Protocol Design

COSC 635 – Network and Data Communications I

Project 2 – Team 2

John Fore (jwfore0), Brenton Kinney (bkinney0), Kyle Voon (kvoon0), Asif Mehboob (smehboob0)

# Scope

This document addresses the specifications and limitations of the error control protocol simulator that our team will be implementing. The simulator will provide an option to use both Stop-And-Wait (SAW) and Go-Back-N (GBN) protocols.

# Objective

The simulator will allow data to be passed between two computers using the User Datagram Protocol (UDP). It will utilize implemented error handling algorithms that will make the simulator’s data transfer more reliable. In general, the simulator will send a large text file over the UDP communication protocol using the chosen error control protocol. The simulator will not implement any error-detection or error-correction mechanisms.

# General Specifications

* The simulator will be implemented using C# on .NET Framework 4.7.2.
* The simulator will provide an option to choose the mode of the session – Client or Server.
  + Server mode will display the user’s IP Address and allow the user to choose a port number.
  + Client Mode will allow the user to enter the server’s IP Address and port to connect to.
* The simulator will provide the user to use either error control protocol for the session.
* The simulator will provide an option to choose a text file to read data from.
* The simulator will allow the user to enter a number to indicate the percent of times the simulator will simulate a failed transmission.
  + The simulator will only assume transmission failures in sending packets and not in acknowledgements.
* The simulator will have default values for each protocol which can be updated by the user.
* The simulator provides a running log of the functions being performed.
* The simulator will allow the user to download the statistics of the transfer session after the transfer is complete.

# Protocol Specifications

## Stop-And-Wait (SAW)

* Maximum Payload Size – 1000 bytes
* Header Fields:
  + Sequence Number
    - Size – 1 byte
    - Possible Values – 0, 1
    - Data Type – 8-bit integer
  + Message Length
    - Size – 2 bytes
    - Possible Values – 1 through 1000

*(We decided to limit the possible values even though the data type supports values up to 65535)*

* + - Data Type – 16-bit integer
  + Packet Status
    - Size – 1 byte
    - Possible Values – 0, 1, 2
    - Value Description:
      * 0 – Start of File
      * 1 – Normal
      * 2 – End of File
    - Data Type – 8-bit integer
* Number of Header Fields – 3
* Total Size of Header – 4 bytes
* Message Sequencing – Alternate between 0 and 1
* Timeout Interval – 3 seconds

## Go-Back-N (GBN)

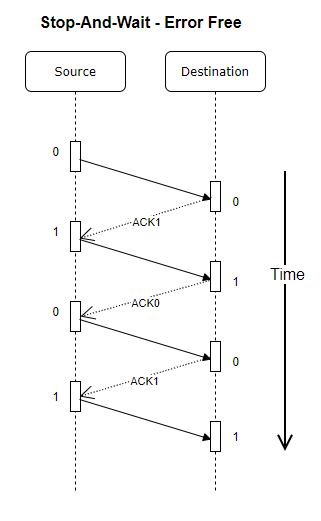
* Maximum Payload Size – 1000 bytes
* Header Fields:
  + Sequence Number
    - Size – 1 byte
    - Possible Values – 0 through 7
    - Data Type – 8-bit integer
  + Message Length
    - Size – 2 bytes
    - Possible Values – 1 through 1000

*(We decided to limit the possible values even though the data type supports values up to 65535)*

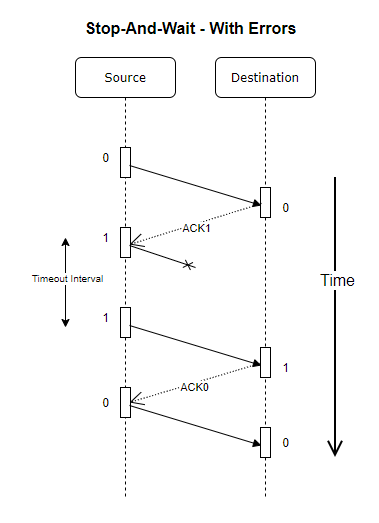
* + - Data Type – 16-bit integer
  + Packet Status
    - Size – 1 byte
    - Possible Values – 3, 4, 5
    - Value Description:
      * 3 – Start of File
      * 4 – Normal
      * 5 – End of File
    - Data Type – 8-bit integer
* Number of Header Fields – 3
* Total Size of Header – 4 bytes
* Message Sequencing – 0 through 7 and then repeat
* Window Size – 7

# Diagrams

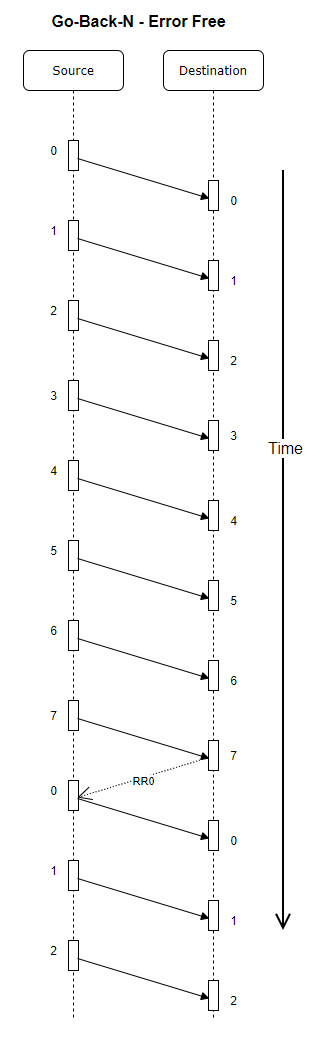
## SAW



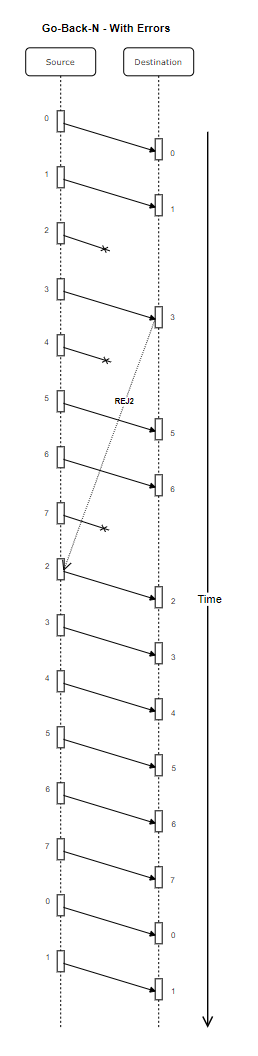
## SAW With Error



## GBN



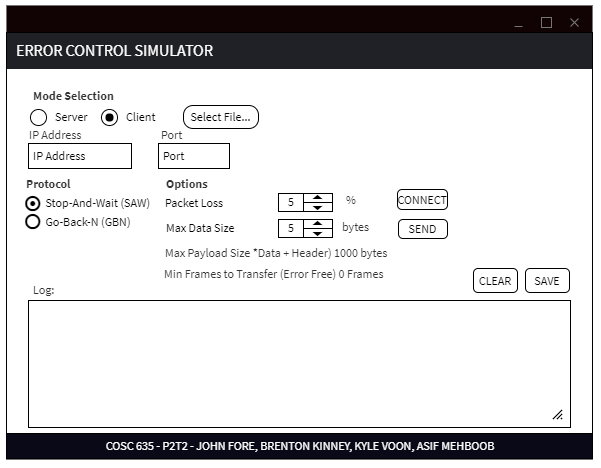
## GBN With Error



# User Interface Specifications

* The User Interface will be implemented using C# WinForms
* The User Interface will have minimization, screen size adjustment and exit application section.
* The User Interface will provide a full GUI with options to choose between Client or Server mode with the use of radial buttons.
* The User interface will display logs of data received or transmitted with options to save or clear logs between client and server modes.
* The server mode will have options for:
  + User input of socket port, and received data save location.
  + Button to listen for socket port interaction.
* The client mode will have options for:
  + User input of IP address and socket port used
  + Radial Buttons for options using Stop-And-Wait (SAW) or Go-Back-N (GBN) protocol.
  + Option to indicate package loss percentage.
  + Button to select file from local computer to transmit to server client.
  + Button to connect to server through socket port.
  + Button to appear once client is connected to server to send data.

## Client



## Server

