# Stop And Wait Algorithm

A COURSE PROJECT REPORT

By

**Anant Duhan (RA1911003010564)**

**Alok Agarwal (RA1911003010562)**

**Shagun Jindal (RA1911003010544)**

Under the guidance of

**Yamini R.**

*In partial fulfilment for the Course*

of

18CSC302J - COMPUTER NETWORKS

in Department Name



**FACULTY OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**Kattankulathur, Chenpalpattu District**

NOVEMBER 2021

**SRMINSTITUTEOFSCIENCEANDTECHNOLOGY**

**(UnderSection3ofUGCAct,1956)**

**BONAFIDECERTIFICATE**

Certified that this project report "**Stop and Wait Protocol** " is the bonafide workof**StudentName(Registerno)**whocarriedouttheprojectworkundermysupervision.

# SIGNATURE SIGNATURE

SubjectStaff **Dr.E.Sasikala,**

# Designation CourseCordinator

**Department AssociateProfessor,**

SRMInstituteofScienceandTechnology **DataScienceandBusinessSystems**Potheri,SRMNagar,Kattankulathur, SRMInstituteofScienceandTechnologyTamilNadu603203 Potheri,SRMNagar,Kattankulathur,

TamilNadu603203

# ABSTRACT

Before understanding the stop and Wait protocol, we first know about the error control mechanism. The error control mechanism is used so that the received data should be exactly same whatever sender has sent the data. The error control mechanism is divided into two categories, i.e., Stop and Wait ARQ and sliding window. The sliding window is further divided into two categories, i.e., Go Back N, and Selective Repeat. Based on the usage, the people select the error control mechanism whether it is **stop and wait** or **sliding-window**. This project implements the stop-and-wait protocol using sockets and threading. The stop-and-wait protocol is a special case of the Go-back-N protocol, with window size = 1. Through this we have illustrated and then tackled the possible errors that can happen through erroneous channels during stop-and-wait protocol communication.

# Computernetworks–CourseProjectFormattingInstructions

1. ChapternumberandChapterheading–**fontsize16,uppercase,bold.**
2. SpacebetweenChapternumberandChapterheading-**doublespacing.**
3. Spacebetweenheadingandcontents–**doublespacing.**
4. Abstractheading–**fontsize16.**
5. Contentofabstract–**fontsize14,doublespacing.**
6. Sampledocumentisgivenbelow,followitfor**fontsize,upper/lowercase,spacing**
7. Sub-headingexampleasfollows.

# REQUIREMENTSPECIFICATION(TimesNewRoman14)

* + 1. **HardwareRequirements(TimesNewRoman12)**

Processor :2.4GHzClockSpeedRAM :1GB

HardDisk :500MB(Minimumfreespace)

## SoftwareRequirements

OperatingSystem :Windows7Platform :Java

BackEnd :MySql

SpecialTools:Opencv,XuggleServer :ApacheTomcat

# ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice ChancellorDr.C.MUTHAMIZHCHELVAN** ,for being the beaconinall our endeavors.

We would like to express my warmth of gratitude to our **RegistrarDr. S.Ponnusamy,**for his encouragement

We express our profound gratitude to our**Dean(CollegeofEngineering and Technology) Dr. T. V.Gopal,**for bringing out novelty in allexecutions.

We would like to express my heartfelt thanks to Chairperson, SchoolofComputing**Dr.RevathiVenkataraman,**forimpartingconfidencetocompletemycourseproject

Wewishtoexpressmysincerethanksto**CourseAuditProfessorDr.M.LAKSHMI,ProfessorandHead,DataScienceandBusinessSystems**and**CourseCordinatorDr.E.Sasikala,AssociateProfessor,DataScienceandBusinessSystems**fortheirconstant encouragementandsupport.

WearehighlythankfultoourmyCourseprojectInternalguide**Subjecthandlingstaffname,Designation,Department,**for**his/her**assistance, timely suggestion andguidance throughout the duration of this courseproject.

Weextendmy gratitude to**Student HOD nameDepartment**andmyDepartmentalcolleaguesfortheirSupport.

Finally, we thank our parents and friends near and dear ones whodirectly and indirectly contributed to the successful completion of our project.Aboveall,IthankthealmightyforshoweringhisblessingsonmetocompletemyCourseproject

# TABLEOFCONTENTS

## CHAPTERS CONTENTS PAGENO.

* + - 1. **ABSTRACT**

## INTRODUCTION

* + - 1. **REQUIREMENTANALYSIS**

## ARCHITECTURE&DESIGN

* + - 1. **IMPLEMENTATION**

## EXPERIMENTRESULTS&ANALYSIS

* + - * 1. RESULTS
        2. RESULTANALYSIS
        3. CONCLUSION&FUTUREWORK

## REFERENCES

# ABSTRACT

Before understanding the stop and Wait protocol, we first know about the error control mechanism. The error control mechanism is used so that the received data should be exactly same whatever sender has sent the data. The error control mechanism is divided into two categories, i.e., Stop and Wait ARQ and sliding window. The sliding window is further divided into two categories, i.e., Go Back N, and Selective Repeat. Based on the usage, the people select the error control mechanism whether it is **stop and wait** or **sliding-window**. This project implements the stop-and-wait protocol using sockets and threading. The stop-and-wait protocol is a special case of the Go-back-N protocol, with window size = 1. Through this we have illustrated and then tackled the possible errors that can happen through erroneous channels during stop-and-wait protocol communication.

**INTRODUCTION**

TheStop-and-Waitprotocolisatechniquethatisusedtoprovidereliability. Inthisprotocol,oneframe(inourcase,onbit)issentata time.Thesenderdoesnotsendanymoreframes,untilitreceivesanacknowledgementfromthereceiverforthesame.Iftheacknowledgementfails to arrive within a given time frame, the sender then re-sends the entireframe.Thisconditionisknownasatimeout.Onthereceiver’sside,itsendsanacknowledgementeachtimeitreceivesaframe.

**REQUIREMENT ANALYSIS**

**3.1 Hardware Requirements**

Processor : 2.4 GHz Clock Speed RAM : 1 GB

Hard Disk : 500 MB (Minimum free space)

**3.2 Software Requirements**

Operating System : Windows 7 Platform : Java

Back End : MySql

Special Tools : Opencv, Xuggle Server : Apache Tomcat

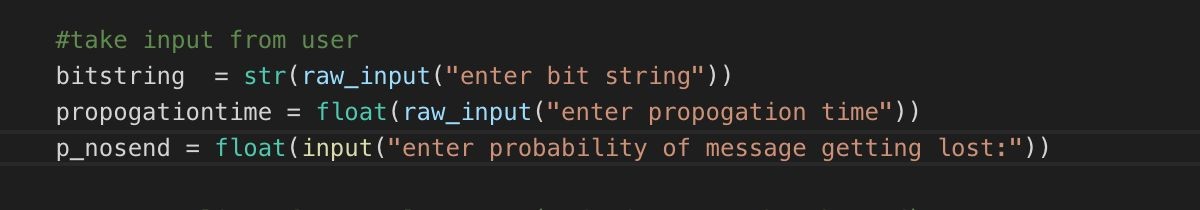
**ISSUES IN COMMUNICATION**

TheStopandWaitprotocolensurereliablecommunication.Inessence, the protocol gives insurance against a noisy (or otherwisedisturbed) channel that might cause a packet to drop while being sent fromthesendersideoftheprogram.Onthecorollary,anoisychannelalsomightcausetheacknowledgementsentback fromthereceiver’ssideto getlost. TheStopandWaitprotocolensurecommunicationinbothofthesecases.

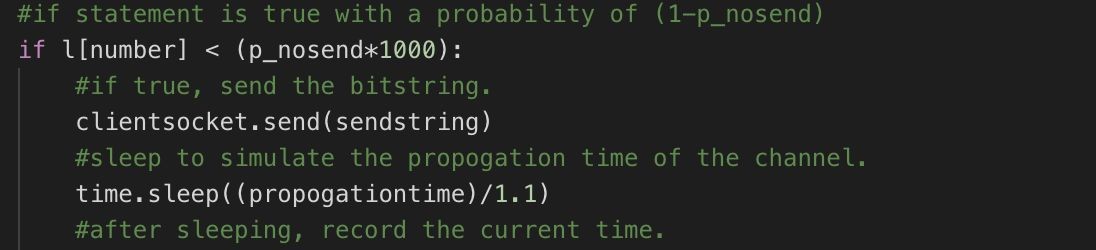
**IMPLEMENTATION**

Ourimplementationoftheprotocolsimulatesapackagedropinanoisychannel.Eachsideoftheprogramaskstheuserforaprobabilityofthepackagegettingdroppedbasedonwhichwethenforciblyensureapackagedrop.

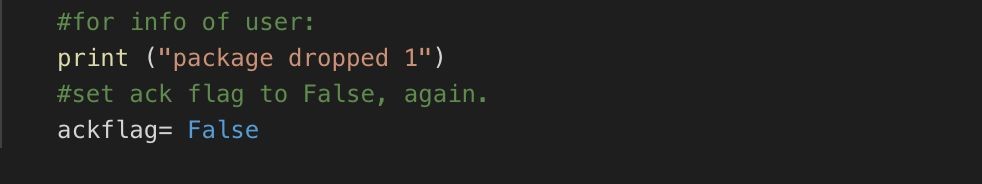
Thesendersideoftheprogramasksforabitstringinputfromtheuser (which is to be transmitted across the channel), the probability ofpackagegettingdroppedandthepropagationtime.



We use sockets in order to send given bitstring from the sender to thereceiver,withasuccessrateenteredbytheuser.Iftrue,thepacketissenttothereceiver’sside.Wetheninduceatime.sleep(),whichismeanttosimulatethepropagationtime.

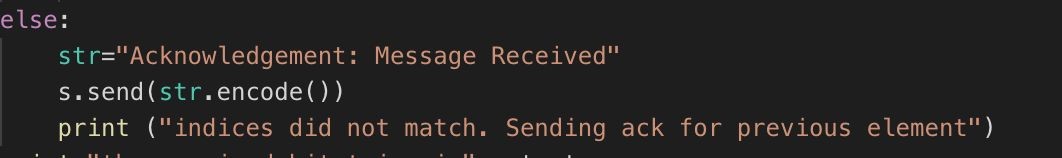


Theothercaseiswhentheframeisdroppedbythechannel.Inthisscenario,we again inducea sleep, inorder to simulate thepackage being sent. As far as the sender “knows”, it has sent the package, but it does notknowthattheframegotlostduetothenoisychannel.Afterwaitingforthegiventimeperiod,thesenderdoesnotreceiveanacknowledgement(since,receiverneverreceivedthepacket).



Afterwaiting,thesenderattemptstosendthesamepacketagain,withthe same probability and repeats until the package is sent successfully, i.e. itreceivesanacknowledgementforit.

Forthereceiver’sside,anacknowledgementissenteverytimeaframe,orbit,isreceived,albeit withasuccessrategiven bytheuser.



Therefore,someacknowledgementsgetlostinthechannel.Therefore,sincesenderdoesnotreceiveacknowledgement,ittimesoutandsendsthesameframeagain.Thereceiver,howeverhadalreadysuccessfullygottentheframe.Inthiscase,sincetheindicesdonotmatch-theacknowledgementforthepreviousframeissent.

**SENDER-CODE**

import socket

from threading import \*

import time

import random

#------------------------------------------------------------------------

#create socket object and bind it.

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

host = "localhost"

port = 8000

s.bind((host, port))

#------------------------------------------------------------------------

#defines the class for client.

class client(Thread):

#initialize everything

def \_\_init\_\_(self, socket, address):

Thread.\_\_init\_\_(self)

self.sock = socket

self.addr = address

self.start()

def run(self):

while True:

#take input from user

bitstring = str(raw\_input("enter bit string"))

propogationtime = float(raw\_input("enter propogation time"))

p\_nosend = float(input("enter probability of message getting lost:"))

#create a list of 1000 elements. (redundant, can be changed)

l = []

for i in range(0,1000):

l = l +[i]

#go through all the bits of the bitstring.

i = 0

while i < len(bitstring):

#for each bit, create a dictionary with the current

#window index (i%2 => 0,1,0,1...) and the bit itself

datadict = {}

datadict = {i%2 : bitstring[i], }

#convert the dictionary to a string

sendstring = str(datadict)

#find a random number between 0,1000 (both included)

number= random.randint(0,1000)

#store current time

time1 = time.time()

#if statement is true with a probability of (1-p\_nosend)

if l[number] < (p\_nosend\*1000):

#if true, send the bitstring.

clientsocket.send(sendstring)

#sleep to simulate the propogation time of the channel.

time.sleep((propogationtime)/1.1)

#after sleeping, record the current time.

time2 = time.time()

#this flag indicates whether an acknowledgement has been received.

ackflag= False

#else statement is true with a probability of (p\_nosend)

#this simulates a packet being sent but getting lost

#along the way (like our lives).

else:

#imaginary send line here.

#wait for propogation time again.

time.sleep(propogationtime/1.1)

#record current time

time2 = time.time()

#for info of user:

print ("package dropped 1")

#set ack flag to False, again.

ackflag= False

while True:

#if the time elapsed is less that prop time,

if time2-time1<= propogationtime:

#store current time

time2= time.time()

#set timeout to listen for an acknowledgement.

clientsocket.settimeout(propogationtime/1.1)

#raises exception when the timeout occurs.

try:

#attempt to listen for ack.

recieved = clientsocket.recv(1024)

print(recieved)

if recieved:

print "ack received"

i = i+1

ackflag = True

break

#this occurs when listening has timed out.

except:

#recheck if it has timed out (quite redundant):

if time2 - time1 >propogationtime and ackflag == False:

print ("timeout")

#at this stage, we again simulate a package drop

#with prob p\_nosend

number= random.randint(0,1000)

#package sent

if l[number] < (p\_nosend\*1000):

clientsocket.send(sendstring)

time1 = time.time()

time2 = time.time()

print "package sent"

#package dropped:

else:

time1 = time.time()

time2 = time.time()

#time.sleep(propogationtime/1.1)

print("package dropped 2")

s.listen(5)

print ('Sender ready and is listening')

while (True):

#to accept all incoming connections

clientsocket, address = s.accept()

print("Receiver "+str(address)+" connected")

#create a different thread for every

#incoming connection

client(clientsocket, address)

**RECEIVER-CODE**

import socket

import random

from ast import literal\_eval

#------------------------------------------------------------------------

#create socket object and bind it.

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

host ="localhost"

port =8000

s.connect((host,port))

#------------------------------------------------------------------------

#take input from user

p\_noack = float(input("enter probability of acknowledgement not being sent"))

count = 0

def try\_ack(previous, current):

if abs(previous-current) == 1:

return True

else:

return False

l = []

for i in range(0,1000):

l = l +[i]

output = ""

while 2:

data=s.recv(8).decode()

print("Received --> "+data)

datadict = literal\_eval(data)

index = list((datadict).keys())[0]

number= random.randint(0,1000)

if count == 0:

count = count +1

current = index

previous = 0

if current == 0:

previous = 1

#simulating the acknowldgement message being lost on the way

else:

count = count +1

previous = current

current = index

print ("p /c :", previous, current)

if try\_ack(previous, current):

output = output + list(datadict.values())[0]

#print ("hello", l[number],p\_noack\*1000 )

if l[number]<(p\_noack\*1000):

print "Ack not sent"

pass

else:

str="Acknowledgement: Message Received"

#print ("!!!!!!!!!")

#print datadict.values()

s.send(str.encode())

else:

str="Acknowledgement: Message Received"

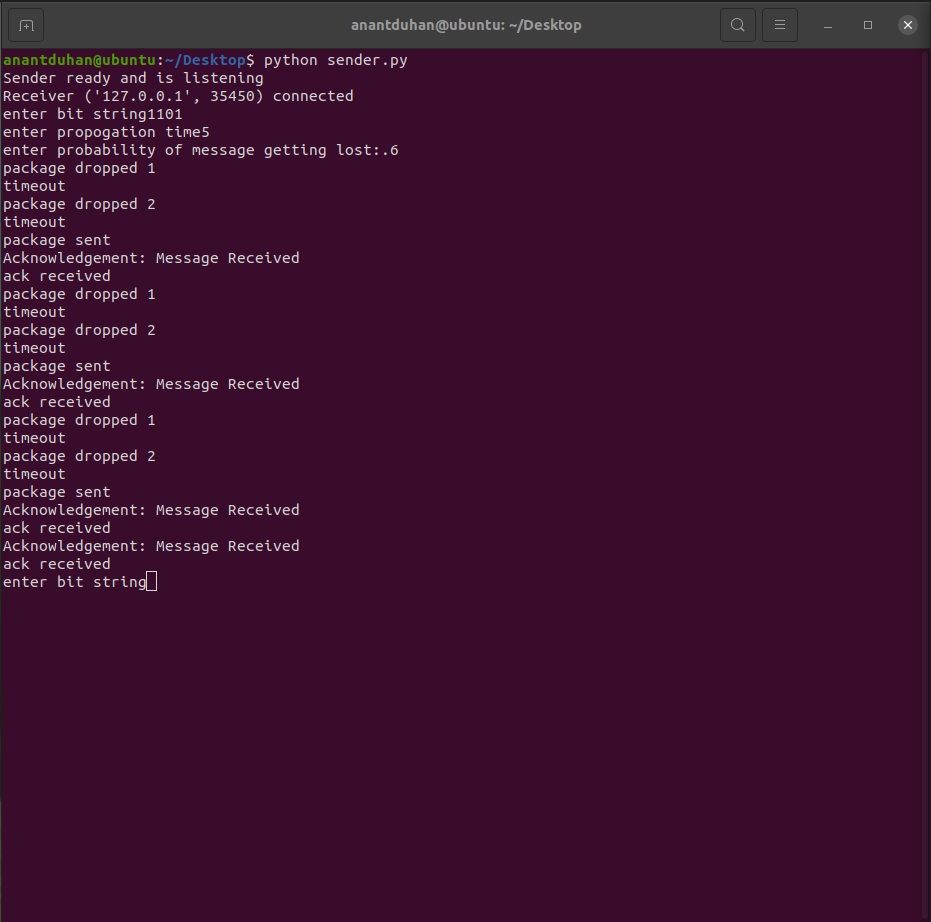
s.send(str.encode())

print ("indices did not match. Sending ack for previous element")

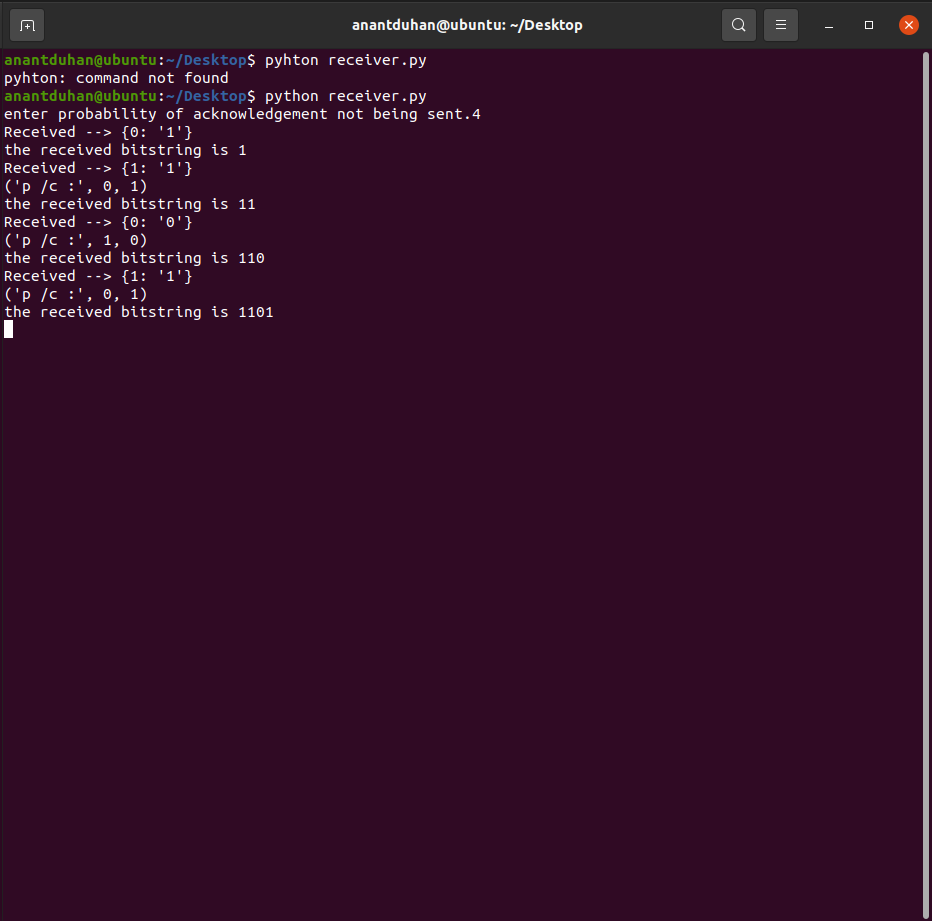
print "the received bitstring is",output

s.close ()

**OUTPUT – SENDER**



**OUTPUT - RECEIVER**



**EXPERIMENT RESULT AND ANALYSIS**

Wehaveimplementedthestop-and-waitprotocolthroughthisproject-andwhilewerealizethatitisindeedareliablemethodofcommunication(itensuresthatthesentframeisreceivedandacknowledgedreliably),wealsorealizedthatitisextremelyinefficient.Theconstant waiting, after sending a single frame makes the protocol slow andtherefore,werealizethattheGo-Back-NprotocolortheSelectiveRepeatprotocolisamuchfasterwaytoensurereliablecommunication.

ReportShouldcontainminimumof25pagesandmaximumof30pages