



图 2-17 Alice 向 Bob 发送一条报文

我们现在仔细观察一下，SMTP 是如何将一个报文从发送邮件服务器传送到接收邮件服务器的。我们将看到，SMTP 与人类面对面交往的行为方式有许多类似性。首先，客户 SMTP（运行在发送邮件服务器上）在 25 号端口建立一个到服务器 SMTP（运行在接收邮件服务器上）的 TCP 连接。如果服务器没有开机，客户会在稍后继续尝试连接。一旦连接建立，服务器和客户执行某些应用层的握手，就像人们在互相交流前先进行自我介绍一样。SMTP 的客户和服务器在传输信息前先相互介绍。在 SMTP 握手的阶段，SMTP 客户指示发送方的邮件地址（产生报文的那个人）和接收方的邮件地址。一旦该 SMTP 客户和服务端彼此介绍之后，客户发送该报文。SMTP 能依赖 TCP 提供的可靠数据传输无差错地将邮件投递到接收服务器。该客户如果有另外的报文要发送到该服务器，就在该相同的 TCP 连接上重复这种处理；否则，它指示 TCP 关闭连接。

接下来我们分析一个在 SMTP 客户（C）和 SMTP 服务器（S）之间交换报文脚本的例子。客户的主机名为 crepes.fr，服务器的主机名为 hamburger.edu。以 C：开头的 ASCII 码文本行正是客户交给其 TCP 套接字的那些行，以 S：开头的 ASCII 码则是服务器发送给其 TCP 套接字的那些行。一旦创建了 TCP 连接，就开始了下列过程。

```

S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr ... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection

```

在上例中，客户从邮件服务器 crepes.fr 向邮件服务器 hamburger.edu 发送了一个报文（“Do you like ketchup? How about pickles?”）。作为对话的一部分，该客户发送了 5 条命令：HELO（是 HELLO 的缩写）、MAIL FROM、RCPT TO、DATA 以及 QUIT。这些命令都是自解释的。该客户通过发送一个只包含一个句点的行，向服务器指示该报文结束了。（按照 ASCII 码的表示方法，每个报文以 CRLF 结束，其中的 CR 和 LF 分别表示回车和换行。）服务器对每条命令做出回答，其中每个回答含有一个回答码和一些（可选