

## 第17章 NNTP简介

由于采用不同的网络传输，NNTP提供了多种C-News新闻交换方法。NNTP代表的是“网络新闻传输协议”，它不是一个特殊的软件包，而是一种因特网标准，是在RFC-977在正式定义的。它以面向流的连接（通常为TCP连接）为基础，这种连接的一端是网络中处于任何一个地方的客户机，另一端是其磁盘上保存netnews（网络新闻）的主机上的服务器。流式连接允许客户机和服务器几乎没有任何延迟地交互式协商文章传输，这样可尽量避免文章的雷同。有了因特网的高传输速率，新闻传输较之前的UUCP网络更为快捷。虽然若干年前，一篇文章抵达Usenet的最后一个站点需要两天甚至更长的时间是常有的事，但现在通常只须花不到两天的时间；而在因特网上，甚至可在数分钟之内就能完成。

许多命令允许客户机恢复、发送或投递文章。发送和投递之间的区别在于：后者还涉及到头信息不完整的文章（在NNTP上投递文章时，服务器始终会加上至少一个头字段，那就是Nntp-Posting-Host:。其中包含客户机的主机名）。文章的恢复可供新闻传输客户机和新闻读者使用。这些都令NNTP成为一个出色的工具，在使用NFS时，无须通过必要的中转，就可为本地网络上的许多客户机提供新闻访问。

NNTP还提供了主动和被动新闻传输程序，通称“推”和“拉”。“推”程序和C-News ihave/sendme协议基本一样。客户机通过ihave <varmsgid>命令，为服务器提供文章，而服务器返回相应的响应代码，指出自己是否已有这篇文章，或者说自己是否想收到这篇文章。如果是，客户机就将该文章发送出去，在一个单独的行上，以一个句点结束。

“推”新闻的不足之处在于：它会为服务器系统带来较大的工作量，因为它必须在自己的历史数据库内搜索每一篇独立的文章。

相反的技术是“拉”新闻。在这种情况下，客户机从一个组请求获得指定日期之后抵达的所有文章的一个列表。这一查询是由NEWNEWS命令来执行的。从消息ID的返回列表中，客户机选择那些自己还没有的文章（依次为每篇新文章使用ARTICLE命令）。

另外还有大量便利的命令可供新闻阅读程序利用，使其能分别接收文章标题和文章主体，甚至从指定范围的文章中，取得每一个标题行。这样一来，我们可就全部新闻（文章）都保存在一台中心主机上。同时，整个网络中的所有用户都利用基于NNTP的客户机程序，读取及发表文章。另外，也可以通过NFS导出新闻目录，具体做法已在第16章详细介绍过了。

对NNTP来说，它存在的一个重要问题在于：某些头脑发达的人士，可以使用虚拟的发送人信息，将文章插入新闻流中。我们将这称为“新闻欺骗”（SMTP——简单邮件传输协议——也存在同样的问题）。然而，通过对NNTP进行一定程度的扩展，便可针对特定的命令，要求证明用户的真实身份。

目前，大家可以找到大量NNTP的变种。其中最著名的恐怕要算NNTP Daemon，亦称作“参考NNTP”。它最初是由Stan Barber和Phil Lapsley编写的，用于阐释RFC-977的细节。其最新的版本是nntpd 1.5.11，后面会以它作为范例详细讲述。此外，大家能拿到它的源码，并可自行编译，或使用由Fred van Kempen的net-std二进制封装提供的nntpd。注意nntpd并没有现

成的二进制文件可供直接使用，这是由于存在大量与具体站点对应的值，必须自行编译。

注意 自本书完稿以后，许多情况都发生了变化。后文提到的 INN封装目前正由ISC（Internet软件协会）负责维护，并被公布为最新的“参考”版本。INN的最新版本是2.2，请访问下述网址了解进一步的情况：<http://www.isc.org/view.cgi?/products/INN/index.phtml>。

在nntpd封装中，包含了一个服务器模块，以及两个客户机模块，分别用于新闻的“拉”与“推”。此外，还有一个用于取代inews的模块。它们要求在一个Bnews环境中使用，但只需作少许调节，就能很好地兼容于C-News环境。但是，假如你计划使用NNTP，让新闻阅读程序访问自己的新闻服务器，那么参考版本其实并不是一个很好的选择。因此，我们将只讨论包含在nntpd封装中的NNTP Daemon；至于具体的客户机程序，则暂时不作讨论。

注意 另外还有一个名为“InterNet News”的封装，简称为INN，由Rich Salz编写。它同时具有NNTP和UUCP新闻传输功能，而且对那些大型的新闻交换中心来说，显得更加稳定。通过NNTP传输新闻消息时，这个版本要明显优于nntpd。目前，INN的版本号是inn-1.4sec。Arjan de Vet为我们提供了一个工具包，专门用来在一台机器上构建INN；你可利用FTP从sunsite.unc.edu网站下载，目录是system/Mail。假如想设置INN，请务必参阅与源码配套提供的说明文档。同时，还应仔细阅读网上公布的INN-FAQ，网址是news.software.b和news.software.inn。

## 17.1 NNTP 服务器的安装

NNTP服务器的名字叫作nntpd，可通过两种方式进行编译。至于具体采用哪种方式，取决于新闻系统预计的负载有多大！注意目前找不到一个现成的、编译好的版本，因为在执行程序中，以“硬编码”的方式，强行设置了许多各个站点都不尽相同的默认选项。所有配置都是通过common/conf.h内的宏定义来完成的。

可考虑将nntpd配置成一个独立运行的服务器，令其在系统引导的时候，自rc.inet2中启动。另外，亦可考虑将其配置成由inetd负责管理。如果是后一种情况，必须在/etc/inetd.conf中设置下述条目：

```
nntp stream tcp nowait news /usr/etc/in.nntpd nntpd
```

假如选择前一种情况，将nntpd配置成独立运行的，那么务必将inetd.conf中的每一行都标注为注释。但无论在哪种情况下，都必须在/etc/services中设置下面这一行：

```
nntp 119 / tcp readnews untp # 网络新闻传输协议
```

为临时保存进入的文章等信息，nntpd还需要在你的新闻spool中，设置一个.tmp目录。创建方法如下：

```
# mkdir /var/spool/news/.tmp  
# chown news.news /var/spool/news/.tmp
```

## 17.2 限制NNTP访问权限

对NNTP资源的访问要受nntp\_access文件的约束，该文件保存在/usr/lib/news目录下。文件中的一系列行描述了授予外部主机的访问权限。每一行都必须采用下述格式：

```
site read | xfer | both | no post | no [!exceptgroups]
```

假如一个客户机连接到 NNTP 端口，nntpd 就会根据其 IP 地址，试着“逆向搜索”出它的完整形式的域名。根据每个条目的设置，按照它们在文件中出现的顺序，会依次检查客户机的主机名及 IP 地址是否相符。这种“相符”既可以是部分的，也可以是完全的。假如一个条目完全相符，便立即接纳它；假如仅部分相符，那么只有在以后再找不出其他相符的条目时，才会接纳它。可用下述几种形式来指定一个具体的站点：

**主机名**——这是一个完整形式的主机域名。如果从字义上讲，它与客户机的规范主机名相符时，就应用这个条目，随后的所有条目都被忽略。

**IP 地址**——这是一个采用点分四段式表达式的 IP 地址。如果客户机的 IP 地址与此相符，就采用这个条目，随后的条目全部忽略。

**域名**——这是一个域名，被指定为 \*.domain。如果客户机的主机名与该域名相符，就说明这个条目是匹配的。

**网络名**——这是 /etc/network 中指定的网络名。如果客户机 IP 地址中的网络编号和与这个网络名关联的网络编号相符，就说明这个条目是匹配的。

**默认**——默认与任何客户机匹配。

带有更多常见站点规格的条目应该在此之前指定，因为任何与之相符的条目都被后来的更与之相符的条目改写。

第二和第三个字段描述了被授予客户机的访问权限。第二个字段详细指出，允许利用“推”(read)恢复新闻，利用“拉”(xfer)传输新闻。采用 both 值，将启用“推”和“拉”，同时无访问拒绝。第三个字段授予客户机投递文章权，也就是说，投递其文章头信息不完整的文章，由新闻软件将其补充完整。如果第二个字段内有 no，第三个字段就会被忽略。

第四个字段是可选的，其中包含一个新闻组列表，中间用逗号隔开，客户机被拒绝对这些新闻组进行访问。

nntp\_access 文件示例如下：

```
#
# by default, anyone may transfer news, but not read or post
default          xfer          no
#
# public.vbrew.com offers public access via modem, we allow
# them to read and post to any but the local.* groups
public.vbrew.com  read          post    !local
#
# all other hosts at the brewery may read and post
*.vbrew.com      read          post
```

### 17.3 NNTP 身份验证

在用大写字母表示 nntp\_access 中的诸如 xfer 和 read 之类的访问特征时，针对不同的操作，nntpd 会要求客户机证明自己的身份。例如，在指定 Xfer 或 XFER 访问许可时，nntpd 不会让客户机向你的站点传输文章，除非它通过了身份验证。

身份验证进程是通过一个新的 NNTP 命令来实施的，该命令名为 AUTHINFO。使用该命

令时，客户机把用户名和密码传递给 NNTP服务器。nntpd把它们和/etc/passwd数据库内保存的进行核对，从而证实它们的身份并证实该用户属于 nntp组。

目前的NNTP身份验证实施尚处于实验阶段，所以其通用性不强。其结果只能适用于纯文本格式的密码数据库；还不能识别影子密码。

## 17.4 nntpd与C-News的沟通

在接收新闻组文章时，nntpd必须将其投递到新闻子系统。再根据该文章是否作为 IHAVE或POST命令结果被接收，分别将它上交给 rnews或inews命令。你可以不调用rnews，而是在编辑时，把它配置为“对进入文章进行批处理”，并将所有批处理文件移到/var/spool/news/in.coming，这些批处理文件便等待relaynews在下次运行时对它们进行处理。

为了能正确执行 ihave /sendme协议，nntpd必须能够访问历史文件。因此，你在编辑时，必须知道确定路径的设置是完全正确的。此外，还应该确定 C-News和nntpd商定了历史文件的格式。C-News利用dbm散列函数来访问它；但是，dbm库也有许多不同的，而且有些互不兼容的实施方案。如果C-News已经链接的dbm库和你的标准libc内的不同，你只好也用这个库来链接nntpd。

nntpd和C-News在数据库格式上的不一致，主要表现在 nntpd打不开系统日志内的错误消息，或通过 NNTP收到重复文件。最好的测验是从你的假脱机区内选出一篇文章，telnet到nntp端口，并将它提供给nntpd，如下所示。当然，你必须用自己准备发送给nntpd的文章之ID来替换 < msg@id >。

```
$ telnet localhost nntp
Trying 127.0.0.1...
Connected to localhost
Escape characters is '^]'.
201 vstout NNTP[auth] server version 1.5.11t (16 November
1991) ready at Sun Feb 6 16:02:32 1194 (no posting)
IHAVE <msg@id
435 Got it.
QUIT
```

这个对话说明了如何正确重建nntpd；消息“Got it”说明它已经有这篇文章了。如果你得到的是“335 OK”这一消息，则说明出于某种原因，历史文件内的查找失败了。键入Ctrl+D，就可中断这个对话。通过查看系统日志，可了解到出错的原因；nntpd把所有类型的消息统统记入系统日志后台程序。兼容性不好的dbm库通常会出现在抱怨dbminit失败的消息内。