文件传输协议FTP

金舒原
jinshuyuan@mail.sysu.edu.cn
计算机学院

本章内容

- 什么是文件传输协议FTP
- FTP 功能
- FTP服务的工作过程
- FTP的访问方式
- FTP的工作原理
- FTP命令及响应编码

什么是文件传输协议FTP

- 文件传输是TCP/IP中使用最广泛的应用之一
- 文件传输协议FTP (File Transfer Protocol)的主要功能是完成从一个系统到另一个系统的完整的文件拷贝
- FTP并不是针对某种具体操作系统或某类具体文件而设计的文件传输协议
- 它通过一些规程，利用网络低层提供的服务，屏蔽了各种计算机系统的细节来完成文件传输的任务
- 它只提供文件传送的一些基本的服务，可以在异构网中任意计算机间传送文件

网络环境下复制文件的复杂性

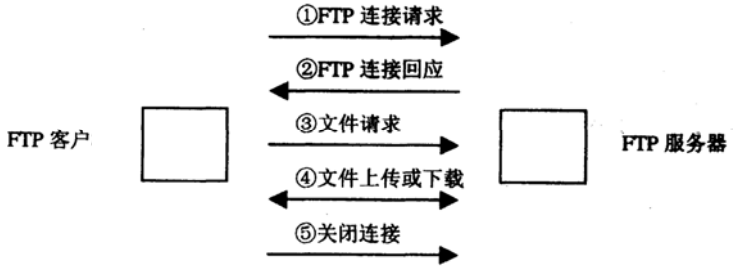
- 由众多的计算机厂商研制出的文件系统多达数百种，且差别很大：
  1. 计算机存储数据的格式不同。
  2. 文件的目录结构和文件命名的规定不同。
  3. 对于相同的文件存取功能，操作系统使用的命令不同。
  4. 访问控制方法不同。
- 要实现不同系统之间的文件复制非常困难。
- FTP 的主要功能是减少或消除在不同操作系统下处理文件的不兼容性。

FTP的功能

- FTP的主要功能包括两个方面：
  - 文件的下载　就是将远程服务器上提供的文件下载到本地计算机上。使用FTP实现的文件下载与HTTP相比较，具有使用简便、支持断点续传和传输速度快的优点
  - 文件的上传　是指客户机可以将任意类型的文件上传到指定的FTP服务器上
- FTP服务支持文件上传和下载

FTP服务的工作过程

- FTP服务采用典型的客户/服务器工作模式
- FTP在数据传输时，使用的传输层协议是TCP



TCP报文段格式

偏移	字节	0								1								2								3							
字节	比特	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	来源连接端口																目的连接端口														
4	32	序列号码																															
8	64	确认号码（当ACK设置）																															
12	96	数据偏移				保留 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	窗口大小														
16	128	校验和																紧急指针（当URG设置）														
20 …	160 …	选项（如果数据偏移 > 5，需要在结尾添加0。） …																															

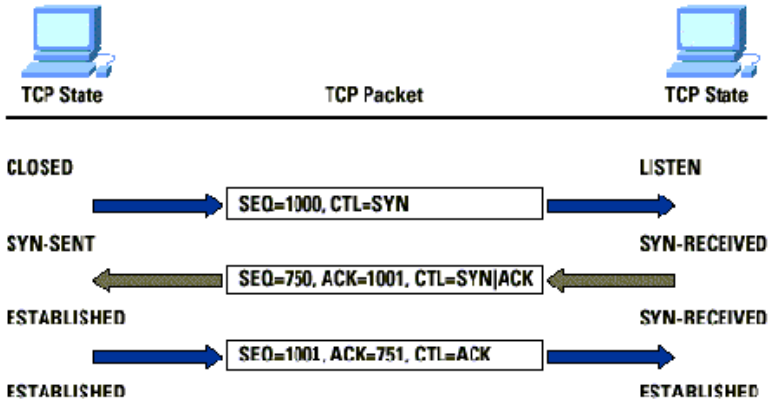
TCP标志位

- NS—ECN-nonce，ECN显式拥塞通知（Explicit Congestion Notification），允许拥塞控制的端对端通知而避免丢包。如果底层网络设施支持，则可能被启用ECN的两个端点使用。在ECN成功协商的情况下，ECN感知路由器可以在IP头中设置一个标记来代替丢弃数据包，以标明阻塞即将发生。数据包的接收端回应发送端的表示，降低其传输速率，就如同在往常中检测到包丢失那样。
- CWR—Congestion Window Reduced，拥塞窗口减, 发送方降低它的发送速率。发送者在接收到一个带有ECE指示的包时,将会使用CWR标志位。定义于RFC 3168（2001）。
- ECE—ECN-Echo, ECN显式拥塞通知回应，定义于RFC 3168（2001）。
- URG—为1表示高优先级数据包，紧急指针字段有效。
- ACK—为1表示确认号字段有效
- PSH—为1表示是带有PUSH标志的数据，指示接收方应该尽快将这个报文段交给应用层而不用等待缓冲区装满。
- RST—为1表示出现严重差错。可能需要重新创建TCP连接。还可以用于拒绝非法的报文段和拒绝连接请求。
- SYN—为1表示这是连接请求或是连接接受请求，用于创建连接和使顺序号同步
- FIN—为1表示发送方没有数据要传输了，要求释放连接。

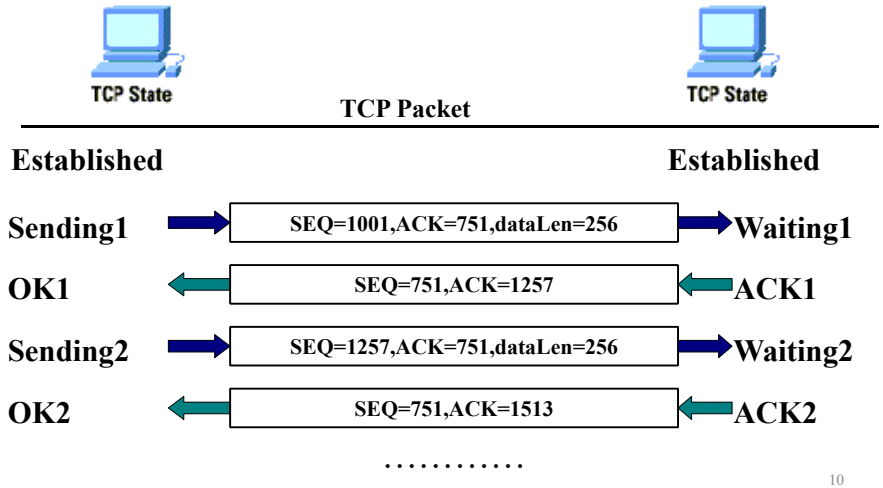
TCP通信过程

- 建立连接
- 数据传输
- 断开连接

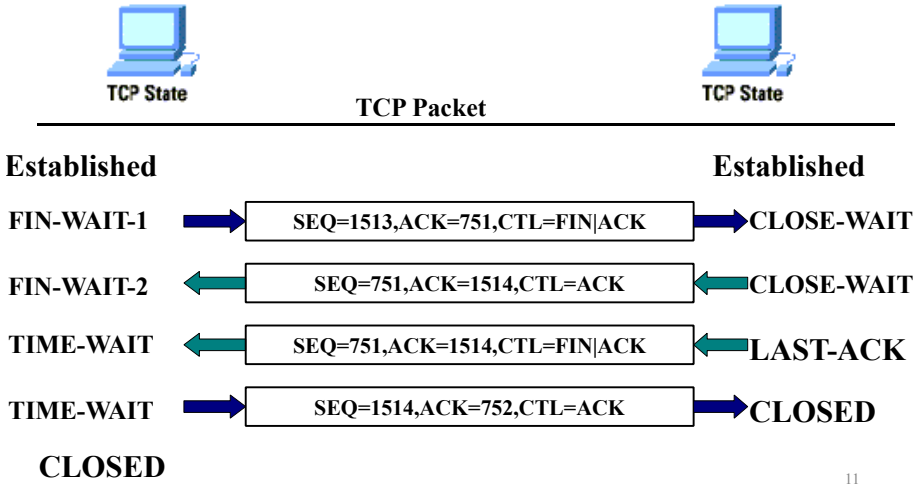
TCP连接建立过程



TCP数据传输过程



TCP连接断开过程

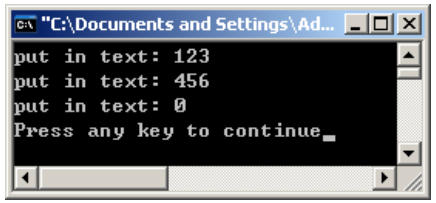


TCP报文段分析实验

- 实验环境
  - 网络中可互相访问的两台主机
  - 例如：
  - 同一局域网中两台机器：172.16.xx.1，172.16.xx.2
  - 不同网络的两台可以相互访问的机器
- 一个C/S模式的程序，实现简单的TCP数据发送与接收（程序代码参见教材“3.5 编程实验 P97-101”）
  - 例如：
  - Client运行在172.16.xx.1
  - Server运行在172.16.xx.2

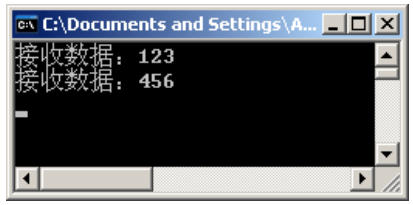
实验场景截图

- Client发送两次数据，内容分别为123和456，然后发送0结束TCP连接。
- 程序截图如下。



客户端发送数据

IP：192.168.1.34



服务器端接收数据

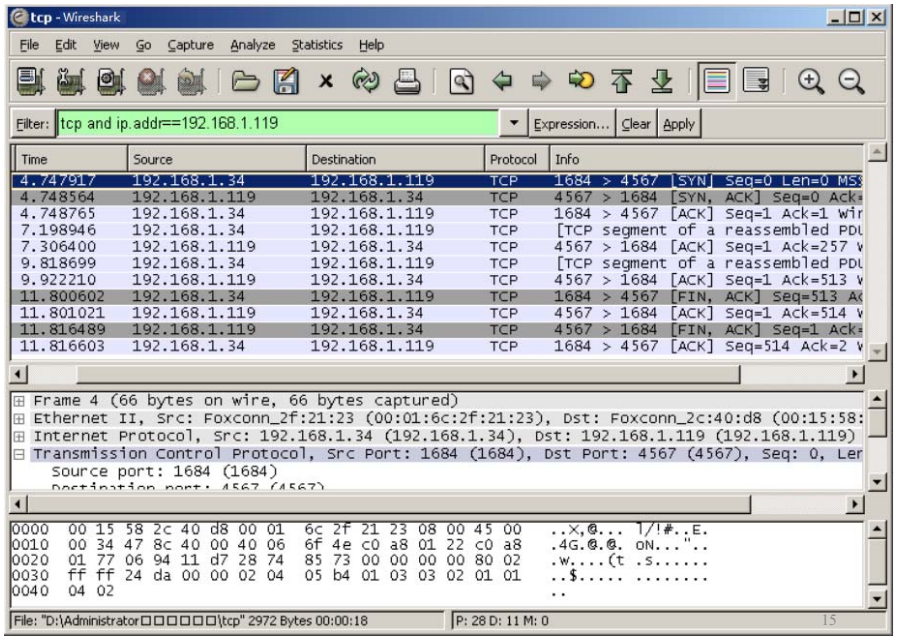
IP：192.168.1.119

捕获报文

- 在Client发送数据之前，在192.168.1.34主机(Client)上开启Wireshark。
- 在捕获前不进行过滤，直接捕获所有数据包。
- 当Client结束TCP连接之后，停止捕获数据包。
- 采用捕获后过滤的方法，在显示过滤器中输入如下过滤规则

  tcp AND ip.addr==192.168.1.119

  其中，192.168.1.119是Server主机。
- 过滤后，共得到11个数据包，见下页图。



TCP报文段详细分析

- 这11个数据包的含义如下：
  - 1~3：三次握手，建立连接
  - 4~5：第一次发送数据
  - 6~7：第二次发送数据
  - 8~11：断开连接
- 下面将对这11个数据包进行详细分析。

1 C→S SYN

SEQ=X+0

与TCP报文格式相对应

```
Source port: 1684 (1684)
Destination port: 4567 (4567)
Sequence number: 0    (relative sequence number)
Header length: 32 bytes
Flags: 0x0002 (SYN)
  0... .... = Congestion Window Reduced (CWR): Not set
  .0.. .... = ECN-Echo: Not set
  ..0. .... = Urgent: Not set
  ...0 .... = Acknowledgment: Not set
  .... 0... = Push: Not set
  .... .0.. = Reset: Not set
  .... ..1. = Syn: Set
  .... ...0 = Fin: Not set
Window size: 262140 (scaled)
Checksum: 0x24da [correct]
Options: (12 bytes)
```


2 S→C SYN,ACK

SEQ=Y+0

ACK=X+1

```
Source port: 4567 (4567)
Destination port: 1684 (1684)
Sequence number: 0    (relative sequence number)
Acknowledgement number: 1    (relative ack number)
Header length: 32 bytes
Flags: 0x0012 (SYN, ACK)
  0... .... = Congestion Window Reduced (CWR): Not set
  .0.. .... = ECN-Echo: Not set
  ..0. .... = Urgent: Not set
  ...1 .... = Acknowledgment: Set
  .... 0... = Push: Not set
  .... .0.. = Reset: Not set
  .... ..1. = Syn: Set
  .... ...0 = Fin: Not set
Window size: 65535
Checksum: 0x1faa [correct]
Options: (12 bytes)
```


3 C→S ACK

SEQ=X+1

ACK=Y+1

三次握手结束

```
Source port: 1684 (1684)
Destination port: 4567 (4567)
Sequence number: 1    (relative sequence number)
Acknowledgement number: 1    (relative ack number)
Header length: 20 bytes
Flags: 0x0010 (ACK)
  0... .... = Congestion Window Reduced (CWR): Not set
  .0.. .... = ECN-Echo: Not set
  ..0. .... = Urgent: Not set
  ...1 .... = Acknowledgment: Set
  .... 0... = Push: Not set
  .... .0.. = Reset: Not set
  .... ..0. = Syn: Not set
  .... ...0 = Fin: Not set
Window size: 256960 (scaled)
Checksum: 0x6584 [correct]
```


4 C→S PSH,ACK

SEQ=X+1, data length=256, next seq=257

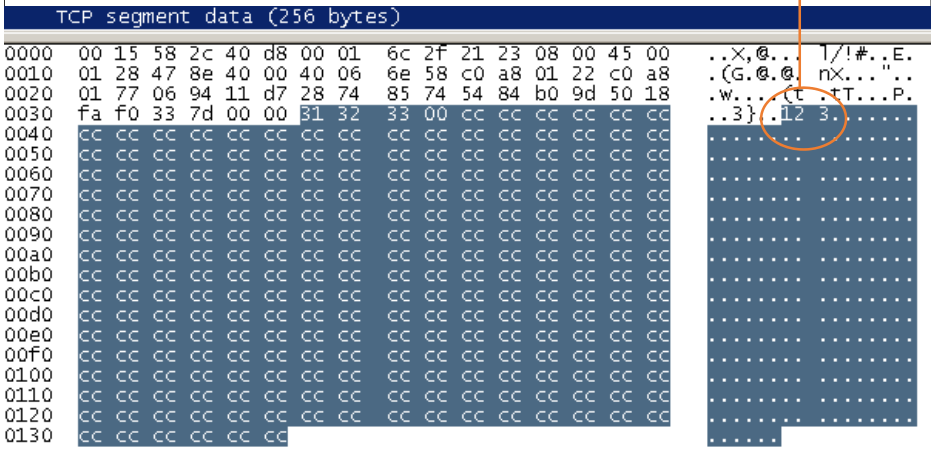
ACK=Y+1

```
Source port: 1684 (1684)
Destination port: 4567 (4567)
Sequence number: 1    (relative sequence number)
[Next sequence number: 257    (relative sequence number)]
Acknowledgement number: 1    (relative ack number)
Header length: 20 bytes
Flags: 0x0018 (PSH, ACK)
  0... .... = Congestion Window Reduced (CWR): Not set
  .0.. .... = ECN-Echo: Not set
  ..0. .... = Urgent: Not set
  ...1 .... = Acknowledgment: Set
  .... 1... = Push: Set
  .... .0.. = Reset: Not set
  .... ..0. = Syn: Not set
  .... ...0 = Fin: Not set
Window size: 256960 (scaled)
Checksum: 0x337d [correct]
TCP segment data (256 bytes)
```

数据内容见下页图

TCP segment data(256 bytes)

这是第一次发送的数据123



5 S→C ACK
SEQ=Y+1
ACK=X+257

第一次传输数据结束

```
Source port: 4567 (4567)
Destination port: 1684 (1684)
Sequence number: 1    (relative sequence number)
Acknowledgement number: 257    (relative ack number)
Header length: 20 bytes
Flags: 0x0010 (ACK)
  0... .... = Congestion Window Reduced (CWR): Not set
  .0.. .... = ECN-Echo: Not set
  ..0. .... = Urgent: Not set
  ...1 .... = Acknowledgment: Set
  .... 0... = Push: Not set
  .... .0.. = Reset: Not set
  .... ..0. = Syn: Not set
  .... ...0 = Fin: Not set
Window size: 65279
Checksum: 0x6075 [correct]
```

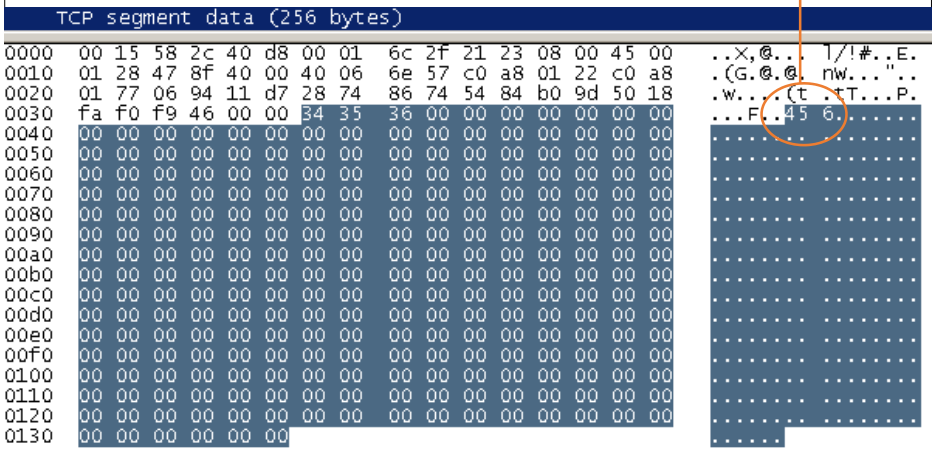

6 C→S PSH,ACK
SEQ=X+257, data length=256, next seq=513
ACK=Y+1

```
Source port: 1684 (1684)
Destination port: 4567 (4567)
Sequence number: 257    (relative sequence number)
[Next sequence number: 513    (relative sequence number)]
Acknowledgement number: 1    (relative ack number)
Header length: 20 bytes
Flags: 0x0018 (PSH, ACK)
  0... .... = Congestion Window Reduced (CWR): Not set
  .0.. .... = ECN-Echo: Not set
  ..0. .... = Urgent: Not set
  ...1 .... = Acknowledgment: Set
  .... 1... = Push: Set
  .... .0.. = Reset: Not set
  .... ..0. = Syn: Not set
  .... ...0 = Fin: Not set
Window size: 256960 (scaled)
Checksum: 0xf946 [correct]
TCP segment data (256 bytes)
```

数据内容见下页图

TCP 报文段数据 (256 bytes)

这是第二次发送的数据456



7 S→C ACK
SEQ=Y+1
ACK=X+513

第二次传输数据结束

```
Source port: 4567 (4567)
Destination port: 1684 (1684)
Sequence number: 1    (relative sequence number)
Acknowledgement number: 513    (relative ack number)
Header length: 20 bytes
Flags: 0x0010 (ACK)
  0... .... = Congestion Window Reduced (CWR): Not set
  .0.. .... = ECN-Echo: Not set
  ..0. .... = Urgent: Not set
  ...1 .... = Acknowledgment: Set
  .... 0... = Push: Not set
  .... .0.. = Reset: Not set
  .... ..0. = Syn: Not set
  .... ...0 = Fin: Not set
Window size: 65023
Checksum: 0x6075 [correct]
```


8 C→S FIN,ACK
SEQ=X+513
ACK=Y+1

```
Source port: 1684 (1684)
Destination port: 4567 (4567)
Sequence number: 513    (relative sequence number)
Acknowledgement number: 1    (relative ack number)
Header length: 20 bytes
Flags: 0x0011 (FIN, ACK)
  0... .... = Congestion Window Reduced (CWR): Not set
  .0.. .... = ECN-Echo: Not set
  ..0. .... = Urgent: Not set
  ...1 .... = Acknowledgment: Set
  .... 0... = Push: Not set
  .... .0.. = Reset: Not set
  .... ..0. = Syn: Not set
  .... ...1 = Fin: Set
Window size: 256960 (scaled)
Checksum: 0x6383 [correct]
```


9 S→C ACK
SEQ=Y+1
ACK=X+514

```
Source port: 4567 (4567)
Destination port: 1684 (1684)
Sequence number: 1    (relative sequence number)
Acknowledgement number: 514    (relative ack number)
Header length: 20 bytes
Flags: 0x0010 (ACK)
  0... .... = Congestion Window Reduced (CWR): Not set
  .0.. .... = ECN-Echo: Not set
  ..0. .... = Urgent: Not set
  ...1 .... = Acknowledgment: Set
  .... 0... = Push: Not set
  .... .0.. = Reset: Not set
  .... ..0. = Syn: Not set
  .... ...0 = Fin: Not set
Window size: 65023
Checksum: 0x6074 [correct]
```


10 S→C FIN,ACK
SEQ=Y+1
ACK=X+514

```
Source port: 4567 (4567)
Destination port: 1684 (1684)
Sequence number: 1    (relative sequence number)
Acknowledgement number: 514    (relative ack number)
Header length: 20 bytes
Flags: 0x0011 (FIN, ACK)
  0... .... = Congestion Window Reduced (CWR): Not set
  .0.. .... = ECN-Echo: Not set
  ..0. .... = Urgent: Not set
  ...1 .... = Acknowledgment: Set
  .... 0... = Push: Not set
  .... .0.. = Reset: Not set
  .... ..0. = Syn: Not set
  .... ...1 = Fin: Set
Window size: 65023
Checksum: 0x6073 [correct]
```

11 C→S ACK
SEQ=X+514
ACK=Y+2

TCP连接已经断开

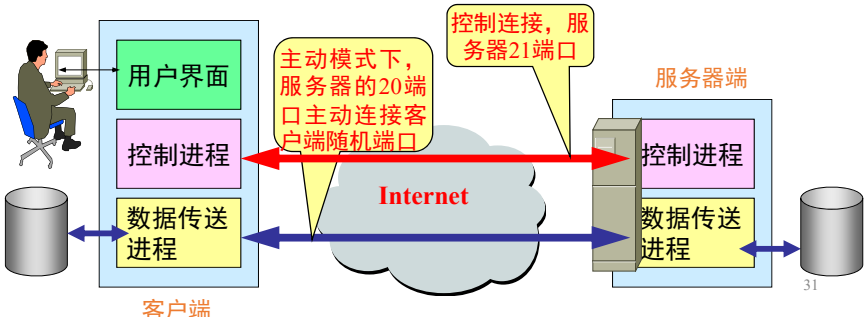
```
Source port: 1684 (1684)
Destination port: 4567 (4567)
Sequence number: 514    (relative sequence number)
Acknowledgement number: 2    (relative ack number)
Header length: 20 bytes
Flags: 0x0010 (ACK)
  0... .... = Congestion Window Reduced (CWR): Not set
  .0.. .... = ECN-Echo: Not set
  ..0. .... = Urgent: Not set
  ...1 .... = Acknowledgment: Set
  .... 0... = Push: Not set
  .... .0.. = Reset: Not set
  .... ..0. = Syn: Not set
  .... ...0 = Fin: Not set
Window size: 256960 (scaled)
Checksum: 0x6382 [correct]
```


FTP的访问方式

- FTP服务分为普通FTP与匿名FTP服务两种类型
  - 普通FTP服务要求用户在登录时提供正确的用户名和用户密码
  - 匿名FTP服务的实质是：提供服务的机构在它的FTP服务器上建立一个公开账户（一般为anonymous），并赋予该账户访问公共目录的权限。如果用户要访问这些提供匿名服务的FTP服务器，可以直接访问而不需要密码。有些FTP服务器可能会要求用户用自己的电子邮件地址作为用户密码
- 为了保证FTP服务器的安全，几乎所有的匿名FTP服务器都只允许用户下载文件，而不允许用户上载文件。

FTP的双重连接

- **FTP在客户和服务器之间要建立双重TCP连接**
- **一条由客户端发起的“控制连接”（21），用来传输FTP命令，在整个会话期间一直保持打开**
- **一条是FTP服务器端发起的“数据连接”（主动模式下是FTP服务器的20端口主动联络客户端的PORT N中的客户端口N，被动模式下是客户端的某个随机端口联络服务器在PASV应答中给出的服务器端口），用来传输FTP数据**



主进程的工作步骤

(1) 服务器主进程在端口号21启动，并等待客户进程连接

(2) 客户进程发出连接请求

(3) 服务器主进程启动从属进程处理客户进程发来的请求。从属进程对客户进程的请求处理完毕后即终止，但从属进程在运行期间根据需要还可能创建其他一些子进程

(4) 进程回到等待状态，继续接受其他客户进程发来的请求。主进程与从属进程的处理是并发地进行

Windows下的FTP客户程序

- 命令行方式：在Windows下键入ftp命令即可打开MS-DOS方式的ftp交互窗，提示符为 ftp >。在ftp交互窗中可以实用各种ftp子命令
- 图形用户界面的FTP客户软件：因特网上提供了多种FTP客户程序
- 浏览器：在Windows下的浏览器软件可用来访问FTP服务器。注意，必须在地址栏输入FTP服务器的URL，如：ftp://abc@211.80.184.8

FTP客户端程序

常用的FTP客户端程序通常有几种类型

- 传统的FTP命令行
- 浏览器
- 资源管理器
- FTP下载工具

常用的FTP交互命令

命令	命令格式	意义
get	get file1 file2	将文件file1下载到本地，并改名为file2
put	put file1 file2	将文件file1上传到服务器，并改名为file2
ls	ls	显示当前目录下的文件
cd	cd abc	进入abc目录
rename	rename file1 file2	将文件file1改名为file2
?	? user	显示user命令的功能
!	!	进入本地操作系统界面（exit返回ftp）
quit	quit	退出ftp

命令行方式使用FTP的常用命令

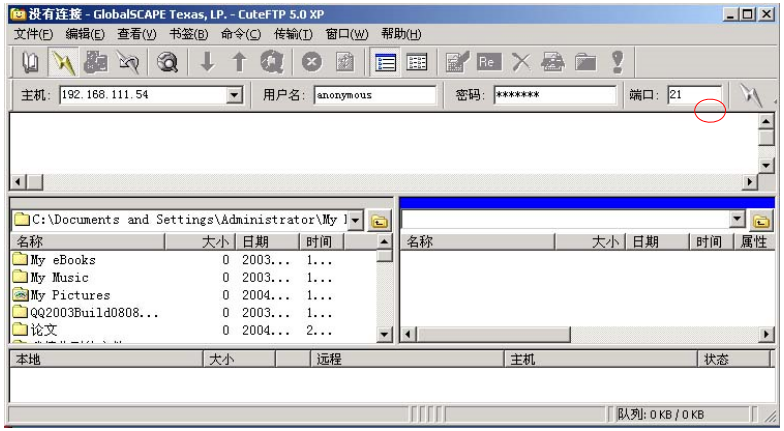
ftp 主机名
FTP>bye 结束与远程计算机的FTP会话并退出ftp
FTP>cd 更改远程计算机上的工作目录
FTP>delete 删除远程计算机上的文件
FTP>dir 显示远程目录文件和子目录列表
FTP>get 使用当前文件转换类型将远程文件复制到本地计算机
格式：get remote-file [local-file]
FTP>help [command]
FTP>ls 显示远程目录文件和子目录的缩写列表
FTP>mkdir 创建远程目录
FTP>mls 显示远程目录文件和子目录的缩写列表
FTP>mput 使用当前文件传送类型将本地文件复制到远程计算机上
FTP>put 使用当前文件传送类型将本地文件复制到远程计算机上
FTP>pwd 显示远程计算机上的当前目录
FTP>quit 结束与远程计算机的FTP会话并退出ftp
FTP>rmdir 删除远程目录
FTP>user 指定远程计算机的用户

FTP的open命令

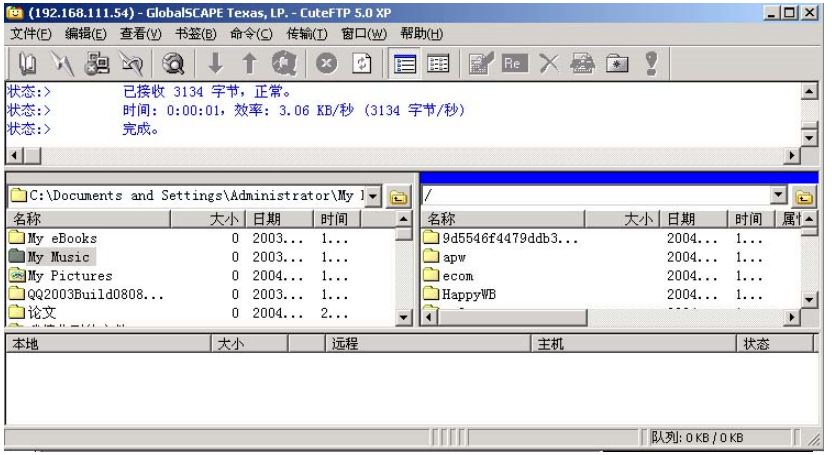
利用浏览器访问FTP站点

- 若要访问的FTP站点为匿名站点，在浏览器的地址栏输入“ftp://FTP站点的IP地址或DNS域名”
- 如果FTP站点提供的是用户访问的方法，在浏览器的地址栏中需要添加用户名和密码信息，格式为：“ftp://用户名：密码@FTP站点的IP地址或DNS域名”。也可以按照匿名访问的方法进行访问，IE浏览器会自动弹出登录身份窗口，提示输入用户名和密码

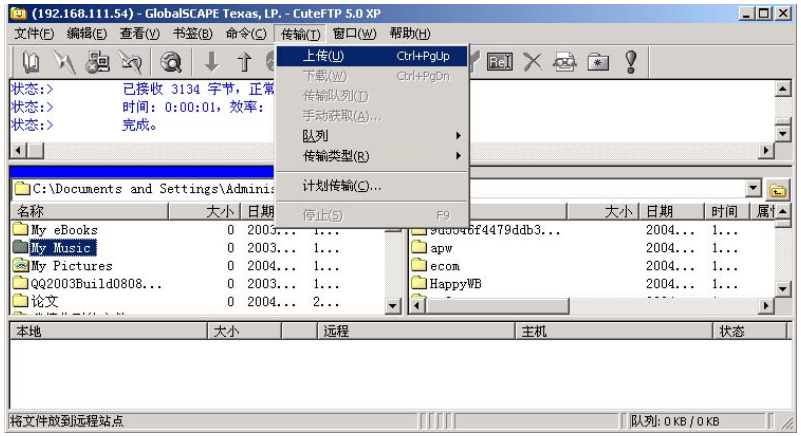
使用专门的FTP客户端软件（以cuteFTP为例 ）



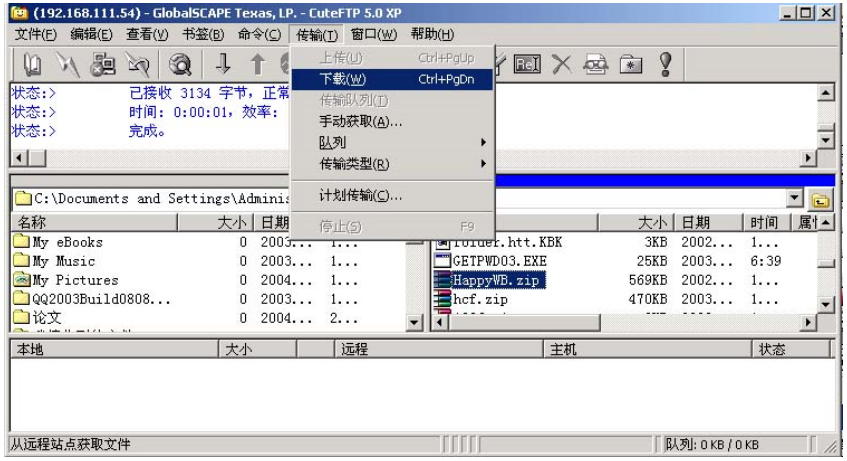
客户端已连接到FTP服务器



客户端上传文件到FTP服务器



将FTP服务器上的文件下载到客户端计算机



FTP常见应答码

应答代码	描述
125	打开数据连接，且此连接可用于数据传输
200	命令被成功执行
211	FTP 服务器忙
212	FTP 服务器返回当前的目录状态给客户端
213	FTP 服务器返回当前的文件状态给客户端
214	FTP 服务器返回用户请求的帮助信息
226	FTP 服务器返回文件传输完成的消息给客户端
331	FTP 服务器返回用户名正确，需要密码的消息给客户端
425	FTP 服务器返回不能打开数据连接的消息给客户端
452	FTP 服务器返回写文件错的消息给客户端，可能是空间不足
500	FTP 服务器返回客户端命令不能识别的消息给客户端
501	FTP 服务器返回客户端命令的参数不能识别的消息给客户端
502	FTP 服务器返回未实现的模式类型的消息给客户端

FTP实验

- 完成在线捕获报文段实验
  - 阅读教材“2.5 FTP协议 P64-69”内容。
  - 完成P51的实例2-1。
- 对ftp协议的两个捕包文件进行分析，并回答相关问题
  - 提示：Wireshark有统计、会话着色、追踪等功能
- 选做：在线捕获TCP报文段实验
  - 实现简单的TCP数据发送与接收（程序代码参见教材P97-101），捕获数据报文，过滤得到其中的数据发送和接收的TCP报文段，并进行分析。
  - 提示：Client和Server的IP地址使用实验室机器的IP地址，Client和Server可以位于同一台机器，也可以位于不同机器。