

第28章 SMTP: 简单邮件传送协议

28.1 引言

电子邮件 (e-mail) 无疑是最流行的应用程序。[Caceres et al.1991]说明, 所有TCP连接中大约一半是用于简单邮件传送协议 SMTP (Simple Mail Transfer Protocol)的 (以比特计算为基础, FTP连接传送更多的数据)。[Paxson 1993] 发现, 平均每个邮件中包含大约 1500字节的数据, 但有的邮件中包含兆比特的数据, 因为有时电子邮件也用于发送文件。

图28-1显示了一个用TCP/IP交换电子邮件的示意图。

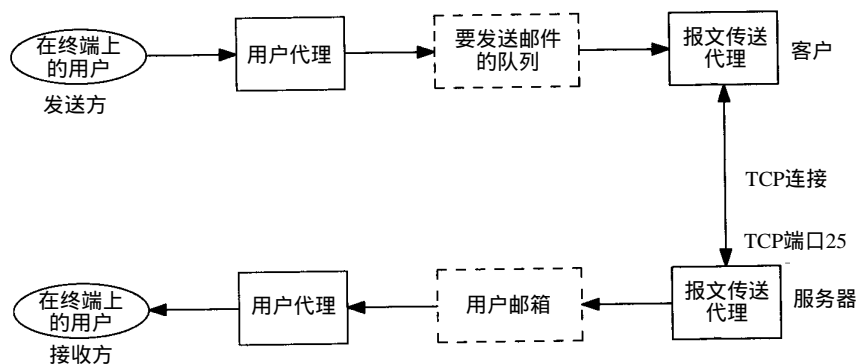


图28-1 Internet电子邮件示意图

用户与用户代理 (user agent) 打交道, 可能会有多个用户代理可供选择。常用的 Unix 上的用户代理包括 MH, Berkeley Mail, Elm 和 Mush。

用TCP进行的邮件交换是由报文传送代理 MTA (Message Transfer Agent) 完成的。最普通的 Unix 系统中的 MTA 是 Sendmail。用户通常不和 MTA 打交道, 由系统管理员负责设置本地的 MTA。通常, 用户可以选择它们自己的用户代理。

本章研究在两个 MTA 之间如何用 TCP 交换邮件。我们不考虑用户代理的运行或实现。

RFC 821 [Postel 1982] 规范了 SMTP 协议, 指定了在一个简单 TCP 连接上, 两个 MTA 如何进行通信。RFC 822 [Crocker 1982] 指定了在两个 MTA 之间用 RFC 821 发送的电子邮件报文的格式。

28.2 SMTP 协议

两个 MTA 之间用 NVT ASCII 进行通信。客户向服务器发出命令, 服务器用数字应答码和可选的人可读字符串进行响应。这与上一章的 FTP 类似。

客户只能向服务器发送很少的命令: 不到 12 个 (相比较而言, FTP 超过 40 个)。我们用简单的例子说明发送邮件的工作过程, 并不仔细描述每个命令。

28.2.1 简单例子

我们将发送一个只有一行的简单邮件, 并观察 SMTP 连接。我们用 -v 标志调用用户代理,

它被传送给邮件传送代理（本例中是 Sendmail）。当设置该标志时，该 MTA 显示在 SMTP 连接上发送和接收的内容。以 >>> 开始的行是 SMTP 客户发出的命令，以 3 位数字的应答码开始的行是从 SMTP 服务器来的。以下就是交互会话：

```
sun % mail -v rstevens@noao.edu          调用我们的代理
To: rstevens@noao.edu                    这是用户代理的输出
Subject: testing                          然后指示我们键入主题
                                          用户代理在首部 and 正文之间加上一行空行
1, 2, 3.                                  这是我们键入的正文
.                                          我们在一行上输入一个句点，说明完成了
Sending letter ... rstevens@noao.edu...   用户代理上详细的输出

Connecting to mailhost via ether...        以下是MTA(Sendmail)的输出
Trying 140.252.1.54... connected.
220 noao.edu Sendmail 4.1/SAG-Noao.G89 ready at Mon, 19 Jul 93 12:47:34 MST

>>> HELO sun.tuc.noao.edu.
250 noao.edu Hello sun.tuc.noao.edu., pleased to meet you

>>> MAIL From:<rstevens@sun.tuc.noao.edu>
250 <rstevens@sun.tuc.noao.edu>... Sender ok

>>> RCPT To:<rstevens@noao.edu>
250 <rstevens@noao.edu>... Recipient ok

>>> DATA
354 Enter mail, end with "." on a line by itself

>>> .
250 Mail accepted

>>> QUIT
221 noao.edu delivering mail

rstevens@noao.edu... Sent
sent.                                     这是用户代理的输出
```

只有5个SMTP命令用于发送邮件：HELO，MAIL，RCTP，DATA和QUIT。

我们键入mail启动用户代理，然后键入主题（subject）的提示；键入后，再键入报文的正文。在一行上键入一个句点结束报文，用户代理把邮件传给MTA，由MTA进行交付。

客户主动打开TCP端口25。返回时，客户等待从服务器来的问候报文（应答代码为220）。该服务器的应答必须以服务器的完全合格的域名开始：本例中为noao.edu（通常，跟在数字应答后面的文字是可选的。这里需要域名。以Sendmail打头的文字是可选的）。

下一步客户用HELO命令标识自己。参数必须是完全合格的客户主机名：sun.tuc.noao.edu。

MAIL命令标识出报文的发起人。下一个命令，RCPT，标识接收方。如果有多个接收方，可以发多个RCPT命令。

邮件报文的内容由客户通过DATA命令发送。报文的末尾由客户指定，是只有一个句点的一行。最后的命令QUIT，结束邮件的交换。

图28-2是在发送方SMTP（客户端）与接收方SMTP（服务器）之间的一个SMTP连接。

我们键入到用户代理的数据是一行报文（“1, 2, 3”），但在报文段12中共发送了393字节的数据。下面的12行组成了客户发送的393字节数据：

```
Received: by sun.tuc.noao.edu. (4.1/SMI-4.1)
       id AA00502; Mon, 19 Jul 93 12:47:32 MST
Message-Id: <9307191947.AA00502@sun.tuc.noao.edu.>
From: rstevens@sun.tuc.noao.edu (Richard Stevens)
Date: Mon, 19 Jul 1993 12:47:31 -0700
```

```

Reply-To: rstevens@noao.edu
X-Phone: +1 602 676 1676
X-Mailer: Mail User's Shell (7.2.5 10/14/92)
To: rstevens@noao.edu
Subject: testing

```

1, 2, 3.

前三行, Received:和Message-Id: 由MTA加上; 下一行由用户代理生成。

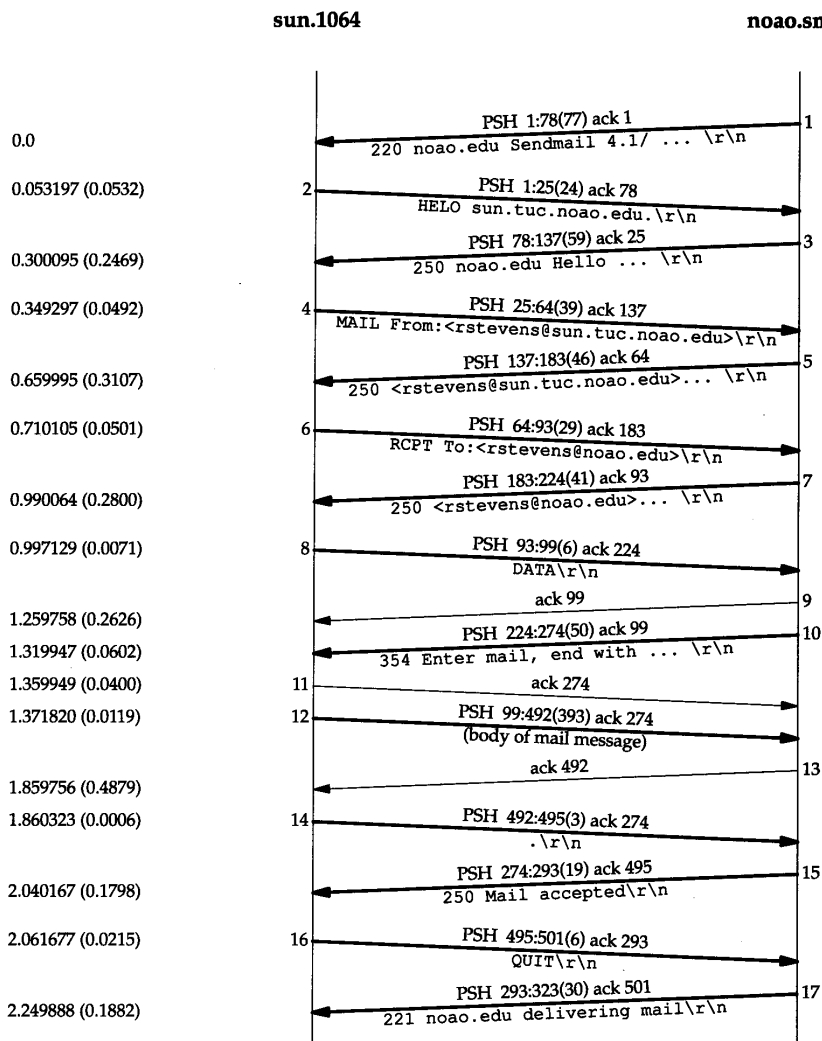


图28-2 基本SMTP邮件交付

28.2.2 SMTP命令

最小SMTP实现支持8种命令。我们在前面的例子中遇到 5个: HELO, MAIL, RCPT, DATA和QUIT。

RSET命令异常中止当前的邮件事务并使两端复位。丢掉所有有关发送方、接收方或邮件的存储信息。

VERFY命令使客户能够询问发送方以验证接收方地址，而无需向接收方发送邮件。通常是系统管理员在查找邮件交付差错时手工使用的。我们将在下一节中给出这方面的例子。

NOOP命令除了强迫服务器响应一个OK应答码（200）外，不做任何事情。

还有附加和可选命令。EXPN扩充邮件表，与VERFY类似，通常是由系统管理员使用的。事实上，许多Sendmail的版本都把这两者等价地处理。

4.4BSD 中的Sendmail版本8不再将两者等同处理。VERFY不扩充别名也不接受.forward文件。

TURN命令使客户和服务器交换角色，无需拆除TCP连接并建立新的连接就能以相反方向发送邮件（Sendmail不支持这个命令）。其他还有三个很少被实现的命令（SEND、SOML和SAML）取代MAIL命令。这三个命令允许邮件直接发送到客户终端（如果已注册）或发送到接收方的邮箱。

28.2.3 信封、首部和正文

电子邮件由三部分组成：

1) 信封（envelope）是MTA用来交付的。在我们的例子中信封由两个SMTP命令指明：

```
MAIL From: <rstevens@sun.tuc.noao.edu>
RCPT To: <estevens@noao.edu>
```

RFC 821指明了信封的内容及其解释，以及在一个TCP连接上用于交换邮件的协议。

2) 首部由用户代理使用。在我们的例子中可以看到 9个首部字段：Received、Message-Id、From、Data、Reply-To、X-Phone、X-Mailer、To和Subject。每个首部字段都包含一个名，紧跟一个冒号，接着是字段值。RFC 822指明了首部字段的格式的解释（以X-开始的首部字段是用户定义的字段，其他是由RFC 822定义的）。长首部字段，如例子中的Received，被折在几行中，多余行以空格开头。

3) 正文（body）是发送用户发给接收用户报文的内容。RFC 822 指定正文为NVT ASCII文字行。当用DATA命令发送时，先发送首部，紧跟一个空行，然后是正文。用DATA命令发送的各行都必须小于1000字节。

用户接收我们指定为正文的部分，加上一些首部字段，并把结果传到MTA。MTA加上一些首部字段，加上信封，并把结果发送到另一个MTA。

内容（content）通常用于描述首部和正文的结合。内容是用户用DATA命令发送的。

28.2.4 中继代理

在我们的例子中本地MTA的信息输出的第1行是：“Connecting to mailhost via ether”（即“通过以太网连接到邮件主机”）。这是因为作者的系统已被配置成把所有非本地的向外的邮件发送到一台中继机上进行转发。

这样做的原因有两个。首先，简化了除中继系统MTA外的其他所有MTA的配置（所有曾使用过Sendmail的人都能证明，配置一个MTA并不简单）。第二，它允许某个机构中的一个系统作为邮件集线器，从而可能把其他所有系统隐藏起来。

在这个例子中，中继系统在本地域（.tuc.noao.edu）中有一个mailhost的主机名，而其他所有系统都被配置成把它们的邮件发往该主机。我们可以执行host命令来看看在DNS中这个名

是如何定义的：

```
sun % host mailhost
mailhost.tuc.noao.edu      CNAME      noao.edu      规范名
noao.edu                   A        140.252.1.54  它的真实IP地址
```

如果将来用于中继的主机改变了，只需改变它的 DNS 名——其他所有单个系统的邮箱配置都无需改变。

目前许多机构都采用中继系统。图 28-3 是修改后的 Internet 邮件图（图 28-2），考虑发送主机和最后的接收主机都可能使用中继主机。

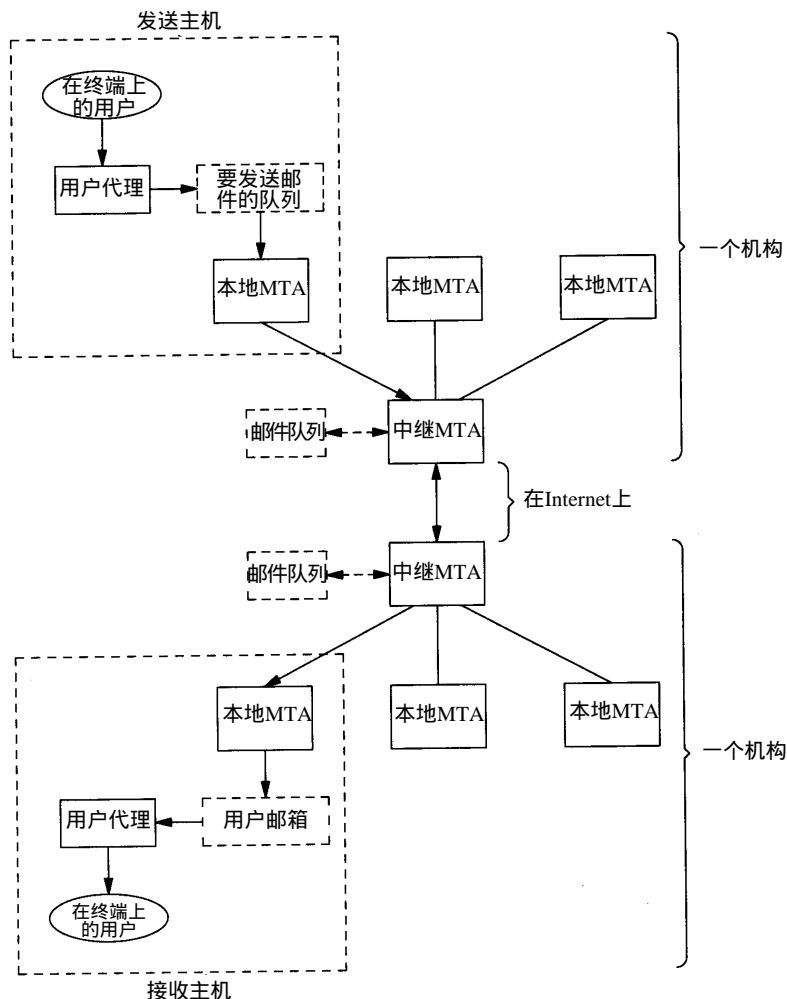


图28-3 在两端都有一个中继系统的Internet电子邮件

在这种情况下，在发送方和接收方之间有 4 个 MTA。发送方主机上的本地 MTA 只把邮件交给它自己的中继 MTA（该中继 MTA 可能在该机构的域中有一个 mailhost 的主机名）。这个通信就在该机构的本地互联网上用 SMTP。然后，发送方机构的中继 MTA 就在 Internet 上把邮件发送到接收方机构的中继 MTA 上，而这个中继 MTA 就通过与接收方主机上的本地 MTA 通信，把邮件交给接收方主机。尽管可能存在其他协议，但这个例子中所有 MTA 均使用 SMTP 协议。

28.2.5 NVT ASCII

SMTP的一个特色是它用NVT ASCII表示一切：信封、首部和正文。正如我们在 26.4节中谈到的，这是一个7 bit的字符码，以8 bit字节发送，高位比特被置为0。

在28.4节中，我们讨论了Internet邮件的一些新特性、允许发送和接收诸如音频和视频数据的扩充SMTP和多媒体邮件（MIME）。我们将看到，MIME和NVT ASCII一起表示信封、首部和正文，只需对用户代理作一些改变。

28.2.6 重试间隔

当用户把一个新的邮件报文传给它的MTA时，通常立即试图交付。如果交付失败，MTA必须把该报文放入队列中以后再重试。

Host Requirements RFC推荐初始时间间隔至少为30分钟。发送方至少4~5天内不能放弃。而且，因为交付失败通常是透明的（接收方崩溃或临时网络连接中断），所以当报文在队列中等待的第1个小时内，尝试两次连接是有意义的。

28.3 SMTP的例子

上面我们说明了普通邮件发送，在这里我们将说明MX记录如何用于邮件发送，以及VRFY和EXPN命令的用法。

28.3.1 MX记录：主机非直接连到Internet

在14.6节中我们提到DNS中的一种资源记录类型是邮件交换记录，称为MX记录。在下面的例子中我们将说明如何用MX记录向不直接连到Internet的主机发送邮件。RFC 974 [Partridge 1986] 描述了MTA对MX记录的处理。

主机mlfarm.com不是直接连到Internet的，但是有一个MX记录指向Internet上的一个邮件转发器。

```
sun % host -a -v -t mx mlfarm.com
The following answer is not authoritative:
mlfarm.com      86388  IN    MX    10 mercury.hsi.com
mlfarm.com      86388  IN    MX    15 hsi86.hsi.com
Additional information:
mercury.hsi.com 86388  IN    A     143.122.1.91
hsi86.hsi.com   172762 IN    A     143.122.1.6
```

有两个MX记录，各有不同的优先级。我们希望MTA从优先级数值低的开始。

```
sun % mail -v ron@mlfarm.com
To: ron@mlfarm.com
Subject: MX test message
```

-v标志看MTA在做什么

在这里键入报文的正文(没显示出来)

一行中的一个句号，结束报文

```
.
Sending letter ... ron@mlfarm.com...
Connecting to mlfarm.com via tcp...
mail exchanger is mercury.hsi.com
Trying 143.122.1.91... connected.
220 mercury.hsi.com ...
```

找到MX记录
先试优先级低的那一个

下面是正常的SMTP邮件传送

从输出中我们看到, MTA发现目的主机有一个MX记录, 并使用具有低优先级数值的MX记录。

在主机sun运行这个例子之前, 它被配置成不使用本地中继主机, 所以我们会看到与目的主机的邮件交换。主机sun还被配置成可使用主机noao.edu(通过拨号SLIP链路)上的域名服务器, 所以我们能使用tcpdump捕获在SLIP链路上进行的邮件发送和DNS通信。图28-4显示了tcpdump输出的开始部分。

```

1  0.0                sun.1624 > noao.edu.53: 2+ MX? mlfarm.com. (28)
2  0.445572 (0.4456)  noao.edu.53 > sun.1624: 2* 2/0/2 MX
                        mercury.hsi.com. 10 (113)

3  0.505739 (0.0602)  sun.1143 > mercury.hsi.com.25: S 1617536000:1617536000(0)
                        win 4096
4  0.985428 (0.4797)  mercury.hsi.com.25 > sun.1143: S 1832064000:1832064000(0)
                        ack 1617536001 win 16384
5  0.986003 (0.0006)  sun.1143 > mercury.hsi.com.25: . ack 1 win 4096
6  1.735360 (0.7494)  mercury.hsi.com.25 > sun.1143: P 1:90(89) ack 1 win 16384

```

图28-4 向一个使用MX记录的主机发送邮件

在第1行, MTA向它的域名服务器查询mlfarm.com的MX记录。跟在2后面的加号“+”意思是设置要求递归的标志位。第2行的响应置位授权比特(跟在2后面的星号“*”), 并包含两个回答RR(两个MX主机名), 0个授权RR, 以及两个附加的RR(两个主机的IP地址)。

第3~5行与主机mercury.hsi.com上的SMTP建立了一个TCP连接。服务器的初始响应220显示在第6行。

由于某种原因, 主机mercury.hsi.com必须把这个邮件报文交付给目的地, mlfarm.com。对于没有连接到Internet上与它的MX站点交换邮件的系统, UUCP协议是一种常用的办法。

在这个例子中, MTA要求一个MX记录, 得到一个肯定的结果, 然后发送邮件。但不幸的是, MTA与DNS之间的交互随不同的实现而不同。RFC 974指定MTA必须首先要求MX记录, 如果没有, 就尝试提交给目的主机(也就是说, 向DNS要主机的记录和IP地址)。MTA也必须处理DNS中的CNAM记录(规范的名)。

作为一个例子, 如果我们从一个BSD/386主机上向rstevens@mailhost.tuc.noao.edu发送邮件, 则MTA(Sendmail)执行以下步骤:

1) Sendmail向DNS询问主机mailhost.tuc.noao.edu的CNAME记录。我们看到存在一个CNAME记录:

```

sun % host -t cname mailhost.tuc.noao.edu
mailhost.tuc.noao.edu    CNAME    noao.edu

```

2) 发布一个要求noao.edu的CNAME记录的DNS查询, 回答是不存在。

3) Sendmail向DNS寻求noao.edu的MX记录并得到一个记录:

```

sun % host -t mx noao.edu
noao.edu                MX      noao.edu

```

4) Sendmail向DNS查询noao.edu的A记录(IP地址), 并得到返回值140.252.1.54(这个A记录大概是由域名服务器为noao.edu返回的, 作为第3步中MX应答的一个附加的RR)。

5) 启动一个到140.252.1.54的SMTP连接并发送邮件。

CNAME查询不是为MX记录(noao.edu)中返回的数据做的。MX记录中的数据不能是

别名——必须是具有一个A记录的主机名。

与只用DNS的SunOS 4.1.3一起发布的Sendmail版本查询MX记录，并且如果没有找到MX记录就放弃。

28.3.2 MX记录：主机出故障

MX记录的另一个用途是在目的主机出故障时可提供另一个邮件接收器。如果看一下主机sun的DNS入口，我们就会看到它有两个MX记录：

```
sun % host -a -v -t mx sun.tuc.noao.edu
sun.tuc.noao.edu      86400      IN      MX      0 sun.tuc.noao.edu
sun.tuc.noao.edu      86400      IN      MX      10 noao.edu
Additional information:
sun.tuc.noao.edu      86400      IN      A       140.252.1.29
sun.tuc.noao.edu      86400      IN      A       140.252.13.33
noao.edu              86400      IN      A       140.252.1.54
```

最低优先级的MX记录表明应该首先尝试直接发送到主机本身，下一个优先级是把邮件发送到主机noao.edu。

在下面的描述中，在关掉目的SMTP服务器后，我们从主机vangogh.cs.berkeley.edu向位于主机sun.tuc.noao.edu的我们自己发送邮件。当端口25上的连接请求到达时，TCP应该响应一个RST，因为没有被动打开的进程为等待该端口而挂起。

```
vangogh % mail -v rstevens@sun.tuc.noao.edu
A test to a host that's down.
.
EOT
rstevens@sun.tuc.noao.edu... Connecting to sun.tuc.noao.edu. (smtp)...
rstevens@sun.tuc.noao.edu... Connecting to noao.edu. (smtp)...
220 noao.edu ...
```

下面是正常的SMTP邮件传送

我们看到MTA尝试联系sun.tuc.noao.edu，然后放弃，并转而联系noao.edu。图28-5显示了TCP用一个RST向到来的SYN响应的tcpdump输出。

```
1 0.0          vangogh.3873 > 140.252.1.29.25: S 2358303745:2358303745(0) ...
2 0.000621 (0.0006) 140.252.1.29.25 > vangogh.3873: R 0:0(0) ack 2358303746 win 0
3 0.300203 (0.2996) vangogh.3874 > 140.252.13.33.25: S 2358367745:2358367745(0) ...
4 0.300620 (0.0004) 140.252.13.33.25 > vangogh.3874: R 0:0(0) ack 2358367746 win 0
```

图28-5 尝试连接一个不在运行的SMTP服务器

第1行vangogh向sun的第1个IP地址140.252.1.29的端口25发送一个SYN。在第2行它被拒绝。然后，vangogh上的SMTP客户尝试sun的第2个IP地址140.252.13.33（第3行），也产生一个RST的返回（第4行）。

SMTP客户不区分第1行它主动打开时所返回的不同差错，而这是导致它在第2行尝试其他IP地址的原因。如果第1次的差错是类似“host unreachable(主机不可达)”，那么第2次尝试或许可行。

如果SMTP客户的主动打开失败的原因是因为服务器主机出故障了，我们将看到客户会向IP地址140.252.1.29重传SYN总共75秒（类似于图18-6）。然后客户向IP地址140.252.13.33发送另一个75秒的其他3个SYN。150秒后客户会移到下一个具有更高优先级的MX记录。

28.3.3 VRFY和EXPN命令

VRFY命令无需发送邮件而验证某个接收方地址是否 OK。EXPN的目的是无需向邮件表发送邮件就可以扩充该表。许多 SMTP实现（如 Sendmail）把两者看成一个，但我们提到新的 Sendmail区分这两者。

作为一个简单测试，我们可以连到一个新的 Sendmail版本，并看到不同之处（已经删除了无关的 Telnet 客户输出）。

```
sun % telnet vangogh.cs.berkeley.edu 25
220-vangogh.CS.Berkeley.EDU Sendmail 8.1C/6.32 ready at Tue, 3 Aug 1993 14:
59:12 -0700
220 ESMTP spoken here

helo bsdi.tuc.noao.edu
250 vangogh.CS.Berkeley.EDU Hello sun.tuc.noao.edu [140.252.1.29], pleased
to meet you

vrfy nosuchname
550 nosuchname... User unknown

vrfy rstevens
250 Richard Stevens <rstevens@vangogh.CS.Berkeley.EDU>

expn rstevens
250 Richard Stevens <rstevens@noao.edu>
```

首先注意到我们故意在 HELO 命令中键入错误的主机名：bsdi，而不是 sun。许多 SMTP 服务器得到客户的 IP 地址，完成一个 DNS 指针查询（14.5 节）并比较主机名。这样允许服务器基于 IP 地址注册到客户的连接，而不是基于用户可能错误键入的名。某些服务器会用幽默的报文回答，如“你是一个骗子”，或“为什么叫你自己……”。在这个例子中我们看到，这个服务器通过指针查询只打印出我们的真实域名以及我们的 IP 地址。

然后我们用一个无效的名字键入 VRFY 命令，服务器就响应 550 差错。下一步我们键入一个有效的名字，服务器用本地主机上的用户名回答。然后我们试试 EXPN 命令，并得到一个不同的回答。EXPN 命令决定到该用户的邮件是否被转发，并打印出转发的地址。

许多站点禁止 VRFY 和 EXPN 命令，有时是因为隐私，有时因为相信这是安全漏洞。例如，我们可以向白宫的 SMTP 服务器试试下面的命令：

```
sun % telnet whitehouse.gov 25
220 whitehouse.gov SMTP/smmap Ready.

helo sun.tuc.noao.edu
250 (sun.tuc.noao.edu) pleased to meet you.

vrfy clinton
500 Command unrecognized

expn clinton
500 Command unrecognized
```

28.4 SMTP 的未来

Internet 邮件发生了很多改变。应当记得 Internet 邮件的三个组成部分：信封、首部和正文。新加入的 SMTP 命令影响了信封，首部中可以使用非 ASCII 字母，正文（MIME）中也加入了结构。本节中我们依次对这三部分的扩充进行讨论。

28.4.1 信封的变化：扩充的SMTP

RFC 1425 [Klensin等, 1993a] 定义了扩充的SMTP的框架, 其结果被称为扩充的 SMTP (ESMTP)。与其他我们已经讨论过的新特性一样, 这些变化以向后兼容的方式被加入, 所以不影响已有的实现。

如果客户想使用新的特性, 首先通过发布一个 EHLO而不是HELO命令启动一个与服务器的会话。相兼容的服务器用 250应答码响应。这个应答通常有好几行, 每行都包含一个关键字和一个可选的参数。这些关键字指定了该服务器支持的 SMTP扩充。新的扩充将在一个RFC中描述并以IANA注册 (在一个多行应答中, 各行数字应答码的后面都要有一个连字符。最后一行的数字应答码后面跟一个空行)。

我们将给出到4个SMTP服务器的初始连接, 其中3个支持扩充的SMTP。我们用Telnet和它们连接, 但删掉了不必要的Telnet客户输出。

```
sun % telnet vangogh.cs.berkeley.edu 25
220-vangogh.CS.Berkeley.EDU Sendmail 8.1C/6.32 ready at Mon, 2 Aug 1993 15:
47:48 -0700
220 ESMTP spoken here

ehlo sun.tuc.noao.edu
250-vangogh.CS.Berkeley.EDU Hello sun.tuc.noao.edu [140.252.1.29], pleased
to meet you
250-EXPN
250-SIZE
250 HELP
```

这个服务器用一个多行220应答作为它的欢迎报文。对EHLO命令的250应答中列出的扩充命令是EXPN、SIZE和HELP。第一个和最后一个来自原来的RFC 821规范, 但它们是可选命令。ESMTP服务器说明除了新命令外, 它们还支持哪些可选的RFC 821命令。

这个服务器支持的SIZE关键字是在RFC 1427 [Klensin, Freed和Moore 1993] 中定义的。它让客户在MAIL FROM命令行中以字节的多少指定报文的大小, 这样服务器就可以在客户开始发送该报文之前, 验证它是否接收该长度的报文。增加这个命令的原因在于, 随着对非ASCII码 (如图像、音频等) 内容的支持, Internet邮件报文的长度在不断增大。

下一个主机也支持ESMTP, 注意250应答指明支持包含一个可选参数的SIZE关键字。这表明该服务器将接受长度不超过461兆字节的报文。

```
sun % telnet ymir.claremont.edu 25
220 ymir.claremont.edu -- Server SMTP (PMDf V4.2-13 #4220)

ehlo sun.tuc.noao.edu
250-ymir.claremont.edu
250-8BITMIME
250-EXPN
250-HELP
250-XADR
250 SIZE 461544960
```

关键字8BITMIME来自于RFC 1426 [Klensin等, 1993a]。它允许客户把关键字BODY加到MAIL FROM命令中, 指定正文中是否包含NVT ASCII字符 (默认的) 或8 bit数据。除非客户收到服务器应答EHLO命令发来的8BITMIME关键字, 否则禁止客户发送任何非NVT ASCII字符 (当我们在本节中谈到MIME时, 我们将看到MIME不要求8 bit传送)。

该服务器也通告了XADR关键字。任何以X开头的关键字都指的是本地SMTP扩充。

另一个服务器也支持 ESMTP, 通知了我们已经看到的 HELP 和 SIZE 关键字。它也支持三个以 X 开头的本地扩充。

```
sun % telnet dbc.mtview.ca.us 25
220 dbc.mtview.ca.us Sendmail 5.65/3.1.090690, it's Mon, 2 Aug 93 15:48:50
-0700
```

```
ehlo sun.tuc.noao.edu
250-Hello sun.tuc.noao.edu, pleased to meet you
250-HELP
250-SIZE
250-XONE
250-XVRB
250 XQUE
```

最后, 我们将看到当客户试图通过向一个不支持 EHLO 的服务器发布 EHLO 命令来使用 ESMTP 时将发生什么。

```
sun % telnet relay1.uu.net 25
220 relay1.UU.NET Sendmail 5.61/UUNET-internet-primary ready at Mon, 2 Aug
93 18:50:27 -0400
```

```
ehlo sun.tuc.noao.edu
500 Command unrecognized
```

```
rset
250 Reset state
```

对 EHLO 命令, 客户收到一个 500 应答而不是 250 应答。客户应发布 RSET 命令, 并跟着一个 HELO 命令。

28.4.2 首部变化: 非ASCII字符

RFC 1522 [Moore 1993] 指明了一个在 RFC 822 报文首部中如何发送非 ASCII 字符的方法。这样做的主要用途是为了允许在发送方名、接收方名以及主题中使用其他的字符。

首部字段中可以包含编码字 (coded word)。它们具有以下格式:

`=?charset?encoding?encoded-text?=`

charset 是字符集规范。有效值是两个字符串 us-ascii 和 iso-8859-x, 其中 x 是一个单个数字, 例如在 iso-8859-1 中的数字 “1”。

encoding 是一个单个字符用来指定编码方法, 支持两个值。

1) Q 编码意思是引号中可打印的 (quoted-printable), 目的是用于拉丁字符集。大多数字符是作为 NVT ASCII (当然最高位比特置 0) 发送的。任何要发送的字符若其第 8 比特置 1 则被作为 3 个字符发送: 第 1 个是字符是 “=”, 跟着两个十六进制数。例如, 字符 é (它的二进制 8 bit 值为 0xe9) 作为三个字符发送: =E9。空格通常作为下划线或三个字符 =20 发送。这种编码的目的在于, 某些文本中除了大多数 ASCII 字符外, 还有几个特殊字符。

2) B 意思是以 64 为基数的编码。文本中的 3 个连续字节 (24 bit) 被编码成 4 个 6 bit 值。用于表示所有可能的 6 bit 值的 64 个 NVT ASCII 字符如图 28-6 所示。当要编码的个数不是 3 的倍数时, 等号 “=” 被用作填充符。

下面两种编码方式的例子取自 RFC 1522:

```
From: =?US-ASCII?Q?Keith_Moore?= <moore@cs.utk.edu>
To: =?ISO-8859-1?Q?Keld_J=F8rn_Simonsen?= <keld@dkuug.dk>
CC: =?ISO-8859-1?Q?Andr=E9_?= Pirard <PIRARD@vml.ulg.ac.be>
Subject: =?ISO-8859-1?B?SWYgeW91IGNhbiByZWZkIHRobXMgeW8=?=
=?ISO-8859-2?B?dSB1bmR1cnN0YW5kIHRobZSBleGFtcGxlLg==?=
```

6 bit 值	ASCII 字符	6 bit 值	ASCII 字符	6 bit 值	ASCII 字符	6 bit 值	ASCII 字符
0	A	10	Q	20	g	30	w
1	B	11	R	21	h	31	x
2	C	12	S	22	i	32	y
3	D	13	T	23	j	33	z
4	E	14	U	24	k	34	0
5	F	15	V	25	l	35	1
6	G	16	W	26	m	36	2
7	H	17	X	27	n	37	3
8	I	18	Y	28	o	38	4
9	J	19	Z	29	p	39	5
a	K	1a	a	2a	q	3a	6
b	L	1b	b	2b	r	3b	7
c	M	1c	c	2c	s	3c	8
d	N	1d	d	2d	t	3d	9
e	O	1e	e	2e	u	3e	+
f	P	1f	f	2f	v	3f	/

图28-6 6 bit值的编码（以64为基数编码）

能处理这些首部的用户代理将输出：

```
From: Keith Moore <moore@cs.utk.edu>
To: Keld Jørn Simonsen <keld@dkuug.dk>
CC: André Pirard <PIRARD@vml.ulg.ac.be>
Subject: If you can read this you understand the example.
```

为说明以64为基数的编码方法是如何工作的，我们看一下主题行中前面 4个编码的字符：SWYg。按照图28-6写出这4个字符的6 bit值（S=0x12,W=0x16,Y=0x18以及g=0x20）的二进制码：

```
010010 010110 011000 100000
```

然后把这24 bit重新分成3个8 bit字节：

```
01001001 01100110 00100000
=0x49      =0x66      =0x20
```

它们是I、f和空格的ASCII表示。

28.4.3 正文变化：通用Internet邮件扩充

我们已经提到RFC 822指定正文是NVT ASCII文本行，没有结构。RFC 1521 [Borenstein和Freed 1993] 把扩充定义为允许把结构置入正文。这被称为 MIME，即通用Internet邮件扩充。

MIME不要求任何扩充，我们在本节前面已作了说明（扩充的 SMTP或非ASCII标题）。MIME正好加入了一些告知收件者正文结构的新标题（与 RFC 822相一致）。正文仍可以用NVT ASCII码来发送，而不考虑邮件内容。虽然我们前面所述的一些扩充可能会和 MIME合在一起产生好的效果——扩充的SMTP SIZE命令，因为MIME报文能变得很长，以及非ASCII标题——这些扩充并不是MIME所要求的。与另一方交换MIME报文所需的一切，就是双方都要有一个能够理解MIME的用户代理。在任何一个MTA中不需要做任何改变。

MIME定义这5个新标题字段如下：

```
Mime-Version:
Content-Type:
Content-Transfer-Encoding:
```

Content-ID:

Content-Description:

作为例子, 下面两个标题行可以出现在一个 Internet 邮件报文中:

Mime-Version: 1.0

Content-Type: TEXT/PLAIN; charset=US-ASCII

当前 MIME 版本是 1.0, 内容类型是无格式 ASCII 码文本, 即 Internet 邮件的默认选择。PLAIN 这个字被认为是内容类型 (TEXT) 的一个子类型, 字符串 charset=US-ASCII 是一个参数。

Text 是 MIME 的 7 个被定义的内容类型之一。图 28-7 总结了 RFC 1521 中定义的 16 个不同的内容类型和子类型。对具体的内容类型和子类型来说都有指定的很多参数。

内容类型	子类型	描 述
text	plain	无格式文本
	richtext	简单格式文本, 如粗体、斜体或下划线等
	enriched	richtext 的简化和改进
multipart	mixed	多个正文部分, 串行处理
	parallel	多个正文部分, 可并行处理
	digest	一个电子邮件的摘要
	alternative	多个正文部分, 具有相同的语义内容
message	rfc822	内容是另一个 RFC 822 邮件报文
	partial	内容是一个邮件报文的片断
	external-body	内容是指向实际报文的指针
application	octet-stream	任意二进制数据
	postscript	一个 PostScript 程序
image	jpeg	ISO 10918 格式
	gif	CompuServe 的图形交换格式
audio	basic	用 8 bit ISDN μ 律格式编码
video	mpeg	ISO 11172 格式

图28-7 MIME 内容类型和子类型

内容类型和用于内容的传送编码是相互独立的。前者由首部字段 Content-Type 指明, 后者由首部字段 Content-Transfer-Encoding 指明。在 RFC 1521 中定义了 5 种不同的编码格式。

1) 7bit, 是默认的 NVT ASCII;

2) quoted-printable, 我们在前面的一个例子中看到有非 ASCII 首部。当字符中只有很少一部分的第 8 bit 置 1 时非常有用;

3) base64, 如图 28-6 所示;

4) 8bit, 包含字符行, 其中某些为非 ASCII 字符且第 8bit 置 1;

5) binary 编码, 无需包含多行的 8 bit 数据。

对 RFC 821 MTA, 以上 5 种编码格式中只有前 3 种是有效的。因为这 3 种产生只包含 NVT ASCII 字符的正文。使用有 8BITMIME 支持的扩充 SMTP 允许使用 8bit 编码。

尽管内容类型和编码是独立的, RFC 1521 推荐有非 ASCII 数据的 text 使用 quoted-printable, 而 image、audio、video 和 octet-stream application 使用 base64。这样允许与符合 RFC 821 的 MTA 保持最大的互操作性。而且, multipart 和 message 内容类型必须以 7bit 编码。

作为一个multipart内容类型的例子，图 28-8显示了一个来自RFC发布清单的邮件报文。子类型是mixed，意思是各部分是顺序处理的，各部分的边界是字符串 NextPart，其前面是行首的两个连字符。

每个边界上可跟一行用于指明下一部分首部字段。忽略报文中第 1个边界之前和最后一个边界之后的所有内容。

因为在第一个边界后面跟着一个空行，而不是首部，所以在第 1个和第2个边界之间的数据的内容类型被假定为具有us-ascii字符集的text/plain。这是新RFC的文字描述。

但是第2个边界后面跟着首部字段。它指定了另一个multipart报文，具有边界OtherAccess。子类型为alternative，有两种不同的选择。第1种OtherAccess选项是用电子邮件获取RFC，第2种选项是用匿名FTP获取。MIME用户代理将列出这两种选项，允许我们选择一个，然后自动地用电子邮件或匿名FTP获取一份复制的RFC。

```
To: rfc-dist@nic.ddn.mil
Subject: RFC1479 on IDPR Protocol
Mime-Version: 1.0
Content-Type: Multipart/Mixed; Boundary="NextPart"
Date: Fri, 23 Jul 93 12:17:43 PDT
From: "Joyce K. Reynolds" <jkrey@isi.edu>
```

--NextPart

第1个边界

A new Request for Comments is now available in online RFC libraries.

. . .

这里的细节在新的RFC中

Below is the data which will enable a MIME compliant Mail Reader implementation to automatically retrieve the ASCII version of the RFCs.

--NextPart

第2个边界

Content-Type: Multipart/Alternative; Boundary="OtherAccess"

一个具有新边界的嵌套的多部分报文

--OtherAccess

```
Content-Type: Message/External-body;
    access-type="mail-server";
    server="mail-server@nisc.sri.com"
```

Content-Type: text/plain

SEND rfc1479.txt

--OtherAccess

```
Content-Type: Message/External-body;
    name="rfc1479.txt";
    site="ds.internic.net";
    access-type="anon-ftp";
    directory="rfc"
```

Content-Type: text/plain

--OtherAccess--

--NextPart--

最后的边界

图28-8 MIME multipart报文的例子

这一部分是 MIME 的一个简要概述。MIME 的详细细节和例子, 见 RFC 1521 和[Rose 1993]。

28.5 小结

电子邮件包括在两端(发送方和接收方)都有的一个用户代理以及两个或多个报文传送代理。可以把一个邮件报文分成三个部分: 信封、首部和正文。我们已经看到这三个部分用 SMTP 和 Internet 标准是如何进行交换的。所有都作为 NVT ASCII 字符进行交换。

我们也看到了一些新的扩充: 用于信封和非 ASCII 首部的扩充 SMTP, 以及使用 MIME 的正文增加了结构。MIME 的结构和编码允许使用已有的 7bit SMTP MTA 交换任意二进制数据。

习题

- 28.1 读 RFC 822, 找到域文字 (domain literal) 的意思。试试用其中一个给自己发送邮件。
- 28.2 除了连接建立和终止外, 要发送一个小的邮件报文的最小网络往返次数是多少?
- 28.3 TCP 是一个全双工协议, 但是 SMTP 用半双工的形式使用 TCP。客户发送一个命令后停止等待应答。为什么客户不一次发送多个命令, 如一行中包括 HELO、MAIL、RCPT、DATA 和 QUIT 命令 (假定正文不是太大)?
- 28.4 当网络在接近其容量运行时, SMTP 的这种半双工操作如何欺骗缓慢的启动机制?
- 28.5 当存在多个具有相同优先值的 MX 记录时, 名服务器是否总能以相同的顺序返回它们?