

文章编号: 2095-2163(2019)02-0126-04

中图分类号: TP311

文献标志码: A

# 流模式下 FTP 文件传输效率分析及改进

张欣艳

(江西财经大学 网络信息中心, 南昌 330013)

**摘要:** 现有的 FTP 实现主要基于标准的流模式传输文件。本文首先分析表明, FTP 流传输在连续传输多个文件时, 由于重复地进行数据连接的建立、释放操作会导致传输效率低下。然后提出一种自动打包的文件传输模式, 将多个文件传输转换为一个文件传输, 以解决流模式效率偏低的问题, 实验结果表明自动打包文件传输模式有效。

**关键词:** FTP; 流模式; 效率分析; 打包传输

## Efficiency analysis and improvement of FTP file transmission on stream mode

ZHANG Xinyan

(Network & Information Administration Center, Jiangxi University of Finance and Economics, Nanchang 330013, China)

**[Abstract]** The existing FTP implementations are mainly based on standard stream mode to transfer files. Firstly, the analysis shows that when FTP stream transfers multiple files continuously, the transmission efficiency is low due to repeated data connection establishment and release operations. Then a file transmission mode of automatic packaging is proposed, which converts multiple file transmission into one file transmission in order to solve the problem of low efficiency of stream mode. The experimental results show that the file transmission mode of automatic packaging is effective.

**[Key words]** FTP; stream mode; efficiency analysis; packaging transmission

### 0 引言

文件传输协议 (File Transfer Protocol, FTP) 可通过计算机网络进行计算机系统之间的文件传送。基于 FTP 协议进行的网络文件传输服务简称 FTP 服务。目前网络上大量共享资源的发布仍然基于 FTP 服务实现。企事业单位内部也常利用 FTP 服务来完成文件的备份或共享。FTP 会话中用于文件传输的数据连接与用于交互的控制连接独立, 需要传送文件时, 通常每个文件的传送都要经历一次数据连接的建立、使用和释放过程, 这种工作机制在特定场景下会带来效率问题。本文将分析这种特定场景及其效率低下的原因, 并着重提出一种新的 FTP 文件传输模式来解决这一问题。

### 1 FTP 服务实现基本原理

在 TCP/IP 协议族中, FTP 是应用层协议之一。FTP 协议最早提出是在 RFC 114 文档中, 现行标准是基于 1985 年的 RFC 959<sup>[1]</sup>。FTP 设计的目标是在同构或异构远程计算机系统之间提供非直接、可靠的透明传输服务, 以促进资源共享。如图 1 所示, FTP 是基于客户端/服务端架构 (简称 C/S 模型) 设计。客户端与服务端之间需要 2 种通信链路来组建

相应的服务, 即: 控制连接与数据连接。其中, 控制连接用于传输客户端发送的用户信息、控制命令及服务器反馈的应答等交互信息。数据连接用于传输数据, 即主要针对客户端与服务端之间的目录信息及文件数据的传输。为了保证数据传输的可靠性, 控制连接和数据连接都是基于 TCP 协议实现。

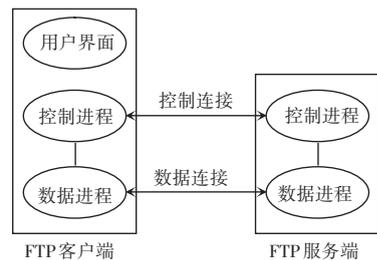


图 1 FTP 使用模型

Fig. 1 FTP usage model

客户端与服务端的一次会话是从建立一条控制连接开始, 并伴随该条控制连接的释放而结束。FTP 服务开始时, 双方首先要建立控制连接, 控制连接在整个会话期间都保持打开, 此后在控制连接上就是双方彼此间的交互, 并以命令请求与应答方式完成登录、目录管理、文件管理、文件上传 (客户端向服务端传输文件) 或下载 (服务端向客户端传输文件) 等各种操作。会话期间有目录信息或文件传

作者简介: 张欣艳 (1976-), 女, 硕士, 工程师, 主要研究方向: 信息技术应用。

收稿日期: 2018-11-20

输需要时,双方才建立数据链路,进行指定内容的传输。根据数据链路的建立方式,FTP 服务的工作模式可分为主动方式和被动方式。这 2 种方式的主要区别在于数据连接建立过程中,主动发起一方是哪一端。主动方式下由服务端先向客户端指定端口发起相应的连接建立过程,而被动方式下由客户端先向服务端指定端口发起数据连接的建立过程。

## 2 FTP 协议传输效率分析

在文件传输模式上,FTP 标准提出了流模式、块模式及压缩模式三种。简言之,第一种实现简单,后两种较为复杂。目前所有 FTP 实现都支持流模式,对其余 2 种模式则几乎都未给予支持。流模式下数据连接按需创建,用完释放,连接释放即表示文件传输结束。客户端和服务端在一次会话期间每当需要传输一个文件时,双方将启动 3 次握手过程建立新的数据连接,之后才转入数据传输,而数据传输后通过释放连接就表示文件传输结束<sup>[2-3]</sup>。研究可知,在流模式和主动方式下从服务端下载一份文件时的协议工作流程详见图 2。当有多个文件需要传输时,就需要不断重复该流程,每个文件重复一次。

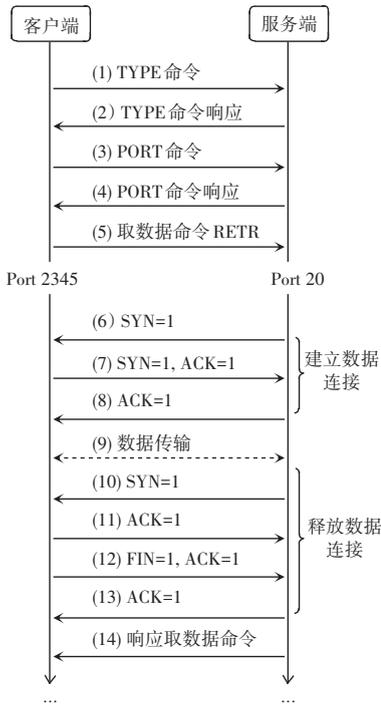


图 2 流模式下的一次数据传输过程

Fig. 2 A data transfer process in streaming mode

下面拟对如上这种流传输模式的工作效率展开分析。设 FTP 服务双方之间的链路带宽为  $e$  字节/s,即发送速率,链路传播时延为  $p$  s,沿途排队时延累计为  $d$ ,沿途系统内部处理时延累计为  $c$ ,则双方

之间的往返时间(Round Trip Time,  $RTT$ ) 的计算公式可表示为:

$$RTT \approx 2(p + d + c), \quad (1)$$

设每次传输的文件大小平均为  $a$  字节、携带命令消息的报文平均大小为  $b$  字节。在网络无差错时, TCP 连接传输  $a$  字节大小的文件可视为整体一次性发出、一次性接收、一次性确认,所以一个文件数据的平均传输时间的计算可使用如下公式:

$$d_1 = a/e + RTT, \quad (2)$$

同样道理,在 TCP 连接下发送一条命令或命令响应消息的平均传输时间的计算可使用如下公式:

$$d_2 = b/e + RTT, \quad (3)$$

一般地,当网络稳定时,  $RTT$  值稳定不变。

由图 2 可知,一次文件传输过程包含了依序进行的 13 条命令或响应消息的传输(称为协议处理部分)和文件本身数据的一次传输(称为数据传输部分)。这里,把链路利用率  $u$  定义为数据传输部分所用时间在一次文件传输过程中的占比,即流模式下  $u$  的计算可参考如下公式:

$$u = d_1 / (d_1 + 13 * d_2). \quad (4)$$

其中,  $d_1, d_2$  的计算见式(2)、式(3)。

为了能直观刻画参数  $u$  与  $a$  之间的关系,需先确定公式(4)中其它参数取值。用 ping 命令分别测试了 3 种典型网络场景下的  $RTT$  值,对此阐述如下。

- (1) 单位校园网内部计算机之间的  $RTT = 1$  ms。
- (2) 单位内部计算机与百度门户网站 ([www.baidu.com](http://www.baidu.com)) 之间的  $RTT = 36$  ms。
- (3) 单位内部计算机与国外斯坦福大学官方网站之间的  $RTT = 239$  ms。

通过抓包软件分析可得,一般 FTP 命令及响应消息的报文平均大小为 80 字节,不妨固定下来,选取  $b = 80$  字节;目前,国内计算机网络端口已普及到百兆速率,根据实际线路状况可取典型值  $e = 70$  Mbps。根据以上参数值,就可绘出  $u$  随文件大小  $a$  的变化曲线,具体如图 3 所示。

由图 3 可知,在近距离 ( $RTT = 1$  ms) 场景下,传输较小文件(例如超过 64 KB)时,传输所用时间的实际利用率不超过 40%,也就是 60%的时间用于传输协议本身的开销;一般场景 ( $RTT = 36$  ms) 下,即使是传输常见大小文件(例如 4 MB 大小),FTP 链路利用效率也仅在 50%左右;如果是长远距离场景或网络拥挤时,随着  $RTT$  值急剧增大(例如  $RTT = 239$  ms),即使传输较大文件(例如 64 M),链路利用效率也仅为 70%左右。综上 3 种场景下的对比分析

结果表明,流传输模式下传输多个文件时线路利用效率与文件大小成正比,与  $RTT$  成反比。由此即可推出,FTP 流模式在一次会话中,批量传输多个中小文件时效率将非常低下,随着未来传输带宽的提高,这一问题也将变得更为突出。

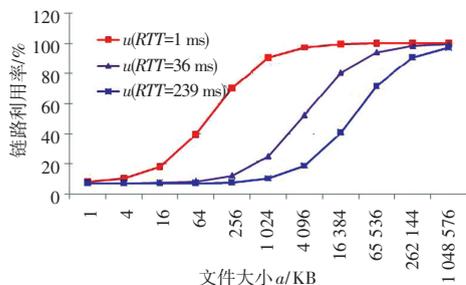


图3 链路利用率与文件大小及  $RTT$  的关系

Fig. 3 Link utilization versus file size and  $RTT$

### 3 基于打包的文件传输模式

目前,已有 FTP 实现已在致力于如何提高客户计算机与服务端计算机之间的总吞吐速率,常用方法有 2 种,即:通过并发连接进行并行传输<sup>[4]</sup>;在数据连接上实时解压缩数据<sup>[5]</sup>。这些方法却仍未能解决前文提出的一次会话期间批量传输多个文件时链路利用率低下的问题。

要提高一次会话中客户端和服务端之间的链路利用效率,可尝试在同一条数据连接上持续传输多个文件,以节约协议处理部分所占用的时间。RFC 959 中特别指出:块模式或压缩模式应通过专门的结束标识符通知接收方,此次文件传输结束,而不需要立即结束当前连接,从而可以做到数据连接的重复利用。但目前已有的 FTP 实现中尚未发现利用同一数据连接传输不同文件的应用实例。

针对客户端和服务端会话期间批量传输文件的需求,本文提出基于打包的文件传输模式,如图 4 所示。打包文件传输模式在系统中增加了 2 个命令,分别是:批量上传(MSTOR)和批量下载(MRETR)。无论上传、还是下载,批量文件传输总是由客户端发起。批量上传时的处理步骤可分述如下。

(1)本地打包。将用户指定的多个文件封装为一个包文件,包文件大小等于原文件大小之和。

(2)向服务端传输包文件。

(3)服务端接收包文件并在当前位置释放出原有文件。

接下来,批量下载时的处理步骤可分述如下。

(1)客户端向服务端传送用户要下载的所有文

件信息。

(2)服务端在本地将客户指定下载的文件封装为一个包文件。

(3)服务端向客户端传输包文件。

(4)客户端接收包文件并在当前位置释放出原有文件。

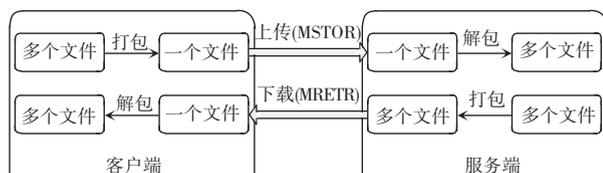


图4 基于打包的文件传输模式

Fig. 4 Packed file transfer mode

由于只会经历一个文件传输过程,批量上传(MSTOR)命令协议处理部分的时间开销,将与执行一次传统 FTP 命令 STOR 一致;稳定带宽下,包文件数据部分的传输时间与所有原文件数据部分传输时间的总和接近,故而 MSTOR 命令将使得多个文件传输仅调用了一次协议处理流程,从而提高了效率。

批量下载(MRETR)命令启动后,客户端首先向服务端传输“客户指定了哪些文件”的信息,为此需要一次额外的数据传输服务,后续步骤则与批量上传类似,也就是服务端打包—传输文件—用户端解包三个步骤,协议处理流程如同执行了一次传统的 FTP 命令 RETR,因此总体上批量下载(MRETR)命令最多需要 2 次协议处理时间开销,即可完成多个文件下载任务,从而提高了效率。

### 4 实验验证

为了验证打包式批量传输模式的有效性,实现了一个面向 Windows 平台的实验性 FTP 服务系统。系统由客户端软件和服务端软件两部分组成。总的来说,服务器端采用多线程并发服务架构,基于 Windows 下的 Socket 编程技术和 MFC 类库开发,具有用户管理、服务配置、目录操作、文件上传与下载等基本功能。客户端采用 WinINET API 开发,具有远程连接、文件浏览及选择、上传或下载指定文件等基本功能。

实验系统在设计时修改了 FTP 协议标准的文件上传处理流程,以达到打包批量传输目的。当用户在客户端选择多个文件进行上传时,客户端将在本地自动把所选文件打包为一个名为 test.zip 的 zip 格式包文件,向服务端传送,服务端接收到文件 test.zip 后,在当前位置进行解包,释放出原有文件。系统中打包和解包操作是通过调用开源的 zip/unzip

工具包及对应类库来实现的,打包操作中将压缩级别设为 0,即不压缩仅进行封装,使得打包和解包用时极少。

在不同物理位置选择了 2 台主机,分别安装客户端和服务端软件。客户端本地准备了大小分别为 1 KB、2 KB、100 KB 及 1 MB 的 4 类文件各 100 个,4 类文件分别存放到 4 个不同文件夹中,作为上传测试数据。同时选择在深夜网络负载极低的时间段进行测试,先用 ping 命令测得主机之间  $RTT = 28 \text{ ms}$ ,然后分别进入各文件夹,选中文件夹下所有文件向服务端事先准备好的对应文件夹进行批量上传操作,记录下上传该类所有文件总共所用时间(单位:秒),测试过程中客户端与服务端之间带宽约为 1.78 Mbps。另外,选择了一个标准 FTP 系统与本文实验系统进行对比实验,测试结果见表 1。由表 1 可知,与标准 FTP 系统对比,实验系统完成同样的任务所用时间明显减少,这表明基于打包的文件传输模式在实现批量传输时的高效性,而且当文件越小时,效率提升越明显。

表 1 批量上传服务所用时间对比

Tab. 1 Time comparison of bulk upload services

	1 KB	2 KB	100 KB	1 MB
标准 FTP 系统	26	27	41	91
实验系统	<1	<1	5	59

(上接第 125 页)

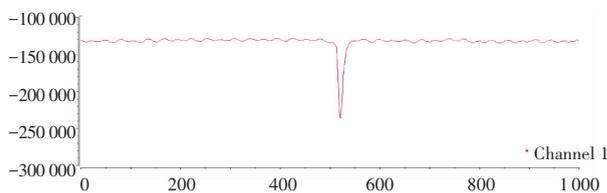


图 6 积分后的信号

Fig. 6 Signal after integration

## 5 结束语

本文提出了一种基于电容传感器的非接触式儿童安全座椅防遗忘检测方法,同时加入了报警系统,可以有效地提醒用户不要将儿童遗忘在车内,同时也保障了不慎被遗忘在车内的儿童的安全。与现有检测装置相比,本装置具有易于安装、可靠性高、安全性能强等优点,具有一定的推广价值。

## 5 结束语

现今,因特网上人们还在频繁使用 FTP 服务或协议进行资源的共享。本文提出的基于打包的批量文件传输模式,能有效减少一次 FTP 会话期间需要连续传送多个文件时的总体所用时间,所以在网络延迟相对发送时间比值大的场合,具有一定的应用价值。例如,在需要传输大量小文件或网络延迟大、带宽高的情况下使用此方法,效果则尤其显著。在 FTP 协议实际应用开发中,研发打包传输模式时还可利用流技术在内存中实现文件打包成流、流数据发送与接收、在内存中将流解包为文件的管道作业方式,从而最大化节约打包和解包操作额外带来的时间和空间消耗。

## 参考文献

- [1] Postel J, Reynolds J. File Transfer Protocol (FTP) [S]. Network Working Group, 1985.
- [2] 郝浩. FTP 原理解析[J]. 计算机与网络, 2016(14):40-41.
- [3] 葛伟伦. FTP 协议两种工作方式下消息交互的分析[J]. 电脑知识与技术, 2016, 12(8):30-31, 35.
- [4] 党海燕, 麻小娟, 詹金珍. 基于 FTP 协议多用户远程数据的访问[J]. 工业仪表与自动化装置, 2017(1):109-111.
- [5] Preston J. Deflate transmission mode for FTP[EB/OL]. [2010-06-30]. <https://tools.ietf.org/html/draft-preston-ftpext-deflate-04>.

## 参考文献

- [1] 黄林, 陈向东, 谢宁宁, 等. 基于 AD7150 微位移电容传感器的研究[J]. 半导体技术, 2012, 37(6):425-428, 441.
- [2] PARK J Y, YEE Y J, NAM H J, et al. Micro machined RF MEMS tunable capacitors using piezoelectric actuators [J]. IEEE MTT-S International Microwave Symposium digest. IEEE MTT-S International Microwave Symposium, 2001, 3(3):2111-2114.
- [3] ZHOU Yiqing, NG T S. Performance analysis on MIMO-OFCDM systems with multi-code transmission [J]. IEEE Transactions on Wireless Communications, 2009, 8(9):4426-4433.
- [4] 江子焯, 裴荣, 吴键. 基于柔性触觉传感器的坐姿压力监测系统的设计[J]. 传感器与微系统, 2018, 37(3):86-88, 91.
- [5] 贺杰, 柳晓飞. 基于 FDC2214 的空调器除霜控制技术[J]. 机电工程技术, 2017(S1):90-93.
- [6] 田旭峰, 董全林, 郭景昊. 一种电磁感应式位移传感器设计[J]. 传感器与微系统, 2018, 37(3):75-77.
- [7] 张思建, 林志赞, 颜钢锋. 基于电容传感器的架空输电线覆冰厚度检测方法[J]. 电力系统自动化, 2011, 35(17):99-102.
- [8] 孙林. 有介质包层的圆柱形平行双导线电容的计算[J]. 电工技术学报, 1995(1):71-74.
- [9] 李慧鹏, 唐若祥, 吕亚宁, 等. 基于电容传感器的精密压电微位移系统研究[J]. 半导体光电, 2018, 39(1):146-150.
- [10] 任艳娜, 冯志慧, 闻素红. 基于 I<sup>2</sup>C 总线的 AD7745 控制与读取方法[J]. 计算机测量与控制, 2012, 20(6):1544-1547, 1572.