# PROJECT DESIGN

**013 April 2014**

**Team C**

**Jamie Lane, Bradley Norman, Daniel Ross**

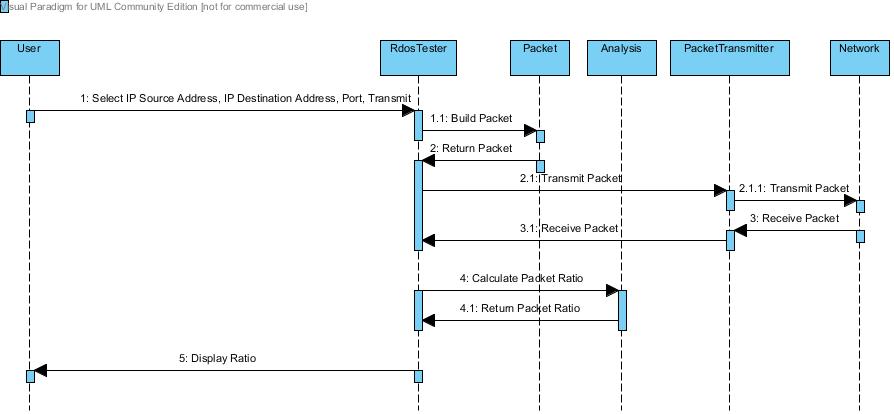
# Introduction

The RDOS Tester design document explains the static and dynamic sides of the design. The dynamic side of the design is shown in event-trace diagrams that demonstrate the following scenarios expected in the system: Start-up, Normal Operation, Error-handling, and Shut-down. The static side of the design is shown in the classes which have functionalities described using pseudocode.

# Event-Trace Diagrams

## Scenario 1: Start-up

## Scenario 2: Normal Operation



## Scenario 3: Error-Handling

## Scenario 4: Shut-down

# Class Design

## Input/Output Subsystem

Class RdosTester

{

// Initialize the text fields

TextField sourceIP1 = new TextField(size);

set sourceIP1 label “Source IP Address”;

TextField sourceIP2 = new TextField(size);

set sourceIP2 label “.“;

TextField sourceIP3 = new TextField(size);

set sourceIP3 label “.“;

TextField sourceIP4 = new TextField(size);

set sourceIP4 label “.“;

TextField destinationIP1 = new TextField(size);

set destinationIP1 label “Destination IP Address“;

TextField destinationIP2 = new TextField(size);

set destinationIP2 label “.“;

TextField destinationIP3 = new TextField(size);

set destinationIP3 label “.“;

TextField destinationIP4 = new TextField(size);

set destinationIP4 label “.“;

TextField port = new TextField(size);

set port label “Port”;

//Create button

Button transmit = new Button;

set transmit label “Transmit”;

listen to button;

//Create a status bar to display messages

Panel statusBar = new panel();

//Put the components in a panel

Panel panel = new panel();

set the layout of the panel;

add sourceIP1, sourceIP2, sourceIP3, sourceIP4 to panel;

add destinationIP1, destinationIP2, destinationIP3, destinationIP4 to panel;

add port to panel;

add button to panel;

add statusBar to panel;

void actionPerformed(action)

{

if button was pushed

{

concatenate sourceIP text fields;

concatenate destinationIP text fields;

Packet originalPacket = new Packet(concatenated sourceIP, concatenated

destinationIP, port);

int originalSize = get originalPacket size();

try to transmit/receive packets

PacketTransmitter packetTransmitter = new PacketTransmitter(originalPacket);

Packet returnedPacket = get packetTransmitter returned packet;

int returnedSize = get returnedPacket size();

double ratio = Analysis(originalSize, returnedSize);

message = “Original Packet/Received Packet Ratio is: “ ratio;

if unsuccessful, message = “Transmission Error. Please try again.”

set statusBar to display message;

}

void createAndShowGUI()

{

//Create the window

Window frame = new Window(window name);

set frame to close when user hits X button;

add the RdosTester main panel to frame;

display the frame;

}

void main()

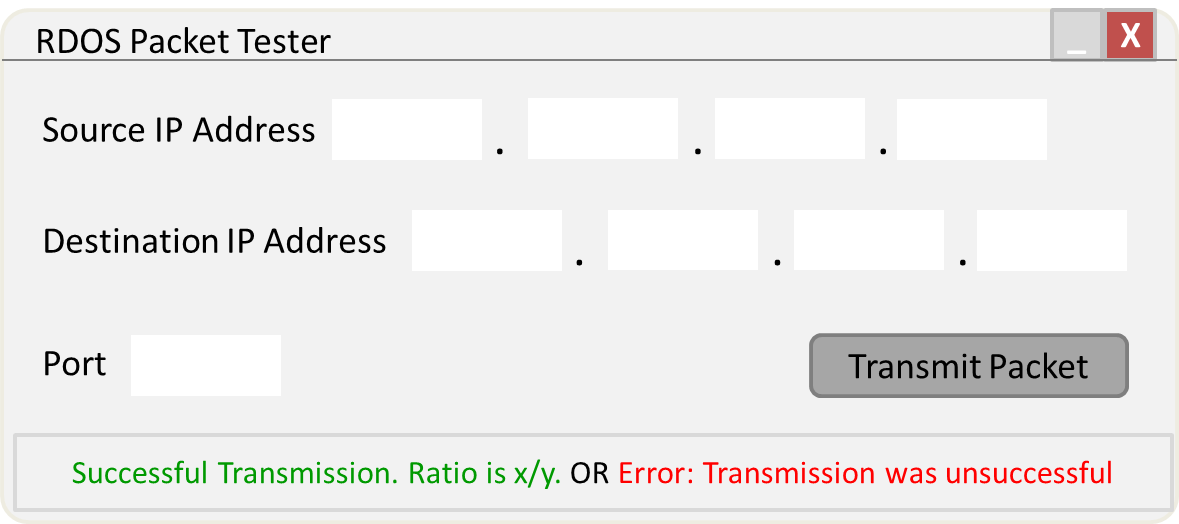
{

run;

createAndShowGUI();

}

}



## Packet Subsystem

## Transmission Subsystem

## Analysis Subsystem