

网络控制系统传输时延分析与测试

桂芳, 全书海

(武汉理工大学 自动化学院, 湖北 武汉 430070)

(nancygui@tom.com)

摘要: 为了了解时延的变化规律, 尽可能减小时延对控制系统性能的影响, 对不同协议下的传输时延进行了测试和分析, 找出了影响时延大小的因素, 并为网络控制系统传输协议的选择提供了参考意见。

关键词: 网络控制系统; 时延; 协议

中图分类号: TP273 **文献标识码:** A

Analysis and test of time-delay in network control systems

GUI Fang, QUAN Shu-hai

(School of Automation, Wuhan University of Technology, Wuhan Hubei 430070, China)

Abstract: In order to find out changing rules of time-delay, time-delay of different protocols were analyzed and measured, and in this way, factors that influence time-delay were founded. Some suggestions on choice of transmission protocols in network control systems were provided.

Key words: network control system; time-delay; protocol

0 引言

网络控制系统是控制技术和网络通讯技术的融合, 稳定、快速、准确依然是网络控制系统中追求的最终目标。由于引入了网络, 就不可避免地涉及到网络延时, 网络时延的存在将大大降低网络控制系统的性能。即使系统仍然保持稳定, 稳定区域也会显著减小。因此, 对网络时延的分析和研究具有重要意义。

本文首先对传输协议的模型进行了分析, 然后用 Visual Basic 6.0 对不同协议, 不同时段的时间进行了测试, 分析了测试结果, 并对网络控制系统传输协议的选择进行了讨论。

1 网络传输时延分析

网络的传输时延可以分为单程时延和往返时延两类。单程传输时延是指从源端机器开始发送数据到目的端机器接收到测试报文所需要的时间。由于两台机器的系统时钟很难精确同步, 一般网络传输时延是指测试报文的往返时延, 即测试报文从源端发送时开始计时, 目的端接收到该报文后立即返回给源端, 源端接收该报文后停止计时, 这个时间间隔即为往返时延。

一次单程时延可由三个部分组成: 源端时延、网络上的传输时延、目的端时延。

1) 源端时延

源端时延包含了源端的预处理时间、源端的等待时间、队列时间和阻塞时间。在源端的预处理时间是机器从外部环境获得数据并将其编码成网络数据格式所需要的时间。这个时间由机器的软硬件特性决定, 一般情况下认为预处理时间是常数或可以忽略。源端的等待时间包括在发送缓冲器队列中的等待时间和网络上其他消息传输阻塞它的时间。影响等待

时间的主要因素有网络协议、消息连接的类别和网络负载。队列时间是消息在源端缓冲区等待直到队列中前面的消息发出的时间。它由队列中前面消息的阻塞时间、消息的周期和处理负载决定。阻塞时间是每一端准备发送消息必须等待的时间, 由网络协议决定。阻塞时间包括其他端点发送消息时的等待时间和发生冲突时需要重发消息的时间。

2) 网络上的传输时间

网络上的传输时间包括帧时间和传播时间。帧时间由数据的大小、开销、填充符和位时间决定。传播时间由信号传输的速度和源端与目的端的距离决定。

3) 目的端时延

目的端的投递处理时间是把网络数据格式解码成物理数据格式并且输出到外部环境的时间。

一般在网络负载变化不大的情况下, 往返时延可以认为是单程时延的两倍。对于整个远程网络控制系统来说, 系统的延时时间除了网络传输的往返时延外, 还包括执行时延, 扰动时延等。设网络远程控制系统总的时延为 T_r , 则 $T_r = T_c + T_d + T_p + T_v$ 。

通信时延 T_c 包括通讯初始化时间在介质中的传输时间。传输信号通过传输介质在两地间的物理传输时间是随距离的增大而增大的。在网络远程控制中, 路由选择的不同会使信息沿不同的线路传输, 从而导致信息传输时间的变化, 不可预测。

执行时延 T_p 包括控制指令的解释、计算、执行时间, 现场图像的处理时间及仿真图像的运行时间等。

数据时延 T_d $T_d = (D_s + D_r) / VL$ 。 D_s 、 D_r 是发送回收的数据总量, VL 是传输速率, 与介质有关。这一项的存在表明数据传输量和带宽对远程作业非常重要。

扰动时延 T_v 主要指传输中不可预测的扰动, 如信息丢

收稿日期: 2005-04-18; 修订日期: 2005-06-24

作者简介: 桂芳 (1981-), 女, 湖北武汉人, 硕士研究生, 主要研究方向: 智能控制与智能信息处理; 全书海 (1955-), 男, 湖北咸宁人, 教授, 博士生导师, 主要研究方向: 汽车电子与电动汽车智能控制、DSP 技术与现场总线、图像信息处理与控制系统的研究与开发。

失或信息次序的混乱。

2 网络时延测试

2.1 时延测试方法

时延测试程序采用 Visual Basic 6.0 编程语言, Visual Basic 中提供了用于 Windows Sockets 编程的通讯控件——Winsock 控件, 可以较方便的实现网络通讯。Windows Socket 规范是一套开放的支持多协议的 Windows 下的网络编程接口, 已成为 Windows 网络编程标准。在 Windows Socket 规范中定义并记录了如何使用 API (Application Programming Interface, 应用编程接口) 与 Internet 协议族的连接。Winsock 控件对 Winsock API 进行了封装, 屏蔽了用 Winsock 编写 TCP/IP 应用的细节。

图 1 是采用 TCP 协议的程序实现框图, 采用 UDP 协议的程序框图与之类似, 只是无需客户端呼叫和服务器侦听, 两端程序运行后即可发送数据。

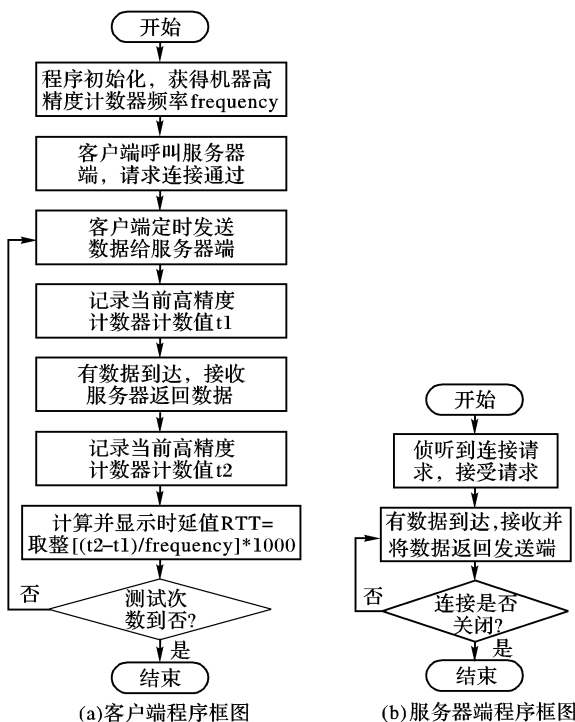


图 1 时延测试程序框图(TCP)

在通讯的两台主机上分别运行服务器端程序和客户端程序后, 客户端程序首先会获得该机器的高精度计数器频率, 为后面的定时和计算时延所用。然后在客户端程序输入服务器端的 IP 地址, 向服务器端发出请求连接呼叫。服务器端侦听到连接请求后接受请求, 两端连接建立, 客户端便可向服务器端发送测试数据, 并相应的记录每次测试数据的时延。基于 UDP 协议的测试程序无需建立连接, 在两端程序运行后, A 端即可向 B 端发送测试数据。

2.1.1 使用 Winsock 控件实现两台主机通讯

Winsock 控件所使用的协议可以是 TCP 或 UDP, 只用设置 Winsock 的 Protocol 属性就可指定协议类型。如: Winsock.Protocol = 0 (0 为 TCP 协议, 1 为 UDP 协议) TCP 是面向连接的传输协议, 在两端传输数据之前要先建立连接。因此在服务器端要用 Listen 方法进入侦听, 等待客户端的连接请求。客户端也要用 Connect 方法向服务器端发出连接请求。而 UDP 协议提供一种基于无连接的数据包传输方式, 设置好源端和目的端的地址和端口后就可以发送数据了。两端通讯的

端口号范围是 0 ~ 65535, 0 ~ 1023 的端口通常定义给已知的通信协议, 因此在两个测试程序中的端口都设置为 1024 以后的端口号。

Winsock 控件通过 SendData 方法来发送数据, 当有数据到达时会产生 DataArrival 事件, 在此事件中使用 GetData 方法就可获取发送过来的数据了。

2.1.2 定时器设置

在实验测试中, 考虑到在实际的网络控制系统中, 控制指令的发送频率较高, 通常大于 100Hz, 因此在程序设计中应实现高精度的定时来控制数据的发送。在 Visual Basic 的基本控件中有一个定时器控件 Timer, 这个控件的 Interval 属性用于设置定时器的定时精度, 从表面上看可以将 Interval 的属性值设为 1, 定时精度为 1ms。实际上 Timer 的定时精度只能达到 1/18s, 约 55ms, Timer 控件是以每 1/18s 产生一次的中断为定时依据。

为了实现高精度的定时, 在本程序设计中采用了高精度性能频率计数器, 它的精度与 CPU 的时钟频率有关。通过调用 Win32 API 函数 QueryPerformanceFrequency 获得高精度性能频率计数器的震荡频率, 函数 QueryPerformanceCounter 获得该计数器的当前震荡次数, 函数的原型如下:

```

BOOL QueryPerformanceFrequency (
    LARGE_INTEGER * lpFrequency)
BOOL QueryPerformanceCounter (
    LARGE_INTEGER * lpPerformanceCount)

```

调用函数 QueryPerformanceFrequency 后, 函数将系统频率计数器的震荡频率保存到 lpFrequency 中, 调用函数 QueryPerformanceCounter 将获得的计数器的震荡次数保存在 lpPerformanceCount 中。在本程序设计中, 首先获得利用 QueryPerformanceFrequency 函数得到的频率计数器的震荡频率(以秒为单位), 若要得到 tms 的定时, 则将计数器的震荡频率除以 1000 后得到每毫秒的计数器的震荡次数再乘以 t 即可得到 tms 的震荡次数, 然后在发送数据程序段中建立一个循环, 在循环中不停调用 QueryPerformanceCounter 函数获得计数器的震荡次数并同先前的频率计数器的震荡次数相减, 将先前的相减结果与 tms 的震荡次数比较, 大于这个次数就发送数据, 那么就实现了比较精确的定时发送数据。

2.1.3 时延计算

在文章前面已提到通过记录下发送数据时刻和接收返回的数据时刻, 相减即可得到往返传输时延 RTT。在程序设计中, 当客户端和服务端可以进行数据通讯了, 则在采用 SendData 方法发送数据后, 调用 QueryPerformanceCounter 函数获得发送数据计数器震荡次数。数据到达服务器端后, 产生 DataArrival 事件, 采用 GetData 接收数据, 并同时数据返回给客户端。客户端接收到数据时, 记下此时的计数器震荡次数, 与先前发送数据计数器震荡次数相减, 再除以频率计数器的震荡频率就得到了一个数据的往返传输时延。

2.2 测试过程

在连入网络的两台计算机上分别运行基于 TCP 协议的客户端程序与服务器端程序以及基于 UDP 协议的 A 端程序与 B 端程序。在不同时段, 发送不同大小的数据包, 发送频率为 10ms 一次, 发送次数 2000 次, 在数据包中还有数据序号, 用来判定是否丢包。作为客户端使用的测试程序界面如图 2 所示。

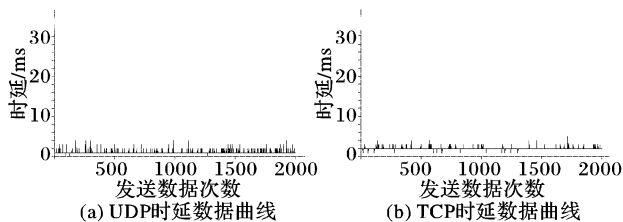
在网络负载较轻时段, 选择发送数据长度为 100Bytes, 采

用 UDP 和 TCP 测试的时延结果如图 3 所示。采用 UDP 测试的最大时延为 4ms, 平均时延为 1.082ms; 采用 TCP 测试的最大时延为 5ms, 平均时延为 2.026ms。



(a) UDP 时延测试界面 (b) TCP 时延测试界面

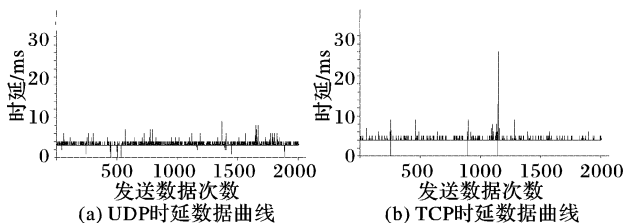
图 2 时延测试界面



(a) UDP 时延数据曲线 (b) TCP 时延数据曲线

图 3 网络负载较轻时段时延测试结果(100Bytes)

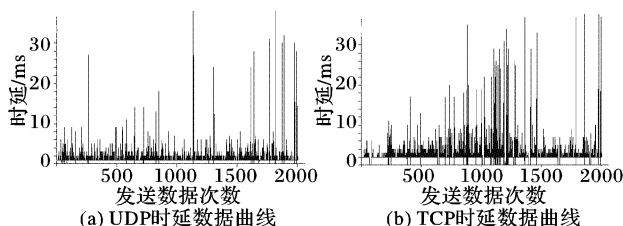
在相同时段将发送数据改为 1K Bytes 后, 采用 UDP 和 TCP 测试的时延结果如图 4 所示。采用 UDP 测试的最大时延为 9ms, 平均时延为 3.234ms; 采用 TCP 测试的最大时延为 26ms, 平均时延为 4.095ms。



(a) UDP 时延数据曲线 (b) TCP 时延数据曲线

图 4 网络负载较轻时段时延测试结果(1KBytes)

在网络负载较重时段, 选择发送数据长度为 100 Bytes, 采用 UDP 和 TCP 测试的时延结果如图 5 所示。采用 UDP 测试的最大时延为 140ms, 平均时延为 1.721ms; 采用 TCP 测试的最大时延为 39ms, 平均时延为 2.885ms。



(a) UDP 时延数据曲线 (b) TCP 时延数据曲线

图 5 网络负载较重时段时延测试结果(1KBytes)

2.3 结果分析

由以上实验结果可以看出, 在不同的时段, 网络时延有不同的特点, 协议选择不同, 网络时延也不同。

1) 同一种传输协议, 当网络负载较轻时段, 网络时延相对比较稳定。而当网络负载较重时段, 相比网络负载轻时网络时延变化较大, 会出现抖动, 不是很稳定。平均时延比上午的略大一点。在局域网内没出现丢包的情况。

2) 当发送数据长度变大, 由 100 字节变为 1K 字节, 网络时延明显增加, 可见在网络传输中, 传输数据的大小会直接影响网络时延。

3) 基于不同的传输协议, 网络时延也有所不同, 基于 UDP 的传输时延明显比 TCP 的传输时延小。

3 网络控制系统传输协议的选择

网络时延的存在会导致控制系统性能下降甚至不稳定, 同时也给控制系统的分析和设计带来了很大的困难。通过前面的分析可以看出传输协议也会对网络时延产生不同程度的影响。

由于 TCP 是面向连接的可靠的传输协议, TCP 数据包的首部有 20 个字节, 在数据传输过程中, 当传输一个数据段时, 发送端会设置一个计时器, 接收端接收到数据后会发送一个应答信号, 若在发送端接到应答信号之前, 计时时间已到, 则认为数据段已丢失, 发送端会重传该数据。TCP 协议还有保证数据完整性功能, 如果接收到数据出现错误, 接收端会向发送端发出请求, 请求重发一份数据拷贝。由于多种原因还会导致到达目的端的报文乱序, 这时目的端的 TCP 会缓存接受到的数据段, 通过 TCP 首部中的序列号完成对接收数据段的重新排序。而 UDP 不是面向连接的协议, 不提供数据传送的保证机制, 也不能确保数据的发送和接收顺序, 将安全和排序的功能移交给上层应用来完成。UDP 数据包首部只有 8 个字节, 由于它不需要用来保证数据安全性的额外开支, 因而它的传输速率要比 TCP 的传输速率快。但可能存在数据丢失现象, 而且可能先发的数据后收到, 使控制系统变为非因果系统。

TCP 协议虽然可以保证数据可靠准确地发送到目的端, 但却因此降低了传输率; 和 TCP 相比 UDP 协议在传输速率上占有优势, 但却不保障可靠数据的传输。在网络控制系统中, 可以选择交替使用 TCP 和 UDP 传输协议来获得较好的数据传输性能。在传输数据前首先采用 TCP 协议, 保证两端可靠连接, 然后在网络负载较小, 时延变化不大时可选用 UDP 协议传送数据, 当网络负载较大时, 需要可靠的传输数据时可将 UDP 协议切换为 TCP 协议传送。

4 结语

为了分析网络时延对控制系统的影响, 文章对网络控制系统传输时延进行了分析, 引起网络时延的因素有很多。通过实验测试, 从实验结果中可以看出网络负载的大小, 传送数据包的大小以及传输协议的选取的不同都会对网络时延产生不同程度的影响。如何克服网络延时对系统的影响是网络控制系统的一个研究重点。

参考文献:

- [1] BRADY K, TARN T-J. Internet-Based Remote Teleoperation[A]. Proceeding of the IEEE International conference on Robotics and Automation[C], 1998. 65-70.
- [2] 熊诚峰, 王旭永. 基于 Internet 的远程控制网络[J]. 自动化仪表, 2001, 22(4): 4-6.
- [3] 邓士普, 王树青. 基于网络的控制系统研究综述[J]. 化工自动化及仪表, 2003, 30(6): 1-5.
- [4] 朱其新, 吴建国, 陆国平. 网络控制系统中的基本问题与时延分析[J]. 南通工学院学报(自然科学版). 2003, 2(3), 40-43.
- [5] 修震, 吴东平, 黄杰, 等. 基于因特网的远程控制中网络延时特性分析[J]. 计算机工程与应用, 2004, 40(3): 129-131.
- [6] 吴恒刚. 网络远程控制系统仿真平台设计与实现[D]. 南京: 南京理工大学, 2004.
- [7] 刘新民, 蔡琼, 白康生. Visual Basic 6.0 程序设计[M]. 北京: 清华大学出版社, 2004.