**Software Design Document**

Ericsson

Chris Berstler

Jeff Geiser

Shane Reetz

Sam Stutsman

1. Overview

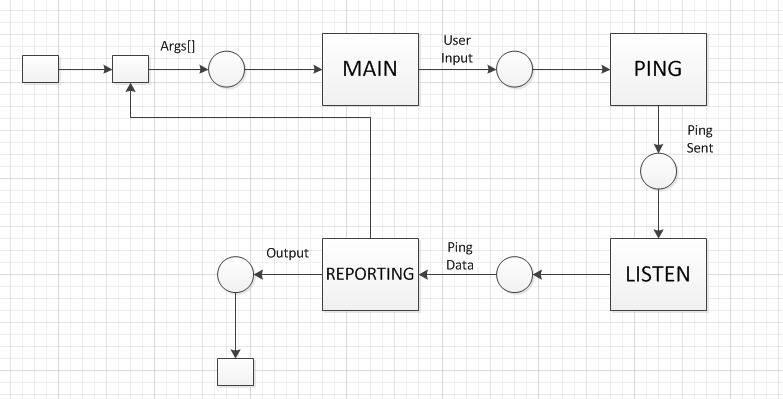
1.1 Introduction

The program tentatively called ePing will create and send out ICMP Echo Requests given various parameters, then display some statistics.

1.2 Scope

This is a high level design document detailing the primary features of ePing and its command line usage.

2. High-Level Design

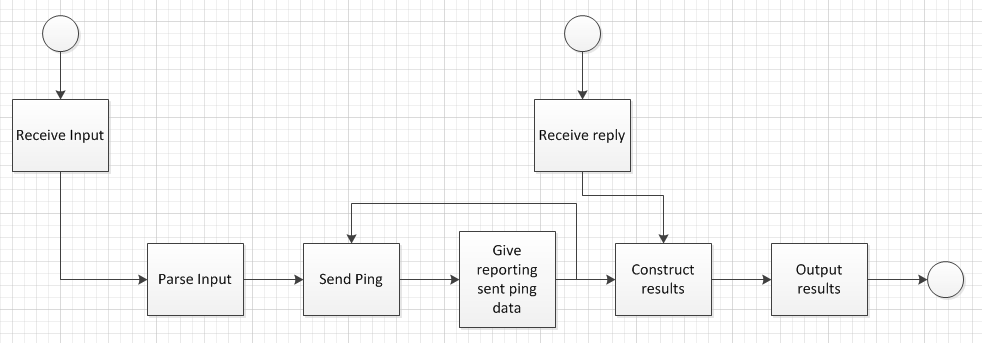


2.1 High-Level Component Design

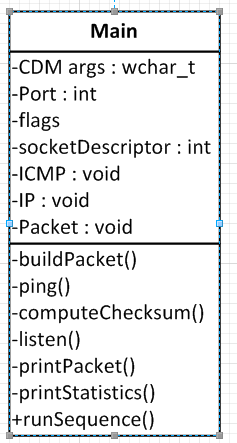
|  |  |
| --- | --- |
| **Component** | **Description** |
| Build | Constructs the packets based on the parameters |
| Main | Handles primary functionality and handles packets |
| Send | Sends packets |
| PrintPacket | Prints packets as they are received in real time |
| PrintStatistics | Prints the statistical summary |
| Listen | Listen for replies |

|  |  |
| --- | --- |
| **Component** | **Related Requirements** |
| Build | * A random ICMP payload size should be possible to command in a given interval * It shall be possible to set an increasing payload size where the payload is increased with a configurable number of bytes for each successful ping. |
| Main | |  |  | | --- | --- | | * Develop a ping utility that can be distributed as a single stand alone executable, in windows, with no installation | * The utility should be able to run under an extended mode where the utility autocompletes user input. | | * It shall be possible to set # of pings that should be sent and print average/standard deviation for the series. |  | | . | * It shall be possible to set a number of pings in the beginning of the series that should be excluded from the summary. | | * The granularity of the measurement should be microseconds if this can be achieved with statistical significance, otherwise milliseconds. | * It should be possible to send UDP packets with the utility, with configurable UDP payload size, a UDP port and a recipient. | | * It shall be possible to set the time between pings and choose if it is the time between each ICMP Echo Request or ICMP Echo reply. | * It shall be possible to set the ping utility in UDP listening mode. When this is done it listens on a configurable UDP port for a UDP Echo Request and returns a UDP Datagram with the same size back to the sender. | | * A random time between pings should be possible to command in a given interval. | * Further UDP settings like different size on return path etc. | | * It shall be possible to set the size of each ping and choose if this is the payload size or the total IP datagram size. | * Add support for IPv6 | |  | |
| Send |  |
| PrintPacket |  |
| PrintStatistics | * The utility should have an option to output results to .csv file |
| Listen |  |

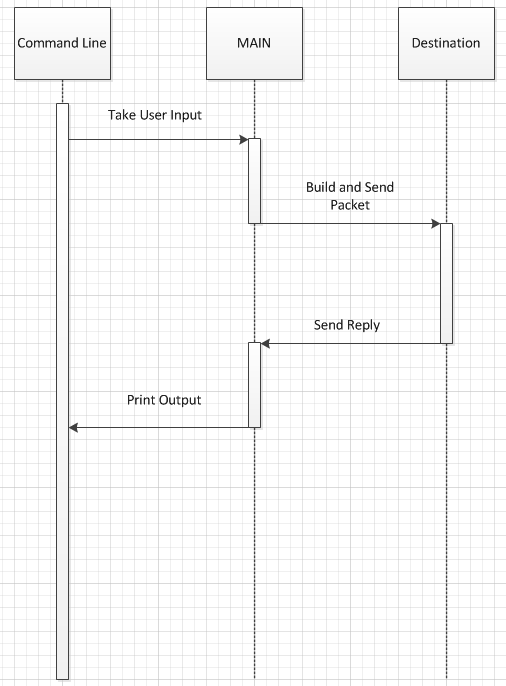
2.2 Activity Diagrams



2.3 Class Diagram



2.4 Sequence Diagram

****

3. User Interface Design

3.1 UI Description

The UI of the program will be a command line interface with various parameters. More than the following parameters listed may end up in the resulting program completed at the end of the semester.

The interface will begin with something along the lines of ‘**ePing <address>**’followed by zero or more of the following flags.

|  |  |  |  |
| --- | --- | --- | --- |
| **Flag** | **Parameters** | **Description** | **Mutually Excludes** |
| -q | <time between requests> | Sets the time between sending multiple requests in milliseconds. | -b -s -t |
| -b | <time between reply and next request> | Sets the time between when a reply is received and when the next ping is sent out | -q -s -t |
| -d | <datagram size> | Sets the size of the entire ICMP packet | -p -l -r |
| -p | <payload size> | Sets the size of the payload of the ICMP packet | -d -l -r |
| -l | <min> <max> | Randomizes the size of the payload between min and max, uses discrete uniform distribution | -d -p -i -r |
| -r | <avg> <std.dev.> | Randomizes the size of the payload using an average and standard deviation distribution | -d -p -i -l |
| -s | <min> <max> | Randomizes the time between sending multiple requests between min and max, uses discrete uniform distribution | -q -b -t |
| -t | <avg> <std.dev.> | Randomizes the time between sending multiple requests using an average and standard deviation distribution | -q -b -s |
| -i | <initial size> <rate of increase> | Sets sequentially increasing size of pings | -d -p -l -r |
| -e | <number of pings to exclude> | Sets the number of pings to be excluded from statistics |  |
| -n | <number of pings> | Sets the number of pings to be sent out |  |

3.2 UI Mockup

The following are some examples of command line inputs and descriptions about what exactly will happen.

|  |  |
| --- | --- |
| ePing yahoo.com | Sends out a single ping to yahoo.com of default size. |
| ePing yahoo.com -n 50 -q 500 -e 5 | Sends 50 pings to yahoo.com which are sent 500 milliseconds apart, and excludes 5 pings from final statistics. |
| ePing yahoo.com -l 30 50 -b 88 -n 100 | Sends 100 pings to yahoo.com which are of random size, uniformly distributed between 30 and 50. The pings are sent 88 milliseconds after a reply is received. |
| ePing yahoo.com -n 8 -r 20 3 -I 30 30 | Will return an error, no ping will be sent. |

4. Data Design

4.1 Program Data

ePing will be able to log statistics into a .csv file and save it on harddrive.

4.2 Data Formats

We will use the struct datatype to hold the data to be packed into the packet.

5. Non-Functional Design

5.1 Security

Consideration has been made into preventing ping abuse. Pings of an unnecessarily large size or too many pings of too large size will not be allowed in order to not choke a destination’s bandwidth and/or steal cycles clock cycles.

5.2 Performance

Overhead will be at a minimum, but there should be no problems running this on any command line on any Internet-capable computer.

**6. Document Review**

|  |  |  |  |
| --- | --- | --- | --- |
| Name: | <Name> | Date: | <Date> |
| Signature: | | | |