

组织: 中国互动出版网 (<http://www.china-pub.com/>)

RFC 文档中文翻译计划 (<http://www.china-pub.com/compters/emook/aboutemook.htm>)

E-mail: ouyang@china-pub.com

译者: 顾国飞 (ggfei ggfei@263.net)

译文发布时间: 2001-3-30

版权: 本中文翻译文档版权归中国互动出版网所有。可以用于非商业用途自由转载, 但必须保留本文档的翻译及版权信息。

Network Working Group
Request for Comments: DRAFT
Replaces: RFC 788, 780, 772

J. Postel
ISI
August 1982

RFC821 简单邮件传输协议 (SMTP)

(RFC821 SIMPLE MAIL TRANSFER PROTOCOL)

目录

1. 介绍	2
2. SMTP 模型	2
3. SMTP 过程	3
3.1. MAIL	4
3.2. 转发	5
3.3. 确认和扩展	6
3.4. 发送信件(mailing)和获得信件(sending)	7
3.5. 打开和关闭	7
3.6. 转发	8
3.7. 域	9
3.8. 改变角色	9
4. SMTP 说明	9
4.1. SMTP 命令	9
RFC 文档中文翻译计划	1

4.1.1. 命令语法.....	9
4.1.2. COMMAND 语法格式.....	13
4.2. SMTP 响应.....	15
4.3. 命令和应答序列.....	16
4.4. 状态图.....	17
4.5. 详细内容.....	18
4.5.1. 最小实现.....	18
4.5.2. 透明性.....	18
4.5.3. 大小.....	19
附录 A TCP 传输服务.....	19
附录 B NCP 传输服务.....	19
附录 C NITS.....	20
附录 D X.25 传输服务.....	20
附录 E 应答码构成方法.....	20
附录 F 一些例子.....	22
参考资料.....	36

1. 介绍

简单邮件传输协议 (SMTP) 的目标是可靠高效地传送邮件, 它独立于传送子系统而且仅要求一条可以保证传送数据单元顺序的通道。附录 A, B, C 和 D 描述了不同传送服务下 SMTP 的使用。在名词表中还定义了本文档中使用的术语。

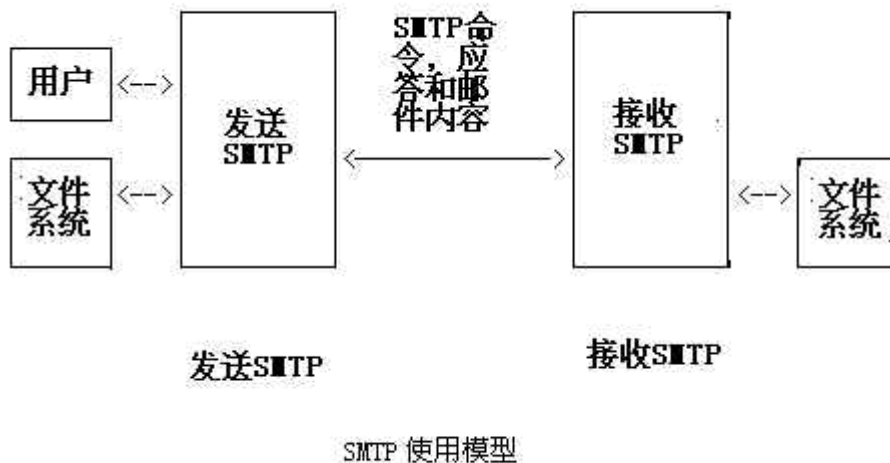
SMTP 的一个重要特点是它能够在传送中接力传送邮件, 传送服务提供了进程间通信环境 (IPCE), 此环境可以包括一个网络, 几个网络或一个网络的子网。理解到传送系统 (或 IPCE) 不是一对一的是很重要的。进程可能直接和其它进程通过已知的 IPCE 通信。邮件是一个应用程序或进程间通信。邮件可以通过连接在不同 IPCE 上的进程跨网络进行邮件传送。更特别的是, 邮件可以通过不同网络上的主机接力式传送。

2. SMTP 模型

SMTP 设计基于以下通信模型: 针对用户的邮件请求, 发送 SMTP 建立与接收 SMTP 之间建立一个双向传送通道。接收 SMTP 可以是最终接收者也可以是中间传送者。SMTP 命令由发

送 SMTP 发出，由接收 SMTP 接收，而应答则反方面传送。

一旦传送通道建立，SMTP 发送者发送 MAIL 命令指明邮件发送者。如果 SMTP 接收者可以接收邮件则返回 OK 应答。SMTP 发送者再发出 RCPT 命令确认邮件是否接收到。如果 SMTP 接收者接收，则返回 OK 应答；如果不能接收到，则发出拒绝接收应答（但不中止整个邮件操作），双方将如此重复多次。当接收者收到全部邮件后会接收到特别的序列，如果接收者成功处理了邮件，则返回 OK 应答。



SMTP 提供传送邮件的机制，如果接收方与发送方连接在同一个传送服务下时，邮件可以直接由发送方主机传送到接收方主机；或者，当两者不在同一个传送服务下时，通过中继 SMTP 服务器传送。为了能够对 SMTP 服务器提供中继能力，它必须拥有最终目的主机地址和邮箱名称。

MAIL 命令参数是回复路径，它指定邮件从何处来；而 RCPT 命令的参数是转发路径的，它指定邮件向何处去。向前路径是源路径，而回复路径是返回路径（它用于发生错误时返回邮件）。

当同一个消息要发往不同的接收者时，SMTP 遇到了向不同接收者发送同一份数据的复制品的问题，邮件命令和应答有一个比较奇怪的语法，应答也有一个数字代码。在下面，例子中可以看到哪些使用实际的命令和应答。完整的命令和应答在第四节。

命令与应答对大小写不敏感，也就是说，命令和应答可以是大写，小写或两者的混合，但这一点对用户邮件名称却不一定是对的，因为有的主机对用户名大小写是敏感的。这样 SMTP 实现中就将用户邮箱名称保留成初始时的样子，主机名称对大小写不敏感。

命令与应答由 ASCII 字母表组成，当传送服务提供 8 位字节传送通道，每 7 位字符正确传送，而最高位被填充为 0。当指定一般的命令或应答格式后，参数会由一些类似于语言的字符串表示出来，如“<string>”或“<reverse-path>”，这里尖括号表示这是一种类似于语言的变量。

3. SMTP 过程

本节提供了 SMTP 中的一些过程。头一个说明的是基本发送过程（定义为发送操作）。下来描述向前传送邮件，确认邮箱名称和扩展邮件列表，发送到终端和打开关闭交换。在本节

的最后是对中断，邮件域的说明。本节的例子只是一部分命令和应答的序列，完整的例子见附录 F。

3.1. MAIL

在 SMTP 发送操作中有三步，操作由 MAIL 命令开始给出发送者标识。一系列或更多的 RCPT 命令紧跟其后，给出了接收者信息，然后是 DATA 命令列出发送的邮件内容，最后邮件内容指示符确认操作。

过程中的第一步是 MAIL 命令，`< reverse-path >`包括源邮箱。

```
MAIL <SP> FROM:<reverse-path> <CRLF>
```

此命令告诉接收者新的发送操作已经开始，请复位所有状态表和缓冲区。它给出反向路径以进行错误信息返回。如果请求被接收，接收方返回一个 250 OK 应答。`<reverse-path>`中不止包括了邮箱，它包括了主机和源邮箱的反向路由，其中的第一个主机就是发送此命令的主机。

过程中的第二步是发送 RCPT 命令。

```
RCPT <SP> TO:<forward-path> <CRLF>
```

此命令给出向前路径标识接收者，如果命令被接收，接收方返回一个 250 OK 应答，并存储向前路径。如果接收者未知，接收方会返回一个 550 Failure 应答。此过程可能会重复若干次。

`<forward-path>`不仅包括邮件，它是主机和目的邮箱的路由表，在其中的第一个主机就是接收命令的主机。过程中的第三步是发送 DATA 命令。

```
DATA <CRLF>
```

如果命令被接收，接收方返回一个 354 Intermediate 应答，并认定以下的各行都是信件内容。当信件结尾收到并存储后，接收者发送一个 250 OK 应答。因为邮件是在传送通道上发送，因此必须指明邮件内容结尾，以便应答对话可以重新开始。SMTP 通过在最后一行仅发送一个句号来表示邮件内容的结束，在接收方，一个对用户透明的过程将此符号过滤掉，以不影响正常的的数据。

注意：邮件内容包括如下提示：Date, Subject, To, Cc, From。

邮件内容指示符确认邮件操作并告知接收者可以存储和再发送数据了。如果此命令被接收，接收方返回一个 250 OK 应答。DATA 命令仅在邮件操作未完成或源无效的情况下失败。

上面所述的过程是一个发送操作。这些命令只能以上面的顺序使用。下例表示了在一个发送操作中这些命令的使用。

SMTP 过程例子 此例是在 Alpha. ARPA 主机的 Smith 发送邮件给 Beta. ARPA 主机的 Jones, Green 和 Brown 的，这里假定主机 Alpha 与主机 Beta 直接相连。

```
S: MAIL FROM:<Smith@Alpha.ARPA>
R: 250 OK
S: RCPT TO:<Jones@Beta.ARPA>
R: 250 OK
S: RCPT TO:<Green@Beta.ARPA>
R: 550 No such user here
S: RCPT TO:<Brown@Beta.ARPA>
R: 250 OK
S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ... 等等
S: <CRLF>.<CRLF>
R: 250 OK
```

此信被前两个人接收，而第三个人在此主机上没有邮箱。

3.2. 转发

下面是一些<forward-path>中目的地址不正确的，但接收者知道正确的目的地址的例子。在这些例子中，下列应答之一应该允许发送方与获得正确地址。

251: 用户不在本地；将向前发送到<forward-path>。

这个应答意味着，接收方 SMTP 知道用户的邮箱在另外的主机上，而且意味着将在未来使用正确的转向路径。请注意，主机或者用户，或者它们两者是不同的。接收方负责传送消息。

551 : 用户非本地，请尝试<forward-path>

这个应答意味着接收 SMTP 知道用户的邮箱在另外的主机上，并意味着使用了正确的转发路径。注意请注意，主机或者用户，或者它们两者是不同的。接收方拒绝接收此用户的信件，发送者必须根据提供的信息重新发送或者向原发送者返回错误信息。下例显示了这些响应的应用。

转发的例子

```
S: RCPT TO:<Postel@USC-ISI.ARPA>
R: 251 User not local; will forward to <Postel@USC-ISIF.ARPA>
或者
S: RCPT TO:<Paul@USC-ISIB.ARPA>
R: 551 User not local; please try <Mockapetris@USC-ISIF.ARPA>
```

3.3. 确认和扩展

SMTP 提供了另外的确认用户名和扩展邮件列表的功能。这些功能由 VREF 和 EXPN 命令完成，它们都以字符串为参数。对于 VREF 命令，字符串参数指的是用户名，对此命令的响应要包括用户的命名和用户的邮箱。对于 EXPN 命令，字符串参数指的是邮件列表，对此命令的响应多于一个，它们要包括所有列表中用户的命名和他们的邮箱。

“用户名”是一个多余的项目，它是故意被加上的。如果主机采用 VREF 命令和 EXPN 命令，最后本地邮箱必须提供用户名使它被主机确认。如果主机选择由另外的字符串作为用户名，也是允许的。

在一些主机中，邮箱列表和一个邮箱的代名有一点不清楚，因为一般的数据结构可能包括两种类型的入口。如果要发出对邮件列表的确认，应该给出确定响应。在接收到这个消息后，主机将把邮件传送到列表上所有的地址上去，如果没有接收到确定响应，就会报告错误。例如，“550 That is a mail list, not a user name”。如果请求用于扩展一个用户名，可能通过返回包括一个名字的列表来形成确定响应，如果没有接收到确定响应，就会报告错误。（例如，“550 That is a user name, not a mailing list”）。

在多个响应的情况下（通常是对于 EXPN 而言的），每个应答指定一个邮箱。在模糊请求的情况下，例如“VRFY Smith”，这里两个 Smith 的响应必须是“553 User ambiguous”。

确认用户名的情况如下例所示：例 3：

确认用户名

S: VRFY Smith R: 250 Fred Smith <Smith@USC-ISIF.ARPA>

或者

S: VRFY Smith

R: 251 User not local; will forward to <Smith@USC-ISIQ.ARPA>

或者

S: VRFY Jones

R: 550 String does not match anything.

或者

S: VRFY Jones

R: 551 User not local; please try <Jones@USC-ISIQ.ARPA>

或者

S: VRFY Gourzenkyinplatz

R: 553 User ambiguous.

邮箱列表要求多个响应的情况如下例所示：

S: EXPN Example-People

R: 250-Jon Postel <Postel@USC-ISIF.ARPA>

R: 250-Fred Fonebone <Fonebone@USC-ISIQ.ARPA>

R: 250-Sam Q. Smith <SQSmith@USC-ISIQ.ARPA>

R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>

R: 250-<joe@foo-unix.ARPA>

R: 250 <xyz@bar-unix.ARPA>

或者

S: EXPN Executive-Washroom-List

R: 550 Access Denied to You.

VERF 和 EXPN 命令的字符串命令参数因为具体实现的不同而不能再加以限制了。在一些系统上, EXPN 命令的参数可能是一个包含邮件列表的文件名, 但是在 Internet 上有许多不同的文件结构。

VERFY 和 EXPN 命令在最小实现中并不包括, 当它们实现时, 它们也不要求被在传送间实现。

3.4. 发送信件(mailing)和获得信件(sending)

SMTP 的主要目的是将邮件发送到用户的邮箱中。由一些主机提供的类似的功能是把邮件送至用户的终端 (如果用户正打开终端)。将邮件送到用户的邮箱中称为发送信件 (mailing); 而送至用户终端则称之为获得信件 (sending)。因为在一些主机上, 这两者的实现十分类似, 所以它们同时被放入了 SMTP 中。然而, 获得信件命令在 SMTP 的最小实现中是没有的。用户应该具有控制向终端上写信息的能力。大部分主机允许用户接受或者拒绝类似的信息。

下面三个命令被定义来支持获得信件。它们被用于邮件命令而不是 MAIL 命令, 指示接收 SMTP 这种操作的特殊意义:

```
SEND <SP> FROM:<reverse-path> <CRLF>
```

SEND 命令要求邮件内容直接传送到用户终端。如果用户未打开终端 (或者未接收终端信息), 450 响应将返回一个 RCPT 命令。如果信息被成功发送, 此操作成功。

```
SOML <SP> FROM:<reverse-path> <CRLF>
```

Send 或者 Mail 命令要求将邮件内容直接发送到用户的终端上 (如果用户在终端上)。如果用户不在终端上, 邮件内容直接进入邮箱。如果邮件被发送到用户终端或者用户信箱, 发送操作成功。

```
SAML <SP> FROM:<reverse-path> <CRLF>
```

Send 和 Mail 命令要求邮件内容直接发送到用户终端上 (如果用户在终端上)。不管怎么样, 信件都会进入信箱。如果信件进入信箱, 发送操作成功。

用于 MAIL 命令的响应和这些命令的响应相同。

3.5. 打开和关闭

当打开传送通道时, 要交换一些信息以确定双方的身份。以下的命令是用于打开和关闭的:

```
HELO <SP> <domain> <CRLF>
```

```
QUIT <CRLF>
```

在 HELLO 命令中, 主机自己发送命令, 此命令可以被解释为: “你好, 我是 XX”。

打开联结的例子

```
R: 220 BBN-UNIX.ARPA Simple Mail Transfer Service Ready
```

```
S: HELO USC-ISIF.ARPA
```

```
R: 250 BBN-UNIX.ARPA
```

关闭联结的例子

```
S: QUIT
R: 221 BBN-UNIX.ARPA Service closing transmission channel
```

3.6. 转发

转发路径可能是如下格式：“@ONE,@TWO:JOE@THREE”，在这里，ONE，TWO 和 THREE 是主机。这种格式用于强调地址和路径的区别。邮箱是绝对地址，路径是关于如何到达的信息。这两个概念不应该被混淆。

概念上，转发路径的元素被移动到回复路径作为从一个 SMTP 服务器到另一个 SMTP 服务器的信息。回复路径是一个反向数据源路径，例如从当前信息的位置到发起者的位置。当一个 SMTP 服务器从转发路径中删除自己的标记并将它插入到回复路径中时，它必须使用它发送环境能够理解的名称来进行，以防它的名称在不同的环境中被理解为不同的名字。

如果当 SMTP 接收到信息的转发路径的第一个元素不是此 SMTP 的标记时，此元素不从转发路径中删除，而被用来决定下一个应该发送到的 SMTP 服务器。在任何情况下，SMTP 都将自己的标记加入反向路径中。

使用源路径时，接收 SMTP 接收转发的邮件并发送到另一接收 SMTP 服务器上。接收服务器可以接受或拒绝转发本地用户的邮件。接收 SMTP 通过将它自己的标记从转发路径移至回复路径的开始处来改变命令参数。这时，接收 SMTP 变成了发送 SMTP，也就建立了到下一个转发路径中 SMTP 的通道，然后，它向这个 SMTP 发送邮件。

在回复路径上的头一个主机应是发送 SMTP 命令的主机，在转发路径上第一个主机应是接收 SMTP 命令的主机。

注意：转发路径和回复路径出现在 SMTP 命令和应答中，但不一定要出现在信息中。也就是说，没有必须要这样的路径特别这种格式出现在信息头的“To:”，“From:”和“CC:”等域中。

如果 SMTP 服务器接受了转发任务，但后来它发现因为转发路径不正确或者其它原理无法发送邮件，它必须建立一“undeliverable mail”信号，将它此信号送到此信的发主者那里。

此信号必须是从此主机的 SMTP 服务上发出的，当然了，此服务器不应该再报告出错信息的错误。一种阻止这种出错报告循环的情况是在信号的邮件命令的回复路径上置空。在传送此信息时，允许将回复路径也置为空。一个 MAIL 命令后的回复路径为空表现为如下形式：

```
MAIL FROM:<>
```

下例中显示了不可传送的邮件信息。此信息是对从 HOSTW 上的 JOE 发出的邮件经过在 HOSTX 需要经过 HOSTZ 到达 HOSTY 时出错的回应。我们看到的例子是在 HOSTX 和 HOSTY 之间发生的。

不可传送邮件信息的例子

```
S: MAIL FROM:<>
R: 250 ok
S: RCPT TO:<@HOSTX.ARPA:JOE@HOSTW.ARPA>
R: 250 ok
S: DATA
R: 354 send the mail data, end with .
S: Date: 23 Oct 81 11:22:33
```

```
S: From: SMTP@HOSTY.ARPA
S: To: JOE@HOSTW.ARPA
S: Subject: Mail System Problem
S:
S: Sorry JOE, your message to SAM@HOSTZ.ARPA lost.
S: HOSTZ.ARPA said this:
S: "550 No Such User"
S: .
R: 250 ok
```

3.7. 域

域是最近被引入 ARPA Internet 邮件系统的。使用域可以使地址空间从一个平面的普通字符串主机名变成全局地址的一个层次结构。主机由一个域名取代，起始主机是由一系列元串组成，它们由逗号按最特殊到一般的顺序排列。

例如，“USC-ISIF.ARPA”，“Fred.Cambridge.UK”和“PC7.LCS.MIT.ARPA”可能是主机-域标识符。

无论域名在 SMTP 中如何使用，只有正式的名称才可以被使用，不可以使用假名或昵称。

3.8. 改变角色

TURN 命令可以用来改变在传输信道上通信的角色的程序。如果程序 A 现在是发送 SMTP，它发送 TURN 命令并接到 OK 应答 (250) 后，它就变为接收 SMTP 了。同理，程序 B 也可以从接收 SMTP 变为发送 SMTP。若要拒绝改变角色，接收方可以发送 502 作为应答。

注意：此命令是可选的。在使用 TCP 的传输信道时，一般不使用此命令。然而，当建立传输信道的代价比较大时，此命令很有用。例如，此命令可以支持一般公共交换电话系统作为传输信道。

4. SMTP 说明

4.1. SMTP 命令

4.1.1. 命令语法

SMTP 命令定义了邮件传输或由用户定义的系统功能。它的命令是由<CRLF>结束的字符串。而在带有参数的情况下，命令本身由<SP>和参数分开，如果未带参数可以直接和<CRLF>连接。邮箱的语法格式必须和接收站点的格式一致。下面讨论 SMTP 命令和应答。

发送邮件操作涉及到不同的数据对象，它们由不同的参数相互连接。回复路径就是 MAIL 命令的参数，而转发路径则是 RCPT 命令的参数，邮件日期是 DATA 命令的参数。这些参数或

者数据对象必须跟在命令后。这种模式也就要求有不同的缓冲区来存储这些对象,也就是说,有一个回复路径缓冲区,一个转发路径缓冲区,一个邮件内容缓冲区。特定的命令产生自己的缓冲区,或使一个或多个缓冲的内容被清除。

HELLO (HELO)

此命令用于向接收 SMTP 确认发送 SMTP。参数域包括发送 SMTP 的主机名。接收 SMTP 通过连接确认命令来向发送 SMTP 确认接收 SMTP。引命令和 OK 响应确认发送和接收 SMTP 进入了初始状态,也就是说,没有操作正在执行,所有状态表和缓冲区已经被清除。

MAIL (MAIL)

此命令用于开始将邮件发送到一个或多个邮箱中。参数域包括回复路径。返回路径中包括了可选的主机和发送者邮箱列表。当有主机列表时,它是一个回复路径源,它说明此邮箱是由在表中的主机一一传递发送(第一个主机是最后一个接收到此邮件的主机)过来的。此表也有作向发送者返回非传递信号的源路径。因为每个传递主机地址都被加在此表起始处,它就必须使用发送 IPCE 而不是接收 IPCE(如果它们不是一个 IPCE 的话)清楚的名称。一些出错信息的回复路径可能就是空的。

此命令清除回复路径缓冲区,转发路径缓冲区和邮件内容缓冲区,并且将此命令的回复路径信息插入到回复路径缓冲区中。

RECIPIENT (RCPT)

此命令用于确定邮件内容的唯一接收者;多个接收者将由多个此命令指定。转发路径中包括一个可选的主机和一个必须的目的地邮箱。当出现主机列表时,这就是一个源路径,它指明邮件必须向列表中的上一个主机发送。如果接收 SMTP 未实现邮件的传递发送,就会返回如未知本地用户(550)的信息给用户。

当邮件被传递发送时,传递主机必须将自己的名称由转发路径的开始处移至回复路径的结束处。当邮件最终到达目的地时,接收 SMTP 将以它的主机邮件格式自己的名称插入目标邮件中。例如,由传递主机 A 接收的带有如下参数的邮件时,

```
FROM:<USERX@HOSTY. ARPA>
```

```
TO:<@HOSTA. ARPA, @HOSTB. ARPA:USERC@HOSTD. ARPA>
```

将会变成如下形式:

```
FROM:<@HOSTA. ARPA:USERX@HOSTY. ARPA>
```

```
TO:<@HOSTB. ARPA:USERC@HOSTD. ARPA>.
```

此命令导致它的转发路径参数加入转发路径缓冲区中。

DATA (DATA)

接收者将跟在命令后的行作为邮件内容。此命令导致此命令后的邮件内容加入邮件内容缓冲区。邮件内容可以包括所有 128 个 ASCII 码字符。邮件内容由只包括一个句号的行结束,也就是如下的字符序列:“<CRLF>.<CRLF>”,它指示了邮件的结束。

邮件内容的结束指示要求接收者现在就处理保存的邮件内容。此过程将回复路径缓冲区,转发路径缓冲区和邮件内容缓冲区的内容全部清空。如果操作成功,接收者必须返回 OK 应答;如果失败也必须返回失败应答。

当接收 SMTP 收到一条信息时,无论是用作转发还是此邮件已经到达目的地,它都必须在邮件内容的开始处加上时间戳这一行,这一行指示了接收到邮件主机和发出此邮件主机的标识,以及接收到邮件内容的时间和日期。转发的信件将有多行这样的时间戳。当接收 SMTP 作最后一站的传送时,它将返回路径信息行插入邮件中。此行包括了发送命令中的<reverse-path>的信息。在这里,最后一站的传送的意思是邮件将被送到目的用户手中,但在一些情况下,邮件可能需要更进一步的加工并由另外的邮件系统传送。

可能在返回路径中的邮箱与实际发送的邮件不一致,这个情况可能发生在需要传送一个

特定的错误处理信箱而不是信件发送者那里。上面所述说明了，最后的邮件内容由一个返回路径行，和在其后的一个或多个时间戳行构成。这些行后面是邮件内容的头和体信息。

当处理后面的邮件数据指示部分成功时就需要特定的说明。这种情况可能发生在发送 SMTP 发现当邮件需要传送给多个用户时，只能成功地向其中的一部分发送信息这种情况下。在这种情况下，必须对 DATA 命令发送 OK 应答，而接收 SMTP 组织并发送一个“不可传递邮件”信息到信息的发送者。在此信息中或者发送一个不成功接收者的列表，或者每次发送一个不成接收者，而发送多次。所有不可传递邮件信息由 MAIL 命令发送。

返回路径和接收时间戳例子

```
Return-Path: <@GHI.ARP, @DEF.ARP, @ABC.ARP:JOE@ABC.ARP>
Received: from GHI.ARP by JKL.ARP ; 27 Oct 81 15:27:39 PST
Received: from DEF.ARP by GHI.ARP ; 27 Oct 81 15:15:13 PST
Received: from ABC.ARP by DEF.ARP ; 27 Oct 81 15:01:59 PST
Date: 27 Oct 81 15:01:01 PST
From: JOE@ABC.ARP
Subject: Improved Mailing System Installed
To: SAM@JKL.ARP
```

This is to inform you that ...

SEND (SEND)

此命令用于开始一个发送命令，将邮件发送到一个或多个终端上。参数域包括了一个回复路径，此命令如果成功就将邮件发送到终端上了。

回复路径包括一个可选的主机列表和发送者邮箱。当出现主机列表时，表示这是一个传送路径，邮件就是经过这个路径上的每个主机发送到这里的（列表上第一个主机是最后经手的主机）。此表用于返回非传递信号到发送者。因为每个传递主机地址都被加在此表起始处，它就必须使用发送 IPCE 而不是接收 IPCE（如果它们不是一个 IPCE 的话）清楚的名称。一些出错信息的回复路径可能就是空的。

此命令清除回复路径缓冲区，转发路径缓冲区和邮件内容缓冲区，并且将此命令的回复路径信息插入到回复路径缓冲区中。

SEND OR MAIL (SOML)

此命令用于开始一个邮件操作将邮件内容传送到一个或多个终端上，或者传送到邮箱中。对于每个接收者，如果接收者终端打开，邮件内容将被传送到接收者的终端上，否则就送到接收者的邮箱中。参数域包括回复路径，如果成功地将信息送到终端或邮箱中此命令成功。

回复路径包括一个可选的主机列表和发送者邮箱。当出现主机列表时，表示这是一个传送路径，邮件就是经过这个路径上的每个主机发送到这里的（列表上第一个主机是最后经手的主机）。此表用于返回非传递信号到发送者。因为每个传递主机地址都被加在此表起始处，它就必须使用发送 IPCE 而不是接收 IPCE（如果它们不是一个 IPCE 的话）清楚的名称。一些出错信息的回复路径可能就是空的。

此命令清除回复路径缓冲区，转发路径缓冲区和邮件内容缓冲区，并且将此命令的回复路径信息插入到回复路径缓冲区中。

SEND AND MAIL (SAML)

此命令用于开始一个邮件操作将邮件内容传送到一个或多个终端上，并传送到邮箱中。如果接收者终端打开，邮件内容将被传送到接收者的终端上和接收者的邮箱中。参数域包括回复路径，如果成功地将信息送到邮箱中此命令成功。

回复路径包括一个可选的主机列表和发送者邮箱。当出现主机列表时，表示这是一个传送路径，邮件就是经过这个路径上的每个主机发送到这里的（列表上第一个主机是最后经手的主机）。此表用于返回非传递信号到发送者。因为每个传递主机地址都被加在此表起始处，它就必须使用发送 IPCE 而不是接收 IPCE（如果它们不是一个 IPCE 的话）清楚的名称。一些出错信息的回复路径可能就是空的。

此命令清除回复路径缓冲区，转发路径缓冲区和邮件内容缓冲区，并且将此命令的回复路径信息插入到回复路径缓冲区中。

RESET (RSET)

此命令指示当送邮件操作将被放弃。任何保存的发送者，接收者和邮件内容应该被抛弃，所有缓冲区和状态表应该被清除，接收方必须返回 OK 应答。

VERIFY (VRFY)

此命令要求接收者确认参数是一个用户。如果这是（已经知道的）用户名，返回用户的全名和指定的邮箱。此命令对回复路径缓冲区，转发路径缓冲区和邮件内容缓冲区没有影响。

EXPAND (EXPN)

此命令要求接收者确认参数指定了一个邮件发送列表，如果是一个邮件发送列表，就返回表中的成员。如果这是（已经知道的）用户名，返回用户的全名和指定的邮箱。此命令对回复路径缓冲区，转发路径缓冲区和邮件内容缓冲区没有影响。

HELP (HELP)

此命令导致接收者向 HELP 命令的发送者发出帮助信息。此命令可以带参数，并返回特定的信息作为应答。此命令对回复路径缓冲区，转发路径缓冲区和邮件内容缓冲区没有影响。

NOOP (NOOP)

此命令不影响任何参数和已经发出的命令。它只是说明没有任何操作而不是说明接收者发送了一个 OK 应答。此命令对回复路径缓冲区，转发路径缓冲区和邮件内容缓冲区没有影响。

QUIT (QUIT)

此命令指示接收方必须发送 OK 应答然后关闭传送信道。接收方在接到 QUIT 命令并做出响应之前不应该关闭通信信道。发送方在发送 QUIT 命令和接收到响应之前也不应该关闭信道。即使出错，也不应该关闭信道。如果连接被提前关闭，接收方应该象接收到 RSET 命令一样，取消所有等待的操作，但不恢复原先已经做过的操作。而发送方应该象接收到暂时错误（4XX）一样假定命令和操作仍在支持之中。

TURN (TURN)

此命令指定接收方要么发送 OK 应答并改变角色为发送 SMTP，要么发送拒绝信息并保持自己的角色。如果程序 A 现在是发送 SMTP，它发出 TURN 命令后接收到 OK (250) 应答，它就变成了接收 SMTP。程序 A 就进入初始状态，好象通信信道刚打开一样，这时它发送 220 准备好服务信号。如果程序 B 现在是接收 SMTP，它发出 TURN 命令后接收到 OK (250) 应答，它就变成了发送 SMTP。程序 A 就进入初始状态，好象通信信道刚打开一样，这时它准备接收 220 准备好服务信号。

若要拒绝改变角色，接收方可以发送 502 应答。

对于这些命令的顺序有一定的限制。对话的第一个命令必须是 HELLO 命令，此命令在此后的会话中也可以使用。如果 HELLO 命令的参数不可接受，必须由返回一个 501 失败应答，同时接收到的 SMTP 必须保持在与刚才一致的状态下。NOOP, HELP, EXPN 和 VRFY 命令可以在会话的任何时候使用。MAIL, SEND, SOML 或 SAML 命令开始一个邮件操作。一旦开始了以后就要发送 RCPT 和 DATA 命令。邮件操作可以由 RSET 命令终止。在一个会话中可以有一个或多个操作。

如果在操作开始参数不可接受，必须返回 501 失败应答，同时接收到的 SMTP 必须保持在与刚才一致的状态下。如果操作中的命令顺序出错，必须返回 503 失败应答，同时接收到的 SMTP 必须保持在与刚才一致的状态下。

会话的最后一个命令必须是 QUIT 命令。此命令在会话的其它时间不能使用。

4.1.2. COMMAND 语法格式

命令是由命令码和其后的参数域组成的。命令码是四个字母组成的，不区别大小写。因为下面的命令的作用是相同的：

```
MAIL Mail mail Mail mAll
```

这对于引导任何参数值的标记也是适用的，如 TO 和 to 就是一样的。命令码和参数由一个或多个空格分开。然而在回复路径和转发路径中的参数是区别大小写的。特别是在一些主机上，“smith”和“Smith”就根本不是一个用户。

参数域由不定长的字符串组成，它由<CRLF>结束，接收方在完全接收到此序列前不会采取任何行动。方括号代表可选的参数域。如果不选择的话，系统选择默认的设置。

```
下面是 SMTP 命令：HELO <SP> <domain> <CRLF> MAIL <SP> FROM:<reverse-path> <CRLF>
RCPT <SP> TO:<forward-path> <CRLF>
DATA <CRLF>
RSET <CRLF>
SEND <SP> FROM:<reverse-path> <CRLF>
SOML <SP> FROM:<reverse-path> <CRLF>
SAML <SP> FROM:<reverse-path> <CRLF>
VRFY <SP> <string> <CRLF>
EXPN <SP> <string> <CRLF>
HELP [<SP> <string>] <CRLF>
NOOP <CRLF>
QUIT <CRLF>
TURN <CRLF>
```

上面参数域的格式在下面给 BNF 的格式给出，其中的“...”代表对于一个域的一次或多次的重复。

```
<reverse-path> ::= <path>
<forward-path> ::= <path>
<path> ::= "<" [ <a-d-l> ":" ] <mailbox> ">"
<a-d-l> ::= <at-domain> | <at-domain> "," <a-d-l>
<at-domain> ::= "@" <domain>
<domain> ::= <element> | <element> "." <domain>
```

```

<element> ::= <name> | "#" <number> | "[" <dotnum> "]"
<mailbox> ::= <local-part> "@" <domain>
<local-part> ::= <dot-string> | <quoted-string>
<name> ::= <a> <ldh-str> <let-dig>
<ldh-str> ::= <let-dig-hyp> | <let-dig-hyp> <ldh-str>
<let-dig> ::= <a> | <d>
<let-dig-hyp> ::= <a> | <d> | "-"
<dot-string> ::= <字符串> | <字符串> "." <dot-string>
<字符串> ::= <字符> | <字符> <字符串>
<quoted-string> ::= "" <qtext> ""
<qtext> ::= "\" <x> | "\" <x> <qtext> | <q> | <q> <qtext>
<字符> ::= <c> | "\" <x>
<dotnum> ::= <snum> "." <snum> "." <snum> "." <snum>
<number> ::= <d> | <d> <number>
<CRLF> ::= <CR> <LF>
<CR> ::= 回车符 (ASCII 码 13) <LF> ::= (ASCII 码 10)
<SP> ::= 空格 (ASCII 码 32) <snum> ::= 由一个, 两个或三个数字组成的介于 0-255 之间的数字
<a> ::= 所有 A-Z 的 52 个大小写英文字母
<c> ::= 128 个 ASCII 字符, 但不包括空格和特殊字符
<d> ::= 0-9 数字
<q> ::= 不包括 <CR>, <LF>, "或\" 的 128 个 ASCII 字符
<x> ::= 所有 128 个 ASCII 字符
<special> ::= "<" | ">" | "(" | ")" | "[" | "]" | "\" | "." | "," | ";" | ":" | "@"
"" 或控制字符
    注意: "\" 是一个转意字符, 它表示在其后的字符代表另外的意义。例如 "Joe\, Smith"
    用于表示单独一个由逗号分隔的用户名。主机通常由转化为地址的名称代表。注意: 域的名称元素是正式的名称, 不能够使用昵称或假名。
    有时候名称的转变机制可能不知道主机, 这就造成了通信的阻塞。为了解决这个问题, 可以采取两种方法: 一种方法是: 在 "#" 后加入一个十进制数表示主机地址; 另一种方法是在其后加入 32 位的 IP 地址, IP 地址的形式是由句号分隔的四个介于 0-255 之间的十进制数。
    时间戳行和返回路径行的格式通常由下面定义:
<return-path-line> ::= "Return-Path:" <SP><reverse-path><CRLF>
<time-stamp-line> ::= "Received:" <SP> <stamp> <CRLF>
<stamp> ::= <from-domain> <by-domain> <opt-info> ";" <daytime>
<from-domain> ::= "FROM" <SP> <域> <SP>
<by-domain> ::= "BY" <SP> <域> <SP>
<opt-info> ::= [<via>] [<with>] [<id>] [<for>]
<via> ::= "VIA" <SP> <连接> <SP>
<with> ::= "WITH" <SP> <协议> <SP>
<id> ::= "ID" <SP> <字符串> <SP>
<for> ::= "FOR" <SP> <路径> <SP>
<连接> ::= 在网络信息中心注册的连接的标准名称
<协议> ::= 在网络中心注册的协议的名称

```

<daytime> ::= <SP> <日> <SP> <时间>
 <日期> ::= <日> <SP> <月> <SP> <年>
 <时间> ::= <小时> ":" <分> ":" <秒> <SP> <时区>
 <dd> ::= 由一个或两个数字组成的每月 1-31 日
 <月> ::= "JAN" | "FEB" | "MAR" | "APR" | "MAY" | "JUN" | "JUL" | "AUG" | "SEP" |
 "OCT" | "NOV" | "DEC"
 <年> ::= 由两位数字表示本世界的年代 00-99
 <小时> ::= 每天的 24 小时, 由 0 到 24
 <分> ::= 每小时的分钟数 0-59
 <秒> ::= 每分钟的秒数 0-59
 <时区> ::= 全球标准时区

返回路径例子

Return-Path: <@CHARLIE. ARPA, @BAKER. ARPA: JOE@ABLE. ARPA>

时间戳行例子

Received: FROM ABC. ARPA BY XYZ. ARPA ; 22 OCT 81 09:23:59 PDT

Received: from ABC. ARPA by XYZ. ARPA via TELENET with X25

id M12345 for Smith@PDQ. ARPA ; 22 OCT 81 09:23:59 PDT

4.2. SMTP 响应

对 SMTP 命令的响应是多样的, 它确定了在邮件传输过程中请求和处理的同步, 也保证了发送 SMTP 知道接收 SMTP 的状态。每个命令必须有且只有一个响应。

SMTP 响应由三位数字组成, 其后跟一些文本。数字帮助决定下一个应该进入的状态, 而文本对人是有意义的。三位的响应已经包括了足够的信息, 不用再阅读文本, 文本可以直接抛弃或者传递给用户。特别的是, 文本是与接收和环境相关的, 所以每次接收到的文本可能不同。在附录 E 中可以看到全部的响应码。正规的情况下, 响应由下面序列构成: 三位的数字, <SP>, 一行文本和一个<CRLF>, 或者也可以是一个多行响应。只有 EXPN 和 HELP 命令可以导致多行应答, 然而, 对所有命令, 多行响应都是允许的。

4.2.1. REPLY CODES BY FUNCTION GROUPS 500 格式错误, 命令不可识别 (此错误也包括命令行过长)

501 参数格式错误

502 命令不可实现

503 错误的命令序列

504 命令参数不可实现

211 系统状态或系统帮助响应

214 帮助信息

220 <domain> 服务就绪

221 <domain> 服务关闭传输信道

421 <domain> 服务未就绪, 关闭传输信道 (当必须关闭时, 此应答可以作为对任何命令的响应)

250 要求的邮件操作完成

251 用户非本地, 将转发向<forward-path>

450 要求的邮件操作未完成, 邮箱不可用 (例如, 邮箱忙)

550 要求的邮件操作未完成, 邮箱不可用 (例如, 邮箱未找到, 或不可访问)

- 451 放弃要求的操作；处理过程中出错
- 551 用户非本地，请尝试<forward-path>
- 452 系统存储不足，要求的操作未执行
- 552 过量的存储分配，要求的操作未执行
- 553 邮箱名不可用，要求的操作未执行（例如邮箱格式错误）
- 354 开始邮件输入，以<CRLF>. <CRLF>结束
- 554 操作失败

4.3. 命令和应答序列

发送者和接收者之间的通信是一问一答的交替对话形式，由发送者控制。这样，发送发出一条命令，接收者发出一个响应。接收者在发送下一条指令前必须等应答。一个重要的应答是连接应答。在连接完成时，接收者通常会发送 220“服务就绪”。发送者在继续发送指令前会等待此应答。注意：每个连接应答必须拥有服务主机的正式名称作为第一部分，其后跟响应码。例如：

```
220 <SP> USC-ISIF.ARPA <SP> Service ready <CRLF>
```

下面列出了成功和失败应答，这些应答必须遵守严格的次序，接收者可以不理睬应答中的文本，但是由数字指定的意义和操作和命令应答序列不能更改。命令响应序列：

每个命令列出了它可能的应答。使用在可能应答前的前缀“P”表示预备的（未用在 SMTP 中），“I”表示中间的，“S”表示成功，“F”表示失败，“E”表示错误。如果 SMTP 接收者必须关闭信道，可以对任何命令作出 421（服务不可用，关闭传输信道）响应。此表基于下面要讲述的状态图：

CONNECTION ESTABLISHMENT（建立连接）

S: 220

F: 421

HELO

S: 250

E: 500, 501, 504, 421

MAIL

S: 250

F: 552, 451, 452

E: 500, 501, 421

RCPT S: 250, 251 F: 550, 551, 552, 553, 450, 451, 452 E: 500, 501, 503, 421

DATA

I: 354 -> data -> S: 250

F: 552, 554, 451, 452

F: 451, 554

E: 500, 501, 503, 421

RSET

S: 250

E: 500, 501, 504, 421

SEND

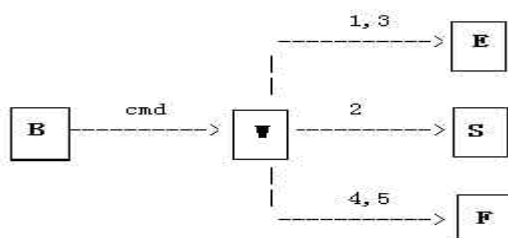
S: 250

F: 552, 451, 452

E: 500, 501, 502, 421
 SOML
 S: 250
 F: 552, 451, 452
 E: 500, 501, 502, 421
 SAML
 S: 250
 F: 552, 451, 452
 E: 500, 501, 502, 421
 VRFY
 S: 250, 251
 F: 550, 551, 553
 E: 500, 501, 502, 504, 421
 EXPN
 S: 250
 F: 550
 E: 500, 501, 502, 504, 421
 HELP
 S: 211, 214
 E: 500, 501, 502, 504, 421
 NOOP
 S: 250
 E: 500, 421
 QUIT
 S: 221
 E: 500
 TURN
 S: 250
 F: 502
 E: 500, 503

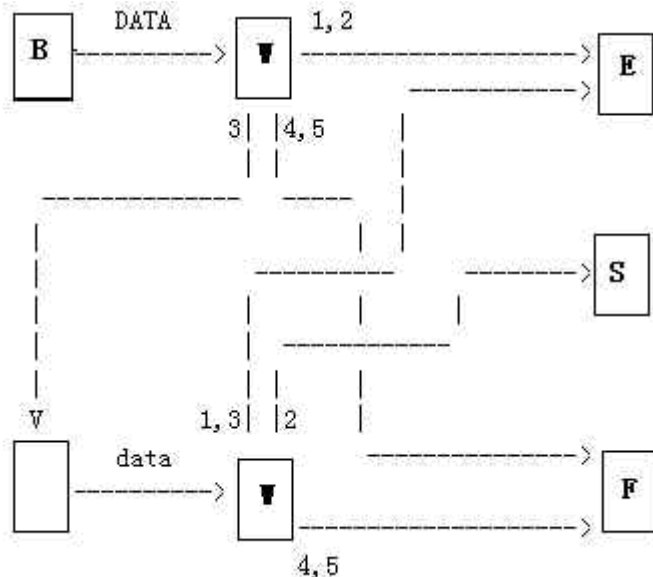
4.4. 状态图

下面状态图是一个简单的 SMTP 实现，每一组命令都有一个状态图。在图中，只使用了响应码的第一位数字作为响应的代表。命令组是对每个命令建立模式然后以结构模式将命令集中起来的。对于每个命令有三种可能的应答：成功 (S)，失败 (F) 和错误 (E)。在状态中，我们使用 B 代表开始，使用 W 代表等待应答。



此状态图使用了如下命令：HELO, MAIL, RCPT, RSET, SEND, SOML, SAML, VRFY, EXPN, HELP, NOOP, QUIT, TURN.

下面是对于 DATA 命令的更复杂的状态图：



注意：这里的邮件内容是多行的，接收者只能收到最后一行时才发出应答。

4.5. 详细内容

4.5.1. 最小实现

为使 SMTP 能够工作，对于接收者来说，这是最少应该实现的命令：

```
COMMANDS - HELO
MAIL
RCPT
DATA
RSET
NOOP
QUIT
```

4.5.2. 透明性

没有对数据透明性的保证，在发送类似“<CRLF>. <CRLF>”结束邮件内容时会发生错误。通常，用户不关心这个“非法”序列。若要所有用户能够透明地使用必须使用以下措施：

1. 在发送邮件之间，发送 SMTP 必须检查邮件的每一行，如果是一个句号，就在行首再加一个句号。
2. 当邮件被接收时，接收 SMTP 必须检查邮件的每一行，如果发现一行仅有一个句号，邮件就此结束，如果一行中有两个句号，那么这一行中就只应该有一个句号，而将第一个句号删除。

发送的邮件内容可以包括所有 128 个 ASCII 字符。所有字符发送到收信者的邮箱，包括格式符号和其它控制字符。如果传输信道提供一个 8 位数据流，7 位的 ASCII 码就可以在其中传送，而将最高位置为 0。一些系统在接收和存储时需要对数据进行格式转换。对于使用不同于 ASCII 字符集的主机或不能以串的形式而只能以记录形式存储的主机更是如此，如果必须进行转换，必须能够再次转换回来，对于用于存储转发的主机更是如此。

4.5.3. 大小

一些对象需要最大和最小大小。也就是说，每个实现必须能够接收大于最小大小的对象，不能发送大于最大大小的对象。对于可能的最大大小，实现技术上并没有限制。

用户 用户名的最大长度是 64 个字节。

域 域的最大长度是 64 个字符

路径 回复路径和转发路径的最大长度是 256 个字符

命令行 命令行的最大长度，包括回车符为 512 个字符

应答行 应答行的最大长度，包括回车符为 512 个字符

文本行 文本行的最大长度，包括回车符和为透明性增加的字符不得超过 1000 个字符

接收缓冲区 接收缓冲区最多可以容纳 100 个接收者

如果出错，应答如下：

500 行过长

501 路径过多

552 接收者过多

552 邮件内容过多

附录 A TCP 传输服务

传输控制协议 (TCP) 在 ARPA Internet 中使用，并遵守网络协议的 US DoD 标准。SMTP 传输信道连接建立在发送进程的端口 U 和接收进程的端口 L 上。一个单一的全双工信道用于传输。被指定用于此协议的服务端口为 25，也就是说 L=25。TCP 连接支持传输 8 位字节，而 SMTP 只需要传输 7 位；这样，每个 8 位字符的最高位被置为 0。

附录 B NCP 传输服务

ARPANET 主机-主机协议 (由网络控制程序实现) 也可以用于 ARPANET。SMTP 传输信道连接建立在发送进程的端口 U 和接收进程的端口 L 上；其后，根据初始连接协议 (ICP) 建立一对简单连接。这一对简单连接被用作传输信道。此协议被指定为连接套接字 25，也就是说 L=25。NCP 连接支持传输 8 位字节，而 SMTP 只需要传输 7 位；这样，每个 8 位字符的最高位被置为 0。

附录 C NITS

也可以使用网络独立传输服务。通过在 NITS 在发送进程和接收进程之间建立传输信道。发送进程执行 CONNECT 原语，然后等待接收 ACCEPT 原语。NITS 连接支持传输 8 位字节，而 SMTP 只需要传输 7 位；这样，每个 8 位字符的最高位被置为 0。

附录 D X.25 传输服务

可以直接使用公共数据网络接收的 X.25 服务，然而，推荐在其上使用可靠的端到端的协议如 TCP。

附录 E 应答码构成方法

三位的应答码每一位都有特定的意义。每一位应答表示是否是成功的，失败的或未完成的。通过这一位，不复杂的 SMTP 发送就可以决定下一步的操作，如果发送方希望大概了解究竟出了什么问题，它可以检测第二位，而第三位则保存了最后更完整的信息。也就是说，从第一位到第三位，接收方可以一步比一步精确地确定接收方的状态。对于第一位有五种可能的表示代表不同的意义：

1yz 部分完成应答

命令被接受，但是要求的操作被中止，原因在应答码中。发送方应该再次发送另一命令指明是否继续操作，或者放弃操作。

2yz 全部完成应答

要求的操作已经完成，可以开始另一个新的请求。

3yz 需要进一步信息的部分完成应答

命令被接受，但是要求的操作被中止，需要接收进一步的信息。发送方应该发送另一条命令指明进一步的信息。

4yz 暂时未完成应答

命令未被接受，要求的操作也未执行，但是发生错误的状态是暂时的，可以再一次请求操作。发送者应该返回命令序列的开始命令（如果有的话）。很难解释这个暂时的意义，特别对于两个不同的站点来说。区别应答是属于些类还是下一类的方法是：如果能够不加任何改变地重复的再一次发送命令，就是本类的，如果不是，就是下一类（5yz）的。

5yz 永久未完成应答

命令未被接受，要求的操作未完成。发送对命令的重复不起作用。即使一些出错条件已经改变，但是用户已经不希望重试，而希望在未来的某个时间再进行操作。

应答的第二位的意义有以下几类：

x0z 语法：此类型的应答是针对以下情况的：语法错误；符合语法但命令不存在功能；未完成或冗余的命令。

x1z 信息：此类型的应答是用于请求信息的，如状态或帮助信息。

x2z 连接：此类型的应答是关于传输信道的。

x3z 未使用。

x4z 未使用。

x5z 邮件系统：此类型的应答指明接收方邮件系统关于请求传送或其它操作的状态的。

第三位给出了更详细的说明。列出的应答表说明了这一点。文本应答是推荐使用的，而不是必须使用的，它的内容是可以根据不同情况而变化的。另一方面，应答码必须严格遵守本节的说明。接收方不应该因为稍稍的不同情况而自己创建新的代码而不使用已经定义的代码。例如，如 NOOP 命令的情况，如果成功执行它后，不用返回任何新的信息，只用返回 250 应答。当发送的命令要求一个未实现的站点指定操作时，应答应该是 502。应答文本可能多于一行；在此情况下，文本必须被标记，接收文本的一方才不致于少读入一行数据。这要求特定的格式说明多行应答。此格式是：每一行，除了最后一行外，都以应答码加一个“-”开始。而最后一行以应答码加空格<SP>开始。如下例：

123-First line

123-Second line

123-234 text beginning with numbers

123 The last line

通常情况下，接收的一方只用寻找应答码加空格的那一行就可以，而忽略前面行的内容。在特殊的情况下，发送方必须知道响应文本的内容，这时接收应答的一方可以通过当时的情况正确地决定是否要知道文本的内容。

附录 F 一些例子

本节提供了一些 SMTP 会话的完整例子。

典型的 SMTP 操作

此类显示邮件如何由在 USC-ISIF 和机上的 Smith 发送到 BBN-UNIX 主机上 Jones, Green 和 Brown 的。这里, 我们假设 USC-ISIF 主机直接和 BBN-UNIX 主机联系。Jones 和 Brown 接收邮件, 而 Green 在 BBN-UNIX 上没有邮箱。

R: 220 BBN-UNIX.ARPA Simple Mail Transfer Service Ready

S: HELO USC-ISIF.ARPA

R: 250 BBN-UNIX.ARPA

S: MAIL FROM:<Smith@USC-ISIF.ARPA>

R: 250 OK

S: RCPT TO:<Jones@BBN-UNIX.ARPA>

R: 250 OK

S: RCPT TO:<Green@BBN-UNIX.ARPA>

R: 550 No such user here

S: RCPT TO:<Brown@BBN-UNIX.ARPA>

R: 250 OK

S: DATA

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

S: Blah blah blah...

S: ...etc. etc. etc.

S: .

R: 250 OK

S: QUIT

R: 221 BBN-UNIX.ARPA Service closing transmission channel

放弃 SMTP 操作

R: 220 MIT-Multics.ARPA Simple Mail Transfer Service Ready

S: HELO ISI-VAXA.ARPA R: 250 MIT-Multics.ARPA

S: MAIL FROM:<Smith@ISI-VAXA.ARPA>

R: 250 OK

S: RCPT TO:<Jones@MIT-Multics.ARPA>

R: 250 OK

S: RCPT TO:<Green@MIT-Multics.ARPA>

R: 550 No such user here

S: RSET

R: 250 OK

S: QUIT

R: 221 MIT-Multics.ARPA Service closing transmission channel

转发邮件

第一步：源主机到转发主机

R: 220 USC-ISIE.ARPA Simple Mail Transfer Service Ready

S: HELO MIT-AI.ARPA

R: 250 USC-ISIE.ARPA

S: MAIL FROM:<JQP@MIT-AI.ARPA>

R: 250 OK

S: RCPT TO:<@USC-ISIE.ARPA:Jones@BBN-VAX.ARPA>

R: 250 OK

S: DATA

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

S: Date: 2 Nov 81 22:33:44

S: From: John Q. Public <JQP@MIT-AI.ARPA>

S: Subject: The Next Meeting of the Board

S: To: Jones@BBN-Vax.ARPA

S:

S: Bill:

S: The next meeting of the board of directors will be

S: on Tuesday.

S: John.

S: .

R: 250 OK

S: QUIT

R: 221 USC-ISIE.ARPA Service closing transmission channel

第二步：转发主机到目的主机

R: 220 BBN-VAX.ARPA Simple Mail Transfer Service Ready

S: HELO USC-ISIE.ARPA

R: 250 BBN-VAX.ARPA

S: MAIL FROM:<@USC-ISIE.ARPA:JQP@MIT-AI.ARPA>

R: 250 OK

S: RCPT TO:<Jones@BBN-VAX.ARPA>

R: 250 OK

S: DATA

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

S: Received: from MIT-AI.ARPA by USC-ISIE.ARPA ;

2 Nov 81 22:40:10 UT

S: Date: 2 Nov 81 22:33:44

S: From: John Q. Public <JQP@MIT-AI.ARPA>

S: Subject: The Next Meeting of the Board

S: To: Jones@BBN-Vax.ARPA

S:

S: Bill:

S: The next meeting of the board of directors will be

S: on Tuesday.

S: John.

S: .

R: 250 OK

S: QUIT

R: 221 USC-ISIE.ARPA Service closing transmission channel

确认和发送

R: 220 SU-SCORE.ARPA Simple Mail Transfer Service Ready

S: HELO MIT-MC.ARPA

R: 250 SU-SCORE.ARPA

S: VRFY Crispin

R: 250 Mark Crispin <Admin.MRC@SU-SCORE.ARPA>

S: SEND FROM:<EAK@MIT-MC.ARPA>

R: 250 OK

S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>

R: 250 OK

S: DATA

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

S: Blah blah blah...

S: ...etc. etc. etc.

S: .

R: 250 OK

S: QUIT

R: 221 SU-SCORE.ARPA Service closing transmission channel

获得和发送邮件 首先确定用户名，然后尝试将邮件发送到用户终端，当它失败时，发送到用户邮箱。

R: 220 SU-SCORE.ARPA Simple Mail Transfer Service Ready

S: HELO MIT-MC.ARPA

R: 250 SU-SCORE.ARPA

```
S: VRFY Crispin

R: 250 Mark Crispin <Admin.MRC@SU-Score.ARPA>

S: SEND FROM:<EAK@MIT-MC.ARPA>

R: 250 OK
S: RCPT TO:<Admin.MRC@SU-Score.ARPA>

R: 450 User not active now

S: RSET

R: 250 OK

S: MAIL FROM:<EAK@MIT-MC.ARPA>

R: 250 OK

S: RCPT TO:<Admin.MRC@SU-Score.ARPA>

R: 250 OK

S: DATA

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

S: Blah blah blah...

S: ...etc. etc. etc.

S: .

R: 250 OK

S: QUIT

R: 221 SU-Score.ARPA Service closing transmission channel
```

上述问题的更有效的作法

```
R: 220 SU-Score.ARPA Simple Mail Transfer Service Ready
```

```
S: HELO MIT-MC.ARPA

R: 250 SU-SCORE.ARPA
S: VRFY Crispin

R: 250 Mark Crispin <Admin.MRC@SU-SCORE.ARPA>

S: SOML FROM:<EAK@MIT-MC.ARPA>

R: 250 OK

S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>

R: 250 User not active now, so will do mail.

S: DATA

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

S: Blah blah blah...

S: ...etc. etc. etc.

S: .

R: 250 OK

S: QUIT

R: 221 SU-SCORE.ARPA Service closing transmission channel
```

邮件列表 首先，两个邮件列表中的每一个在不同主机的不同会话上扩展，然后，通过转发主机向列表上的用户发送邮件。

第一步：扩展第一个列表

```
R: 220 MIT-AI.ARPA Simple Mail Transfer Service Ready

S: HELO SU-SCORE.ARPA

R: 250 MIT-AI.ARPA
```

S: EXPN Example-People

R: 250-<ABC@MIT-MC.ARPA>

R: 250-Fred Fonebone <Fonebone@USC-ISIQ.ARPA>

R: 250-Xenon Y. Zither <XYZ@MIT-AI.ARPA>

R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>

R: 250-<joe@foo-unix.ARPA>

R: 250 <xyz@bar-unix.ARPA>

S: QUIT

R: 221 MIT-AI.ARPA Service closing transmission channel

第二步：扩展第二个列表

R: 220 MIT-MC.ARPA Simple Mail Transfer Service Ready

S: HELO SU-SCORE.ARPA

R: 250 MIT-MC.ARPA

S: EXPN Interested-Parties

R: 250-Al Calico <ABC@MIT-MC.ARPA>

R: 250-<XYZ@MIT-AI.ARPA>

R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>

R: 250-<fred@BBN-UNIX.ARPA>

R: 250 <xyz@bar-unix.ARPA>

S: QUIT

R: 221 MIT-MC.ARPA Service closing transmission channel

第三步：通过转发主机向包括于两个列表中的所有用户发送邮件

R: 220 USC-ISIE.ARPA Simple Mail Transfer Service Ready

S: HELO SU-SCORE.ARPA

R: 250 USC-ISIE.ARPA

S: MAIL FROM:<Account.Person@SU-SCORE.ARPA>

R: 250 OK

S: RCPT TO:<@USC-ISIE.ARPA:ABC@MIT-MC.ARPA>

R: 250 OK

S: RCPT TO:<@USC-ISIE.ARPA:Fonebone@USC-ISIQA.ARPA>

R: 250 OK

S: RCPT TO:<@USC-ISIE.ARPA:XYZ@MIT-AI.ARPA>

R: 250 OK

S: RCPT

TO:<@USC-ISIE.ARPA,@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>

R: 250 OK

S: RCPT TO:<@USC-ISIE.ARPA:joe@FOO-UNIX.ARPA>

R: 250 OK

S: RCPT TO:<@USC-ISIE.ARPA:xyz@BAR-UNIX.ARPA>

R: 250 OK

S: RCPT TO:<@USC-ISIE.ARPA:fred@BBN-UNIX.ARPA>

R: 250 OK

S: DATA

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

S: Blah blah blah...

S: ...etc. etc. etc.

S: .

R: 250 OK

S: QUIT

R: 221 USC-ISIE.ARPA Service closing transmission channel

转发的情况

R: 220 USC-ISIF.ARPA Simple Mail Transfer Service Ready

S: HELO LBL-UNIX.ARPA

R: 250 USC-ISIF.ARPA

S: MAIL FROM:<mo@LBL-UNIX.ARPA>

R: 250 OK

S: RCPT TO:<fred@USC-ISIF.ARPA>

R: 251 User not local; will forward to <Jones@USC-ISI.ARPA>

S: DATA

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

S: Blah blah blah...

S: ...etc. etc. etc.

S: .

R: 250 OK

S: QUIT

R: 221 USC-ISIF.ARPA Service closing transmission channel

第一步：尝试第一台主机上的邮箱

R: 220 USC-ISIF.ARPA Simple Mail Transfer Service Ready

S: HELO LBL-UNIX.ARPA

R: 250 USC-ISIF.ARPA

S: MAIL FROM:<mo@LBL-UNIX.ARPA>

R: 250 OK

S: RCPT TO:<fred@USC-ISIF.ARPA>

R: 251 User not local; will forward to <Jones@USC-ISI.ARPA>

S: RSET

R: 250 OK

S: QUIT

R: 221 USC-ISIF.ARPA Service closing transmission channel

第二步：尝试第二台主机上的邮箱

R: 220 USC-ISI.ARPA Simple Mail Transfer Service Ready

S: HELO LBL-UNIX.ARPA

R: 250 USC-ISI.ARPA

S: MAIL FROM:<mo@LBL-UNIX.ARPA>

R: 250 OK

S: RCPT TO:<Jones@USC-ISI.ARPA>

R: OK

S: DATA

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

S: Blah blah blah...

S: ...etc. etc. etc.

S: .

R: 250 OK

S: QUIT

R: 221 USC-ISI.ARPA Service closing transmission channel

许多接收者的情况

R: 220 BERKELEY.ARPA Simple Mail Transfer Service Ready

S: HELO USC-ISIF.ARPA

R: 250 BERKELEY.ARPA

S: MAIL FROM:<Postel@USC-ISIF.ARPA>

R: 250 OK

S: RCPT TO:<fabry@BERKELEY.ARPA>

R: 250 OK

S: RCPT TO:<eric@BERKELEY.ARPA>

R: 552 Recipient storage full, try again in another transaction

S: DATA

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

S: Blah blah blah...

S: ...etc. etc. etc.

S: .

R: 250 OK

S: MAIL FROM:<Postel@USC-ISIF.ARPA>

R: 250 OK

S: RCPT TO:<eric@BERKELEY.ARPA>

R: 250 OK

S: DATA

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

S: Blah blah blah...

S: ...etc. etc. etc.

S: .

R: 250 OK

S: QUIT

R: 221 BERKELEY.ARPA Service closing transmission channel

名词表:

<CRLF>

回车

<SP>

空格

ASCII

美国标准信息交换码

主机

拥有 SMTP 进程或邮箱的网络计算机

发送 SMTP 进程

与接收 SMTP 进程一起工作的进程。发送 SMTP 开始传输服务连接，它发出 SMTP 命令，接收应答，管理邮件的传送

用户

希望获得邮件服务的人（或以人的名义出现的进程），还有就是邮件的接收者。

会话

当传输信道打开时进行的一系列信息交换

传输服务

可靠的面向流的数据通信服务。例如：NCP，TCP，NITS。

传输信道

在发送 SMTP 和接收 SMTP 之间建立的全双工的用于交换命令，应答和邮件内容的信道

字符

可显示字符串

行

以一个<CRLF>结束的邮件内容

应答

接收 SMTP 对发送 SMTP 的通过传输信道发送的对某一命令的（成功或不成功的）响应。应答的一般格式是应答码加一段文本。通常情况下，应答码供机器使用，而文本用于人类用户使用

邮件内容

一系列的字符串，它们符合 ARPA Internet 文本信息格式标准的标准字符集

邮件内容结束标记

标明邮件内容结束的特定字符

邮箱

指定应该向何处发向用户的信件地址（字符串）。它通常由用户名和主机名表示

命令

由发送 SMTP 发送到接收 SMTP 的要求一个邮件服务操作的请求

域

邮件系统中主机地址字符串的层次式表示

接收 SMTP 进程

与发送 SMTP 进程一起工作的进程。它等待通过传输服务建立的连接。它接收发送 SMTP 发出的命令，给出应答并执行相应的操作

操作

一个信息由一个接收者发送到另一个或多个接收者的一系列操作

参考资料

[1] ASCII

ASCII, "USA Code for Information Interchange", United States of America Standards Institute, X3.4, 1968. Also in: Feinler, E. and J. Postel, eds., "ARPANET Protocol Handbook", NIC 7104, for the Defense Communications Agency by SRI International, Menlo Park, California, Revised January 1978.

[2] RFC 822

Crocker, D., "Standard for the Format of ARPA Internet Text Messages," RFC 822, Department of Electrical Engineering, University of Delaware, August 1982.

[3] TCP

Postel, J., ed., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", RFC 793, USC/Information Sciences Institute, NTIS AD Number A111091, September 1981. Also in: Feinler, E. and J. Postel, eds., "Internet Protocol Transition Workbook", SRI International, Menlo Park, California, March 1982.

[4] NCP

McKenzie, A., "Host/Host Protocol for the ARPA Network", NIC 8246, January 1972. Also in: Feinler, E. and J. Postel, eds., "ARPANET Protocol Handbook", NIC 7104, for the Defense Communications Agency by SRI International, Menlo Park, California, Revised January 1978.

[5] Initial Connection Protocol

Postel, J., "Official Initial Connection Protocol", NIC 7101, 11 June 1971. Also in: Feinler, E. and J. Postel, eds., "ARPANET Protocol Handbook", NIC 7104, for the Defense Communications Agency by SRI International, Menlo Park, California, Revised January 1978.

[6] NITS

PSS/SG3, "A Network Independent Transport Service", Study Group 3, The Post Office PSS Users Group, February 1980. Available from the DCPU, National Physical Laboratory, Teddington, UK.

August 1982
Simple Mail Transfer Protocol

RFC 821

[7] X.25

CCITT, "Recommendation X.25 - Interface Between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks," CCITT Orange Book, Vol. VIII.2, International Telephone and Telegraph Consultative Committee, Geneva, 1976.