



哈尔滨工业大学  
Harbin Institute of Technology

# 计算机网络 课程实验报告

实验名称	可靠数据传输协议-停等协议的设计与实现					
姓名	李劲光		院系	计算机科学与技术学院		
班级	1903201		学号	L190202102		
任课教师	聂兰顺		指导教师	聂兰顺		
实验地点			实验时间			
实验课表现	出勤、表现得分(10)		实验报告 得分(40)		实验总分	
	操作结果得分(50)					
教师评语						



实验目的：

（注：实验报告模板中的各项内容仅供参考，可依照实际实验情况进行修改。）

本次实验的主要目的。

理解可靠数据传输的基本原理；掌握停等协议的工作原理；掌握基于 UDP 设计并实现一个停等协议的过程与技术。

实验内容：

概述本次实验的主要内容，包含的实验项等。

- 1) 基于 UDP 设计一个简单的停等协议，实现单向可靠数据传输（服务器到客户的数据传输）。
- 2) 模拟引入数据包的丢失，验证所设计协议的有效性。
- 3) 改进所设计的停等协议，支持双向数据传输；（选作内容，加分项目，可以当堂完成或课下完成）
- 4) 基于所设计的停等协议，实现一个 C/S 结构的文件传输应用。（选作内容，加分项目，可以当堂完成或课下完成）

实验过程：

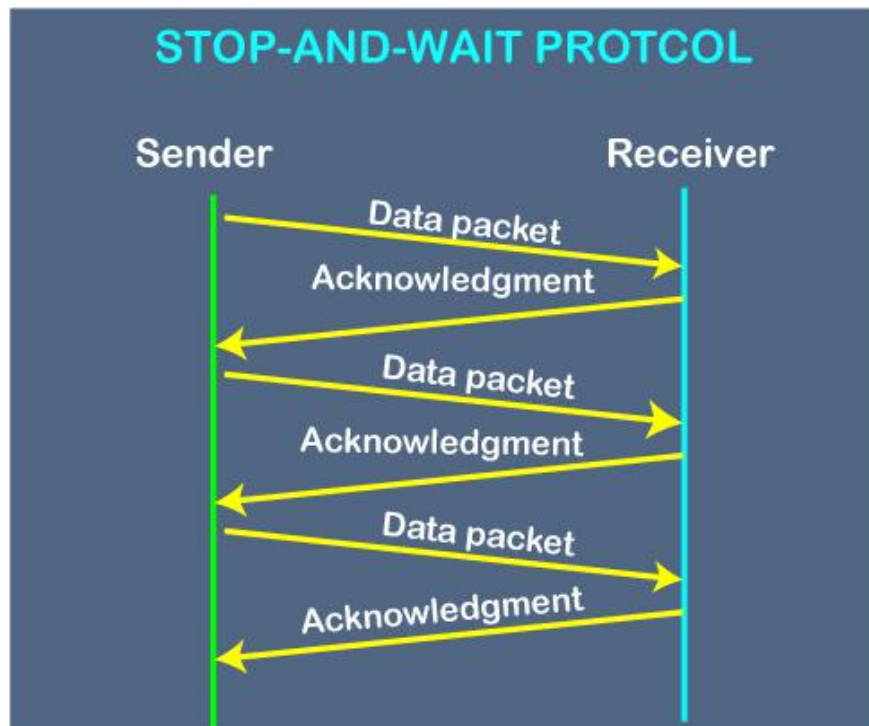
以文字描述、实验结果截图等形式阐述实验过程，必要时可附相应的代码截图或以附件形式提交。

发送方：

1. 发送方一次发送一个数据包。
  2. 发送方只有在收到前一个数据包的确认后才发送下一个数据包。
- 因此，发送方停止等待协议的思想非常简单，即一次发送一个数据包，在收到确认之前不发送另一个数据包。

接收端：

1. 接收然后消费数据包。
  2. 当数据包被消费时，接收方向发送方发送确认。
- 因此，接收端停止等待协议的思想也很简单，即消费数据包，一旦数据包被消费，就发送确认。这被称为流量控制机制。



上图显示了停止和等待协议的工作。如果有发送方和接收方，则发送方发送数据包，该数据包称为数据包。发送方在没有收到第一个数据包的确认之前不会发送第二个数据包。接收方发送对收到的数据包的确认。收到确认后，发送方发送下一个数据包。这个过程一直持续到所有的数据包都没有被发送。该协议的主要优点是简单，但也有一些缺点。例如，如果有 1000 个数据包要发送，那么 1000 个数据包不能一次全部发送，因为在 Stop and Wait 协议中，一次发送一个数据包。

#### Sender Site Algorithm of Simplex Stop and Wait Protocol

```
begin
  canSend = True;           //Allow the first frame to be sent
  while (true)              //check repeatedly
  do
    Wait_For_Event();        //wait for availability of packet
    if ( Event(Request_For_Transfer) AND canSend) then
      Get_Data_From_Network_Layer();
      Make_Frame();
      Send_Frame_To_Physical_Layer();
      canSend = False;
    else if ( Event(Acknowledgement_Arrival)) then
      Receive_ACK();
      canSend = True;
    end if
  end while
end
```

# Receiver Site Algorithm of Simplex Stop and Wait Protocol

```
begin
  while (true)      //check repeatedly
  do
    Wait_For_Event();    //wait for arrival of frame
    if ( Event(Frame_Arrival) then
      Receive_Frame_From_Physical_Layer();
      Extract_Data();
      Deliver_Data_To_Network_Layer();
      Send_ACK();
    end if
  end while
end
```

## 实验结果:

采用演示截图、文字说明等方式, 给出本次实验的实验结果。

## Server:

1. 一次发送一个数据包。
2. 只有在收到对前一个数据包的确认后才发送下一个数据包。

```
50696E67s-MacBook-Pro:Stop and Wait 50696e67$ python sender.py
[*] Server is listening ..
[*] Receiver connected at 127.0.0.1:58010
ENTER BIT STRINGS : test msg
[+] Acknowledgement: Message Received
[+] ACK received
[-] Package dropped (1)
[-] Timeout!! - package resend
[+] Acknowledgement: Message Received
[+] ACK received
[+] Acknowledgement: Message Received
[+] ACK received
[+] Acknowledgement: Message Received
[+] ACK received
[+] Acknowledgement: Message Received
[+] ACK received
[+] Acknowledgement: Message Received
[+] ACK received
[+] Acknowledgement: Message Received
[+] ACK received
[-] Package dropped (1)
[-] Timeout!! - package dropped 2
[-] Timeout!! - package dropped 2
[-] Timeout!! - package dropped 2
[-] Timeout!! - package resend
[+] Acknowledgement: Message Received
[+] ACK received
```

Receiver:

1. 接收并消费一个数据包后发送确认。

```
50696E67s-MacBook-Pro:Stop and Wait 50696e67$ python rcver.py
Received data : {0: 't'}      print ("[-] Timeout!!", end=" - ")
Received bitstring : t       if not random.randint(0, 1):
Received data : {1: 'e'}      clientsocket.send(sendstring.encode())
Received bitstring : te      print("package resend")
Received data : {0: 's'}      else:
Received bitstring : tes      print("package dropped 2")
Received data : {1: 't'}      current_time = time.time()
Received bitstring : test     compare_time = time.time()
Received data : {0: ' '}
Received bitstring : test
Received data : {1: 'm'}
Received bitstring : test m
Received data : {0: 's'}
Received bitstring : test ms
Received data : {1: 'g'}
Received bitstring : test msg
clientsocket.listen(5)
address = s.accept()
client(clientsocket, address)
```

问题讨论:

对实验过程中的思考问题进行讨论或回答。

由于数据丢失而出现问题，假设发送方发送数据，数据丢失。接收方长时间等待数据。由于接收方没有收到数据，所以它不发送任何确认。由于发送方没有收到任何确认，因此它不会发送下一个数据包，发生此问题是由于数据丢失。

由于丢失确认而出现问题，假设发送方发送数据并且它已被接收方接收。在接收到数据包时，接收者发送确认。在这种情况下，确认在网络中丢失，因此发送方没有机会收到确认。在停止等待协议中，发送方也没有机会发送下一个数据包，直到收到前一个数据包的确认才能发送下一个数据包。

由于延迟数据或确认引起的问题，假设发送方发送数据并且它也被接收方接收。然后接收方发送确认，但在发送方的超时期限之后收到确认。由于收到确认较晚，因此确认可能被错误地视为对某些其他数据包的确认。

心得体会:

结合实验过程和结果给出实验的体会和收获。

通过此次实验,我对停等协议的有了更加深刻的认识,了解错误控制机制。使用错误控制机制,以便接收到的数据应该与发送方发送的数据完全相同。