개 이상의 수신자를 받아드리는 것을 거절하는 경우 100 수신자 “청크”로 전송할 수 있어야 한다.

이러한 제한값을 초과하여 발생한 오류는 응답 코드를 사용하여 보고할 수 있다. 응답 코드의 몇 가지 예는 다음과 같다.

500 줄이 너무 김.
또는
501 경로가 너무 김.
또는
452 수신자가 너무 많음(아래 참조).
또는
552 메일 데이터가 너무 많음.

답장 코드 552를 제공하면서 SMTP 서버가 RCPT 명령의 개수에 대한 구현 제한(“수신자가 너무 많음”)을 명시하는 RFC 821 [30]은 오류를 잘못 사용한 것이다. 이 조건에 맞는 올바른 답장 코드는 452이다. 클라이언트는 아래 논리를 적용하여 이 경우의 552 코드를 영구적인 오류가 아닌 임시 오류로 간주해야 한다.

규격에 부합되는 SMTP 서버에서 이 조건을 접하는 경우 수신자 버퍼에는 최소 100개의 성공적인 RCPT 명령이 있어야 한다. 서버가 메시지를 받을 수 있는 경우 이러한 최소 100개의 주소가 SMTP 클라이언트의 대기열에서 제거될 것이다. 클라이언트가 452

응답을 수신한 주소를 재전송하려 하는 경우 그러한 주소 중 최소 100개는 SMTP 서버의 수신자 버퍼에 저장될 수 있을 것이다. 무언가의 배달을 수행할 수 있는 각 재전송 시도는 이러한 수신자 중 최소 100명을 보낼 수 있을 것이다.

SMTP 서버에 RCPT 명령 개수에 대한 구현 제한이 있고 이 제한이 명시된 경우 서버는 452 응답 코드를 사용해야 한다.(그러나 클라이언트도 위에서 언급한 바와 같이 552에 대비해야 한다.) 서버에 RCPT 명령 개수에 대해 구성된 사이트-정책 제한이 있는 경우 5XX 응답 코드를 대신 사용할 수 있다. 이 경우는 특정 메시지 본문이 다수의 메일 트랜잭션을 통해 전송되었을 지라도 해당 메시지 본문에 대한 총 수신자 수가 강제로 맞춰져야 하는 경우에 가장 적합할 것이다.

5.5.3.2 시간 제한

SMTP 클라이언트는 시간 제한 메커니즘을 제공해야 한다. 전체 메일 트랜잭션의 시간을 맞추려 하기보다는 명령단위로 시간 제한을 사용해야 한다. 시간 제한은 쉽게 구성할 수 있어야 하며 가급적이면 SMTP 코드를 다시 컴파일하지 않고도 구성할 수 있어야 할 것이다. 이렇게 시간 제한을 구현하기 위해 각 SMTP 명령과 데이터 전송에 사용되는 각 버퍼에 대해 타이머가 설정된다. 후자의 방법은 전체 제한 시간이 본래부터 메시지의 크기에 비례함을 의미한다.

사용 중인 메일 릴레이 호스트에 대한 다양한 경험에 비추어 볼 때 명령당 최소 제한 시간 값은 다음과 같아야 한다.

초기 220 메시지: 5 분
SMTP 클라이언트 프로세스는 실패한 TCP 연결과 초기 220 인사 메시지 수신에서의 지연을 구분해야 한다. 많은 SMTP 서버는 TCP 연결을 받아들이지만 시스템이 더 많은 메일을 처리할 수 있도록 220 메시지의 배달을 지연시킨다.

MAIL 명령: 5 분

RCPT 명령: 5 분
메시지가 받아들여질 때까지 메일링 목록과 별칭의 처리가 연기되지 않는다면 보다 긴 제한시간이 필요하다.

DATA 시작: 2 분

DATA 명령에 대해 “354 시작 입력(Start Input)” 응답을 기다리는 시간이다.

데이터 블록: 3 분

청크 데이터를 전송하는 각 TCP SEND 호출이 완료되기를 기다리는 시간이다.

DATA 종료: 10 분

"250 정상(OK)" 응답을 기다리는 시간이다. 수신자가 메시지 데이터를 종료하는 최종 구두점을 받으면 수신자는 사용자 우편함에 메시지를 전달하기 위한 처리를 수행한다. 메시지가 성공적으로 보내졌고 서버가 배달 책임을 받아들인 이후이므로, 이 시점에서의 사소한 시간 제한은 매우 비경제적인 것이며 일반적으로 여러 개의 메시지 사본이 배달되어질 것이다. 자세한 설명은 7.1절을 참조하기 바란다.

SMTP 서버는 발신자로부터 다음 명령을 기다리는데 최소 5분의 제한 시간이 설정되어야 한다.

5.5.4 재시도 정책

호스트 SMTP 구현의 일반적인 구조에는 사용자 우편함, 전송중인 메시지를 대기열에 저장하기 위한 하나 이상의 영역 및 메일을 송신 및 수신하는 하나 이상의 데몬 프로세스가 포함된다. 정확한 구조는 호스트 상의 사용자 요건과 호스트에서 지원하는 메일링 목록의 개수 및 크기에 따라 달라질 것이다. 이 절에서는 높은 수준의 트래픽을 지원하는 메일러에 특히 유용하다고 판명된 몇 가지 최적화 방법을 설명한다.

모든 대기열 저장(queuing) 정책은 명령별로 모든 작업에 대한 제한 시간을 포함해야 한다. 대기열 저장 정책은 어떠한 상황에서도 오류 메시지에 대한 응답으로 오류 메시지를 전송해서는 안 된다.

5.5.4.1 송신 정책

SMTP 클라이언트의 일반적인 모델은 보내어질 메일의 전송을 정기적으로 시도하는 하나 또는 그 이상의 프로세스이다. 일반적인 시스템에서 메시지를 작성하는 프로그램은 새로 작성된 송신 메일에 대하여는 즉각적인 주의를 요청하기 위한 방법을 제공하는 반

면, 즉시 전송할 수 없는 메일은 대기열에 저장하고 발신자에 의해 정기적으로 재시도 되도록 한다. 메일 대기열 항목에는 메시지 자체뿐만 아니라 엔벨로프 정보도 들어 있을 것이다.

발신자는 한 번의 시도가 실패한 후 특정 대상에 대한 재시도를 연기해야 한다. 일반적으로, 재시도 간격은 최소 30분이 되어야 한다. 그러나, SMTP 클라이언트에서 배달되지 않은 이유를 판별할 수 있는 경우에는 보다 복잡하고 가변적인 방법이 효과적일 것이다.

재시도는 메시지가 전송되거나 발신자가 포기할 때까지 계속된다. 일반적으로 포기 시간은 최소 4~5일이어야 한다. 재시도 알고리즘의 매개변수는 설정할 수가 있어야 한다.

클라이언트는 대기열에 저장된 메일에 대하여 단순히 재전송을 시도하기보다는 도달할 수 없는 호스트의 목록과 이에 상응하는 연결 시간 제한 정보를 유지해야 한다.

다양한 경험에서 볼 때 오류는 대개 일시적이므로(대상 시스템의 고장 또는 연결 중단) 가급적이면 처음 한 시간 동안에는 두 번 연결을 시도한 후 실패한 메시지를 대기열에 저장하고 이후부터는 두 시간이나 세 시간 간격으로 재시도하는 정책을 사용할 것을 제안한다.

SMTP 클라이언트는 SMTP 서버와 협력하여 대기열 저장 정책에서의 지연 시간을 단축시킬 수 있다. 예를 들어, 메일이 특정 주소에서 수신되는 경우 해당 호스트를 위해 대기열에 저장된 메일은 바로 전송될 가능성이 높다. 이러한 원리를 이용하면 많은 경우에 있어 ETRN [9]와 같이 명시적 “지금 대기열 전송” 기능을 사용할 필요가 없게 된다.

이 정책은 지연 시간 대 리소스 사용을 최적화하기 위해 호스트당 다수의 주소 사용(아래 참조)에 대한 결과로서 세부적으로 변경될 수 있다.

SMTP 클라이언트는 현재 동작되지 않는 각각의 목적지 호스트를 위해 커다란 메시지 대기열을 가질 수 있다. 이러한 메시지 모두가 모든 재시도 주기에 따라 재시도 된다면 인터넷 오버헤드가 과도해질 수 있고 송신 시스템이 장시간 마비 될 것이다. SMTP 클라이언트는 일반적으로 몇 분의 제한 시간이 경과된 후에만 재시도가 실패했는지 확인할 수 있으므로 연결당 제한 시간이 1분밖에 되지 않을 지라도 동일한 호스트의 대기열에 저장된 수십 개 또는 수백 개의 메시지에 대해 재시도 되는 경우 지연 시간이 매우 길어

질 것이다.

이와 동시에, SMTP 클라이언트는 서버로부터 부정적인 응답을 캐시에 저장할 때 각별히 주의해야 한다. 최악의 경우, EHLO가 하나의 SMTP 연결에서 중에 여러 번 실행되는 경우 서버로부터 서로 다른 답장을 받을 수 있다. 보다 중요한 것은 MAIL 명령에 대해 5yz 응답을 캐시에 저장해서는 안 된다는 점이다.

메일 메시지가 여러 명의 수신자에게 전달되어야 하며 메시지의 사본이 전송될 SMTP 서버가 여러 명의 수신자에 대해 동일할 경우, 메시지에 대한 하나의 사본만 전송되어야 한다. 즉, SMTP 클라이언트는 MAIL, RCPT, DATA, ..., MAIL, RCPT, DATA 명령 순서 대신에 MAIL, RCPT, RCPT,... RCPT, DATA 순서를 사용해야 한다. 그러나, 주소가 매우 많을 경우 MAIL 명령당 RCPT 명령의 수에 대한 제한이 있을 수 있다. 가급적이면 효율적인 이 기능을 구현하는 것이 바람직하다.

마찬가지로, 제 시간에 배달될 수 있도록 하기 위해 SMTP 클라이언트는 동시에 여러 개의 송신 메일 트랜잭션을 지원할 수 있다. 그러나, 모든 리소스가 메일에만 사용되지 않도록 호스트를 보호할 수 있는 적절한 제한이 있을 수 있다.

5.5.4.2 수신 정책

SMTP 서버는 항상 SMTP 포트 상에서 수신 상태를 유지해야 한다. 그러기 위해서는 SMTP용으로 다수의 수신 TCP 연결을 지원해야 한다. 제한이 있을 수도 있지만 한 번에 두 개 이상의 SMTP 트랜잭션을 처리할 수 없는 서버는 이 규격의 의도와는 일치하지 않는다.

위에서 설명한 바와 같이, SMTP 서버가 특정 호스트 주소에서 메일을 수신하는 경우 서버는 해당 호스트 주소에 대해 보류 중인 모든 메일을 재시도하기 위해 고유의 SMTP 대기열 저장(queuing) 메커니즘을 활성화할 수도 있다.

5.5.5 널 역경로를 사용하는 메시지

널 역경로와 함께 전송되어야 할 기존 및 제안된 표준에서 필요로 하는 알림 메시지에는 몇 가지 종류가 있다. 즉, 4.7절에 설명된 비배달 알림 메시지, 다양한 종류의 배달 상

태 알림 메시지(Delivery Status Notification: DSN) [24] 및 메시지 처리 알림 메시지(Message Disposition Notification: MDN) [10]등이 있다. 이러한 모든 종류의 메시지는 이전 메시지에 대한 알림 메시지이며 이전 메일 메시지의 역경로로 전송된다. (그러한 알림 메시지가 배달되지 못할 경우 이는 일반적으로 알림 메시지가 배달될 주소로 지정된 호스트의 메일 시스템에 문제가 있음을 나타낸다. 이러한 이유로 일부 호스트에서는 메일 시스템 문제를 해결할 수 있는 사람에게, 예를 들면, 포스트마스터 별칭을 통해, 전송되지 못한 그러한 알림 메시지를 전달하도록 MTA를 설정한다.)

다른 모든 종류의 메시지(예를 들면 표준화 진행 RFC에서 널 역경로를 가지는 것을 필요로 하지 않는 메시지)는 널이 아닌 유효한 역경로와 함께 전송되어야 한다.

자동화된 전자 메일 처리기의 구현은 사람은 널 역경로를 사용하는 다양한 종류의 메시지가 올바르게 처리되도록 각별히 주의해야 한다. 특히 그러한 메시지의 목적지가 되는 시스템은 널 역경로를 사용하는 메시지에 응답해서는 안 된다.

6. 주소결정과 메일처리

(4.6절과 4.7절에서 설명한 바와 같이) SMTP 클라이언트가 처리를 위해 메일이 배달되어질 도메인을 사전적으로 식별하고 나면, 도메인 이름[22]을 결정하기 위해 DNS 조회가 수행되어야 한다. 이 이름들은 정식 도메인 이름(FQDN)일 것이다. 부분 이름이나 로컬 별칭으로부터 FQDN을 추론하기 위한 방법은 이 규격에 포함되지 않으며, 많은 문제가 발생하는 것으로 알려져 있으므로 일반적으로는 사용하지 않는 것이 좋다. 이 조회에서는 먼저 이름과 관련된 MX 레코드를 찾는다. MX 레코드 대신에 CNAME 레코드가 발견되면 결과 이름이 시작 이름인 것처럼 처리된다. MX 레코드를 찾을 수 없지만 A RR이 발견되면, 우선 순위가 0인 해당 호스트를 가리키는 암시적 MX RR과 관련되어 있는 것처럼 A RR이 처리되며 해당 호스트를 가리킨다. 제공된 이름에 대해 하나 이상의 MX RR이 발견되면, SMTP 시스템은 ARR들이 MX 레코드에 의해 지정된 것이 아니라면, 해당 이름과 관련된 어떠한 A RR도 사용해서는 안 된다. 위의 “암시적 MX″ 규칙은 MX 레코드가 존재하지 않는 경우에만 적용된다. MX 레코드가 존재하지만 이러한 레코드 중 사용할 수 있는 레코드가 없다면 이 상황은 오류로 보고되어야 한다.

조회가 성공적으로 수행되면 다수의 MX 레코드 또는 멀티홈(multihoming) 아니면 두 가지 모두로 인한 매핑 결과는 단일 주소보다는 여러 개의 배달 주소 목록을 얻게 된다.

신뢰할 수 있는 메일 전송을 제공하기 위해 SMTP 클라이언트는 배달 시도가 성공할 때까지 순서대로 이 목록에 있는 각각의 관련 주소를 시도(및 재시도)할 수 있어야 한다. 그러나, 또한 시도될 수 있는 다른 주소의 개수에 대한 제한 설정을 구성할 수도 있다. 어떤 경우든 SMTP 클라이언트는 최소 두 개의 주소에 대해 시도해야 한다.

호스트 주소의 등급을 매기는 데는 다수의 MX 레코드와 멀티홈(multihomed) 호스트라는 두 가지 종류의 정보가 사용된다.

다수의 MX 레코드에는 정렬에 사용되어야 할 우선 순위를 나타내는 표시가 들어 있다.(아래 참조) 높은 우선 순위보다는 낮은 우선 순위를 사용하는 편이 더 낫다. 우선 순위가 동일한 대상이 여러 개이고 그 중 하나를 선택하는 데 있어 명백한 이유(예를 들면, 쉽게 도달되는 주소라서 선택할 수도 있음)가 없는 경우 발신자 SMTP는 특정 조직을 위해 사용되는 다수의 메일 익스체인저(exchanger) 상에 로드를 분산시키기 위해 대상들을 무작위로 선택해야 한다.

목적지 호스트(아마도 우선되는 MX 레코드에서 얻은)가 멀티홈의 형태일 수도 있는데, 이 경우 도메인 이름 결정자(resolver)는 선택 가능한 IP 주소의 목록을 반환할 것이다. 이 기능은 필요할 경우 우선 순위를 낮추어 이 목록을 정렬해야 할 도메인 이름 결정자 인터페이스로의 책임이므로 SMTP는 제공된 순서대로 시도해야 한다.

여러 개의 다른 주소를 시도할 수 있는 기능이 요구될지라도 특정 설치 제품의 경우 다른 주소의 사용을 제한하거나 사용하지 못하게 할 수도 있다. 발신자가 멀티홈 호스트의 서로 다른 주소를 사용하여 재시도해야 하는 지에 대한 문제는 지금까지도 쟁점이 되어 왔다. 다수의 주소 사용을 지지하는 측의 주된 주장은 적시적인 배달의 가능성을 최대화 시켜주고, 실제에 있어서도 가끔씩은 배달 가능성을 줄여준다. 이를 반대하는 측에서는 그로 인해 불필요한 리소스 사용이 발생할 수도 있다고 주장하고 있다. 가급적이면 5.5.4.1절에 설명된 송신 방법을 통해 리소스를 사용하는 것이 바람직하다.

SMTP 서버가 메일 익스체인저에서 지정된 목적지를 가지는 메시지를 수신하는 경우 서버는 메시지를 릴레이 하거나(아마도 MAIL FROM 및 RCPT TO 주소를 다시 작성한 후) 메시지를 최종적으로 배달하거나 아니면 SMTP에서 제공한 전송 환경 이외의 메커니즘을 사용하여 전달할 수 있다. 물론, 뒤에 두 가지 방법에서는 MX 레코드의 목록을 세부적으로 살펴보아야 할 필요가 없다.

주소를 다시 작성하지 않고 메시지를 릴레이 해야 한다고 판단한 경우에는 배달할 후보를 결정하기 위해 MX 레코드를 정렬해야 한다. 이 레코드는 먼저 우선 순위에 따라 정렬되는데, 가장 낮은 번호의 레코드가 제일 앞에 온다. 그리고 나면 릴레이 호스트가 메일 트랜잭션에서 알려지게 될 이름 또는 주소 중 하나에 대한 목록을 검사해야 한다. 일치하는 레코드가 발견되면 해당 우선 순위 수준의 모든 레코드와 높은 번호의 레코드를 버려야 한다. 이 때 남은 레코드가 없을 경우 오류 조건이 되고 메시지는 배달할 수 없으므로 반송되어야 한다. 남은 레코드가 있으면 레코드에 대해 시도해야 하며 위에서 설명한 바와 같이 최고로 선호되는 레코드에 대해 가장 먼저 시도한다.

7. 문제 발견 및 처리

7.1 신뢰할 수 있는 배달과 전자 메일 응답

수신자 SMTP가 (DATA에 대한 응답으로 “250 OK” 메시지를 전송하여)하나의 메일을 받아들인다는 것은 메시지를 배달하거나 및 릴레이 할 책임을 수락하는 것이다. 이러한 책임은 신중히 받아들여야 한다. 추후에 있을 호스트 장애 또는 예측할 수 있는 리소스 부족 등과 같은 사소한 이유로 메시지를 잃어서는 안 된다.

메시지를 수락한 후 배달 실패가 발생하면 수신자 SMTP는 절차에 따라 알림 메시지를 만들어 메일로 전송해야 한다. 이 알림 메시지는 엔벨로프에 널(“<>”) 역경로를 사용하여 전송해야 한다. 이 알림 메시지의 수신자는 엔벨로프의 반환-경로(또는 Return-Path: 줄)에 지정된 주소여야 한다. 그러나, 이 주소가 널(“<>”)일 경우 수신자 SMTP는 알림 메시지를 보내지 말아야 한다. 분명히 말하지만, 이 절의 어떤 부분도 원할 경우 지역에서 발생한 널 주소 이벤트에 대한 정보를 로그에 기록하거나 전송할 지에 대한 내부 결정(즉, 수신자 SMTP와 동일한 시스템환경의 일부)을 억제할 수 없으며 억제해서도 안 된다. 주소가 명시적인 소스 라우트인 경우 최종 홉으로 분리되어야 한다.

예를 들어, 다음의 명령을 사용하여 도착된 메시지에 대해 오류 알림 메시지를 전송해야 한다고 가정하자.

```
MAIL FROM:<@a,@b:user@d>
```

알림 메시지는 다음 명령을 사용하여 전송되어야 한다.

RCPT TO:<user@d>

SMTP에서 메시지를 수락한 이후에 발생하는 부수적인 배달 실패는 피할 수 없을 것이다. 예를 들면, 대상이 메일링 목록이라서(앞의 RCPT 설명 참조), 또는 서버가 릴레이로 작동하고 있으며 배달 시스템에 대한 액세스 권한이 없어서와 같은 이유들로 인해 발생되는 "경미한" 오류 때문에 수신 SMTP 서버가 RCPT 명령(들)에 포함된 모든 배달 주소를 확인할 수는 없을 수도 있다.

시간 제한의 결과로서 메시지가 중복하여 수신되지 않도록 하려면 수신자 SMTP는 데이터의 최종 <CRLF>.<CRLF> 끝 표시기에 응답하는 데 필요한 시간을 최소화해야 한다. 이 문제에 대한 설명은 RFC 1047 [28]을 참조하기 바란다.

7.2 루프 탐지

메일 시스템에서 가장 좋은 루프 탐지 방법은 아닐 지라도, 단순히 메시지에서 "Received:" 헤더의 수를 세는 방법은 효과적인 루프 탐지 방법으로 판명되었다. 이 기술을 사용하는 SMTP 서버는 일반적으로 100개의 Received 항목과 같은 높은 거절 임계값을 사용해야 한다. 어떠한 방법을 사용하든 서버는 사소한 루프를 찾아서 중지시키기 위한 규정을 제공해야 한다.

7.3 변칙에 대한 보완책

안타깝게도, 인터넷 메일 프로토콜에 대한 변형, 독창적인 해석 및 공공연한 위반 행위가 발생하고 있다. 혹자들은 이러한 일이 너무나 자주 발생한다고 이야기한다. 잘 작동되는 SMTP 수신자 또는 릴레이가 잘못 구성된 메시지를 거부해야 하는 지, 이러한 메시지를 변경되지 않은 상태로 전달해야 하는지, 아니면 성공적인 배달(또는 후속 답장)의 가능성을 증가시키기 위해 복구해야 하는지에 대한 논쟁은 구조화된 네트워크 메일이 등장할 무렵부터 시작되어 아직도 삭으러들 조짐을 보이지 않고 있다. 거절해야한다고 주장하는 측에서는 시도된 복구가 거의 완벽하지 않으며 잘못된 메시지에 대한 거절만이 잘못된 소프트웨어를 복구하는 유일한 방법이라고 주장한다. "어떠한 일이 있어도 복구"하거나 "배달"해야 한다고 주장하는 측에서는 사용자가 가능하다면 메일이 전달되기를 바라며 시장에서도 그렇게 하도록 상당한 압력을 가하고 있다고 주장한다. 실제로, 실제 개발자가 어떤 주장을 선호하는 지에는 상관없이, 특정 벤더에게 있어 이러한 시장 압력은 표

준에 대한 엄격한 준수보다 더 중요할 수도 있다.

잘못 구성된 메시지와 관련된 문제는 분리된 UA 메일 판독 프로토콜 [3, 26, 5, 21]의 도입으로 더욱 심화되었다. 이러한 프로토콜은 SMTP를 우편 프로토콜로 사용하고 SMTP를 이러한 클라이언트 호스트(흔히 인터넷에 일시적으로만 연결됨)의 릴레이 시스템으로 사용하도록 부추겼다. 예전에는 이러한 클라이언트 시스템 중 많은 시스템의 경우 일부 메커니즘과 SMTP(와 사실상 메일 형식 프로토콜 [7])에 의해 추측할 수 있는 정보가 부족했다. 어떤 시스템에서는 적절한 시간 추적 정보를 유지할 수 없었으며, 다른 시스템에는 시간대의 개념이 없었다. 또한 고유 이름 또는 주소를 식별하지 못하는 시스템도 있었다. 물론, RFC 822의 인증된 주소 개념을 밑받침하는 가정을 만족할 수 있는 시스템은 하나도 없었다.

이러한 열악한 SMTP 클라이언트에 대해, 이제 많은 SMTP 시스템에서는 불완전하거나 올바르지 않은 형태로 클라이언트에 전달되는 메시지를 완벽하게 만들고 있다. 이 방법은 일반적으로 서버가 클라이언트를 식별하거나 인증할 수 있으며 서버와 클라이언트 간에 사전 합의가 있을 경우에 적합하다고 간주된다. 이와는 대조적으로, 사용자 또는 클라이언트 시스템에 대한 정보가 거의 없거나 전혀 없는 릴레이 또는 배달 SMTP 서버에서 사용된 수정 작업에 대해서는 기껏해야 많은 관심만 있을 뿐이다.

처리되어질 메시지에 대한 다음의 변경 사항은 발기(originating) SMTP 서버에서 필요할 경우에 적용하거나 SMTP의 대상으로 사용되는 서버에서 초기 우편 프로토콜로 적용할 수 있다.

- 아무 것도 표시되지 않는 경우 메시지 id 필드를 추가한다.

- 아무 것도 표시되지 않는 경우 날짜, 시간 또는 시간대를 추가한다.

- 올바른 FQDN 형식으로 주소를 교정한다.

서버가 클라이언트에 대해 알고 있는 정보가 적을수록 이러한 변경은 교정되지 못할 가능성이 크며 수정 작업을 수행할 것인지 여부와 그 방법을 고려할 때는 보다 세심한 주의와 보수적인 사고를 적용해야 한다. 이러한 변경은 중간 릴레이 기능을 제공하는 SMTP 서버에서는 적용되지 말아야 한다.

모든 경우에 있어 올바른 정보를 제공하면서 올바르게 작동하는 클라이언트는 SMTP 서버에 의한 수정을 선호한다. 모든 경우에 있어 가급적이면 서버에서 수행하는 작업에 대한 설명(추적 필드 또는 헤더 설명에)을 제공하는 것이 바람직하다.

8. 보안고려사항

8.1 메일 보안 및 스푸핑(spoofing)

본래 SMTP 메일은 매우 평범한 사용자조차 수신 및 릴레이 SMTP 서버와 직접 협정하여 다른 곳에서 왔다고 생각되도록 원래의 수신자를 속이는 메시지를 만들 수 있다는 점에서 불안한 시스템이다. 전문가가 "스푸핑(spoofed)" 작동을 찾아낼 수 없도록 하는 한 메시지를 생성하는 일은 다소 어려울 수도 있지만 그러한 것을 알고 있는 누군가를 하지 않게 막기에는 역부족이다. 결과적으로, 인터넷 메일의 지식이 증가될수록, 전송 수준에서 SMTP 메일이 본래 인증할 수 없거나 또는 수행할 수 없는 무결성 검사 등에 대한 지식 또한 그만큼 증가된다. 실제 메일 보안은 디지털 서명([14]와 PGP [4] 또는 S/MIME [31] 참조)을 사용하는 것과 같이 메시지 본문에 포함되는 종단 간 방법에만 의존하고 있다.

전송 수준에서 인증(예를 들면, SMTP 클라이언트에서 SMTP 서버로)을 제공하는 다양한 프로토콜 확장이나 구성 옵션이 위에서 언급한 기존의 상황을 다소 개선한다. 그러나, 신중하게 설계된 트러스트 환경에서의 신중한 책임의 전달을 통해 수행되지 않을 경우 그러한 확장이나 구성-옵션은 전송 시스템의 무결성에 의존하지 않고 디지털로 서명된 메시지를 사용하는 종단 간 메커니즘보다 더 약한 보안력을 지니게 된다.

사용자가 자신의 주소가 아닌 다른 유효한 주소를 가리키도록 하는 엔벨로프 반환 경로 및 헤더 "From:" 필드를 설정하기 더 어렵게 만드는 노력이 크게 잘못 인식되고 있다. 이러한 노력은 다른 사용자를 대신하여 하나의 사용자가 메일을 전송하거나 오류(또는 정상) 답장이 특수한 주소로 지정되어야 하는 합법적인 사용을 방해한다. (사용자가 메시지별로 이러한 필드를 쉽게 변경할 수 있도록 하는 방법을 제공하는 시스템은 메시지 데이터 안에서 Sender 필드가 지각 있게 생성될 수 있도록 사용자에게 기본 적이고도 영구적인 우편함 주소를 설정하려 시도해야 한다.)

이 규격은 메일을 위조하려 하는 무례한 사용자에 대한 약간의 보호 영역을 제공하고

자 하는 바람 때문에 유용한 기능을 사용할 수 없게 해서는 안 된다고 주장할 뿐 SMTP와 관련된 인증 문제에 관해 더 이상의 설명을 제공하지 않는다.

8.2 "숨은(Blind)" 참조

메시지 헤더에 나타나지 않은 주소가 몇 가지 이유에서 SMTP 서버로의 RCPT 명령에 나타날 수 있다. 가장 일반적인 두 가지 경우는 메일링 주소를 "목록 익스플로더(exploder)"로 사용하는 경우와 (하나의 주소가 여러 개의 주소로 결정된다.) "숨은 참조"를 표시하는 경우이다. 특별히 두 개 이상의 RCPT 명령이 존재하는 경우 이러한 메커니즘의 목적 중 일부에 위배되지 않도록 하기 위해 SMTP 클라이언트와 서버는 헤더에 전체 RCPT 명령 인수 집합을 추적 헤더의 일부나 정보 또는 개인용 확장 헤더로서 복사해서는 안 된다. 이 규칙이 실제로는 지켜지지 않으며 강요할 수도 없으므로, "숨은 참조"를 인식하는 송신 SMTP 시스템의 경우 단일 RCPT 명령만을 포함하는 개별 메시지 트랜잭션으로서 각 숨은 참조 사본을 전송하는 것이 유용할 수 있다.

SMTP 트랜잭션 ("엔벨로프")에서의 "역방향"(MAIL, SAML 등의 명령에서) 또는 "순방향"(RCPT) 주소와 헤더의 주소 사이에는 고유 관계가 없다. 수신 시스템은 그러한 관계를 추론하거나 그러한 관계를 사용하여 배달할 메시지의 헤더를 변경하려 해서는 안 된다. 널리 알려져 있는 "Apparently-to" 헤더는 이러한 원리에 위배될 뿐만 아니라 의도하지 않은 정보 노출의 공통 소스이기도 하므로 사용해서는 안된다.

8.3 VRFY, EXPN 및 보안

4.5절에서 설명한 바와 같이, 개인 사이트에서는 보안상의 이유로 VRFY 또는 EXPN, 아니면 두 개 명령 모두를 사용하지 못하게 할 수도 있다. 위에 대한 당연한 결론으로서, 이를 허용하는 구현 제품은 사실상 확인되지 않은 주소를 확인한 것처럼 보이게 해서는 안 된다. 사이트에서 보안상의 이유로 이러한 명령을 사용할 수 없게 하는 경우 SMTP 서버는 성공적인 검증인지 실패한 검증인지 혼동될 수 있는 코드보다는 252 응답을 반환해야 한다.

구문에 대해서만 주소를 확인한 후 VRFY 명령에 열거된 주소를 포함하여 250 답장 코드를 반환하는 것은 이 규칙에 위배된다. 물론, 주소가 유효한지 여부에 상관없이 항상 550을 반환하여 VRFY를 "지원"하는 구현 제품도 마찬가지로 규칙과 일치하지는 않는다.

최근 몇 년 동안, 메일링 목록의 내용은 “스패머(spammer)"를 위한 주소 정보의 원천으로 인기를 끌어왔다. 목록 관리자가 목록 자체의 부적절한 사용에 대한 보호를 설치했으므로 EXPN을 사용하여 주소를 ”수집“하는 행위가 늘고 있다. 구현 제품에서는 여전히 EXPN을 지원해야 하지만 사이트는 이 명령을 지원할 경우와 지원하지 않을 경우의 장/단점을 신중히 평가해야 한다. 인증 메커니즘이 SMTP에 도입됨에 따라, 일부 사이트의 경우 인증된 요청자에 대해서만 EXPN 명령을 사용하도록 선택하고 있다.

8.4 발표 시 정보 노출

인사 응답이나 HELP 명령에 대한 응답으로 서버의 유형과 버전(과 때로는 서버 도메인 이름조차)정보를 알려줌으로서 디버깅을 할 수 있다는 장점과 잠재적인 적대적 공격 시에 유용한 정보가 노출된다는 단점 사이에 절충에 대한 논쟁이 계속되어 왔다. 디버깅 정보의 유용성은 의심할 필요가 없다. 서버의 정확한 ID를 숨김으로서 알려진 취약점을 숨기려는 희망보다는 SMTP 서버를 실제적으로 보호하는 것이 훨씬 낳다라는 점을 지적하면서, 그러한 기능을 만드는 것이 유용하다라고 주장하는 사람들은 더 많은 보호 방법을 제공해야 할 것이다. 사이트는 이 문제와 관련하여 장/단점을 신중히 평가해야 할 것이다. 구현 제품은 가급적이면 어떤 방법으로든 다른 네트워크 호스트에서 사용할 수 있도록 하기 위한 유형 및 버전 정보를 최소한으로 제공하는 것이 좋다.

8.5 추적 필드에서 정보 노출

공용 인터넷 상에 직접 존재하지 않는 LAN에 있는 호스트에서 메일이 만들어지는 경우와 같이 어떤 경우에는 이 규격에 따라 만들어진 추적(“Received") 필드가 일반적으로는 사용할 수 없는 호스트 이름 및 유사한 정보를 노출시킬 수도 있다. 대개의 경우 이는 문제가 되지 않지만 이름 노출을 특별히 중요시하는 사이트에서는 이 사실을 알고 있어야 한다. 또한, 다수의 수신자가 포함된 경우 다른 사람에게 ”숨은 참조“ 수신자의 ID가 우연히 노출되지 않도록 FOR 절(선택 사항)을 절대 사용하지 않거나 주의하여 사용해야 한다.

8.6 메시지 전달에서 정보 노출

4.4절에서 설명했듯이, 우편함과 관련된 대체 주소를 나타내기 위해 251이나 551 응답 코드를 사용할 경우 중요한 정보가 우연하게 노출될 수도 있다. 이러한 문제를 중요시하는 사이트는 적절한 방법을 선택했는 지와 선택된 방법을 위한 구성이 적절한지를 확인해야 한다.

8.7 SMTP 서버 작동의 범위

이 내용은 신중하게 제정된 원칙으로서 SMTP 서버가 서버를 제공하는 사이트가 민감해하는 작동상의 또는 기술적인 이유로 메일 승인을 거부할 수도 있음을 명시하고 있다. 그러나, 사이트와 설치 제품간에 협력이 있어야 인터넷을 원활하게 해 준다. 사이트에서 트래픽을 거부할 수 있는 권한의 이점을 남용하는 경우에는 전자 메일 가용성(인터넷의 강점 중 하나)을 통한 정보 편재(ubiquity) 특성이 위협을 받게 될 것이다. 사이트에서 받아서 처리할 트래픽을 선택할 수 있는 경우 각별한 주의를 기울여야 하며 균형을 유지해야 한다.

최근 몇 년 동안, 임의의 사이트를 통한 릴레이 기능의 사용이 메일의 실제 출처를 숨기기 위한 적대적 행위의 일부로서 사용되어 왔다. 일부 사이트의 경우 알려진 소스 또는 식별 가능한 소스로의 릴레이 기능으로만 제한하기로 결정하였으므로, 구현 제품은 이러한 유형의 필터링 수행기능을 제공해야 한다. 메일이 이러한 이유 또는 다른 정책상의 이유로 거부되는 경우에 해당되면 EHLO, MAIL 또는 RCPT에 대한 응답으로 550 코드를 사용해야 한다.

9. IANA 고려사항

IANA는 이 규격을 지원하면서 세 개의 레지스트리를 유지할 것이다. 첫 번째 레지스트리는 SMTP 서비스 확장, 확장과 관련된 키워드, 필요할 경우 매개변수와 및 동사로 구성된다. 3.2.2절에서 명시한 바와 같이 이 레지스트리에는 “X″로 시작되는 항목이 없다. 이러한 항목은 이 목적으로 IESG에서 특별히 승인한 표준(Standard Track) 또는 실험(Experimental) RFC에 정의되어 있는 서비스 확장(과 관련 키워드, 매개변수 또는 동사)에 대해서만 만들어질 수 있다.

두 번째 레지스트리는 IPv4 주소(RFC 821과 이 문서에 명시됨)와 IPv6 주소(이 문서에 명시됨)의 태그가 아닌 도메인 리터럴 형태를 나타내는 “태그”로 구성된다. 추가 리터럴 유형은 사용되기 전에 표준화되어야 한다. 현재는 예상되는 추가 리터럴이 없다.

RFC 821에서 설정되고 이 규격에서 새로 작성한 세 번째 레지스트리는 4.4절에서 설명한 시간 스탬프(“Received:” 헤더)의 ”via” 및 “with” 하위 절과 함께 사용될 링크와 프로토콜 식별자(ID)의 레지스트리이다. 이 문서에 지정된 식별자에 추가되는 링크와 프로토콜 식별자는 표준화 또는 RFC 문서에 명시되고 IESG에서 승인한 실험적(Experimental) 프로토콜 확장을 통해서만 등록될 수 있다.

부록 I. TCP 전송 서비스

TCP 연결은 8비트 바이트의 전송을 지원한다. SMTP 데이터는 7비트 ASCII 문자이다. 각 문자는 상위 비트가 0으로 소거된 8비트 바이트로 전송된다. 서비스 확장은 전체 8비트 데이터 비트를 SMTP 명령이나 응답에서가 아닌 메시지 본문의 일부로서 전송할 수 있도록 이 규칙을 변경할 수 있다.

부록 II. RFC822 헤더에서 SMTP 명령 생성

일부 시스템의 경우 메일 의뢰 프로토콜에서(만) RFC 822 헤더를 사용하거나, 그렇지 않으면 메시지가 UA에서 MTA로 전달되었을 때 RFC 822 헤더로부터 SMTP 명령을 생성한다. MTA-UA 프로토콜은 인터넷 표준에서는 다루지 않는 개인적인 요소이지만, 이 접근방법에는 문제가 있다. 예를 들면, 개념적으로 메일 엔벨로프에 속하는 정보가 헤더 정보로부터 처리 시, 조기에 분리되어 별도로 유지되지 않는 경우 “숨은 참조” 사본 및 재배포 목록을 처리할 때 문제가 반복된다.

가급적이면 UA가 메시지 자체와는 별도로 초기(“의뢰 클라이언트”) MTA에 엔벨로프를 제공하는 것이 좋다. 그러나, 엔벨로프가 제공되지 않는 경우 SMTP 명령은 다음과 같이 생성되어야 한다.

1. TO, CC 또는 BCC 헤더 필드의 각 수신자 주소가 (대기열에 저장하거나 배달하는데 필요할 경우 다수의 메시지 사본을 생성하는) RCPT 명령으로 복사되어야 한다. 여기에는 RFC 822 ″그룹“에 열거된 모든 주소가 포함된다. 그리고 나면 모든 BCC 필드가 헤더에서 제거되어야 한다. 이 프로세스가 완료되고 나면 최소한 하나의 To:. Cc: 또는 Bcc: 헤더가 유지되는 지를 확인하기 위해 남은 헤더를 검사해야 한다. 유지되는 헤더가 없을 경우 추가 정보가 없는 Bcc: 헤더가 [32]에 명시된 바와 같이 삽입되어야 한다.

2. MAIL 명령에 사용되는 반환 주소는 가능한 경우 의뢰하는(로컬) 사용자의 시스템 ID에서 파생되거나 그렇지 않을 경우 “From:″ 헤더 필드에서 파생되어야 한다. 또한 사용할 수 있는 시스템 ID가 있고 그 ID가 From 헤더 필드의 주소와 다르면 ID가 Sender 헤더 필드로 복사되어야 한다. (이미 존재하는 모든 Sender 필드를 제거해야 한다.) 시스템은 의뢰자가 엔벨로프 반환 주소를 변경하는 방법을 제공할 수 있지만, 그 사용을 권한이 부여된 사용자로만 제한하려 할 것이다. 이렇게 할 경우 메일 위조를 막지는 못하겠지

만 메일 사고는 줄일 수 있다. 8.1절 참조.

MTA가 이러한 방식으로 사용되는 경우 MTA는 전송되는 메시지가 유효하다는 것을 보장할 책임이 있다. 이 유효성을 확인하고 도착 시 유효하지 않은 메시지를 처리(또는 반송)하는 방법은 MUA-MTA 인터페이스의 일부이므로 이 규격에서는 다루지 않는다.

표준 RFC 822 정보에 기초한 의뢰 프로토콜은 단독으로 외부(비SMTP) 메일 시스템으로부터 SMTP 환경으로 메시지를 연결하는 데 사용되어서는 안 된다. 엔벨로프 생성에 대한 추가 정보는 보충 헤더이든 외부 시스템의 엔벨로프이든 다른 환경의 소스에서 얻어야 한다.

헤더 “To:”와 ”Cc:” 필드만을 사용하여 메시지를 연결하려는 시도는 반복적으로 메일 루프와 인터넷 메일 환경의 올바른 기능을 방해하는 기타 작동을 일으켜 왔다. 이러한 문제는 특별히 메시지가 인터넷 메일링 목록에서 비롯되어 엔벨로프 정보를 사용하여 외부 환경으로 배포되는 경우에 일반적으로 발생되었다. 이러한 메시지가 헤더-전용 리메일러(remailer)에 의해 처리되는 경우 인터넷 환경(과 메일링 목록)으로 되돌려 지는 것은 거의 불가피하다.

부록 III. 소스 라우트

예전에는 <역경로>가 호스트와 소스 우편함을 포함하는 소스 라우팅 목록의 역구성이었다. <역경로>에서 첫 번째 호스트는 MAIL 명령을 전송하는 호스트여야 한다. 마찬가지로, <순경로>는 호스트와 목적지 우편함의 소스 라우팅 목록일 수 있다. 그러나, 일반적으로 라우팅 정보를 제공하는 도메인 이름 시스템에 의지하여 <순경로>는 우편함과 도메인 이름만을 포함한다. 소스 라우트의 사용은 금지된다. 하지만, 4.3절과 부록 VI.2절에서 설명한 바와 같이 서버는 소스 라우트를 수신하고 처리할 수 있어야 하는 반면, 클라이언트는 소스 라우트를 전송해서는 안되며 이 절은 내용을 제공하기 위한 목적으로만 수록되었다.

릴레이에 사용하려는 경우 순경로는 ″@ONE,@TWO:JOE@THREE″ 형태의 소스 라우트가 될 수 있다. 이 때, ONE, TWO 및 THREE는 정식 도메인 이름이어야 한다. 이 형태는 주소와 라우트의 차이를 강조할 때 사용된다. 우편함은 절대 주소이며, 라우트는 그 주소에 도달하는 방법에 대한 정보이다. 이 두 가지 개념을 혼동해서는 안 된다.

소스 라우트가 사용되는 경우 순경로와 역경로를 생성 및 업데이트하기 위한 메커니즘을 이해하려면 RFC 821과 아래 내용을 참조해야 한다.

SMTP 서버는 자신의 식별자(도메인 이름이나 메일 익스체인저(exchanger)로 동작하는 대상 도메인의 이름)가 나타나는 경우 식별자를 순경로에서 역경로의 시작 부분으로 옮겨서 명령 인수를 변환한다.

순경로와 역경로는 SMTP 명령과 응답에 나타나지만 메시지에 반드시 나타날 필요는 없다. 즉, 메시지 헤더의 “To:", "From:", "CC:" 등의 필드에 표시하기 위해 이러한 경로와 특히 이 구문은 필요하지 않다. 반대로, SMTP 서버는 메시지 헤더 필드에서 최종 메시지 배달 정보를 추출해서는 안 된다.

호스트 목록이 있을 경우 그 목록은 "역" 소스 라우트로서 메일이 목록 상의 각 호스트(목록에서 첫 번째 호스트가 가장 최근의 릴레이임)를 통해 릴레이 되었음을 나타낸다. 이 목록은 발신자에게 배달되지 않았음을 알리는 메시지를 보낼 때 소스 라우트로 사용된다. 각 릴레이 호스트가 자신의 이름을 목록의 시작 부분에 추가할 때 메일이 수신된 전송 환경에 알려져 있는 이름이 아닌 메일을 릴레이 하는 전송 환경에 알려져 있는 이름을 사용해야 한다(두 개 이름이 서로 다를 경우).

부록 IV. 시나리오

이 절에서는 몇 가지 유형의 SMTP 세션을 보여주는 전체 시나리오를 제공한다. 이 예에서 “C:"는 SMTP 클라이언트 측에서 전송하는 내용을 나타내며, ”S:"는 SMTP 서버 측에서 전송하는 내용을 나타낸다.

IV.1 일반적인 SMTP 트랜잭션 시나리오

이 SMTP 예는 bar.com 호스트의 Smith가 호스트 foo.com의 Jones, Green 및 Brown에게 전송하는 메일을 보여준다. 여기서는 호스트 bar.com이 호스트 foo.com에 직접 연결한다고 가정한다. Jones와 Brown의 경우 이 메일이 받아드려졌지만 Green의 경우에는 호스트 foo.com에 우편함이 없다.

S: 220 foo.com 단순 메일 전송 서비스 준비
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 여기에 그런 사용자 없음

C: RCPT TO:<Brown@foo.com>

S: 250 OK

C: DATA

S: 354 메일 입력 시작. <CRLF>.<CRLF>로 끝남.

C: Blah blah blah...

C: ...etc. etc. etc.

C: .

S: 250 OK

C: QUIT

S: 221 foo.com 서비스에서 전송 채널을 닫음

IV.2 중지된 SMTP 트랜잭션 시나리오

S: 220 foo.com 단순 메일 전송 서비스 준비

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 여기에 그런 사용자 없음

C: RSET

S: 250 OK

C: QUIT

S: 221 foo.com 서비스에서 전송 채널을 닫음

IV.3 릴레이 된 메일 시나리오

1단계 -- 소스 호스트에서 릴레이 호스트로

S: 220 foo.com 단순 메일 전송 서비스 준비

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:<@foo.com:Jones@XYZ.COM>

S: 250 OK

C: DATA

S: 354 메일 입력 시작. <CRLF>.<CRLF>로 끝남.

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 서비스에서 전송 채널을 닫음

2단계 -- 릴레이 호스트에서 대상 호스트로

S: 220 xyz.com 단순 메일 전송 서비스 준비

C: EHLO foo.com

S: 250 xyz.com is on the air

C: MAIL FROM:<@foo.com:JQP@bar.com>

S: 250 OK

C: RCPT TO:<Jones@XYZ.COM>

S: 250 OK

C: DATA

S: 354 메일 입력 시작. <CRLF>.<CRLF>로 끝남.

C: Received: from bar.com by foo.com ; Thu, 21 May 1998

C: 05:33:29 -0700

C: Date: Thu, 21 May 1998 05:33:22 -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 서비스에서 전송 채널을 닫음.

IV.4 확인 및 송신 시나리오

S: 220 foo.com 단순 메일 전송 서비스 준비
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: SEND FROM:<EAK@bar.com>
S: 250 OK
C: RCPT TO:<Admin.MRC@foo.com>
S: 250 OK
C: DATA
S: 354 메일 입력 시작. <CRLF>.<CRLF>로 끝남.
C: Blah blah blah...
C: ...etc. etc. etc.
C: .
S: 250 OK
C: QUIT
S: 221 foo.com 서비스에서 전송 채널을 닫음.

부록 V. 다른 게이트웨이 문제

일반적으로, 인터넷과 다른 메일 시스템 사이에 놓여있는 게이트웨이는 포함된 두 개의 메일 시스템 간 경계 상에 놓여 있는 어떤 계층적 의미론을 보존해야 한다. 맵핑을 통해 지름길을 채택하고자 하는 게이트웨이 변환 방법(하나의 시스템에서의 엔벨로프 정보로부터 다른 메시지의 헤더나 본문으로의 매핑 같은 일반적으로 중요한 면에서 부적절한 것으로 판명되었다. 엔벨로프와 헤더 모두를 지원하지 않는 환경과 인터넷 메일을 변환하는 시스템은 정보 손실이 거의 불가피하다는 사실을 이해한 채로 수행되어야 한다.

부록 VI. RFC821에서 사용이 금지된 기능

RFC 821의 몇 가지 기능은 문제가 있는 것으로 판명되었으므로 인터넷 메일에서 사용해서는 안 된다.

VI.1 TURN

RFC 821에 설명된 이 명령은 클라이언트와 서버의 역할 전환을 요청하는 호스트의 강력한 인증이 없을 경우 메일을 올바른 대상에서 다른 곳으로 전환하는 데 쉽게 사용될 수 있으므로 중요한 보안 문제를 일으킨다. 따라서 이 명령의 사용은 금지된다. SMTP 시스템은 서버가 클라이언트를 인증할 수 있는 경우가 아니라면 이 기능을 사용하면 안 된다.

VI.2 소스 라우팅

RFC 821에서는 일련의 릴레이를 통해 하나의 호스트에서 다른 호스트로 메일을 전달하기 위해 명시적인 소스 라우팅의 개념을 이용했다. 보통의 메일 트래픽에서 소스 라우트를 사용해야 하는 필요성은 도메인 이름 시스템 “MX” 레코드의 도입으로 사라졌으며, 소스 라우트 사용에 대한 의미있는 변명도 RFC 11234에서 ”@“ 뒤에 오는 주소는 모두 정식 도메인 이름이어야 한다는 명확한 요구사항의 도입으로 사라졌다. 따라서 소스 라우트를 사용하기 위한 유일한 이유를 말한다면 오래된 SMTP 클라이언트 또는 MUA에 대한 지원과 메일 시스템 디버깅을 위해서일 것이다. 그러나, 소스 라우트는 여전히 후자의 경우에 유용하게 사용되며, 관련 DNS 레코드 문제와 같은 심각하지만 일시적인 문제 발생 시에 메일을 라우팅 할 때도 유용할 수 있다.

SMTP 서버는 이 문서의 본문과 RFC 1123에서 명시한 바와 같이 소스 라우트 구문을 계속 허용해야 한다. 필요할 경우 SMTP 서버는 소스 라우트를 무시하고 주소에서 대상 도메인만을 사용할 수도 있다. 소스 라우트를 사용하는 경우 메시지는 주소에 표시된 첫 번째 도메인으로 전송되어야 한다. 특히, 서버는 소스 라우트 안에서 지름길(shortcut)을 추측해서는 안 된다.

클라이언트는 방화벽 주위에서 디버깅하거나 잠재적으로 릴레이하는 경우 또는 메일 시스템 구성 오류 등의 특수한 상황을 제외하고는 명시적인 소스 라우팅을 사용해서는 안 된다.

VI.3 HELO

4.1절과 5.1.1절에 설명된 바와 같이, 서버가 EHLO를 허용할 경우 HELO보다는 EHLO를 사용하는 것이 바람직하다. 서버는 이전의 클라이언트를 지원하기 위해 계속해서 HELO를 받아들이고 처리해야 한다.

VI.4 # 리터럴

RFC 821에서는 “#” 접두사가 붙은 십진 정수 호스트 값을 인터넷 주소를 나타내기 위해 제공했다. 실제로는 TCP/IP를 도입한 이후부터 이 형태는 더 이상 사용되지 않고 있다. 따라서 이 형태는 사용하지 말아야 한다.

VI.5 날짜와 연도

SMTP 클라이언트 또는 서버가 메시지에 날짜를 삽입하는 경우(예를 들면, 추적 필드에) 네 자리 연도를 사용해야 한다. 두 자리 연도의 사용은 금지된다. 세 자리 연도는 인터넷 메일 시스템에서 절대로 사용할 수 없다.

VI.6 송신과 메일링

메시지를 사용자의 우편함에 전달하기 위한 메커니즘을 지정하는 것 외에도, RFC 821은 사용자의 터미널 화면에 메시지를 직접 전달하기 위한 추가 명령(선택 사항)을 제공했다. 이러한 명령(SEND, SAML, SOML)은 거의 구현되지 않으며, 구현되는 경우라도 워크스테이션 기술의 변경과 다른 프로토콜의 도입으로 인해 구현된 환경에서도 사용되지 않고 있다.

클라이언트는 SEND, SAML 또는 SOML을 서비스로 제공해서는 안 된다. 서버는 이러한 명령을 구현할 수 있다. 서버가 이러한 명령을 구현하는 경우 RFC 821에 명시된 구현 모델이 사용되어야 하며 EHLO 명령에 대한 응답으로 명령 이름이 공포되어야 한다.