用 W5500实现嵌入式 TFTP服务器



作者：高永彪，余育槐

一 实验背景

之前一段时间专门研究了固件升级的方法，主要是通过网页或者上位机软件实现远程固件升级。最近正好在研究 TFTP 简单文件传输协议，于是我就尝试给 设备添加联网功能，通过 TFTP 实现网络更新固件，而后发现这种升级方式所占 设备内存小，可以穿越多数防火墙，并且不需要去设备现场，在办公室通过网络 就能将成千上万用户或设备的固件升级，简单高效。

其实现在很多设备都已经具有网络固件升级功能，例如我们经常用到的电视机顶盒、家用无线路由器等设备。很多设备升级内核都是通过 TFTP 协议上传的，因为 TFTP 实现非常的简单，比如自己家里用的路由器就可以通过 TFTP 协议升级。

二 **TFTP** 基础普及

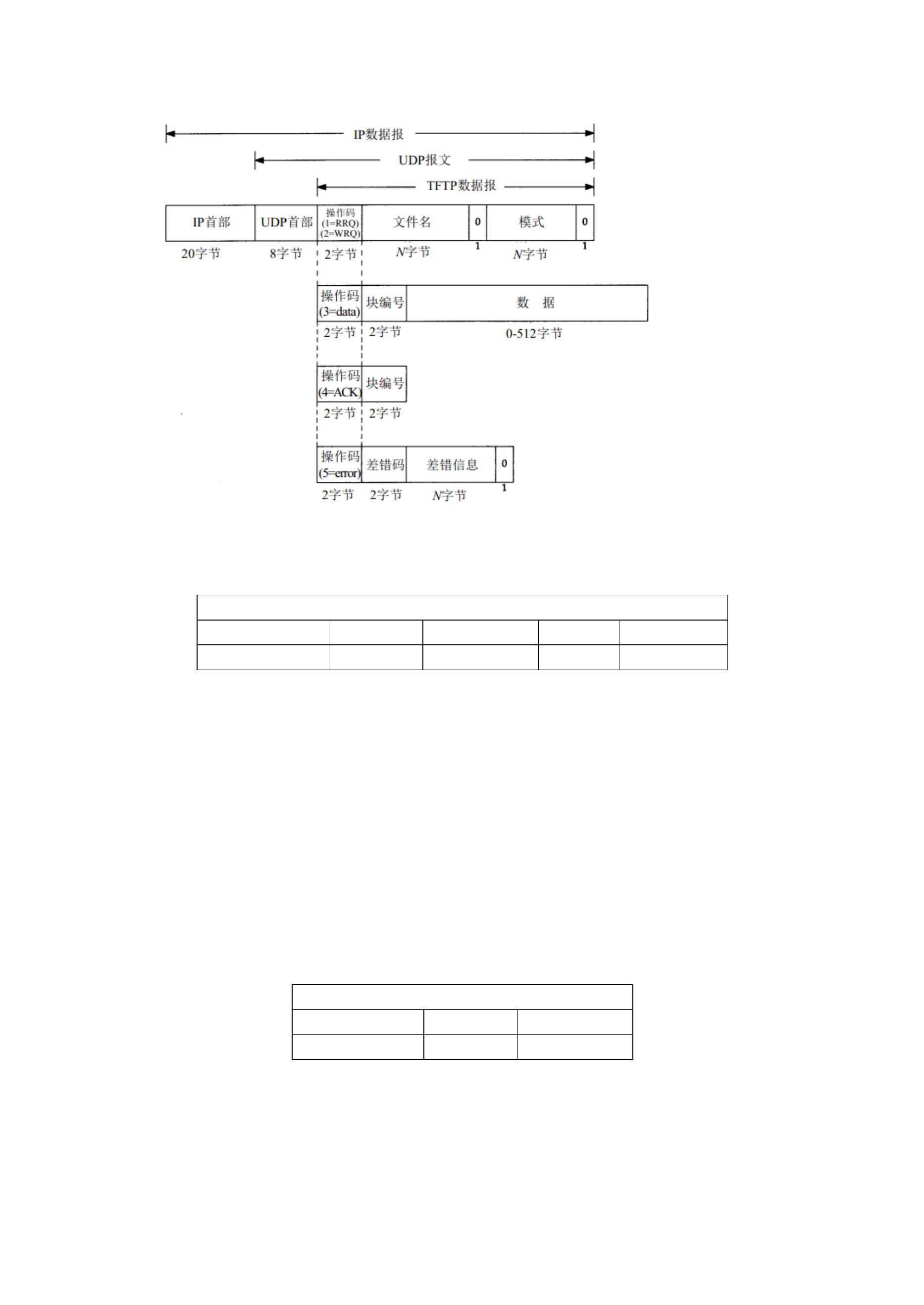
TFTP是 TCP/IP协议族中的一个用来在客户机与服务器之间进行简单文件传 输的协议，基于 UDP协议实现，端口号为 69。通过 TFTP协议，可以实现网络中 两台计算机之间的文件上传与下载，如文件备份，为无盘工作站下载引导文件， 下载初始化代码到打印机、集线器和路由器。当然，还有就是我们本次用到的对 设备进行固件升级。

TFTP 协议是专为小文件传输设计的，提供不复杂、开销不大的文件传输服务，缺乏标准 FTP 协议的许多特征。TFTP 只能从远程服务器上读、写文件（邮 件）或者读、写文件传送给远程服务器。它不能列出目录并且当前不提供用户认 证。当前 TFTP 有 3 种传输模式： netASCII 模式即 8 位 网络 ASCII 码 ； octet即八位组模式；邮件模式，这种模式现在已经废止不用了。主机双方也可 以自己定义其它模式。

TFTP基于 UDP协议实现，而 UDP使用 IP。因此一个 TFTP包中会有如图 1所示的以下几段：本地媒介头，IP 头，UDP 数据报头，TFTP 数据报。TFTP 在 IP头中不指定任何数据，但是它使用 UDP中的源和目标端口以及包长度域。由 TFTP

使用的包标记（TID）在这里被用做端口，因此 TID必须介于 0到 65,535之间。图中显示了 5种 TFTP报文格式，每个报文格式 TFTP报文的头两个字节表示操作 码。之后对于不同的报文格式存在差异。

图 1 TFTP报文格式



下面分别对每个报文包进行分析：

RRQ和 WRQ包的报文格式如表 1所示。

RRQ/WRQ包

Opcode Filename 0 Mode 0

2 bytes string 1 byte string 1 byte

表 1 RRQ和 WRQ包的报文格式

RRQ(读请求)报文由客户使用，用来建立一条从服务器读数据的连接。WRQ（写

请求）报文由客户使用，用来建立一条把数据写到服务器的连接，它的格式与

RRQ相同，RRQ包的操作码为 1，WRQ包的操作码为 2。Filename（文件名字段）

说明客户要读或写的位于服务器上的文件，文件名是 NETASCII码字符，以 0结 束。Mode(模式字段)是一个 ASCII码串 netascii或 octet（大小写可任意组合），同样以 0 字节结束。netascii 表示数据是以成行的 ASCII 码字符组成，以两个 字节—回车字符后跟换行字符（称为 CR / LF）作为行结束符。 OCTET模式用于 传输文件，这种文件在源机上以 8位格式存储。在使用 MAIL模式时，用户可以在 FILE处使用接收人地址，这个地址可以是用户名或用户名@主机的形式，如果是后一种形式，允许主机使用电子邮件传输此文件。如果使用 MAIL 类型，包必

须以 WRQ开始，否则它与 NETASCII完全一样。DATA包的报文格式如表 2所示。

DATA包

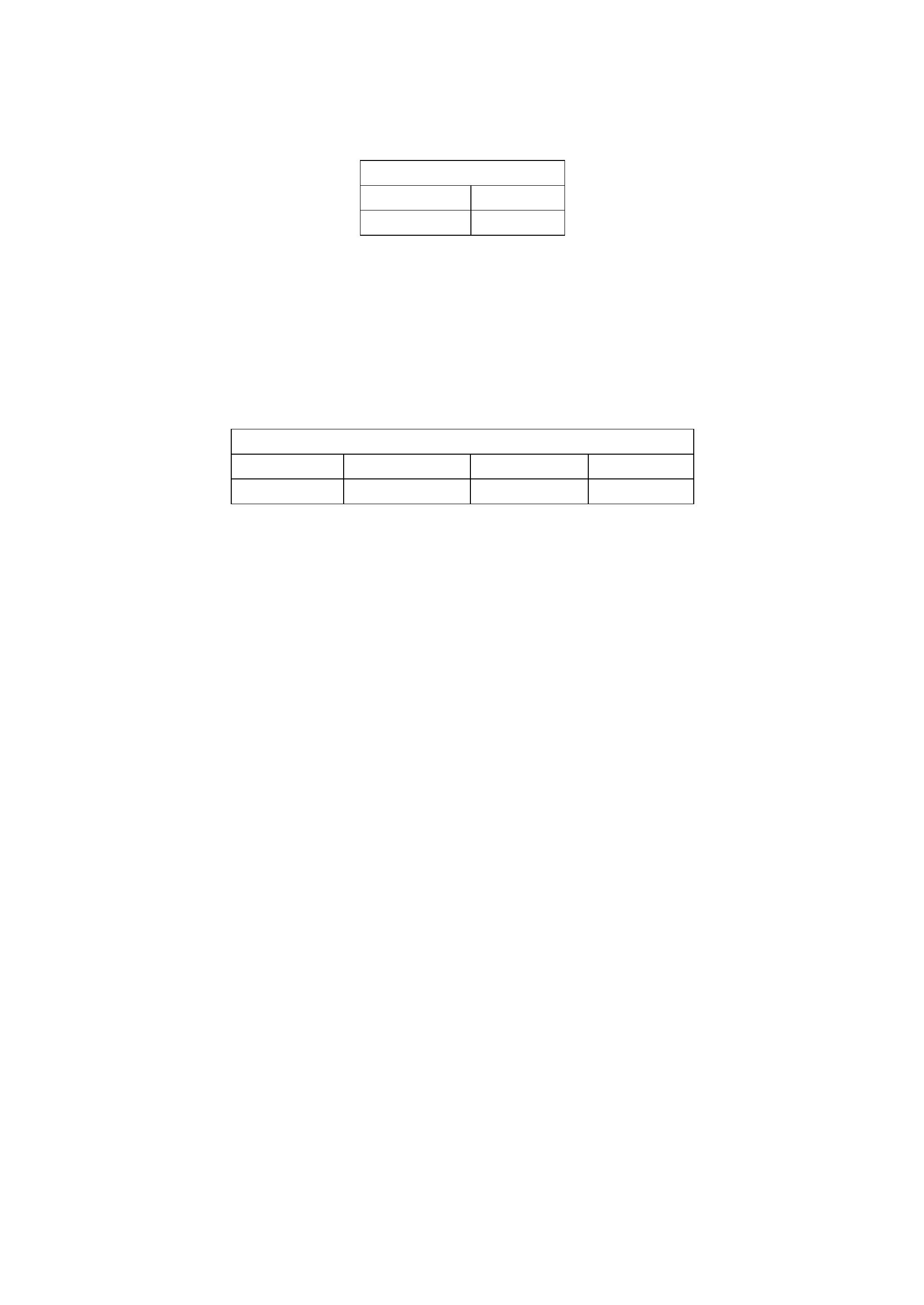
Opcode Block 0

2 bytes 2 bytes Data

表 2 DATA包的报文格式

DATA 数据包的 opcode 为 3，它还包括有一个数据块号和数据。数据块号域从 1开始编码，每个数据块加 1，这样接收方可以确定这个包是新数据还是已经 接收过的数据。数据域从 0字节到 512字节。如果数据域是 512字节则它不是最 后一个包，如果小于 512字节则表示这个包是最后一个包。如果最后一个包正好

为 512字节，则再发送一个 0字节的包用于表示结束。



ACK包的报文格式如表 3所示。

ACK包

Opcode Block

2 bytes 2 bytes

表 3 ACK包的报文格式

ACK包用于确认数据包已收到。ACK包的操作码为 4。当接收方收到一个数据 包后，会向发送方发送一个 ACK包；而发送方则会在收到一个 ACK包后继续发送下一个包。若发送完未能收到 ACK包，则会使用超时机制，重新发送刚才的数据 包。除了 ACK和用于中断的包外，其它的包均需得到确认。发出新的数据包等于确认上次的包。WRQ和 DATA包由 ACK或 ERROR数据包确认，而 RRQ数据包由 DATA或 ERROR数据包确认。

ERROR包的报文格式如表 4所示。

ERROR包

Opcode ErrorCode ErrMsg 0

2 bytes 2 bytes string 1 byte

表 4 ERROR包的报文格式

一个 ERROR包的操作码是 5。此包可以被其它任何类型的包确认,错误码指定

错误的类型。它用于服务器不能处理读请求或写请求的情况。在文件传输过程中

的读和写差错也会导致传送这种报文，接着停止传输。差错编号字段给出一个数

字的差错码，跟着是一个 ASCII表示的差错报文字段，可能包含额外的操作系统

说明的信息。错误的值和错误的意义如下：

0 未定义，请参阅错误信息1．文件未找到

2．访问非法

3．磁盘满或超过分配的配额

4．非法的 TFTP操作

5．未知的传输 ID

6．文件已经存在

7．没有类似的用户

三 **TFTP** 嵌入式系统实现方法

TFTP 协议执行过程中，任何一个传输进程都以 WRQ（请求写入远程系统）或 RRQ（请求读取远程系统）开始，收到一个确定应答并建立一个连接。创建连接时，通信双方随机选择一个 TID，因为是随机选择的，因此两次选择同一个 ID 的可能性就很小了。每个包包括两个 TID，发送者 ID和接收者 ID。这些 ID用于 在 UDP 通信时选择端口，在第一次请求的时候它会将请求发到 TID 69，也就是服务器的 69 端口上。应答时，服务器使用一个选择好的 TID 作为源 TID，并用上一个包中的 TID作为目的 ID进行发送。这两个被选择的 ID在随后的通信中会 被一直使用。

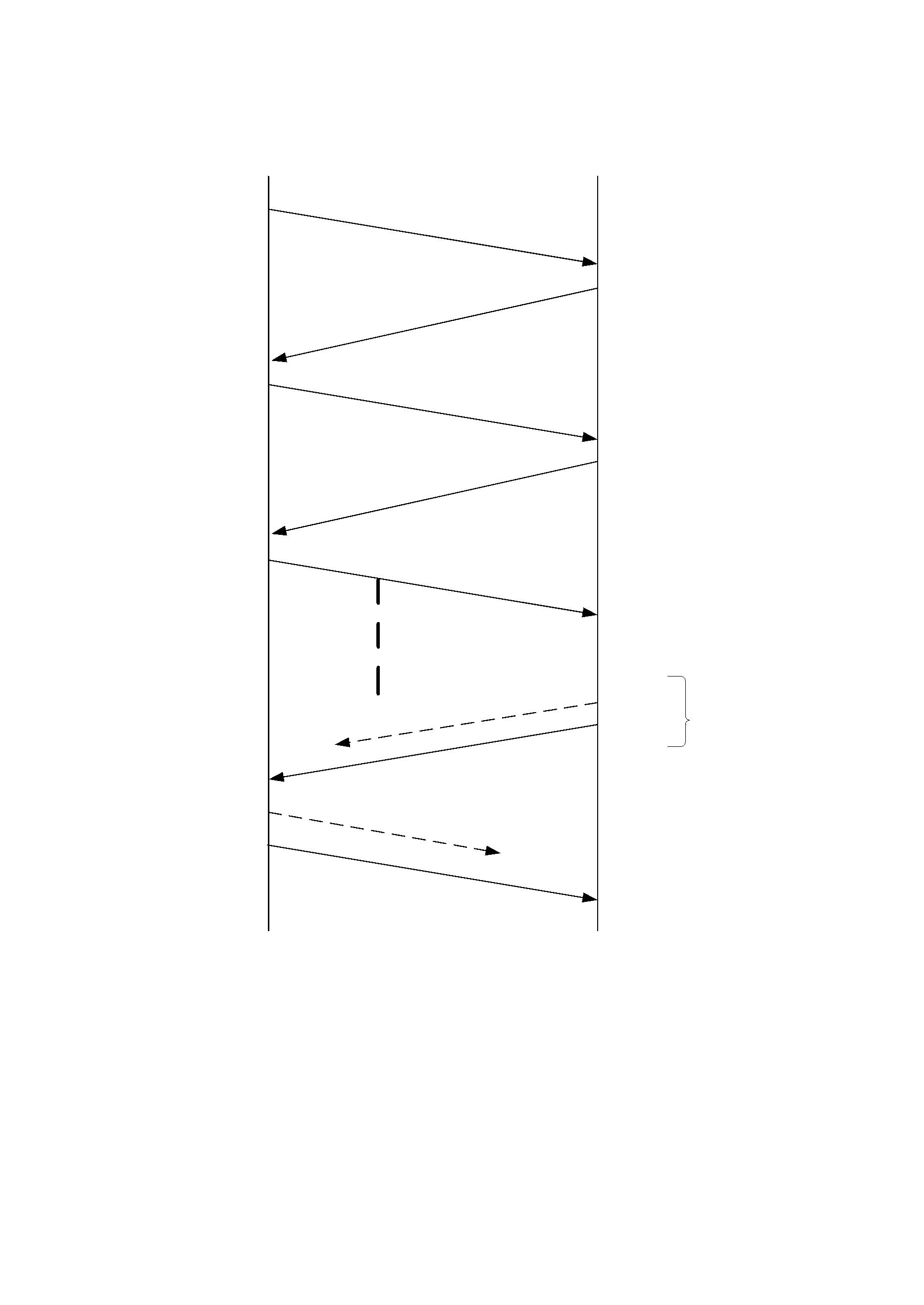
连接成功以后文件就以固定的 512 字节块的长度进行传送。每个数据包都包含一个数据块，块号从 1开始而且是连续的。因此对于写入请求的确定是一个

比较特殊的情况，因此它的包的包号是 0。在发送下一个包之前，数据块必须得 到确认响应包的确认。如果一个数据包的大小小于 512字节，则表明传输结束。 如果包在网络中丢失，接收端就会在超时以后重新传输最后一个未被确认的数据 包（可能是数据也可能是确认响应），这就导致丢失包的发送者重新发送丢失包。 通信的双方都是数据的发出者与接收者，一方传输数据接收应答，另一方发出应 答接收数据。发送者需要保留一个包在手头用于重新发送，由 LOCK 确认响应保证所有过去的包都已经收到。大部分的错误会导致连接中断，错误由一个错误的数据包引起。这个包不会被确认，也不会被重新发送，因此另一方无法接收到。如果错误包丢失，则使用超时机制。错误主要是由下面三种情况引起的：不能满足请求，收到的数据包内容错误(不能由延时或重发解释)，对需要资源的访问丢 失（如硬盘满）。TFTP只在一种情况下不中断连接，这种情况是源端口不正确， 在这种情况下，指示错误的包会被发送到源机。这个协议限制很多，这都是为了 实现起来比较方便而进行的。



TFTP 的工作过程很像停止等待协议，发送完一个文件块后就等待对方的确 认，确认时应指明所确认的块号。发送完数据后在规定时间内收不到确认就要重 发数据 PDU（协议数据单元），发送确认 PDU 的一方若在规定时间内收不到下一 个文件块，也要重发确认 PDU。这样保证文件的传送不致因某一个数据报的丢失 而告失败。通过下边的图片来了解 TFTP 协议的通信流程：

TFTP客户端 TFTP服务器



Sendto

读写请求

Recvfrom

Sendto

回应

Recvfrom

Sendto ACK

Recvfrom

Sendto 数据包

Recvfrom

Sendto ACK

Recvfrom

Sendto

数据包（丢失）

TimeoutSendto

数据包

Recvfrom

Sendto

ACK（丢失）

Sendto

ACK

Recvfrom

图 2 TFTP通信流程

四 测试 **TFTP** 客户端

了解了 TFTP协议之后，下面就让我们通过 WIZnet W5500EVB做一个嵌入式TFTP客户端的简单实验。

1.实验目的：建立一个 TFTP客户端

2.硬件环境：板载 LED

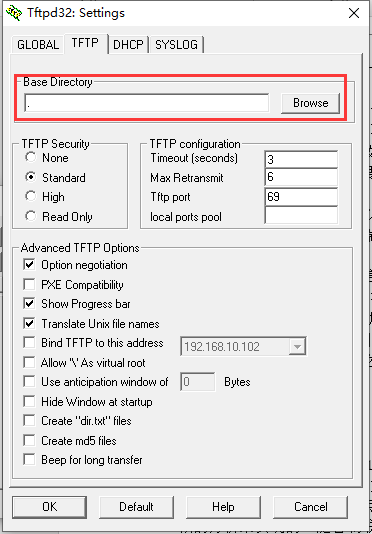
3.开发工具: MDK5（版本不一样，需要稍加改动）

4.测试软件：串口调试助手，TFTP32（可从网络下载）,tool文件夹中有。

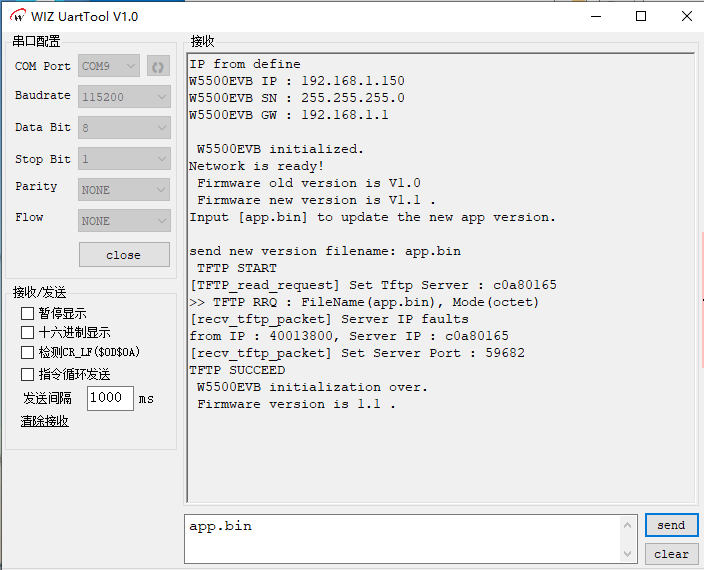
下面以W5500为TFTP客户端，讲述如何测试实现TFTP通信过程。

1. 在网上下载Tftp32软件，不需安装直接点击Tftpd32软件就可以应用。

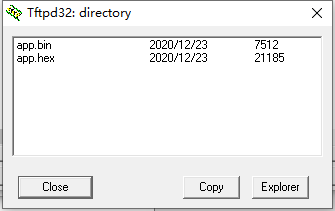
2. 配置TFTP服务器信息。如图3所示，Setting——>TFTP。接着在Base 工作。一定注意需下载的文件路径要与你的文件位置保持一致，否则服务器找不到文件而提示错误。



3.串口发送文件的名字，可以看到如图4所示文件下载过程。发送文件名为app.bin,接着就是TFTP服务器与客户端之间文件传输过程，如果传输成功会提示TFTP SUCCEED信息。



4.然后观察TFTP32软件提示信息，如图所示。点击Show Dir弹出Tftp32 Directory对话框，可以看到相关的文件下载信息。



5.观察APP应用程序是否成功下载并启动，本次操作已LED流水灯为例。串口信息提示TFTP SUCCEED,说明文件下载成功，之后又打印了APP的串口提示信息。说明APP更新成功，观察开发板也发现LED灯在按照设定的要求进行闪烁。

五总结

本文主要实现了基于STM32F103+W5500的嵌入式系统TFTP的设计方案，并展示了如何用TFTP32软件进行简单文件的下载传输。虽然只是简单讨论TFTP协议的相关理论及实现，但稍微复杂一点的FTP协议，POP3协议，SNMP协议也都可以通过类似的分析来实现的。随着物联网事业的发展，越来越多的嵌入式设备都将拥有联网功能，相信TFTP协议的作用将越来越重要。在当下物联网时代，想必还有其他应用也会遇到类似问题，希望本文能对做网络设备开发的朋友有所帮助。