

基于有限状态机的UDP传输设计

张菁

(长安大学电子与控制工程学院, 西安 710061)

摘要: 针对传统用户数据报协议(UDP)传输可靠性低的问题, 提出一种利用有限状态机(FSM)控制UDP传输数据的设计方法。为传输过程中的通信双方分别设计不同的状态; 构造数据包格式, 每个数据包携带命令字以表明不同状态; 增加数据包校验位, 保证数据传输的一致性; 通过正向替换和逆向替换消除并恢复数据中存在的保留关键字; 在发送过程中采用自动重发机制和请求重发机制, 保证数据包正常有序的传输; 通过增加包头包尾使数据包大小可变, 以增强数据传输的灵活性。实验结果证明了该设计方法的有效性。

关键词: 用户数据报协议; 有限状态机; 自动重发; 请求重发; 正向替换; 逆向替换

UDP Transmission Design Based on Finite State Machine

ZHANG Jing

(School of Electronic and Control Engineering, Chang'an University, Xi'an 710061, China)

【Abstract】 To remedy the defect of User Datagram Protocol(UDP) transmission in reliability, this paper proposes a design method using Finite State Machine(FSM) to control UDP transmission data. The different states are designed for the two sides of the transmission process. The format of data packet is conceived, each data packet carries the a command word to respect the different state. The data packet increased checking parity can ensure data consistency in transmission, to remove and recover a reserved keyword of data though the positive replacement and reverse replacement. It uses the automatic retransmission mechanism and the request retransmission mechanism to ensure normal and orderly transformation in the process of sending data, the data packet may become a variable-size packet thought increasing the header and the ender, which enhance the flexibility of data transmission. Experimental results prove the validity of the design method.

【Key words】 User Datagram Protocol(UDP); Finite State Machine(FSM); automatic retransmission; request retransmission; positive replacement; reverse replacement

DOI: 10.3969/j.issn.1000-3428.2011.17.016

1 概述

传输控制协议(Transmission Control Protocol, TCP)是面向连接的协议, 在正式收发数据前, 必须和对方建立可靠的通信连接。用户数据报协议(User Datagram Protocol, UDP)是与TCP相对应的、面向非连接的协议。它不与对方建立连接, 直接发送数据包。由于UDP协议没有连接的过程, 因此其通信效率较高, 但却降低了可靠性。UDP适用于对可靠性要求不高的应用环境。但是随着网络传输业务的快速发展, 在某些对可靠性和高效性要求都较高的应用环境中, TCP能满足可靠性要求, 但是达不到高效性; UDP可以解决高效性, 但是可靠性较低。为解决这一矛盾, 本文设计利用有限状态机(Finite State Machine, FSM)模式控制UDP的传输, 实现UDP高效可靠的进行数据传输。

2 有限状态机原理

有限状态机系统^[1]是指在不同阶段会呈现出不同运行状态的系统, 这些状态是有限的、不重叠的。这样的系统在某时刻一定会处于其所有状态中的一个状态, 此时它接收一部分允许的输入, 产生一部分可能的响应, 并且迁移到一部分可能的状态。FSM包含以下5个要素:

(1)状态(State): 是一个系统在其生命周期中某一时刻的运行情况, 此时, 系统会执行一些动作或等待一些外部输入。

(2)条件(Guard): 状态机对外部消息进行响应的时候, 除了需要判断当前的状态, 还要判断跟这个状态相关的一些条件是否成立。这种判断称为Guard。Guard通过允许或禁止某

些操作来影响状态机的行为。

(3)事件(Event): Event就是在一定的时间和空间上发生的对系统有意义的事情。

(4)动作(Action): 当一个Event被状态机系统分发的时候, 状态机用Action来进行响应, 比如修改一下变量的值、进行输入输出、产生另外一个Event或迁移到另外一个状态等。

(5)迁移(Transition): 从一个状态切换到另一个状态被称为Transition。引起状态迁移的事件称为触发事件(Triggering Event)或触发(Trigger)。

FSM的状态迁移如图1所示。

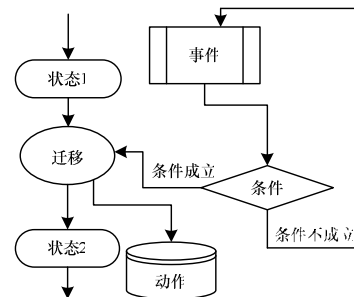


图1 FSM的状态迁移

作者简介: 张菁(1972—), 女, 讲师、博士研究生, 主研方向: 网络传输, 信息处理

收稿日期: 2011-03-24 **E-mail:** jingzhang@chd.edu.cn

3 通信双方的状态机设计

参照 FSM 要素标准设计数据传输的状态机, 首先将数据传输过程分解为 6 个过程, 数据发送前相互通知准备, 数据发送, 数据发送完毕相互通知结束, 针对每一个过程设计一对状态机^[2]。

(1)开始发送状态设计为 P_PREPARE_SEND_Message (准备发送信息)。

(2)接收方接收到信息后回应 P_CAN_SEND_Message(可以发送信息)。

(3)发送方进入下一状态为 P_SEND_Message_DATA(发送信息)。

(4)接收方接收数据后回应 P_RECEIVED_Message_DATA(数据正确接收)。

(5)数据发送完毕后发送方状态改为 P_FINISHED_SEND_Message(信息发送完毕)。

(6)接收方回应 P_FINISHED_SAVED_Message(信息保存完毕)。

设计出各个阶段的 State 后, 根据发送数据的实际过程设计出各个状态迁移需要的条件。首先发送方发送 P_PREPARE_SEND_Message 数据进入准备发送信息状态, 等候接收方的回应是否准备好接收数据, 接收方接收到要求发送数据的信息后, 做好接收信息的准备, 同时回应 P_CAN_SEND_Message, 通信双方都完成了发送数据前的准备。接收数据为状态迁移的 Event 要素, 判断接收到的数据是状态迁移的 Guard 要素, 如果条件要求进入下一状态, 做好发送或接收数据的准备工作就是状态迁移的 Action, 该动作完成将触发状态进入下一状态, 就形成的状态的 Transition。进入发送数据状态, 发送方开始将数据一次或分几次发出, 一次或逐次发送 P_SEND_Message_DATA, 接收方接收数据后回应 P_RECEIVED_Message_DATA。发送方检查数据发送完毕作为进入发送结束状态的事件, 该事件发生触发发送方进入发送数据结束状态, 同时发送 P_FINISHED_SEND_Message, 接收方接收该数据后处理接收的所有数据, 回应 P_FINISHED_SAVED_Message 同时结束数据接收, 接收方接收到 P_FINISHED_SAVED_Message 结束数据发送完成一次数据发送^[3]。

图 2 通过时间轴的推移演示状态机的迁移过程。

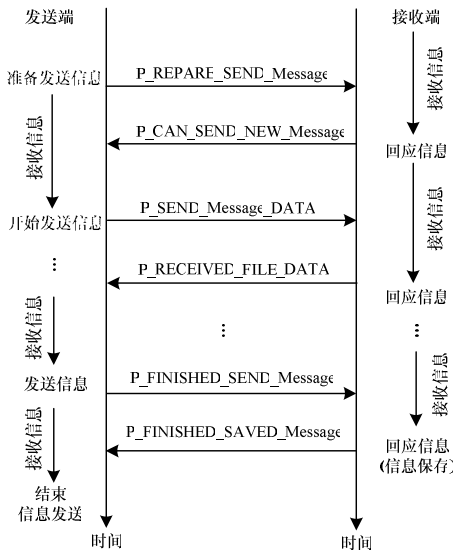


图 2 状态机设计及迁移过程

回应信息成为状态机迁移的 Event, 回应不同的信息成为状态机的 Guard, 同时发送剩余的信息成为状态机的 Action, 同时触发状态机进行 Transition。

4 数据包结构设计

在数据通信过程中, 数据包应该包含有控制信息和数据信息, 其中控制信息用于维持整个通信正常进行, 数据信息是实际传输的信息。为了使通信双方对发送接收的数据进行正常的分析处理, 设计各个状态的数据包格式, 通信双方遵照设计的包格式封装数据进行传输。数据包前后增加保留关键字作为包头包尾, 可以使数据包成为大小可变化的, 改变常规以数据包大小来分解数据的做法。数据包大小可以变化, 每次发送数据可以根据实际情况填写数据信息。

参照状态机的设计, 数据包结构设计如图 3 所示, 其中 H 表示 16 进制。

准备发送新的信息						
包头	命令字	发送序号	信息长度	信息内容	校验位	包尾
8EH	11H	XXH	XXH	8FH
可以接收数据信息						
包头	命令字	验证序号	请求序号	校验位	包尾	
8EH	21H	XXH	XXH	XXH	8FH	
发送信息数据						
包头	命令字	发送序号	信息长度	信息内容	校验位	包尾
8EH	12H	XXH	XXH	8FH
可以接收数据信息						
包头	命令字	验证序号	请求序号	校验位	包尾	
8EH	22H	XXH	XXH	XXH	8FH	
数据发送完毕						
包头	命令字	发送序号	信息长度	信息内容	校验位	包尾
8EH	13H	XXH	XXH	8FH
已经保存文件完毕						
包头	命令字	验证序号	请求序号	校验位	包尾	
8EH	23H	XXH	XXH	XXH	8FH	

图 3 数据包结构设计

数据包中以保留关键字 8E、8F 作为包头包尾。数据包命令字对应各个发送接收状态, 与状态机设计一一对应。

- 11H → P_PREPARE_SEND_Message
- 21H → P_CAN_SEND_Message
- 12H → P_SEND_Message_DATA
- 22H → P_RECEIVED_Message_DATA
- 13H → P_FINISHED_SEND_Message
- 23H → P_FINISHED_SAVED_Message

发送序号代表信息发送的次序, 接收方可以根据发送序号来组织数据信息, 同时检查数据序号是否混乱, 发送次序采用连续的、唯一不重复的一组数据。验证序号与发送序号编码规则是相同的, 接收方正确接收数据后将发送序号填入验证序号中, 将验证序号加 1 作为请求序号填入数据包。信

息长度表示本次数据包中数据信息的长度。信息内容是数据包中的数据信息。校验码是对整个数据包进行数据校验计算,逐个字节异或计算出校验码。

数据包信息数据中如果包含有包头(8EH)或包尾(8FH)保留关键字,在接收端分解数据出现无法准确的确定包头包尾的位置。为了避免这个现象,在增加包头、包尾前对数据进行正向替换,首先对于要发送的数据逐位判断是否为8E、8F、8D,发现数据8E,整个信息数据加长1位,8E后面的数据整体后移一位,将8E替换成8D5E,同样的原理将8F替换成8D5F,8D替换成8D5D,将进行过正向替换(用8D5E替换8E,用8D5F替换8F,用8D5D换8D定义为正向替换)的数据增加包头包尾进行发送。接收端接收数据后判断包头包尾的位置后截取中间的数据,同样逐位进行判断,如发现有8D,判断后续位是否为5E、5F、5D,如果存在则进行逆向替换(用8E替换8D5E,用8F替换8D5F,用8D替换8D5D定义为逆向替换),将经过正向替换处理的数据恢复原样^[4]。

5 重发机制设计

由于实际网络传输可能出现数据丢失的现象,这种情况如果发生在状态迁移过程中,则可能使状态机陷入沉睡状态无法进行正常的状态迁移。如果发送端发送的数据丢失,则接收端无法产生接收信息的事件,或是接收端回应信息在传输中丢失,同样无法在接收端产生接收消息的事件,没有事件的触发,状态机就在一个状态下一直等待。另外由于网络传输的延迟性,可能出现后发的数据包先到接收方,在接收方出现乱序的现象,为避免这种情况发生^[5],本文引入2种重发机制:自动重发机制,请求重发机制^[6]。

自动重发机制如图4所示。

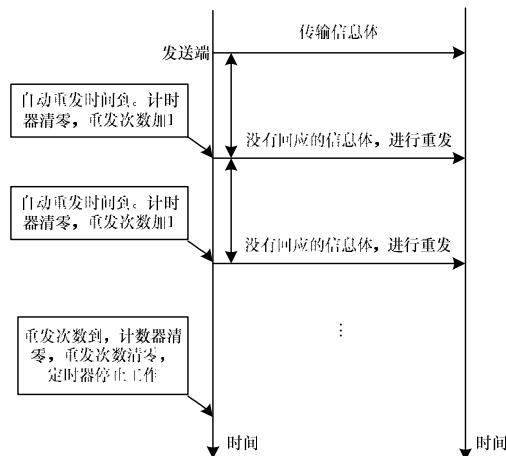


图4 自动重发机制

通信发送方维持一个自动重发定时器,发送方在给对端发送消息时,检查自动重发定时器是否启动,如果没有启动,就会启动这个定时器,同时将本次发送的信息存入临时队列,定时器规定时间内接收到正常回应,则定时器器归零,分解接收数据中的验证序号,把临时队列中发送序号与验证序号相同的数据删除;如果定时器到规定时间没有接收到回应信息,将定时器归0,并将队列中的没有收到回应的消息重发一遍,重发次数加1。如果重发次数到达指定次数,置重发次数为0,重发定时器为0,本次发送数据失败。

请求重发机制如图5所示。通信接收方记录着每次收到信息的序号,同时计算下一次收到的数据的序号。如果接收

方收到的数据包的序号不是自己所期望的序号时,接收方将会向发送方发送一条请求重发消息,请求从自己期望的序号开始进行重发。发送方收到这个请求重发消息后,将立即进行重发。该机制主要用于处理消息包出现先发后到的现象。

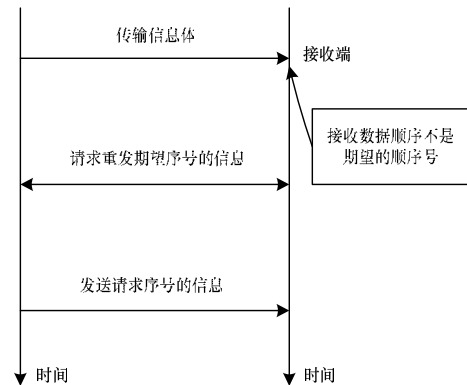


图5 请求重发机制

6 传输实验结果

实验环境为2台PC机,分别是PC1和PC2,实验从PC1传输文件到PC2,骤如下:

(1)PC1选择传输文件,向PC2发送携带命令字为11H的数据包,PC1接收了命令字为21H的数据包。

(2)PC1向PC2发送命令字为12H加文件内容数据数据包,接收到命令字为22H的数据包,循环发送直至文件发完。

(3)PC1发送文件完毕,向PC2发送命令字为13H的数据包,接收到命令字为23H的数据包。传输完毕。

实验结果表明,传输过程严格按照设计的状态机进行变化,整个传输过程可控。

7 结束语

本文提出了一种基于有限状态机的UDP传输设计方法。采用UDP传输,能保证传输的高效性,引入状态对传输过程加以控制,保证了传输的可靠性。由于现今计算机普遍安装防火墙软件,可能会阻挡UDP数据的传输,因此下一步将利用网络地址转换(Network Address Translation, NAT)技术穿越防火墙,实现数据高效、可靠、无阻挡的传输。

参考文献

- [1] 史雪飞,王志良,张琼.基于有限状态机矩阵模型的人工情绪模型[J].计算机工程,2010,36(18):24-25.
- [2] 鲁玲,刘大年.消除状态机毛刺策略探讨[J].电子技术应用,2006,(9):118-119.
- [3] 张国印,叶在伟,曲丽君.一种UDP穿越NAT的新方案[J].计算机工程,2008,34(12):112-113.
- [4] 田泽.嵌入式系统开发与应用教程[M].北京:北京航空航天大学出版社,2005.
- [5] 顾栋梁,周健,程克勤.基于Netfilter的连接限制的研究与实现[J].计算机工程,2009,35(10):162-163.
- [6] Eppinger J L. TCP Connections for P2P Apps: A Software Approach to Solving the NAT Problem[R]. Carnegie, UK: Carnegie Mellon University, Technical Report: CMU-ISRI-05-104, 2005.

编辑 顾姣健