

# RFC 821 内容介绍

网易公司邮件组  
彭勇华 Jeff Peng  
[pangj@earthlink.net](mailto:pangj@earthlink.net)

# 一. SMTP模式



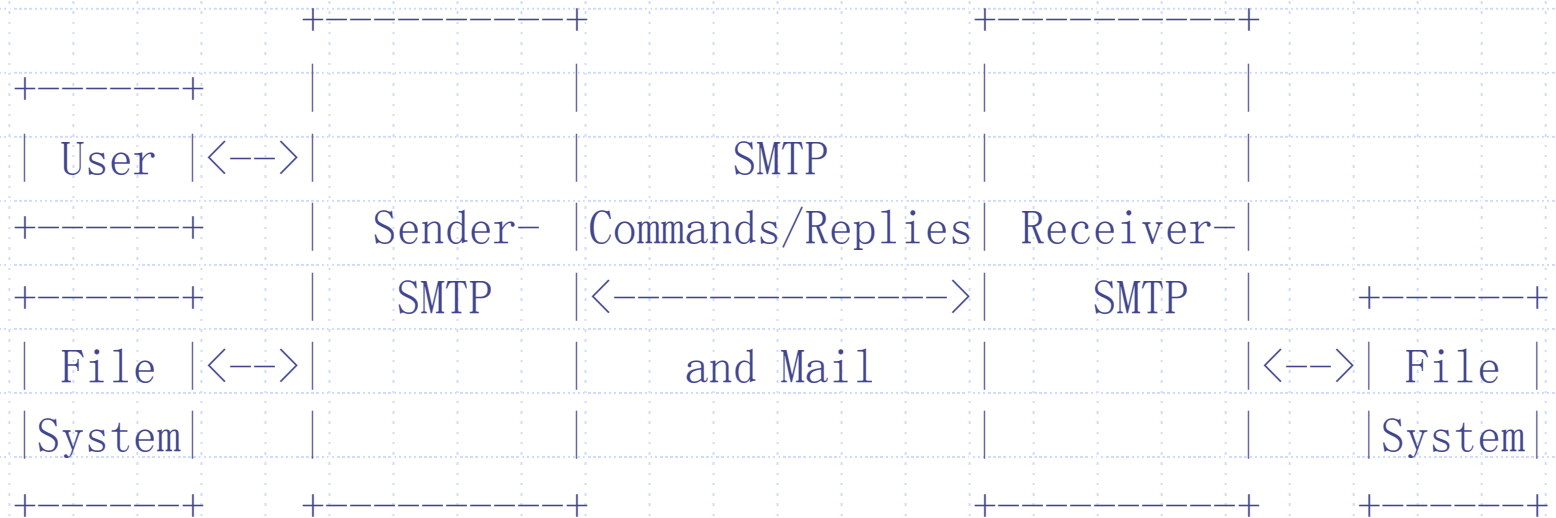
# 1. SMTP模式介绍

1) SMTP设计基于下列通信模式：用户开始mail请求，发送smtp建立双向传输通道到接受smtp。接受smtp可能是最终目的地，或者是一个中继。SMTP命令由发送SMTP产生，发送到接受SMTP。相应的SMTP响应会由接受SMTP产生，并返回到发送SMTP。

# 1. SMTP模式介绍

一旦传输通道建立，发送SMTP发送一条mail命令指示本邮件的sender。假如接受smtp可以接受邮件，它会响应一个OK。然后发送SMTP发送一个RCPT命令，标明收件人。假如接受SMTP能接受它，就响应一个OK；如不能，它拒绝该收件人（但不是整个邮件会话）。发送方和接受方可以协商几个收件人。当收件人协商妥当后，发送mail data。假如接受SMTP成功的处理data，它响应OK。

# 1. SMTP模式介绍



Sender-SMTP

Receiver-SMTP

Model for SMTP Use

# 1. SMTP模式介绍

2) MAIL命令后跟reverse-path（退信路径），它标明邮件来自何处。RCPT命令后是forward-path（转发路径），它标明邮件发向何处。

# 1. SMTP模式介绍

3) 当同一消息发往多个收件人时, SMTP鼓励对同一目的主机的所有收件人, 仅传送一份data拷贝。

# 1. SMTP模式介绍

4) SMTP命令和响应有严格的语法，见后面所示。



# 1. SMTP模式介绍

5) 命令不区分大小写。但mailbox的用户名区分大小写，主机名不区分大小写。

## 二. SMTP会话流程



# 1. 发送邮件

- 1) 有3个发信流程: MAIL, RCPT, DATA, 第一步是MAIL命令。退信路径包含源mailbox:

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

该命令指示SMTP接受方, 新的mail事务开始了, 重置所有状态表和缓冲, 包括任何收件人或mail data。给出退信路径用于报告错误。退信路径可以包含源路由表, 它是由主机和mailbox组成的反向源路由列表。第一个host应该是发送此命令的host。(RFC2821不鼓励源路由表。)

# 1. 发送邮件

2) RCPT给出转发路径，标明一个收件人。假如可接受，接受方返回250 OK，并存储转发路径。假如收件人未知，接受方返回550失败响应。RCPT命令可重复多次。跟退信路径一样，转发路径是主机和mailbox的源路由列表。第一个host应该是接受此命令的host。（RFC2821不鼓励源路由表。）

# 1. 发送邮件

3) DATA后，如果命令接受，对端返回354，完成文本发送后，对端返回250 OK。指示DATA数据终结的符号是一个单行，只包括一个句点。请注意mail data包括邮件头部和主题，邮件头部的参数有Date,Subject,To,Cc,From等。

DATA命令仅在如下情况下失败：a) mail事务不完整（例如没有收件人）；b) 资源不可用。

# 1. 发送邮件

## 4) 会话示例:

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: ...etc. etc. etc.

S: <CRLF>.<CRLF>

R: 250 OK

## 2. 转发

1) 251 User not local; will forward to  
<forward-path>

用户非本地，但是接受者会负责转发；

551 User not local; please try  
<forward-path>

用户非本地，接受者不负责转发，由发送者转发到目的地址，或者直接退信给用户。

### 3.VERIFYING 和EXPANDING

- 1) VRFY是验证用户是否存在的， EXPN是验证邮件列表是否存在的。



## 4.SENDING 和MAILING

- 1) 定义如下: The delivery to the user's mailbox is called "mailing", the delivery to the user's terminal is called "sending".

## 4.SENDING 和MAILING

如下3个是sending命令，它们用以代替MAIL命令：

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

SOML <SP> FROM:<reverse-path> <CRLF> (send  
or mail)

SAML <SP> FROM:<reverse-path> <CRLF> (send  
and mail)

## 5. 打开和关闭会话

1) 主要指如下2个命令:

HELO <SP> <domain> <CRLF>

QUIT <CRLF>

## 6. 中继

- 1) 转发路径可以是源路由方式  
“"@ONE,@TWO:JOE@THREE”，这里ONE,TWO,THREE是主机。这种方式强调一个地址和一个路由之间的区别。  
mailbox是绝对的地址，但路由是如何到达该地址的信息。两者不应混淆。

## 6. 中继

2) 概念上，当message从一个SMTP中继到另一个时，转发路径的元素被移到退信路径。退信路径是反向源路由（从消息的当前位置到消息起源处的路由）。

## 6. 中继

3) 假如消息到达某SMTP时，转发路径的第一个元素不是该SMTP自身，则转发路径的元素不会删除，它用于决定下一跳SMTP。任何情形下，该SMTP把它自己写入退信路径。

## 6. 中继

4) 退信路径里的第一个主机应该时发送SMTP命令的主机；转发路径里的第一个主机应该时接受SMTP命令的主机。

## 6. 中继

5) 请注意转发路径 (forward-path) 和退信路径 (reverse-path) 出现在smtp命令里, 但不必出现在消息里, 也就是说, 不必在to:,from:,cc:里包含这些内容。



## 6. 中继

6) 如果退信给用户错误，接受到退信的smtp不应再给发送退信的smtp发送错误通知。阻止退信循环的方式是，发送退信的服务器用空发件人来发送。命令如下：MAIL FROM: <>

## 6. 中继

7) 示例：如下命令里：

```
S: RCPT TO:<@HOSTX.ARPA:JOE@HOSTW.ARPA>  
@HOSTX.ARPA:是邮件relay的路由。
```

# 7. DOMAINS

因特网域名系统。

## 8. 改变角色

- 1) TURN命令用于反转角色。假如程序A当前是发送SMTP，它在发送TRUM命令并接受到250 OK响应后，就变成了接受SMTP。如果拒绝角色转换，接受者应发送502响应。该命令可选，不应在双方是tcp会话通道时使用，在公共交换电话系统作为通道时可用。

# 三. SMTP规范



# 1. SMTP命令

1) 通用的几个写法:

a) <CRLF>: 回车换行符, 等同于Unix系统的"\r\n";

b) <SP>: 空格符

# 1. SMTP命令

2) HELO命令和OK响应，确认双方位于初始状态，也就是说没有事务在处理，状态表和buffer都清空。

# 1. SMTP命令

3) MAIL命令后跟退信路径，退信路径由可选的主机列表，和sender的邮箱组成。当指定了主机列表时，这是个退信的源路径。每个relay（中继）主机增加它自己到该列表的开头处，它必须使用IPCE指定格式的名字。

该命令清空退信路径buffer，转发路径buffer，以及mail data buffer，并插入退信路径到缓冲里。



# 1. SMTP命令

4) RCPT命令跟的参数与mail类似，可以指定mail relay的源路径。假如接受smtp拒绝relay，它返回550。当邮件relay时，中继主机必须从转发路径的开头处删除自己，并将自己放在退信路径的开头。

# 1. SMTP命令

例如，如下命令：

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

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

经过下一跳后，变成：

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

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

该命令导致转发路径参数增加到转发路径的buffer里。

# 1. SMTP命令

5) DATA命令：该命令导致mail data消息增加到mail data buffer里。mail data可以包含任何128 ASCII字符码。mail data以只包含一个句点的单行结束，"`<CRLF>.<CRLF>`"。

# 1.SMTP命令

6) DATA命令：当接受smtp收到一条中继消息或最终目的的消息时，它在mail data的开头处插入时间戳。这个时间戳暗示发送主机的身份，以及接受主机的身份（正在插入时间戳的主机），以及message接受的时间日期。中继消息有多个时间戳的行。

# 1.SMTP命令

7) DATA命令：当接受SMTP完成某条消息的最终投递时，它在mail data的开头处插入退信路径。这个退信路径保留着<reverse-path>里的信息（来自mail命令）。最终投递意味着邮件离开SMTP世界，交由deliveragent投递。

# 1.SMTP命令

8) DATA命令：有可能退信的mailbox不同于实际sender的mailbox。例如，错误响应可以投递到某个特殊的错误处理邮箱，而不是消息发送邮箱。

# 1.SMTP命令

9) DATA命令：上述步骤意味着最终的mail data以退信路径开头，紧跟着一个或多个时间戳行，然后时间戳后面就是信件的头部和body。

# 1.SMTP命令

10) DATA命令：假如data命令完成后，接受主机发现多个rcpt对象里，有一部分是可投递的，一部分不可投递的，在该形式下，响应DATA命令的必须是OK。但是，接受SMTP必须发送一封“undeliverable mail”通知到原始发送者。



# 1.SMTP命令

11) DATA命令：退信路径和时间戳的示例：

Return-Path:

<@GHI.ARPA,@DEF.ARPA,@ABC.ARPA:JOE@ABC.ARPA>

Received: from GHI.ARPA by JKL.ARPA ; 27 Oct 81 15:27:39  
PST

Received: from DEF.ARPA by GHI.ARPA ; 27 Oct 81 15:15:13  
PST

Received: from ABC.ARPA by DEF.ARPA ; 27 Oct 81 15:01:59  
PST

Date: 27 Oct 81 15:01:01 PST

From: JOE@ABC.ARPA

Subject: Improved Mailing System Installed

To: SAM@JKL.ARPA

# 1.SMTP命令

12) SEND命令：见前述说明

13) SOML命令：见前述说明

14) SAML命令：见前述说明

# 1.SMTP命令

15) RSET命令：该命令指示当前mail传输abort，所有存储的sender,recipients,mail data都丢弃，所有缓冲和状态表清空。

# 1.SMTP命令

16) VRFY命令：验证用户

17) EXPN命令：验证邮件列表

18) HELP命令：系统帮助

# 1.SMTP命令

19) NOOP命令：该命令没有任何影响，它只让接受者响应一个OK。

20) QUIT命令：终止会话

21) TURN命令：反转角色

# 1.SMTP命令

## 22) 关于命令顺序:

session的第一个命令必须是HELO，可以重复使用HELO。HELO不接受，返回501。

NOOP，HELP，EXPN以及VRFY命令可用在session的任何阶段。

MAIL, SEND, SOML, or SAML开始一个mail事务。Mail事务由这几个命令开头，然后是一个或多个RCPT命令，接着是DATA命令，按顺序来。邮件事务可用RSET命令来中断。在同一session里，可以有0个或多个邮件事务。

假如命令参数不接受，返回501；假如顺序错误，返回503。

最后一个命令是QUIT命令。

## 2. 命令语法

1) 命令由4位alpha字符组成，不区分大小写。

某些邮件头里的符号表现参数也不区分大小写，例如TO或to。但这些参数的值有的区分大小写，例如退信路径和转发路径里的参数是大小写敏感的。某些主机的用户"smith"与"Smith"不同。

## 2. 命令语法

2) 语法列表:

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>



## 2. 命令语法

### 3) 关于HELO domain:

这里的host必须是官方主机名，非昵称或别名。如果没有指定主机名，则可以是10进制的主机地址，或12.34.56.78的32位的internet地址。

## 3. SMTP响应

- 1) SMTP响应由3位数字紧随一段文本组成。数字对sender mta有意义，而文本是让人工可读的。3位数字在设计上包含足够的信息，机器不需要后面的文本，甚至可以丢弃它。

# 3. SMTP响应

## 2) SMTP响应CODE含义:

500: 语法错误, 命令不可识别 (可能包含命令太长的错误)

501: 语法错误, 参数不正确

502: 命令不能执行

503: 顺序错

504: 命令参数不可执行

### 3. SMTP响应

211: 系统状态, 或系统帮助的应答

214: 帮助消息, 仅对人工有用

220: 服务准备好

221: 服务关闭传输通道

421: 服务不可用, 关闭传输通道

### 3. SMTP响应

250: 请求OK

251: 用户非本地, 将转发至<forward-path>

450: 请求动作未采取: mailbox不可用, 例如 mailbox繁忙

550: 请求动作未采取: mailbox不可用, 例如 mailbox不存在或不可访问

451: 请求动作中断: 处理时发生错误

551: 用户非本地, please try <forward-path>

452: 请求动作未采取: 信箱空间余额不足

# 3. SMTP响应

552: 请求动作中断: 信箱空间满

553: 请求动作未采取: mailbox名字不允许, 例如Mailbox语法错误

354: 开始mail输入, 以<CRLF>.<CRLF>结束

554: 事务失败。

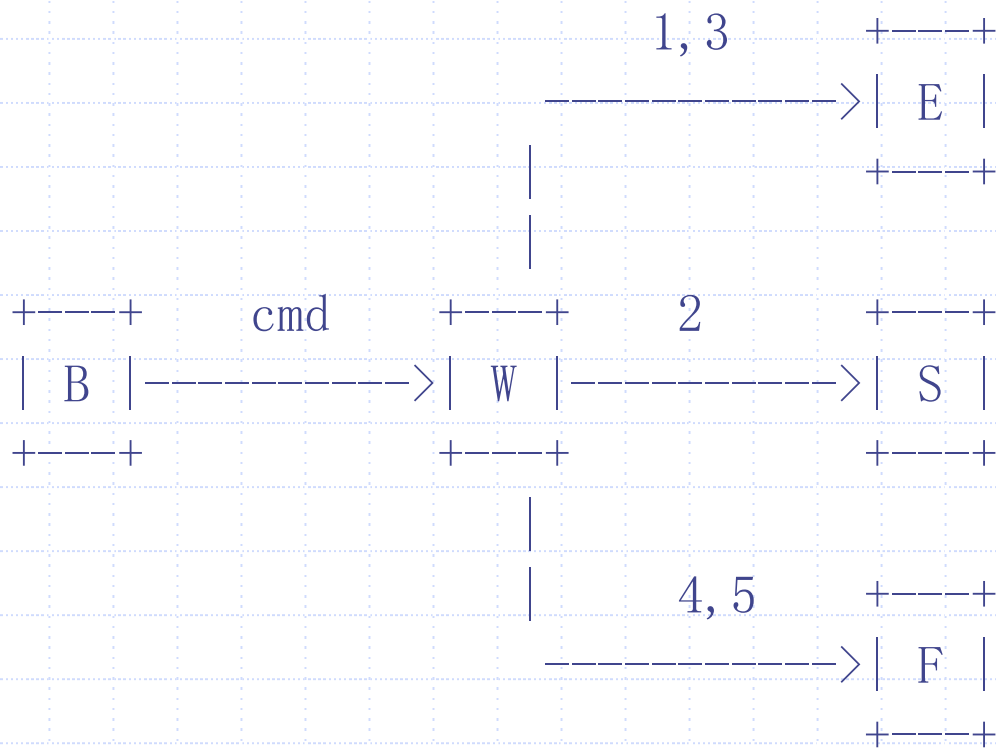
(基本上, 2xx的是成功, 45x,55x的是失败, 50x的是错误)

## 3. SMTP响应

3) 关于各sender命令对应的响应code, 在  
<page37>页有非常详细的说明。

# 4. SMTP状态图

## 1) 最基本的SMTP状态图:





## 4. SMTP状态图

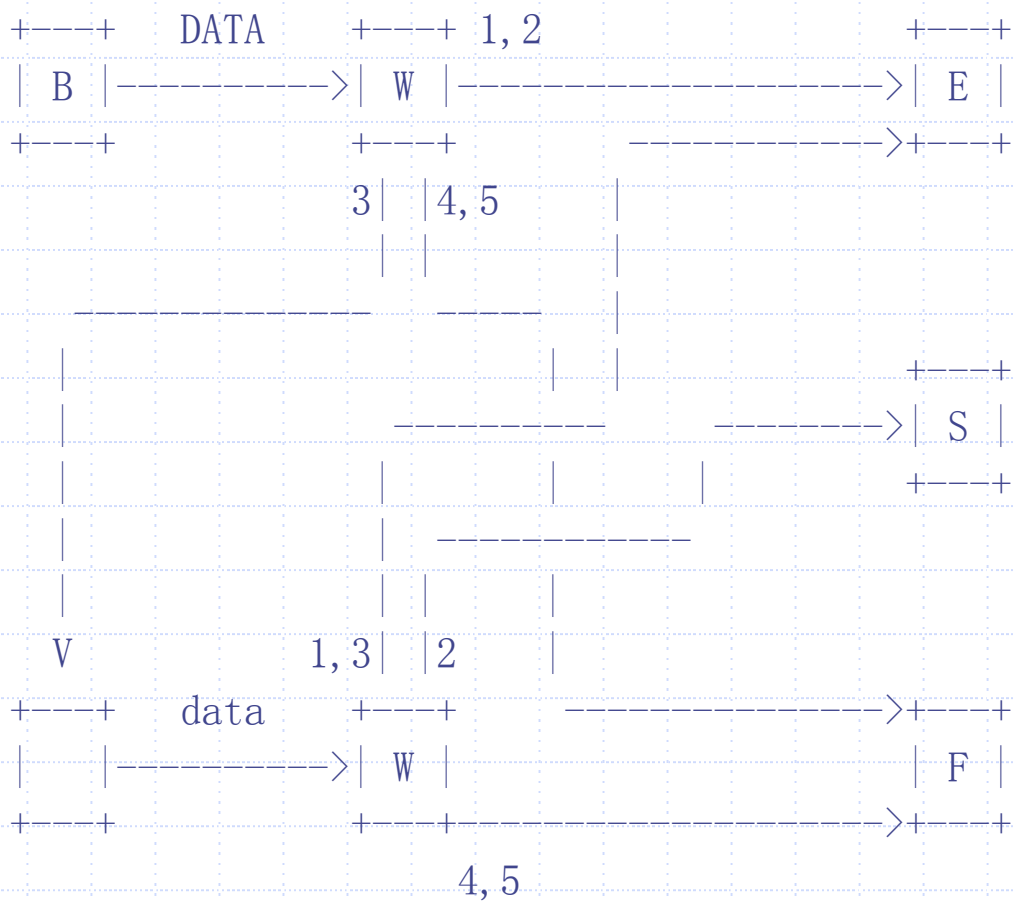
B:Begin W:Wait S:Success E>Error F:False

CMD:

HELO, MAIL, RCPT, RSET, SEND, SOML, SAML, VRFY, EXPN,  
HELP,  
NOOP, QUIT, TURN.

# 4. SMTP状态图

2) data命令状态图:



## 5. 其他细节

1) 接受端最少必须接受如下smtp命令:

COMMANDS -- HELO

MAIL

RCPT

DATA

RSET

NOOP

QUIT

## 5. 其他细节

2) 关于结束data内容的"<CRLF>.<CRLF>" :

a) 假如mail文本的第一个字节是个句点，则发送mta会将这个句点写入该行的开头。

b) 当接受mta接受到文本行时，如果某行只有1个句点，则标志着mail结束。如果某行第一个字符是句点，但该行还有其他内容，则删除这个句点。

这样保证SMTP会话过程对用户透明，即用户不用知道这个"."的细节。

## 6. 关于SIZE

如下size限制表明：接受mta必须能够接受这些size内的目标，但发送mta不能发送超过这些size的目标。

user: 用户名的最大长度是64字符；

domain: 最大长度64字符

path: 退信路径或转发路径的最大长度是256字符

command line: 命令行的最大长度是512字符

reply line: 最大响应行是512字符

text line: 文本行的最大长度是1000字符

recipients buffer: 最多收件人100个

## 6. 关于SIZE

超出size后可发送如下响应码:

500 Line too long.

501 Path too long

552 Too many recipients.

552 Too much mail data.

## 四. SMTP响应code的原理



# 1. SMTP响应code原理说明

3个字节的smtp响应code有特殊意义：第1个数字指示响应是good,bad或不完整的。发送smtp通过检查第1个数字，就可知道下一步的动作。如果发送smtp想知道大概发生了什么错误（例如mail系统错误或命令语法错误）就可检查第2个数字，第3个数字是保留给最终的信息分级的。



# 1. SMTP响应code原理说明

第1个数字有5个值，意义如下：

**1xx：** 肯定的预备响应

命令被接受，但请求动作被暂停挂起，本次响应的信息未被确认。发送smtp应该发送另一个命令，指示继续或中断本次动作。

没有smtp命令允许这种类型的应答，所以也就不会有继续或中断的动作了。

**2xx：** 肯定的完成响应

请求动作成功完成，可以开始新的请求。

# 1. SMTP响应code原理说明

## 3xx: 肯定的交互式响应

命令已被接受，但请求动作被暂停挂起，等待更多信息。发送smtp应该发送其他的命令指明这个信息。这个响应用于命令序列组里。

## 4xx: 临时的否定响应

命令未被接受，请求动作没有发生。然而，错误条件是临时的，可以再次请求。发送方应该返回到命令序列的最开始。所谓临时的这里存在矛盾，因为2个不同的站点，对这个临时时间的理解不同。每个应答可以有不通的时间值，但发送端应该重新请求。

## 5xx: 永久的否定响应

命令未被接受，请求动作不再发生。

# 1. SMTP响应code原理说明

第2个数字的意义:

x0x: 语法 -- 这个0代表语法错误。

x1x: 信息 -- 对信息请求的响应, 例如status或help。

x2x: 连接 -- 对传输通道的响应

x3x: 还未指定

x4x: 还未指定

x5x: 邮件系统 -- 这些响应表明接受方邮件系统的状态

# 1. SMTP响应code原理说明

第3个数字的意义:

第3个数字体现响应code的具体意义。第3个数字后跟响应文本，推荐每个响应附带文本，但不强制要求。但是，响应code必须严格遵循本节规范。接受方不应自主创建自己的code。

例如，NOOP成功执行后不会提供给发送smtp任何新信息，它应返回250。当请求不可执行的非站点规范的动作时，返回502。若请求命令可执行，但请求参数不可执行，则返回504。

# 1. SMTP响应code原理说明

请求文本可以长于1行，在此情况下，多行文本必须做上标识，以便发送smtp能知道何时终止读对端的响应。多行响应的格式要求每一行（除了最后一行），以响应code开始，紧跟一个短横线'- '，然后是文本。最后一行以响应code开始，紧跟一个空格，然后是可选的文本，最后是<CRLF>。

# 1. SMTP响应code原理说明

例如：

123-First line

123-Second line

123-234 text beginning with numbers

123 The last line

很多情形下发送smtp仅需要最后一行的信息，忽略前面的行。在少许情形下，所有的reply文本对发送smtp都重要。