

第30章 其他的TCP/IP应用程序

30.1 引言

本章中我们描述了另外一些很多实现都支持的 TCP/IP 应用程序。有些很简单，易于全面了解（Finger 和 Whois），而另一个则相当复杂（X 窗口系统）。我们只提供了这个复杂应用程序的一个简短的概述，集中介绍其对 TCP/IP 协议的使用。

另外，我们提供一些 Internet 上资源发现工具的概述。包括一组在 Internet 上导航的工具，可以帮助寻找一些我们不知道确切位置和名字的信息。

30.2 Finger 协议

Finger 协议返回一个指定主机上一个或多个用户的信息。它常被用来检查某个人是否登录了，或者搞清一个人的登录名以便给他发送邮件。RFC1288 [Zimmerman 1991] 指明了这个协议。

由于两个原因，很多站点不支持一个 Finger 服务器。第一，Finger 服务器的一个早期版本中的一个编程错误被 1988 年声名狼藉的 Internet 蠕虫病毒利用，作为进入点之一（RFC1135 [Reynolds 1989] 和 [Curry 1992] 更详细地描述了蠕虫）。第二，Finger 协议有可能会泄露一些很多管理员认为是有关用户的私有信息（登录名、电话号码，他们上次的登录时间，等等）。RFC1288 的第 3 节给出了这个有关服务安全方面的细节。

从一个协议的角度来看，Finger 服务器有一个知名的端口 79。客户对这个端口做一个主动打开，然后发送一个在线的请求。服务器处理这个请求，把输出发送回去，然后关闭连接。查询和响应都是采用 NVT ASCII，类似于我们在 FTP 和 SMTP 协议中所看到的。

尽管大多数的 Unix 用户都是使用 finger (1) 客户来访问 Finger 服务器，我们将从使用 Telnet 客户与 Finger 服务器直接相连开始，看看客户发出的每一条在线命令。如果客户的查询是一个空行（在 NVT ASCII 中，空行以一个回车符 CR 跟着一个换行符 LF 来传输），它就是一个请求查询所有在线用户信息的命令。

```
sun % telnet slip finger
Trying 140.252.13.65 ...
Connected to slip.
Escape character is '^['.
```

Telnet 客户输出前三行

这儿我们键入回车作为 Finger 客户的命令

```
Login      Name      Tty  Idle  Login Time   Office   Office Phone
rstevens  Richard Stevens  *co   45   Jul 31 09:13
rstevens  Richard Stevens  *c2   45   Aug  5 09:41
Connection closed by foreign host.
```

Telnet 客户的输出

office 和 office phone 的空白输出字段是从用户的口令 (password) 文件记录的选项字段中取出的（在这个例子中，这两个字段的值没有提供）。

服务器必须在最后做一个主动的关闭操作，因为服务器返回的是一个可变长度的信息。

当客户收到文件结束字符时, 就知道服务器的输出结束了。

当客户的请求由一个用户名组成时, 服务器只以该用户的信息作为响应。下面是另一个例子, 这个例子中删去了Telnet客户的输出:

```
sun % telnet vangogh.cs.berkeley.edu finger
rstevens                      这是我们键入的客户请求
Login: rstevens                Name: Richard Stevens
Directory: /a/guest/rstevens   Shell: /bin/csh
Last login Thu Aug 5 09:55 (PDT) on ttyq2 from sun.tuc.noao.edu
Mail forwarded to: rstevens@noao.edu
No Plan.
```

当一个系统完全禁止了Finger服务时, 因为没有进程被动打开端口 79, 所以客户的主动打开将从服务器接收到一个RST。

```
sun % finger @svr4
[svr4.tuc.noao.edu] connect: Connection refused
```

一些站点在端口 79提供了一个服务器, 但服务器只是向客户输出信息, 而不理睬客户的任何请求:

```
sun % finger @att.com
[att.com]                      这一行是Finger客户输出的; 其余行是服务器输出的
-----
There are no user accounts on the AT&T Internet gateway.
To send email to an AT&T employee, send email to their name
separated by periods at att.com. If the employee has an email
address registered in the employee database, they will receive
email - otherwise, you'll receive a non-delivery notice.
For example: John.Q.Public@att.com
sun % finger clinton@whitehouse.gov
[whitehouse.gov]
```

```
Finger service for arbitrary addresses on whitehouse.gov is not
supported. If you wish to send electronic mail, valid addresses are
"PRESIDENT@WHITEHOUSE.GOV", and "VICE-PRESIDENT@WHITEHOUSE.GOV".
```

对一个组织来说, 另一种可能就是实现一个防火墙网关: 在组织内部和 Internet之间的一个路由器, 负责过滤 (也就是扔掉) 特定的 IP数据报 ([Cheswick and Bellovin 1994] 详细讨论了防火墙网关)。防火墙网关可以被配置成扔掉从 Internet进来的这样一些数据报, 这些数据报是目的端口为 79的TCP报文段。

对于Finger的服务器和Unix的Finger客户还有其他的实现。欲知详情, 请参考 RFC1288和有关finger(1)的手册。

RFC1288指出提供了Finger服务器的、具有TCP/IP连接的自动售货机应该对客户的空行请求响应以现有产品的列表。对于由一个名字组成的客户请求, 它们应该响应以一个数目或者与这个产品有关的可用项的列表。

30.3 Whois协议

Whois协议是另一种信息服务。尽管任何站点都可以提供一个 Whois服务器, 在InterNIC站点 (rs.internic.net) 的服务器是最常使用的。这个服务器维护着所有的 DNS域和很多连接在Internet上的系统的系统管理员的信息 (另一个可用的服务器在 nic.ddn.mil, 不过只包含了有关MILNET的信息)。不幸的是信息有可能是过期的或不完整的。RFC954 [Harrenstein, Stahl,

and Feinler 1985] 说明了 Whois 服务。

从协议的角度来看，Whois 服务器有一个知名的 TCP 端口 43。它接受客户的连接请求，客户向服务器发送一个在线的查询。服务器响应以任何可用的信息，然后关闭连接。请求和应答都以 NVT ASCII 来传输。除了请求和应答所包含的信息不一样，Whois 服务器和 Finger 服务器几乎是一样的。

最常用的 Unix 客户程序是 whois(1) 程序，尽管我们可以使用 Telnet 自己手工键入命令。开始的命令是只包含一个问号的请求，服务器会返回所支持的客户请求的具体信息。

当 NIC 在 1993 年改变为 InterNIC 时，Whois 服务器的站点也从 nic.ddn.mil 移到了 rs.internic.net。很多厂商仍然装载了采用 nic.ddn.mil 版本的 whois 客户程序。为了和正确的服务器联系上，你可能需要指明命令行参数 -h rs.internic.net。

另外，我们可以使用 Telnet 登录 rs.internic.net 站点，登录名采用 whois。

我们将使用 Whois 服务器来查询一下本书的作者（已经删去了无关的 Telnet 客户输出）。第一个请求是查询所有匹配 “stevens” 的名字。

```
sun % telnet rs.internic.net whois
stevens
```

这是我们键入的客户命令

我们省略了其他 25 个 “stevens” 的信息

```
Stevens, W. Richard (WRS28)    stevens@kohala.com    +1 602 297 9416
```

```
The InterNIC Registration Services Host ONLY contains Internet
Information (Networks, ASN's, Domains, and POC's).
Please use the whois server at nic.ddn.mil for MILNET Information.
```

名字后面的括号中的三个大写字母跟着一个数字，(WRS28)，是个人的 NI 句柄。下一个查询包含一个感叹号和一个 NIC 句柄，用于获得有关这个人的进一步信息。

```
sun % telnet rs.internic.net whois
!wrs28
```

我们键入的客户请求

```
Stevens, W. Richard (WRS28)    stevens@kohala.com
Kohala Software
1202 E. Paseo del Zorro
Tucson, AZ 85718
+1 602 297 9416
```

```
Record last updated on 11-Jan-91.
```

很多有关 Internet 变量的其他信息也可以查找。例如，请求 net 140.252 将返回有关 B 类地址 140.252 的信息。

白页

使用 SMTP 的 VRFY 命令、Finger 协议以及 Whois 协议在 Internet 上查找用户类似于使用电话号码簿的白页查找一个人的电话号码。在目前阶段，诸如上述的工具已经广泛可用了，为了提高这种服务的研究正在进行当中。

[Schwartz and Tsirigotis 1991] 包含了正在 Internet 上试验的不同白页服务的其他信息。一个叫作 Netfind 的特别工具可以通过使用 Telnet，以 netfind 登录到 bruno.cs.colorado 或者 ds.internic.net 站点来访问。

RFC1309 [Weider, Reynolds, and Heker 1992] 提供了对 OSI 目录服务 X.500 的概述，并且比较了它与当前的 Internet 技术（Finger 和 Whois）的相同点和不同点。

30.4 Archie、WAIS、Gopher、Veronica和WWW

前两节我们讨论的工具——Finger、Whois和一个白页服务——是用来查找人的信息的。还有一些工具是用来定位文件和文档的，本节中对这些工具给出了一个概述。我们只提供了一个概述，因为对每一个工具的细节的研究超出了本书的范围。我们给出了在 Internet上找到这些工具的方法，鼓励你去试一试，找找看哪些工具可以帮助你。还有一些其他的工具正在被开发。[Obraczka, Danzig, and Li1993] 概述了在Internet上的资源发现服务。

30.4.1 Archie

本书中使用的很多资源都是使用匿名 FTP得到的。问题是如何找到有我们想要的程序的FTP站点。有时候我们甚至不知道精确的文件名，但知道几个很可能在文件名中出现的关键词。

Archie提供了Internet上几千个FTP服务器的目录。我们可以通过登录进一个 Archie服务器，搜索那些名字中包含了一个指定的常规表达式的文件。输出是一个与文件名匹配的 FTP服务器的列表。然后我们可以使用匿名 FTP去那个站点取得想要的文件。

全世界有很多 Archie服务器。一个比较好的开始点是使用 Telnet以archie名字登录进 ds.internic.net，然后执行命令 servers。这个命令的输出提供了所有 Archie服务器以及它们地址的一个列表。

30.4.2 WAIS

Archie帮助我们查找名字中包含关键字的文件，但有时候我们需要查找包含一个关键字的文件或数据库。即，想查找一个内容中包含一个关键字的文件，而不是文件名字中包含关键字。

WAIS (Wide Area Information Servers广域信息服务系统)知道几百个包含了有关计算机主题和其他一般性主题信息的数据库。为了使用 WAIS，我们要选择需要查找的数据库，指明关键字。尝试WAIS服务请使用Telnet，以wais名字登录quake.think.com站点。

30.4.3 Gopher

Gopher是其他 Internet资源服务如 Archie、WAIS和匿名FTP的一个菜单驱动的前端程序。Gopher是最容易使用的工具之一，因为不管它使用了哪个资源服务，它的用户界面都是一样的。

为了尝试Gopher，请使用Telnet，以gopher名字登录is.internic.net站点。

30.4.4 Veronica

就像 Archie是一个匿名 FTP服务器的索引一样，Veronica(Veronica Very Easy Rodent-Oriented Netwide Index to Computerized Archives)是一个Gopher标题的索引。一次 Veronica搜索一般要查找几百个Gopher服务器。

我们必须通过一个 Gopher客户来访问 Veronica服务。选择 Gopher的菜单项 “Beyond InterNIC: Virtual Treasures of the Internet”，然后在下一个菜单中选择 Veronica。

30.4.5 万维网WWW

万维网使用一个称为超文本的工具，使得我们可以浏览一个大的 / 全球范围的服务和文档。信息和关键字一起显示，不过关键字被突出显示[⊖]。我们可以通过选择关键字得到更多的信息。

为了访问WWW，请使用Telnet登录info.cern.ch站点。

30.5 X窗口系统

X窗口系统(X Window System)，或简称为X，是一种客户-服务器应用程序。它可以使得多个客户（应用）使用由一个服务器管理的位映射显示器。服务器是一个软件，用来管理显示器、键盘和鼠标。客户是一个应用程序，它与服务器在同一台主机上或者在不同的主机上。在后一种情况下，客户与服务器之间通信的通用形式是 TCP，尽管也可以使用诸如 DECNET 的其他协议。在有些场合，服务器是与其他主机上客户通信的一个专门的硬件（一个 X终端）。在另一种场合，一个独立的工作站，客户与服务器位于同一台主机，使用那台主机上的进程间通信机制进行通信，而根本不涉及任何网络操作。在这两种极端情况之间，是一台既支持同一台主机上的客户又支持不同主机上的客户的工作站。

X需要一个诸如TCP的、可靠的、双向的流协议（X不是为不可靠协议，如UDP，而设计的）。客户与服务器的通信是由在连接上交换的8 bit字节组成的。[Nye 1992] 给出了客户与服务器在它们的TCP连接上交换的150多个报文的格式。

在一个Unix系统中，当X客户和X服务器在同一台主机上时，一般使用 Unix系统的本地协议，而不使用TCP协议，因为这样比使用TCP的情况减少了协议处理时间。Unix系统的本地协议是同一台主机上的客户和服务器之间可以使用的一种进程间通信的形式。回忆一下在图2-4中，当使用TCP作为同一台主机上进程间的通信方式时，在IP层以下发生了这个数据的环回(loopback)，隐含着所有的TCP和IP处理都发生了。

图30-1显示了三个客户使用一个显示器的可能的脚本。一个客户与服务器在同一台主机上，使用Unix系统的本地协议。另外两个位于不同的主机上，使用TCP。一般来说，其中一个客户是一个窗口管理程序(window manager)，它有权限管理显示器上窗口的布局。例如，窗口管理程序允许我们在屏幕上移动窗口，或者改变窗口的大小。

在这里客户和服务器这两个词猛一看含义相反了。对于Telnet和FTP的应用，我们把客户看作是在键盘和显示器上的交互式用户。但是对于X，键盘和显示器是属于服务器的。服务器被认为是提供服务的一方。X提供的服务是对窗口、键盘和鼠标的访问。对于Telnet，服务是登录远程的主机。对于FTP，服务是服务器上的文件系统。

当X终端或工作站引导时，一般启动X服务器。服务器创建一个TCP端点，在端口6000 + n 上做一个被动打开，其中 n 是显示器号（一般是0）。大多数的 Unix服务器也使用名字 /tmp/.X11-unix/X n 创建一个Unix系统的插口，其中 n 还是显示器的号。

当一个客户在另一台主机上启动时，它创建一个TCP端点，对服务器上的端口6000 + n 做一个主动打开。每个客户都得到了一个自己与服务器的连接。服务器负责对所有的客户请求进行复用。从这点开始，客户通过TCP连接向服务器发送请求（例如，创建一个窗口），服务

⊖ 例如通过使用不同的颜色——译者注。

器返回应答, 服务器也发送事件给客户 (鼠标按钮按下, 键盘键按下, 窗口暴露, 窗口大小改变, 等等)。

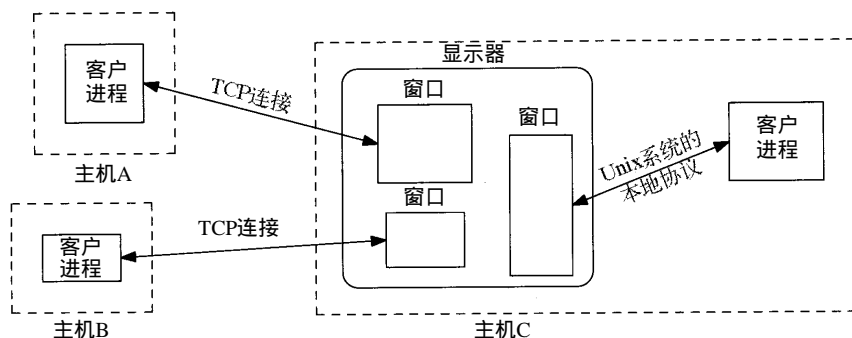


图30-1 使用一个显示器的三个X客户

图30-2将图30-1重新画, 但强调了客户与X服务器进程间的通信, X服务器进程轮流管理着每个窗口。图中没有显示的是X服务器管理键盘和鼠标。

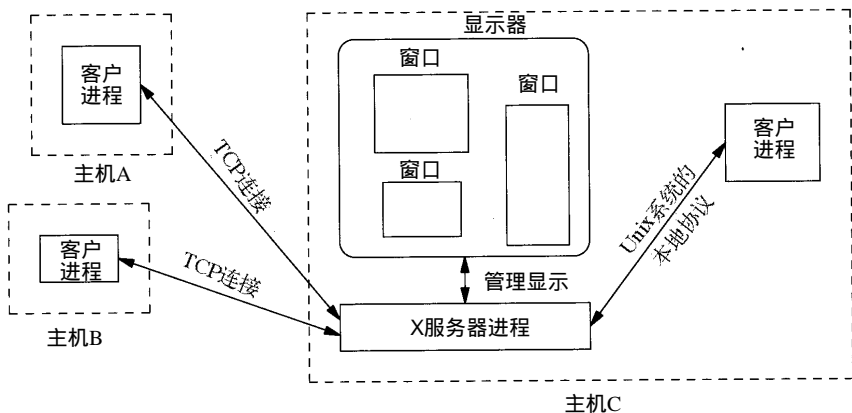


图30-2 使用一个显示器的三个客户

单个服务器处理多个客户请求的这种设计与我们在 18.11 节描述的正常的 TCP 并发服务器设计不同。例如, 每次一个新的 TCP 连接请求到达, FTP 和 Telnet 服务器都会产生一个新的进程, 因此, 每个客户都和一个不同的服务器进程通信。然而, 对于 X, 运行在同一台主机或者在不同主机上的所有客户都和同一个服务器通信。

通过 X 客户和它的服务器之间的 TCP 连接可以交换很多数据。传输数据的数目依赖于特定的应用程序设计。例如, 如果我们运行 Xclock 客户, Xclock 在服务器的一个窗口中显示客户机当前的时间和日期。如果我们指定每隔 1 秒修改一次时间, 那么每隔 1 秒, 就会有一个 X 报文通过 TCP 连接从客户传输到服务器。如果我们运行 X 终端模拟程序, Xterm, 我们敲的每一个键都会变成一个 32 字节的 X 报文 (加上标准的 IP 和 TCP 首部就是 72 字节), 在相反方向上的回送字符将是一个更大的 X 报文。[Droms and Dyksen 1990] 检查了不同的 X 客户与一个特定的服务器之间的 TCP 流量。

30.5.1 Xscope 程序

Xscope 是检查 X 客户与它的服务器之间交换的信息的一个方便的程序。大多数的 X 窗口实

现都提供这个程序。它处在客户与服务器之间，双向传输所有的数据，同时解析所有的客户请求和服务器应答。图30-3显示了这种设置。

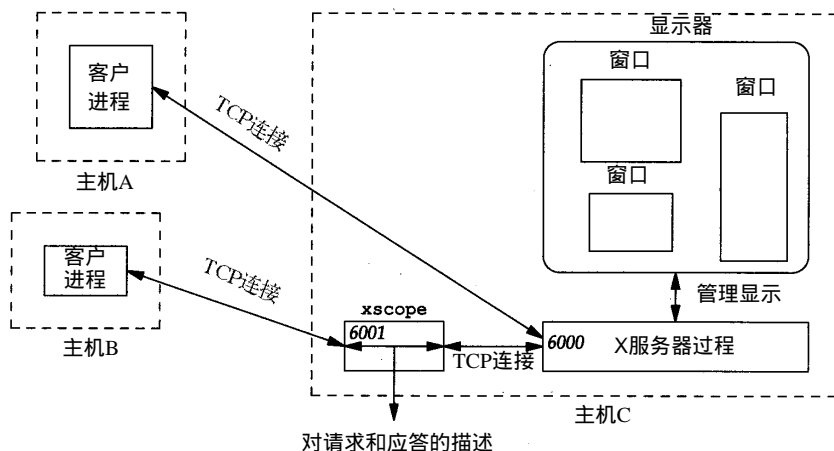


图30-3 使用xscope 监视一个X连接

首先，我们在服务器所在的主机上启动 xscope 进程，但是 xscope 不是在端口 6000 而是在端口 6001 上监听 TCP 的连接请求。然后我们在另一台主机上启动一个客户，指明显示器号为 1，而不是 0，使得客户与 xscope 相连，而不直接与服务器相连。当客户的连接请求到达时，xscope 创建与端口 6000 上的真正的服务器的一个 TCP 连接，在客户与服务器之间复制所有的数据，同时生成请求与应答的一个可读的描述。

我们将在 sun 主机上启动 xscope，然后在主机 svr4 上运行 xclock 客户。

```
svr4 % DISPLAY=sun:1 xclock -digital -update 5
```

这条命令在主机 sun 的一个窗口中以数字形式显示时间和日期。我们指明了一个每 5 秒的更新时间。

```
Thu Sep 9 10:32:55 1993
```

我们对 xscope 指明一个 -q 选项以产生最小的输出。为了看到每个报文的所有字段，可以使用不同的冗长级别。下面的输出显示了前三个请求和应答。

```
sun % xscope -q
0.00: Client --> 12 bytes
0.02:                                     152 bytes <-- X11 Server
0.03: Client --> 48 bytes
      .....REQUEST: CreateGC
      .....REQUEST: GetProperty
0.20:                                     396 bytes <-- X11 Server
      .....REPLY: GetProperty
0.30: Client --> 8 bytes
0.38: Client --> 20 bytes
      .....REQUEST: InternAtom
0.43:                                     32 bytes <-- X11 Server
      .....REPLY: InternAtom
```

客户的第 1 个在时刻 0.00 的报文和服务器在时刻 0.02 的响应是客户与服务器之间标准的连接建立过程。客户标识它的字节顺序以及它希望的服务器版本。服务器响应以有关自己的不

同的信息。

下一个在时刻0.03的报文包含了两个客户请求。第1个请求在服务器上创建一个客户可以在其中画的图形上下文。第2个请求从服务器上得到一个属性(RESOURCE_MANAGER属性)。属性可以用于客户之间的通信,经常是在一个应用程序和窗口管理程序之间。服务器在时刻0.20的应答包含了这个属性。

下面两个在时刻0.30和0.38的客户报文形成了返回一个原子的单个请求(每个属性具有一个唯一的整型标识符称为原子)。服务器在时刻0.43的应答包含了这个原子。

如果不提供有关X窗口系统更多的细节是不可能进一步理解这个例子的,但这又不是本节的目的。在这个例子中,在窗口被显示之前,客户总共发送了1668个字节组成的12个报文段,服务器总共发送了1120个字节组成的10个报文段。耗费的时间为3.17秒。从这以后客户每5秒发送一个平均44个字节的小请求,请求更新窗口。这样一直持续到客户被终止。

30.5.2 LBX: 低带宽X

为了将X用于局域网,对X协议使用的编码进行了优化,因为在局域网中花在对数据进行编码和解码的时间比最小化传输的数据更重要。尽管这种推断对以太网是适用的,但对于低速的串行线,如SLIP和PPP链路,就存在问题了(2.4节和2.6节)。

定义一个称为低带宽X(LBX)的标准的工作正在进行当中,它使用了下面的技术来减少网络流量的数目:快速缓存、只发送与前面分组的不同部分以及压缩技术。标准的规范和在第6版的X窗口系统中的一个样本实现应该会在1994年的早些时候完成。

30.6 小结

我们介绍的前两个应用,Finger和Whois,是用来获得用户信息的。Finger客户查询一个服务器,经常是为了找到某个人的登录名(以便给他们发电子邮件),或者去看一下某个人是否登录了。Whois客户一般与InterNIC运行的服务器联系,查找关于一个人、机构、域或网络号的信息。

我们简单描述了其他一些Internet资源发现服务:Archie、WAIS、Gopher、Veronica和WWW,帮助我们在Internet上定位文件和文档。还有一些资源发现工具正在被开发。

本章的最后简单浏览了另一个TCP/IP的重要客户程序,X窗口系统。我们看到X服务器管理一个显示器上的多个窗口,处理客户与其窗口的通信。每个客户都有它自己的与服务器的TCP连接,一个单个的服务器为一个给定的显示器管理着所有的客户。通过Xscope程序,我们看到怎样把一个程序放在一个客户与服务器之间,输出有关两者之间交换的报文的信息。

习题

- 30.1 试用Whois找到网络号为88的A类网络的拥有者。
- 30.2 试用Whois找到管理whitehouse.gov域的DNS服务器。这个应答与DNS给出的答案相匹配吗?
- 30.3 在图30-3中,你认为xscope进程必须和X服务器运行在同一台主机上吗?