**Software Requirements Specification**

Ericsson

Chris Berstler

Jeff Geiser

Shane Reetz

Sam Stutsman

**1.** **Project Overview**

**1.1 Vision**

The purpose of this project is to create a flexible ping utility for the testing of various telecommunications networks. Having this utility available will reduce the amount of scripting required network technicians and increase the value of their tests by providing additional options and feedback. Users will be able to set several options about how the packets will be sent out, and what kind of results summary they will receive back.

**1.2 Overall Description**

This software will allow the user to run the utility from a command line and, if time permits, through a GUI. Running this utility will be very similar to other popular ping utilities such as the Microsoft implementation, Linux implementation, and hrPing. There are many functions which may prove useful in one way or another, however due to the short period of time allowed for development, only the most pressing of requirements given by Ericsson will be implemented.

**2. Software Requirements**

**2.1 High-Level Requirements**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Added** | **Description** | **Status** | **Dependencies** |
| H100 | Onset | Application shall be platform independent | **C** | L100 |
| H200 | Onset | Application shall run on the windows command line | **C** |  |
| H300 | Onset | Shall not require installation | **C** | L100 |
| H400 | Onset | The application shall have minimal overhead | **C** |  |
| H500 | Onset | Graphical User Interface | **T** |  |

**2.2 Low-Level Requirements**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Added** | **Slogan** | **Description** | **Sponsor comments** | **Verification/Validation** |
| L100 | Onset | Application | Develop a ping utility that can be distributed as a single stand-alone executable. The utility should run under a windows environment without any installation. |  | Attempt to use utility on Mac, Linux, and Windows Operating systems |
| L200 | Onset | Basic ICMP | It shall be possible to set how many pings that should be sent and the application should print average and standard deviation for the series. |  | Verify that the number of pings sent equals number of pings requested via packet sniffing software. |
| L300 | Onset | Output | The utility should have an option for sending the output to a .csv file. If that option is used only the results of the file should be printed there, not the summary. |  | CSV file |
| L400 | Onset | Time Granularity | The granularity of the measurement should be microseconds if this can be achieved with statistical significance, otherwise milliseconds |  |  |
| L500 | Onset | Time Settings | It shall be possible to set the time between each ping and choose if it is the time between each ICMP Echo Request or the time between the last ICMP Echo reply and the next ICMP Echo Request. |  | Implementing and testing time will be verification for time between ICMP Echo Requests, validation will be completion of requirement |
| L600 | Onset | Random Time | A random time between pings should be possible to command in a given interval. |  | Implementing and testing time will allow for testing of random time intervals |
| L700 | Onset | Ping Size | 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. | **As you mentioned it might be necessary to put some restrictions on final output throughput.** | A packet sniffer will be able to determine if our packet sizes match what we anticipate them to be sending them out |
| L800 | Onset | Random Payload | A random ICMP payload size should be possible to command in a given interval. | **I'm thinking a normal distribution with just a mean and a standard deviation as input, e.g. ePing.exe -RP [200,50]** | A packet sniffer will be able to determine if our packet sizes match what we anticipate them to be sending them out |
| L900 | Onset | Increasing Payload | 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. |  | A packet sniffer will be able to determine if our packet sizes match what we anticipate them to be sending them out |
| L1000 | Onset | Excluded Pings | It shall be possible to set a number of pings in the beginning of the series that should be excluded from the summary. |  | Statistics done by hand excluding whichever pings should match statistics done by program. |
| L1100 | Onset | Basic UDP Mode Sending | It should be possible to send UDP packets with the utility. As a first step it should be possible to configure UDP payload size, a UDP port and a recipient. |  | UDP Listening and Broadcasting will require extensive testing on both host and server-ends |
| L1200 | Onset | Basic UDP Mode Listening | 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. |  | UDP Listening and Broadcasting will require extensive testing on both host and server-ends |
| L1300 | Onset | Advanced UDP Mode | Further UDP settings like different size on return path | **Will be discussed later if time admits.** | UDP Listening and Broadcasting will require extensive testing on both host and server-ends |
| L1400 | Onset | IPv6 | Make sure the application can handle IPv6. | **Let’s discuss this when most of the basics are covered.** | Simple IPv6 requests/look ups will be needed to ensure that they IPv6 is being sent properly. A packet sniffer will also aid in determining IP version. |
| L1500 | Onset | Extended mode | The utility should be able to run under an extended mode where the utility autocompletes user input. |  | Autocompletion will be validated and approved at Ericsson’s discression. |

**3. Dependencies and Constraints**

|  |  |
| --- | --- |
| **Dependencies and Constraints** | **ID** |
| Product language will be in English only | 100 |

**4. Acronyms and Definition**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| ICMP | Internet Control Message Protocol |
| UDP | User Datagram Protocol |
| IPv4 | Internet Protocol version 4, the old standard |
| IPv6 | Internet Protocol version 6, the new standard |
| .csv | Comma Separated Value file extension |
| GUI | Graphical User Interface |

**5. Requirements Review**

