Implementing Security on NACK based Protocol

CSC8370 Data Security

Anitha Subramanian
002509602

Professor - Dr.Robert Harrison
Department of Computer Science

December 8, 2020

Abstract

By default, communication packets in NACK reliable protocol[3] are not encrypted. So, the project proposed here will be imposing security only for the packets which received NACK from the receiver so that the same packet will not be lost/stolen again and also reduce the transmission time for forthcoming data packets. This is also to avoid encryption cost on every data we sent over the network from sender to receiver. This problem specifically prevents the attack on data after it gets NACK.

Table of Contents

1 Introduction

A protocol is a standard set of rules that allow electronic devices to communicate with each other. Network protocols are the reason we can easily communicate with people all over the world, and thus play a critical role in modern digital communications. There are various number of protocols like communication, network management, security protocol, etc., Security protocols are also called cryptographic protocols to ensure that the network and data sent over it are protected from unauthorized users[4]. But the aim of this project is to provide security on NACK based protocol[3] whether it be TCP/IP or any other protocol. NACK or Negative Acknowledgement is a protocol message sent when something goes wrong between sender and receiver in the network. It is been assumed that hosts in the network should not give out information gratuitously. There are various attacks possible between sender and receiver over network communication. So there comes this ACK/NACK concept[5]. Since using ACK for every message will be expensive, this project is proposing to use NACK. This project will be imposing security techniques only on the data sent over the network rather than imposing network security. [**?**]

2 Importance, Impact and Benefits of Solution

2.1 Importance

In a network, data will be transmitted as packets from sender to receiver. Whenever the packets are lost/stolen, (by any means like eavesdropping or unauthorized attack or etc.,) there will be acknowledgment from receiver to send the packets again. There are 'n' number of security measures can be taken on transmitting messages from sender to receiver but there is no case for negative acknowledgements. Even though there is a very little probability of losing the data again, there is still a chance that the resent packets will get lost/stolen again. So to avoid the data loss, crypto techniques can be done only on the packets/data which received NACK. So whichever protocol is implementing negative acknowledgment for their communication, they can use this method to securely re-transmit the lost/stolen data packet. [**?**]

2.2 Impact

With this solution, the probability of losing/stealing the same data/packet repeatedly over the network can be avoided. The cost of using the crypto techniques will be lesser since we will be using the technique only for Negative acknowledgment data from receiver compared to using for the entire communication. So we can consider this as cost effective solution for data security in NACK based protocol[3].

2.3 Benefits

Applications using NACK relying protocol can use this solution for effective and secure communication. So applications like messenger/chat application, e-commerce applications, etc., Since most of the real world applications rely on stream oriented communication protocols, but only very few following NACK based. So this solution will be really helpful for those kind of applications.

3 Assumptions

Exchange of messages between sender and receiver can be anything and can be in any order. But ACK and NACK relying protocols work based on the sequence of packets being transmitted between sender and receiver. So to simulate sequence, numbers have been added to the packets to identify their sequences. Whereas in real scenario, there will be automatic tracking of message sequences (based on time). And I generated packets on my own to depict the real world scenario.

And also for simulating packet lost/stolen, random probability had been generated in the code. Based on that probability, total number of packets lost/stolen will be changed.For example, 1/10 packets will get lost one time and 3/10 will be other time. This is just to simulate the real world scenario.

And regarding crypto techniques, after encryption i sent the secret key along with encrypted data separated by comma. And at receiver side, the transmitted data has been divided into 2 by comma delimiter and first part is secret key, second part is encrypted data which needs to be decrypted. This is purely based on the assumption that key and encrypted data has been separated by comma and it has been known only to sender and receiver. That is message confidentiality established between sender and receiver.

4 Methodology

This is the new solution proposed for implementing security on NACK based protocol. Existing solutions are available to handle/prevent the attacks in TCP/IP with ACK[5]. But there is no solution available for TCP/IP with NACK. So I thought of implementing security on NACK based communication.

Applications using NACK relying protocol can use this solution for effective and secure communication. So applications like messenger/chat application, e-commerce applications, etc., Since most of the real world applications rely on stream oriented communication protocols, but only very few following NACK based. So this solution will be really helpful for those kind of applications.

Several comprehensive defenses has been used nowadays to prevent the attacks. Some of them are,

4.1 Authentication

Some form of Cryptographic authentication is being used by applications nowadays. As usual, there are lot of approaches but the well known is Needham-Schroeder algorithm (exist for both private-key and public-key cryptosystems)[1]. This algorithm relies on each host sharing a key with an authentication server. Whichever host wish to establish a connection obtains a session key from the authentication server and passes a sealed version along to the destination. This pre-authenticated connection can ensure safety[1].

4.2 Encryption

This is been done by Authenticated Encryption with Associated Data (AEAD) and using encryption frame[2]. Sender breaks data into series of chunks. Each chunk is placed in a data field of plain text value and then encrypted to yield a frame's ciphertext field. An associated data value is constructed for each frame with Frame ID and sent over the network. The receiver reconstructs the data and Frame ID values using decryption[2].

Still there are lot of approaches used around the world but these two are the most common.

I have implemented Encryption technique alone on NACK protocol to impose security.

4.3 Theory behind Solution

This project have been implemented in the following ways,
• Create a NACK only reliable protocol with sender, receiver and router. The data will be sent in packets from sender to receiver via router.
• Adding router to the program to mimic the real world scenario.
• Router will store the incoming data packets from sender and follows "Store and go" procedure to direct to receiver.
• Receiver will receive the data packets and check for the sequence number. If any data lost/stolen, it will send NACK message.
• To simulate data stolen/lost, I will be dropping the packets manually with some probability.
• When NACK message is received, Router will re-transmit the data to receiver after encrypting it. This is to avoid data loss/steal again.
• I have displayed how many data has been stolen/lost in the overall transmission of data packets.
• When the packet is lost while transmitting from sener to receiver, receiver will send the NACK to sender. Sender will encrypt the same packet again with secret key and resent it.
• At receiver side, decryption will be done with the secret key and it will check for the expected packet again.

This method will ensure less cost of cryptographic security for the entire network by imposing only for NACK messages. Advantage of using NACK is there will be less transmission acknowledgement between receiver and sender thereby original data transmission time will be lesser.

5 Solution

5.1 Design

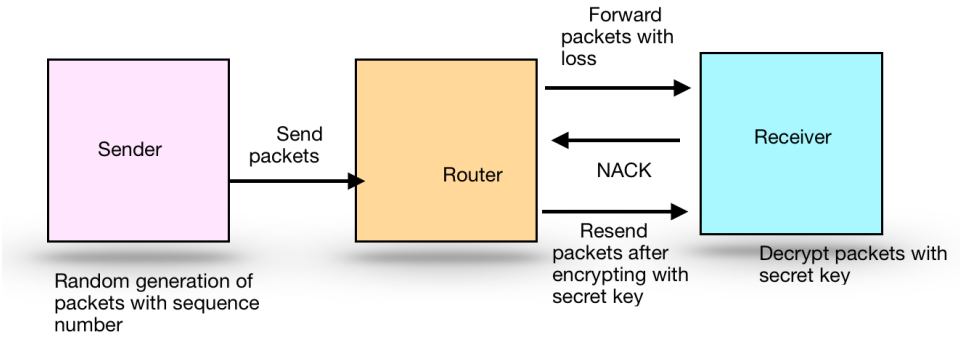
To implement NACK based transmission control protocol, I have created three programs which includes tcpReceiver, tcpRouter and tcpSender. This is based on socket programming which establishes connection between sender, router and receiver.

tcpReceiver - Acts like a server, which will accept connection to router for sending packets. This will intimate the router if there is a packet loss by sending NACK. This will decrypt the encrypted lost packet resent from router.

tcpRouter - Acts like a middleman, which will get the packets from sender, then store it and forward to receiver. This will drop packets based on some probability (calculated as per assignment question). Encryption will be done here before re-transmitting lost packets.

tcpSender - Acts like a client, which will send packets to router.

Below is the design of protocol. This Router will store the packets that receives from sender and then forward those to receiver.



5.2 Program Structure

tcpSender.java

• Established Socket connection using Socket senderSocket = new Socket(host, PORT);
• Generated random messages as packet. I used the structure of packet as 0 Packet, 1

Packet. . . where number shows the sequence number.
• Create ObjectOutputStream object to send the packets to router. Send "Exit" to show the end of packet sending.
• Close the Socket connection.

tcpRouter.java
• Establish both Server Socket and Socket connection to create communication between sender and receiver.
• Server Socket for receiving packets from Sender.
• Socket for sending packets to Receiver.
• Identify the probability of losing packets here. My panther id ends with 02 and as I work alone for this assignment I reversed the number to show some loss.
• Create random number to lose the packet randomly.
• Get packets from sender and store it here.
• Send the packets to Receiver.
• If the flag value received is greater than 0, then it means NACK and that number shows the sequence of packet lost.
• To show the random loss of packet, incremented the packet counter without sending.
• When NACK is received from Receiver, call the function encryptmessage() to encrypt the lost packet with secret key.

tcpReceiver.java
• Establish Server Socket connection to create communication with Router.
• Get the packet from router and check for null or lost packet.
• If the packet is lost, send the flag.
• Router will re-send the lost packet by encrypting with secret key. Here the receiver will decrypt it with the secret key shared.
• Check if the received packet is expected by comparing the sequence number.
• If they are same, get the next packet. Continue step 2 to 4 until it receives the exit.
• Close the socket connection.

6 Results

6.1 Data Generation

This project will be simulating data packets as sequence number (eg., 1 to 100 to show 100 data). This is also to address the data along with sequence number. So the data will be sent like 1,2,3,..100. If it receives like 1,2,4,5 then NACK message will be sent from receiver to say that 3 data packet is stolen/lost. In real world applications, we will be adding sequence number to data packets, here am using the sequence numbers as test data to satisfy both the requirements.

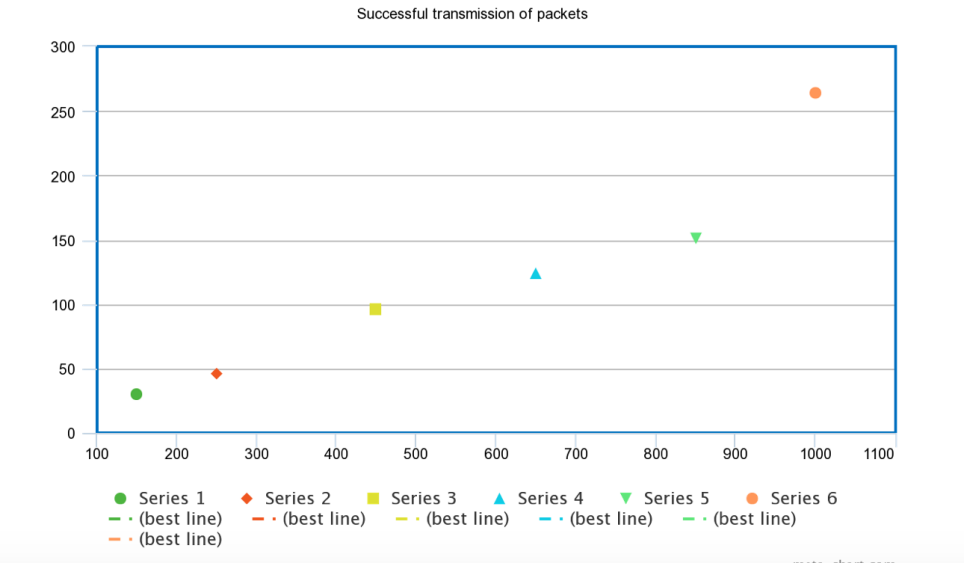
6.2 Data Limitation

Message packets should start only with numbers to show the sequence of packets being sent between sender and receiver. I have tested only for 10 packets for better clarity and understanding. But, we can increase to any number and test.
Programming language used - Java
Protocol - TCP/IP
Acknowledgement - NACK
Crypto techniques used - Encryption/Decrytion
Java library for encoder/decoder - java.util.Base64
Java libraries for Cryptography - javax.crypto

6.3 Testing Strategy

I have used sequence numbers to test for the correct message packet at receiver side and sent NACK based on that. And also used console window to check for message transmission.
Lost packets are recovered and encrypted in 1 try in this program.

Y = 150 = lost 30 packets = Retired and Encryption done for 30 times
Y = 250 = lost 46 packets = Retired and Encryption done for 46 times
Y = 450 = lost 96 packets = Retired and Encryption done for 96 times
Y = 650 = lost 124 packets = Retired and Encryption done for 124 times
Y = 850 = lost 171 packets = Retired and Encryption done for 151 times
Y = 1000 = lost 264 packets = Retired and Encryption done for 264 times



We could see there is potential loss in message packets sent between router and receiver. So to avoid future loss of same packets again and again, we are implementing this solution.

7 Future Work and Directions

Since networking and communications are the most important factor that drives this world, this work can be implemented mainly for login applications or login page of websites to prevent attacking on password. Since password is the sensitive and prone to attacks, we can implement this NACK based security on this kind of data. Suppose password is lost/stolen, there is high possibility of stealing the new password again. So in that case, we can implement this strategy to prevent attacks on communication.

Regarding directions for using this project, since most of the time communication happens between sender and receiver through routers and there is a high possibility of eavesdropping/packet loss, this implementation has been done. I have just implemented with one router and it is using store and forward method for routing the packets. We can use any method in router to forward the packets to receiver. And I used simple encryption technique to show the data security on network communication. While implementing for real world, more secure and robust crypto techniques can be used to ensure higher level of security.

8 Lessons Learned

• At router, Writing code to simulate data loss or data steal. (ie.,dropping/losing packets/) Implementing eavesdropping was really difficult.
• Imposing security only on particular data.
• Sending key along with encrypted data.
• Identifying which crypto technique suitable for this NACK reliable protocol communication.

References

[1] Prabhaker Mateti. "(PDF) Chapter 1 Security Issues in the TCP/IP Suite." Chapter 1 Security Issues in the TCP/IP Suite. Accessed November 18, 2020. https://www.researchgate.net/public

[2] A. Bittau, D. Giffin, M. Handley, D. Mazieres, Q. Slack, and E. Smith. "Cryptographic Protection of TCP Streams (Tcpcrypt)." IETF Tools. Accessed November 17, 2020. https://tools.ietf.org/html/rfc8548.

[3] Praveen Gupta, Srinivasan Balasubramanian, and Radhachandran Padmanaban. "US20050175012A1 - System and Method for Transmitting and Receiving Data Frames in a NAK-Based Win- dow Protocol." Google

Patents. Google. Accessed November 17, 2020. https://patents.google.com/paten

[4] Sen, Jaydip. "Theory and Practice of Cryptography and Network Security Protocols and Technologies." IntechOpen, July 17, 2013. https://www.intechope and-practice-of-cryptography-and-network-security-protocols-and-technologies.

[5] Guru. "TCP 3-Way Handshake (SYN, SYN-ACK,ACK)." Guru99. Accessed November 18, 2020. https://www.guru99.com/tcp-3-way-handshake.html.