<https://www.cnblogs.com/liang-ling/p/5833489.html>

# 1. 概述

     Telnet协议是TCP/IP协议族中的一员，是Internet远程登陆服务的标准协议。Telnet协议的目的是提供一个相对通用的，双向的，面向八位字节的通信方法，允许界面终端设备和面向终端的过程能通过一个标准过程进行互相交互。应用Telnet协议能够把本地用户所使用的计算机变成远程主机系统的一个终端。

Telnet协议具有如下的特点：

**1. 适应异构**

      为了使多个操作系统间的Telnet交互操作成为可能，就必须详细了解异构计算机和操作系统。比如，一些操作系统需要每行文本用ASCII回车控制符（CR）结束，另一些系统则需要使用ASCII换行符（LF），还有一些系统需要用两个字符的序列回车-换行（CR-LF）；再比如，大多数操作系统为用户提供了一个中断[程序](http://www.xuebuyuan.com/" \o "程序" \t "_blank)运行的快捷键，但这个快捷键在各个系统中有可能不同（一些系统使用CTRL+C，而另一些系统使用ESCAPE）。如果不考虑系统间的异构性，那么在本地发出的字符或命令，传送到远地并被远地系统解释后很可能会不准确或者出现错误。因此，Telnet协议必须解决这个问题。

      为了适应异构环境，Telnet协议定义了数据和命令在Internet上的传输方式，此定义被称作网络虚拟终端NVT（Net Virtual Terminal）。它的应用过程如下：

* 对于发送的数据：客户机软件把来自用户终端的按键和命令序列转换为NVT格式，并发送到服务器，服务器软件将收到的数据和命令，从NVT格式转换为远地系统需要的格式；
* 对于返回的数据：远地服务器将数据从远地机器的格式转换为NVT格式，而本地客户机将将接收到的NVT格式数据再转换为本地的格式。

**2. 传送远地命令**

      我们知道绝大多数操作系统都提供各种快捷键来实现相应的控制命令，当用户在本地终端键入这些快捷键的时候，本地系统将执行相应的控制命令，而不把这些快捷键作为输入。那么对于Telnet来说，它是用什么来实现控制命令的远地传送呢？

      Telnet同样使用NVT来定义如何从客户机将控制功能传送到服务器。我们知道USASCII字符集包括95个可打印字符和33个控制码。当用户从本地键入普通字符时，NVT将按照其原始含义传送；当用户键入快捷键（组合键）时，NVT将把它转化为特殊的ASCII字符在网络上传送，并在其到达远地机器后转化为相应的控制命令。将正常ASCII字符集与控制命令区分主要有两个原因：

* 这种区分意味着Telnet具有更大的灵活性：它可在客户机与服务器间传送所有可能的ASCII字符以及所有控制功能；
* 这种区分使得客户机可以无二义性的指定信令，而不会产生控制功能与普通字符的混乱。

**3. 数据流向**

将Telnet设计为应用级软件有一个缺点，那就是：效率不高。这是为什么呢？下面给出Telnet中的数据流向：

      数据信息被用户从本地键盘键入并通过操作系统传到客户机程序，客户机程序将其处理后返回操作系统，并由操作系统经过网络传送到远地机器，远地操作系统将所接收数据传给服务器程序，并经服务器程序再次处理后返回到操作系统上的伪终端入口点，最后，远地操作系统将数据传送到用户正在运行的应用程序，这便是一次完整的输入过程；输出将按照同一通路从服务器传送到客户机。

因为每一次的输入和输出，计算机将切换进程环境好几次，这个开销是很昂贵的。还好用户的键入速率并不算高，这个缺点我们仍然能够接受。

**4. 强制命令**

      我们应该考虑到这样一种情况：假设本地用户运行了远地机器的一个无休止循环的错误命令或程序，且此命令或程序已经停止读取输入，那么操作系统的缓冲区可能因此而被占满，如果这样，远地服务器也无法再将数据写入伪终端，并且最终导致停止从TCP连接读取数据，TCP连接的缓冲区最终也会被占满，从而导致阻止数据流流入此连接。如果以上事情真的发生了，那么本地用户将失去对远地机器的控制。

      为了解决此问题，Telnet协议必须使用外带信令以便强制服务器读取一个控制命令。我们知道TCP用紧急数据机制实现外带数据信令，那么Telnet只要再附加一个被称为数据标记(date mark)的保留八位组，并通过让TCP发送已设置紧急数据比特的报文段通知服务器便可以了，携带紧急数据的报文段将绕过流量控制直接到达服务器。作为对紧急信令的相应，服务器将读取并抛弃所有数据，直到找到了一个数据标记。服务器在遇到了数据标记后将返回正常的处理过程。

**5.  选项协商**

      由于Telnet两端的机器和操作系统的异构性，使得Telnet不可能也不应该严格规定每一个telnet连接的详细配置，否则将大大影响Telnet的适应异构性。因此，Telnet采用选项协商机制来解决这一问题。

      Telnet选项的范围很广：一些选项扩充了大方向的功能，而一些选项制涉及一些微小细节。例如：有一个选项可以控制Telnet是在半双工还是全双工模式下工作（大方向）；还有一个选项允许远地机器上的服务器决定用户终端类型（小细节）。

      Telnet选项的协商方式也很有意思，它对于每个选项的处理都是对称的，即任何一端都可以发出协商申请；任何一端都可以接受或拒绝这个申请。另外，如果一端试图协商另一端不了解的选项，接受请求的一端可简单的拒绝协商。因此，有可能将更新，更复杂的Telnet客户机服务器版本与较老的，不太复杂的版本进行交互操作。如果客户机和服务器都理解新的选项，可能会对交互有所改善。否则，它们将一起转到效率较低但可工作的方式下运行。所有的这些设计，都是为了增强适应异构性，可见Telnet的适应异构性对其的应用和发展是多么重要。

# 2. 原理

Telnet协议的主体由三个部分组成：

* 网络虚拟终端（NVT，Network Virtual Terminal）的定义；
* 操作协商定义；
* 协商有限自动机；

## 2.1. 网络虚拟终端（NVT）

### 2.1.1. NVT工作原理

顾名思义，网络虚拟终端（NVT）是一种虚拟的终端设备，它被客户和服务器所采用，用来建立数据表示和解释的一致性。

### 2.1.2. NVT的定义

1. NVT的组成

     网络虚拟终端NVT包括两个部分：

* 输出设备：输出远程数据，一般为显示器
* 输入设备：本地数据输入

2. 在NVT上传输的数据格式

在网络虚拟终端NVT上传输的数据采用8bit字节数据，其中最高位为0的字节用于一般数据，最高位为1的字节用于NVT命令

3. NVT在TELNET中的使用

      TELNET使用了一种对称的数据表示，当每个客户机发送数据时，把它的本地终端的字符表示影射到NVT的字符表示上，当接收数据时，又把NVT的表示映射到本地字符集合上。

      在通信开始时，通信双方都支持一个基本的NVT终端特性子集（只能区分何为数据，何为命令），以便在最低层次上通信，在这个基础上，双方通过NVT命令协商确定NVT的更高层次上的特性，实现对NVT功能的扩展。

      在TELNET中存在大量的子协议用于协商扩展基本的网络虚拟终端NVT的功能，由于终端类型的多样化，使得TELNET协议族变得庞大起来。

## 2.2. 操作协商

### 2.2.1. 为什么要协商操作选项

      当定义了网络虚拟终端设备后，通信的双方就可以在一个较低的层次上实现数据通信，但基本的NVT设备所具有的特性是十分有限的，它只能接收和显示7位的ASCII码，没有最基本的编辑能力，所以简单的NVT设备是没有实际应用意义的；为此TELNET协议定义了一族协议用于扩展基本NVT的功能，目的是使NVT能够最大限度地达到用户终端所具有的功能。

      为了实现对多种终端特性的支持，TELNET协议规定在扩展NVT功能时采用协商的机制，只有通信双方通过协商后达成一致的特性才能使用，才能赋予NVT该项特性，这样就可以支持具有不同终端特性的终端设备可以互连，保证他们是工作在他们自己的能力以内。

### 2.2.2. 操作协商命令格式

      TELNET的操作协商使用NVT命令，即最高位为1的字节流，每条NVT命令以字节IAC（0xFF）开始。原理如下：

      只要客户机或服务器要发送命令序列而不是数据流，它就在数据流中插入一个特殊的保留字符，该保留字符叫做“解释为命令”（IAC  ，Interpret As Command) 字符。当接收方在一个入数据流中发现IAC字符时，它就把后继的字节处理为一个命令序列。下面列出了所有的Telnet NVT命令，其中很少用到。

表1   TELNET 命令

|  |  |  |
| --- | --- | --- |
| 名称 | 编码 | 说明 |
| EOF | 236 | 文件结束符 |
| SUSP | 237 | 挂起当前进程 |
| ABORT | 238 | 中止进程 |
| EOR | 239 | 记录结束符 |
| SE | 240 | 子选项结束 |
| NOP | 241 | 空操作 |
| DM | 242 | 数据标记 |
| BRK | 243 | 终止符（break） |
| IP | 244 | 终止进程 |
| AO | 245 | 终止输出 |
| AYT | 246 | 请求应答 |
| EC | 247 | 终止符 |
| EL | 248 | 擦除一行 |
| GA | 249 | 继续 |
| SB | 250 | 子选项开始 |
| WILL | 251 | 选项协商 |
| WONT | 252 | 选项协商 |
| DO | 253 | 选项协商 |
| DONT | 254 | 选项协商 |
| IAC | 255 | 字符0XFF |

其中常用的TELNET选项协商如下：

* WILL (option code) 251 指示希望开始执行，或者确认现在正在操作指示的选项。
* WON'T (option code) 252 指出拒绝执行或继续招待所指示的选项。
* DO (option code) 253 指出要求对方执行，或者确认希望对方执行指示的选项。
* DON'T (option code) 254 指出要求对方停止执行，或者确诊要求对方停止执行指示的选项。

那么对于接收方和发送方有以下几种组合：

表2   TELNET 选项协商的六种情况

|  |  |  |
| --- | --- | --- |
| 发送者 | 接收者 | 说明 |
| WILL | DO | 发送者想激活某选项，接受者接收该选项请求 |
| WILL | DONT | 发送者想激活某选项，接受者拒绝该选项请求 |
| DO | WILL | 发送者希望接收者激活某选项，接受者接受该请求 |
| DO | DONT | 发送者希望接收6者激活某选项，接受者拒绝该请求 |
| WONT | DONT | 发送者希望使某选项无效，接受者必须接受该请求 |
| DONT | WONT | 发送者希望对方使某选项无效，接受者必须接受该请求 |

选项协商需要3个字节：IAC，然后是WILL、DO、WONT或DONT；最后一个标识字节用来指明操作的选项。常用的选项[代码](http://www.xuebuyuan.com/" \o "代码" \t "_blank)如下：

表3   TELNET 选项代码

|  |  |  |
| --- | --- | --- |
| 选项标识 | 名称 | RFC |
| 1 | 回应（echo) | 857 |
| 3 | 禁止继续 | 858 |
| 5 | 状态 | 859 |
| 6 | 时钟标识 | 860 |
| 24 | 终端类型 | 1,091 |
| 31 | 窗口大小 | 1,073 |
| 32 | 终端速率 | 1,079 |
| 33 | 远端流量控制 | 1,372 |
| 34 | 行模式 | 1,184 |
| 36 | 环境变量 | 1,408 |

      通常情况下，客户机向服务器发送字符而服务器将其回显到用户的终端上，但是，如果网络的时延回引起回显速度太慢，用户可能更愿意让本地系统回显字符。在客户机允许本地系统回显前，它要向服务器发送以下序列：

IAC  DONT ECHO

服务器收到请求后，发出3个字符的响应：

IAC WONT ECHO

表示服务器已经按请求同意关闭回显。

## 2.3. 子选项协商

      除了“打开”或“关闭”以外，有些选项还需要更多的信息，例如对于指明终端类型来说，客户必须发送一个字符串来标识终端类型，所以要定义子选项协商。

RFC 1091定义了终端类型的子选项协商。举个例子：

客户发送字节序列来请求打开选项：

<  IAC，WILL，24>

24是终端类型的选项标识符。如果服务器同意该请求，响应为：

<  IAC，DO，24 >

接着服务器发送

<  IAC，SB，24，1，IAC，SE>请求客户给出其终端类型。

SB是子选项开始命令，下一个字节24表示该子选项为终端类型选项。下一个字节1表示：发送你的终端类型。客户的响应为：

<  IAC，SB，24，0，'I'，'B'，'M'，'P'，'C'， IAC，SE>

第四个字节0的含义是“我的终端类型为”。

# 3. 实现

      整个协议软件分为三个模块，各模块的功能如下：

1. 与本地用户的输入/输出模块：处理用户输入/输出；

2. 与远地系统的输入/输出模块：处理与远程系统输入/输出；

3. TELNET协议模块：实现TELNET协议，维护协议状态机。

      telnet客户机要做两件事：

* 读取用户在键盘上键入的字符，并通过tcp连接把他们发送到远程服务器上
* 读取从tcp连接上收到的字符，并显示在用户的终端上