

# 异步通信协议及控制规程设计实例

杨秀合 (解放军总医院计算机室)

## 一、引言

在开发异步通信软件时,主要问题是通信协议的设计和通信控制规程的设计。我们在开发“客户/服务器模式的信息系统支撑环境”时,采用异步通信连接客户机和服务器,异步通信是这一系统的基础,负责服务器上的数据库服务程序和客户机上的DOS应用程序的数据交换,它须具备较高的可靠性和通信效率,以便保证整

个系统正确工作。在开发这个通信软件时,我们对通信协议和通信控制规程进行了精心考虑,本文就这两个问题进行描述性说明。

注:以下讲到通信时,均指异步通信。

## 二、协议设计

在开发通信软件时,我们设计了一套帧协议。所谓帧是由具有一定含义的字节(下面均以字节为单位进行

讨论)组成的字节串,它携带有关通信方的信息。通信双方以报文包为单位进行数据交换,负责通信的软件根据报文包的长度把它分解成一个或多个数据帧传给对方,另一方把接收的数据帧组装成报文。对上层软件来说,通信双方是通过报文进行通信的,而在通信线路上流动的是包含报文中一段信息的数据帧,如图1所示。

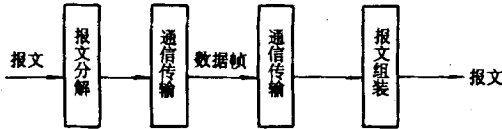


图1 报文与数据帧

为了保证可靠地传输报文,设计了七个具有不同含义和功能的协议帧,它包括一个数据帧(用于携带报文信息段)和六个控制帧(携带通信中的控制信息),共同完成数据帧的正确传输。

1.协议帧的格式

每种类型的帧都包括两字节的帧头,表示帧的类型和用于自校验,在下面用加重框表示。其中,第一个字节表示帧的类型,第二个字节是第一个字节的补码,这两个字节的模2和等于零,即自校验的。这样设计是为了防止因帧头错而产生误操作,并且使传输错误得到极早发现。在通信处理中,接收方对接收的前两个字节做模2和,检查帧头是否正确,只有当帧头正确时才继续处理后续字节。每种协议帧除表示帧头的两个字节外,还附有其他信息字节。

(1)数据帧

TYPE	NTYPE	COUNT	SEQ	NUMS	SNO	DATA	CHKSUM
------	-------	-------	-----	------	-----	------	--------

1B    1B    1B    1B    1B    1B    <256B    2B

其中:

TYPE和NTYPE组成帧头,各占一个字节,NTYPE是TYPE的补码,下面类同。这里TYPE的值为零,表示为数据帧。

COUNT表示此帧所携带数据(DATA)的按字节长度,占一个字节。

SEQ为本帧的序号,占一个字节,它只取0、1两个值,用于同步。

NUMS为本报文包含的数据帧数,占一个字节。一个报文的长度最大可为64K字节。

SNO为本帧在报文包中的帧号,占一个字节。

SNO、NUMS 包括了一个数据帧相对于报文的完整信息,用于报文的分解和组装。

DATA 为携带的数据,其长度由COUNT说明。

CHKSUM是除TYPE、NTYPE外其它各字节的检验和,用于通信查错,占两个字节。

(2)确认帧 ACK

TYPE	NTYPE	SEQ	NSEQ
------	-------	-----	------

1B            1B            1B            1B

这里 TYPE 的值为 1。SEQ 是应答的帧序号,

NSEQ 是 SEQ 的补码,各占一个字节。在通信过程中,只有当发送方的数据帧序号 SEQ 和接收方应答的确认帧的序号 SEQ 相同时,才认为是一次正确的通信。

(3)否认帧 NAK

TYPE	NTYPE	SEQ	NSEQ
------	-------	-----	------

1B            1B            1B            1B

其中,TYPE 的值为 2。SEQ 是否认的帧序号,NSEQ 是 SEQ 的补码。在通信时,只当对方应答的 NAK 帧的 SEQ 与发送的数据帧的 SEQ 相同时,才算是一个可接收的否认帧。

(4)初始化帧 START

TYPE	NTYPE
------	-------

1B            1B

这里 TYPE 的值为 3,只包含一个两字节的帧头。START 帧用于通信双方的同步,下面将叙述 START 帧在通信同步中的作用。

(5)初始化确认帧 STARTACK

TYPE	NTYPE
------	-------

1B            1B

这里 TYPE 的值为 4,同 START 帧一样它只包含一个双字节的帧头。在通信过程中,STARTACK 帧与 START 帧是匹配使用的,做为 START 帧的应答帧。

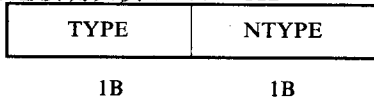
(6)断开连接帧 STOP

TYPE	NTYPE
------	-------

1B            1B

这里 TYPE 的值为 5。它同 START 帧一样,用于同步控制,关于这一点下面将详细说明。

(7)断开连接确认帧 STOPACK



其中,TYPE 的值为 6。在通信过程中,STOPACK 帧与 STOP 帧是匹配出现的,做为 STOP 帧的应答帧。

2.帧协议的特点

(1)通信速度:在异步通信过程中,一般采用中断方式(DOS下)接收线路上的数据。在帧协议方案下,采用的是中断与查询相结合的方式接收线路上的数据,即由帧头引起中断后,马上改为查询方式接收后续字符,接收完一帧后再改为中断方式,我们称之帧中断方式,如图2所示。而在字符协议下,每接收一个字符都要进入一次中断,为了接收一个帧,需要多次进入中断。大家知道,查询方式较中断方式快,考虑到通信速度,采用帧协议方案是比较好的。

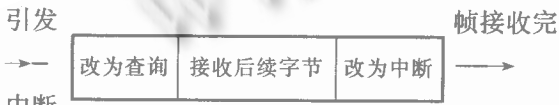


图2 接收一个协议帧的全过程

(2)通信效率:在通信过程中,发送一个数据帧后,对方回答一个4字节的应答帧,且数据帧的长度是可变的(随DATA段的长度变化而变化),其通信效率的概率平均值为91.4%。

(3)把一组相关的信息组装成一个完整的帧,便于同步控制、查错和报文的分解与组装。

三、通信控制规程

在通信控制规程设计中,设置了严格的同步控制、应答控制及多方面的超时控制,使得设计出的通信软件非常可靠。下面就这三个问题进行说明。

1.应答控制

收和发采用“停.等”方式,即:“发送→等应答→发送”的方式进行数据传输。

(1)发送方

发送完一帧数据(数据帧)后,等待对方的应答帧。只有当接收到对方的确认帧(ACK)后,方可发送下一帧;如果接收到否认帧(NAK),需要重发本数据帧,重发次数可以设定。

(2)接收方当收到一帧数据后,检查传输是否正确。如果正确,则发送确认帧ACK;否则发送否认帧NAK,通知对方重发它的数据帧。

图3和图4分别为发送流程和接收流程。

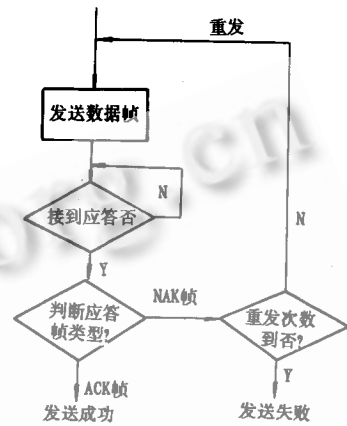


图3 发送流程

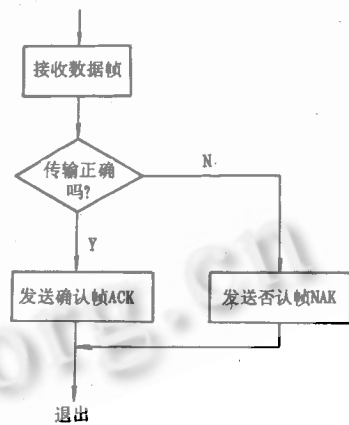


图4 接收流程

2.同步控制

同步控制用于保证通信双方协同地完成报文交换,在报文交换过程中,双方必须时刻处在同一个同步点。在这里同步控制问题体现在如下几个方面:

- ①通信一方在向对方发送报文时,进行初始化,当确知对方已准备好接收数据后,方可进行报文的发送。
- ②在通信过程中,可随时进行初始化,以便将通信双方置为已知状态。
- ③当发送完报文后(或通信过程中出现了无法恢复的混乱时),需要关闭与接收方的通信联系,此时双方就不能进行数据交换了,直到通信一方再进行初始化。

④通信双方都持有一个发送序号  $sseq$  和一个接收序号  $rseq$ , 取值 0, 1, 用来标识通信双方的同步情况。

⑤通信双方都设置了一个通信连接开关, 用于标识通信双方联系的状态, 控制通信方是否可以进行数据的发送或接收, 它一般由欲通信方进行设置, 当连接状态为开时, 方可进行数据的发送与接收。

欲通信方在发送报文时, 需进行这样的三步操作(如图 5):

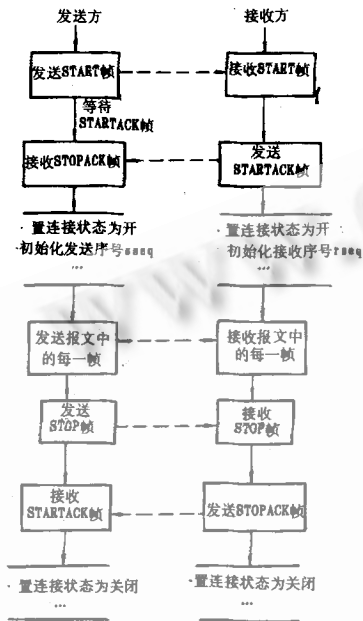


图 5 报文传输的全过程

①进行通信状态的初始化: 发送方向对方发送 START 帧, 等对方应答 STARTACK 帧, 初始化本地状态, 打开通信连接开关; 另一方收到 START 帧后, 向对方应答 STARTACK 帧, 初始化本地状态, 打开通信连接开关, 准备好接收数据。②进行报文的传输。

③关闭通信联系: 向对方发送 STOP 帧, 等收到 STOPACK 帧后, 关闭连接状态; 另一方收到 STOP 帧后, 关闭连接状态, 并应答 STOPACK 帧。

$sseq, rseq$ : 在通信过程中, 通过数据帧或 ACK、NAK 帧传给对方, 以便确认双方是否处于同一状态。发送方发送数据时, 把本地的  $sseq$  同时发送给对方, 接收方接收到数据帧后, 需要检查发送方的  $sseq$  与本地的

接收序号  $rseq$  是否相等, 只有相等时才做接收处理; 接收方接收到数据帧后, 通过应答帧 (ACK、NAK) 把本地的  $rseq$  传给对方, 发送方再把接收方的  $rseq$  同本地的  $sseq$  相比较。如果相等才认为是对本帧的应答, 否则继续等待对方的应答。如图 6 所示。

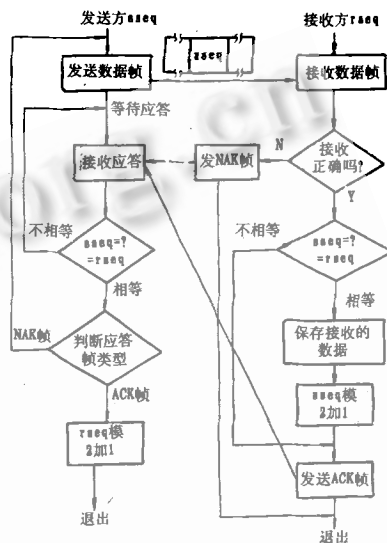


图 6 通过  $sseq, rseq$  进行同步的过程

### 3. 超时控制

在通信过程中, 有可能因线路干扰或程序问题而导致字符丢失、应答丢失或不正常的线路延迟, 因此需要进行超时控制, 以处理此类通信故障。

(1)字符间超时: 在接收协议帧时, 前后两个字符间的延迟大于给定的时间。当出现字符超时时, 需通过延时摔掉协议帧中的后续字符, 以防止误操作。

(2)数据帧的接收超时: 在接收一个数据帧时, 从接收第一个字符开始到最后一个字符所用的时间超过一个给定的值, 这个控制是为了检查通信过程中的异常情况。

(3)应答超时: 从发出一个协议帧到接收到相应应答帧的延迟超过某一给定时间。

图 7 给出了这三种超时控制在接收数据和发送数据时的应用情况。(a)为接收流程; (b)为发送流程。

在实际应用时, 需根据通信线路的波特率、双机的

CPU 时钟和双机上的运行环境来设定超时参数,表 1 是基于如下条件设定的三组超时值。

- ① 机器 A 是 PC286, 机器 B 是 PC486 / 33。
- ② 机器 A 运行 DOS 操作系统, 机器 B 运行 UNIX 操作系统。

表 1:

序号	字符间超时	数据帧接收超时	应答超时	波特率
1	1 秒	3 秒	10 秒	>9600
2	2 秒	4 秒	13 秒	>2400
3	3 秒	5 秒	19 秒	>1200

这些值是实验参数,并不一定十分准确。但在①②所述情况,能起到较好的超时控制作用。

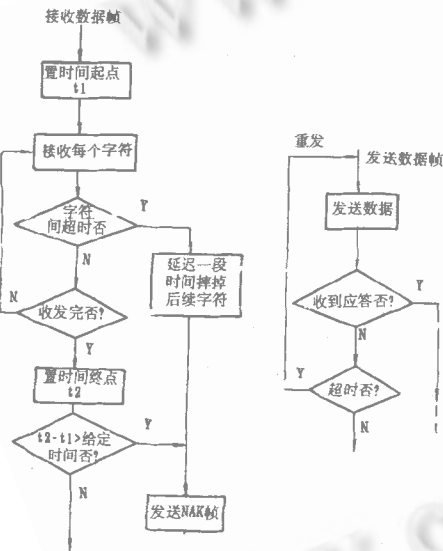


图 7

4.其它

在我们开发异步通信软件中,通过三种超时控制和校验和(或自校验)处理通信中的字符丢失、应答帧丢失、误码等差错及其它线路故障。此外,我们在设计每个帧时,也考虑了差错控制。

(1)帧头表示了帧的类型,代表了一个帧的重要信

息。因此,帧头必须正确传输,传输错误时要提早发现,以便通信双方能正确地理解每一帧。通信一方接收到每一帧后,首先通过检查表示帧头的两个字节之和是否为零来判别帧头是否被正确传输。只有当正确地获得了关于帧类型(第一个字节)的信息以后,才去理解帧头之后的每一个字节。

(2)在数据帧中加入两个字节的校验和,它是数据帧中除帧头以外的每个字节的模 2 和。发送方把校验和及数据帧中的其它部分一起传给对方,接收方把接收到的数据帧除帧头以外的每一个字节做模 2 和操作,然后再与所接收到的校验和进行比较,检查传输中是否出现了错误。

四、结束语

依照上面的协议及规程控制,我们开发了通信软件做为“客户/服务器模式的信息系统支撑环境①”的通信模块。图 8 是这一系统的结构简图。90 年我们开始开发此系统,93 年 5 月份通过了电子工业部鉴定。此系统做为信息系统的运行环境或/和开发环境已被包括我院在内的五家单位所使用。从鉴定的测试结果以及目前的运行情况看,此系统的通信软件可靠性高、效率高。因此,这里讨论我们所设计的协议和控制规程对用户是很有借鉴作用的。



图 8 “客户/服务器模式信息系统支撑环境”结构简图

关于“客户/服务器模式的信息系统支撑环境”的讨论请参见文献[1]。

参考文献:

[1] 任连仲等,“一种集中、分布式信息系统结构及其支撑环境”计算机系统应用,1992年,第1期。