International Journal of
Computational Learning & Intelligence
Vol. 02, Issue 04 (October) 2023

RESEARCH ARTICLE OPEN ACCESS

Stop and Wait Protocol Using Python

Shivaraj L S . Likhith G . Yashas Kumar S . Omkar S . Naveen Chandra Gowda
School of Computer Science and Engineering
REVA University, Bengaluru, India

DOI: **10.5281/zenodo.10254217**
Received: 18 October 2023 / Revised: 16 November 2023 / Accepted: 01 December 2023

Abstract – The Stop and Wait protocol is a widely used method for achieving reliable data transmission in an unreliable communication channel. This protocol ensures data accuracy by utilizing acknowledgments and retransmission. Thisabstract presents an implementation of the StopandWait protocol using Python. The process involves establishing a connection between a sender andreceiver, dividing the data into packets, andtransmitting them one at a time. The sender waits for acknowledgments from the receiver and retransmits any lost packets. The receiver verifies the integrity of the received packets and sends acknowledgments to the sender. This iterative process continues until all data is successfully transmitted. The Python implementation provides a practical and versatile approach for simulating and testing reliable data transmission, enabling developers to evaluate the protocol's effectiveness and performance under different conditions and network scenarios.

Index Terms – Socket programming, Server –side, Client-side, Stop -Wait-Mechanism, Packet Structure, Error Handling, Low-Bandwidth Networks, Serial Communication, CRC Generator

I. INTRODUCTION

Stop and wait protocol is an error control protocol, in this protocol the sender sends data packets one at a time and waits for positive acknowledgment from the receiver's side, if acknowledgment is received then the sender sends the next data packet else it'll resend the previous packet until a positive acknowledgment is not received. Whenever packets of data need to be transmitted from point A (the transmitter) to point B (the receiver), there is always a chance that something bad happens to them while they move through the medium between A and B (the channel): some packets may be corrupted or even

lost entirely. To cope with this, ARQ (Automatic Repeat Request) protocols have been used to provide a more reliable way of communication between the transmitter and the receiver.

Applications

- Low-Bandwidth Networks: Essential in low-data-rate environments, ensuring reliable transmission with acknowledgment.
- Serial Communication: Used in serial communication to guarantee accurate one-bit-at-a-time data delivery.

Where it is required

The Stop-and-Wait protocol is required to ensure reliable data transmission in situations where acknowledgment of successful reception is necessary before sending the next piece of data. This prevents data loss, corruption, and helps maintain the order of transmitted information, making it suitable for scenarios where reliability is prioritized over high throughput.

Why it is required

The Stop-and-Wait protocol is a simple flow control mechanism used in communication systems. It ensures reliable data transfer by having the sender wait for an acknowledgment from the receiver before sending the next piece of data. This approach prevents data loss and maintains the sequential order of transmitted information. While straightforward, it may not be the most efficient for high-throughput scenarios, but it finds applications in low-bandwidth networks, serial communication, and situations where reliability is paramount.

II. LITERATURE SURVEY

The stop and wait protocol is a simple error control protocol that is used to ensure reliable data transmission over unreliable channels. It works by transmitting data one packet at a time and waiting for an acknowledgment (ACK) from the receiver before transmitting the next packet. If the sender does not receive an ACK within a certain time period, it will retransmit the packet. The stop and wait protocol is simple to implement and understand, but it is not very efficient. This is because the sender must wait for an ACK from the receiver before transmitting the next packet, which can lead to long delays. There are a number of Python implementations of the stop and wait protocol available. One popular implementation is the stop and wait module, which is available on the Python Package Index. The stop and wait module provides a simple and easy-to-use interface for implementing the stop and wait protocol.

Another popular Python implementation of the stop and wait protocol is the srpc module, which is also available on the Python Package Index. The srpc module provides a more sophisticated implementation of the stop and wait protocol than the stop and wait module. The srpc module supports a number of features that are not available in the stop and wait module, such as flow control and congestion control. The stop and wait protocol is a useful tool for implementing reliable data transmission over unreliable channels. It is simple to implement and understand, but it is not very

efficient. There are a number of Python implementations of the stop and wait protocol available, including the stop and wait module and the srpc module.

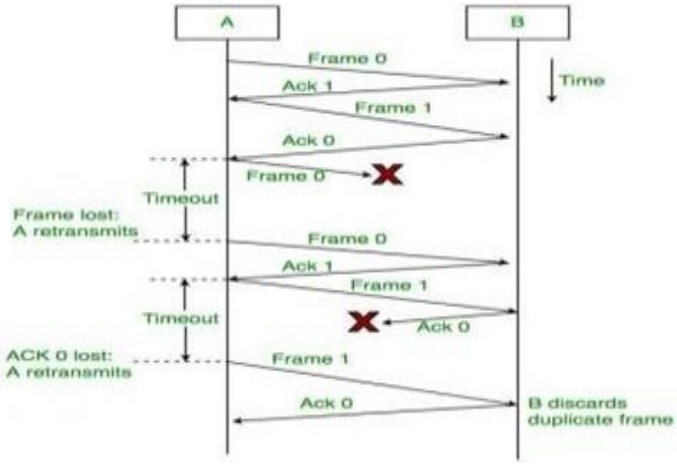

Fig. 1: working of stop and wait protocol

III. PROPOSED WORK

After studying the various techniques used to perform encipherment, we have done the comparison based on the following important factors: Error is introduced as follows :Generate a random number, say r1. Perform r1 % 2. If you get a 0 do not introduce error and send original bits. If you get a 1, introduce error.To decide which bit will be in error, generate another random number, say r2. Perform r2 %(size of received frame). Assume you get a value i as the outcome. Flip the i-th bit. Now send it to the receiver. Modulo two-division is required to calculate the CRC.configuration data that are shared by the reciver and the sender.

Client: The client class will have five methods. Constructor: To connect to the server using a socket at the given IP address and port number. asciiToBin: To convert ASCII string to binary string. appendZero: To append (k – 1) 0's to the end of binary data. encode: To generate and append CRC at the end of actual data bits. sendfile : This method reads n characters from the input file at a time &Creates the data packet to be sent by calling the encode method.Calls induce_error method to randomly introduce an error in the data packet.Sends the data packet and waits for the acknowledgment.If the acknowledgment received is positive, then move on to the next n bits,else resend the current data packet.When the file is completely read, then send a flag to tell the receiver to stop waiting for the next frame.Terminate the session.

Server: The server class will have six methods.Constructor: To listen for client request at the given IP address and port number. iszero : To determine if a string represents 0.isCurrupted : To determine if the received data is corrupted by performing modulo 2 division by the CRC_GENERATOR.decode : To extract the data bits from the received data packet and convert them to their ASCII values.log : To log the entry of each frame in the logfile.

receive_file : This method receives the data packet from the sender &Checks its validity by calling the is Currupted function. If the data packet is valid, then it decodes it and copies it in a file at server's end and sends a positive acknowledgement to the sender and logs the entry of the data packet in the logfile .else, it sends a negative acknowledgement to the sender. If the received data packet is the end of file, then it terminates all the connections and returns.

IV. RESULTS AND PERFORMANCE EVALUATION

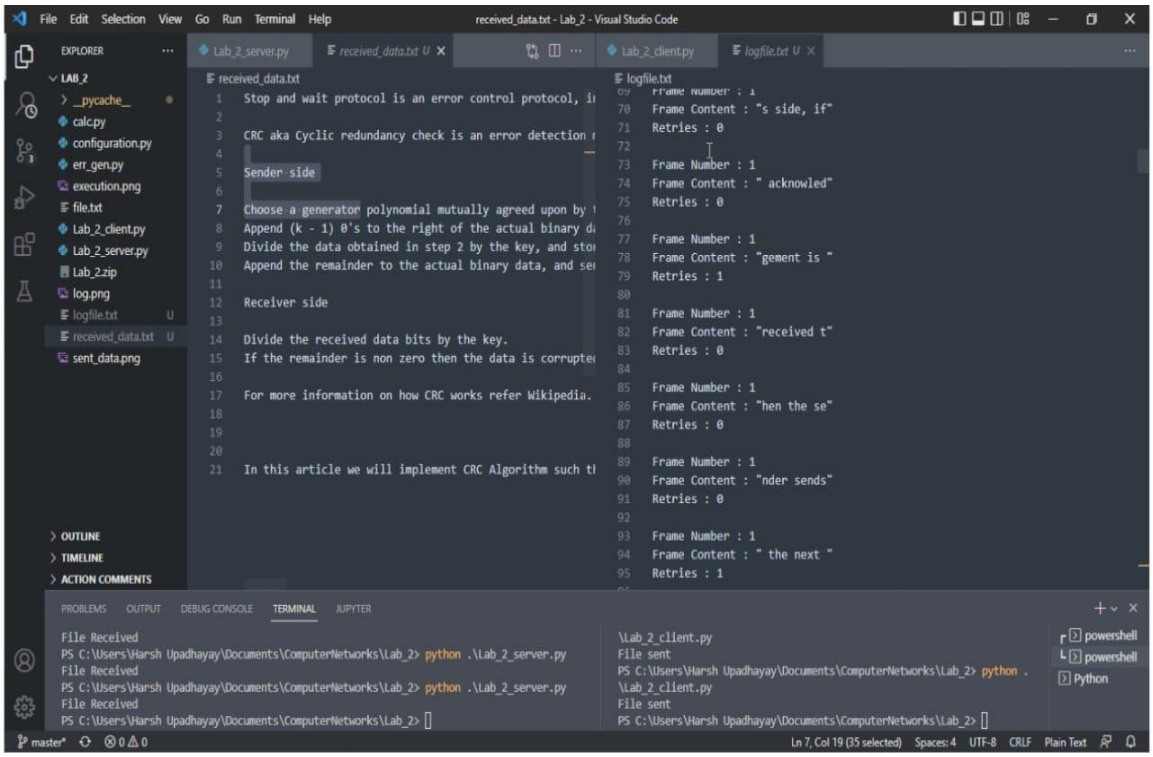

Fig. 2: log file showing input and output

Input and output files should match. A log file should show how many frames were in error and how many retries were done for each frame.

V. CONCLUSION

The Stop and Wait protocol is a reliable method for transmitting data between a sender and receiver in an unreliable communication channel. By using acknowledgments and retransmission, it ensures that data is accurately delivered. Implementing the protocol in Python involves establishing a connection, splitting the data into packets, and sending them one at a time. The sender waits for acknowledgments from the receiver and resends any lost packets. The receiver checks the integrity of the received packets and sends acknowledgments to the sender. This process continues until all data is successfully transmitted. Implementing the Stop and Wait protocol in Python allows for effective testing and simulation of reliable data transmission.

REFERENCES

1. A. Prasetyo, “Stop-and-Wait ARQ,” Dunia Informatika, [Online]. Available: http://duniainformatikaindonesia.blogspot.co.id/2013/03/Stop-and-Wait-arq.html. [Diakses 30 7 2016].
2. Ramadhan, Z., & Siahaan, A. P. U. (2016). Stop-and-Wait ARQ Technique for Repairing Frame and Acknowledgment Transmission. *International Journal of Engineering Trends and Technology*, *38*(7), 384-387.
3. Varthis, E. G., & Fotiadis, D. I. (2006). A comparison of stop-and-wait and go-back-N ARQ schemes for IEEE 802.11 e wireless infrared networks. *Computer communications*, *29*(8), 1015-1025.
4. Ramadhan, Z., & Siahaan, A. P. U. (2016). Stop-and-Wait ARQ Technique for Repairing Frame and Acknowledgment Transmission. *International Journal of Engineering Trends and Technology*, *38*(7), 384-387.
5. Raja, D. K., Kumar, G. H., Basha, S. M., & Ahmed, S. T. (2022). Recommendations based on integrated matrix time decomposition and clustering optimization. *International Journal of Performability Engineering*, *18*(4), 298.
6. Bukate, R. R. (2014). PM Ingale dan US Shid,“ARQ Strategies and Protocols for Relay Co-operative System,”. *International Journal For Advance Research In Engineering And Technology*, *2*(11), 24-28.
7. Ahmed, S. T., & Basha, S. M. (2022). *Information and Communication Theory-Source Coding Techniques-Part II*. MileStone Research Publications.
8. I. Ma'ruf, “Prinsip Kerja Stop-and-Wait Flow Control,” Blogspot, [Online]. Available: http://irham93.blogspot.co.id/2013/06/prinsip-kerja-Stop-and-Wait-flow-control.html. [Diakses 30 7 2016].