BLEKINGE INSTITUTE OF TECHNOLOGY

ACCEPTANCE TEST PLAN

TCP Evaluation in Semi-Live Streams

“Team Enigma”

Anirudh Kodaru

Hemanth Kumar Ravuri

Nandini Chowdary Godavarthi

Naren Naga Pavan Prithvi Tanneedi

Reventh Thiruvallur Vangeepuram

Sasidhar Podapati

Sathvik Katam

Srinand Kona

SriKavya Chavali

Vaibhav Bajaj

Venkata Sathya Sita J S Ravu

Version 1.3
Publication Date: 2015/08/24

1. PREFACE:

This is the Acceptance Test Plan of the project titled, "TCP evaluation in Semi-Live streams" and is the first version v1.0.
The current version of this document aims at a detailed description of the unit test plans depending upon the user functional requirements in the Software Requirement Specification document of the on-going project.
The reader should be able to have an idea about the various tests performed to test the different units of the tool. The glossary and abbreviations section gives the vocabulary of the terms used in the document and other technical information. The unit test plan for each user functional requirement in SRS is covered in this document.

Release v1.3 on 2015-08-24

- In section 3 of the document, requirement string numbers in all the unit test plans are updated.
- In section 3 of the document, order of the tests is updated according to the design document.

Release v1.2 on 2015-06-01

- In section 3, all the Unit Test Plans have been updated.

Release v1.1 on 2015-05-20

- In section 3 of the document all the tables are provided with respective table numbers and captions.
- In section 3 of the document the tests are divided upon the user functional requirements mentioned in SRS document.
- In section 3 of the document the test string numbers are changed from the previous version of the document.
- In section 3 of the document new tests are added depending upon the user functional requirements.

Release v1.0 on 2015-05-14

- Initial release

2. GLOSSARY AND ABBREVATION:

SRS - Software Requirement Specification

RRD – Round Robin Database
The data in RRD is mainly a time series data like network bandwidth, CPU load etc., which is stored in circular buffers.

DPMI – Distributed Passive Measurement Infrastructure
This structure allows for an efficient use of passive monitoring equipment in order to supply researchers and network managers with up-to-date and relevant data. [1]

API – Application Programming Interface
A set of routines, protocols and tools for building software applications.

RTT – Round Trip Time
The total time taken for a packet to be sent from source to destination and the acknowledgement received from destination to source.

GUI – Graphical User Interface
Allows the user to interact with electronic devices through graphical icons and audio-visual indicators.

TCP – Transmission Control Protocol
Protocol used to exchange data between two communicating hosts through an established connection.

3. ACCEPTANCE TEST PLAN:

ATP_1: Login Authentication Test

Test	ATP_1
Purpose	Login Authentication Test
Requirements	REQ_USRF_1
Environment	The webpage must be available for the user for accessing the tool.
Operation	• The user first enters the username on the webpage of the tool. • Then the user enters the authentication password of the tool. • Next the user clicks on the login button
Expected Result	The user will be authenticated into the tool.
Result	pass
Comment	

Table 3.1: ATP_1: Login Authentication test

ATP_2: Metric Selection

Test	ATP_2
Purpose	Metric Selection
Requirements	REQ_USRF_7
Environment	The tool must be authenticated by the user.
Operation	• First, the user must log on to the webpage of the tool. • After logging in, the user will be directed to the dashboard of the tool. • The dashboard will contain different options of the tool. • The user will be able to select metric on the dashboard.
Expected Result	The user will be able to see the result regarding the selected metric of his choice on the dashboard.
Result	pass
Comment	

Table 3.2: ATP_2: Metric Selection

ATP_3: Consumer login

Test	ATP_3
Purpose	Consumer login
Requirements	REQ_USRF_8
Environment	The tool must be authenticated by the user.
Operation	• Firstly the user must log on to the webpage of the tool. • The user will click on the consumer login option on the dashboard. • The user will enter the details of the consumer on the webpage in the requested fields. • The user will click on the submit button. • The user will now go back to the previous page and click on the any metric required.
Expected Result	The user will be able to see the result of the selected metric.
Result	pass
Comment	

Table 3.3: ATP_3: Metric Selection

ATP_4: Generation of graphs

Test	ATP_4
Purpose	Generation of graphs
Requirements	REQ_USRF_4
Environment	The connection between RRD and frontend must be established properly.
Operation	After the user logs onto the tool and the tool runs successfully, the user will click on the statistics button on the dashboard.
Expected Result	The graph will be displayed on the GUI.
Result	pass
Comment	

Table 3.4: ATP_4: Generation of Graphs

ATP_5: Captured Packets are TCP packets

Test	ATP_5
Purpose	Captured Packets are TCP packets
Requirements	REQ_USRF_2
Environment	There should continuous flow of data stream.
Operation	• Open the terminal in the system on an UBUNTU based operating system. • In the terminal, the user must connect with the consumer of DPMI. • The system needs to connect to the consumer of the DPMI. This can be done by entering the username of the consumer in the network via, ***ssh <username>@<IP address>*** • Then the password for connecting to the consumer will be asked. The user must enter the password of the consumer. Password: ***password*** • Then the TCP packets are captured and stored in a ***cap*** file by using the following terminal command: ***sudo capdump -i eth2 -tcp -o <filename.cap> <stream>*** • The ***cap*** file is converted into a ***pcap*** file by using the following terminal command: ***cap2pcap -o <filename.pcap> <filename.cap>*** • Now the ***pcap*** file is run through the t-shark command using the following terminal command: ***tshark -r <filename.pcap> -O tcp***
Expected Result	The format of the captured packets is observed as TCP packets.
Result	pass
Comment	

Table 3.5: ATP_5: Captured packets are TCP packets

ATP_6: Analysis of TCP packets

Test	ATP_6
Purpose	Analysis of TCP packets
Requirements	REQ_USRF_3
Environment	There should be a continuous flow of data.
Operation	• The user, after logging in on the dashboard, must click on the run button. • Now the user will be selecting the required parameters on the dashboard like RTT or socket setup time or data rate per stream.
Expected Result	The requested parameters of TCP packets are displayed on WEB GUI
Result	pass

Comment	

Table 3.6: ATP_6: Analysis of TCP packets

ATP_7: Threshold Notifications as e-Mails

Test	ATP_7
Purpose	Threshold Notifications as e-mails
Requirements	REQ_USRF_9
Environment	The e-mail address is provided by the user.
Operation	• The user will define the threshold levels on the dashboard. • If the values exceed the threshold levels the traps will be sent. • Open the terminal and go into /var/log/syslog
Expected Result	Threshold notifications are seen as traps in the syslogs.
Result	pass
Comment	

Table 3.7: ATP_7: Threshold notifications as e-Mails

ATP_8: Threshold Notifications as traps

Test	ATP_8
Purpose	Threshold Notifications as traps
Requirements	REQ_USRF_9
Environment	User must define a threshold level for the selected input.
Operation	• The user will define the threshold levels on the dashboard. • If the Threshold levels are exceeded then trap is generated and sent to the user. • Type the following command to see the traps that u have logged onto ***logfile –Lf the pathfile where u want the traps to be logged into*** • The path file can be text file (.txt) or log file (.log)
Expected Result	Threshold notifications as traps are generated.
Result	
Comment	

Table 3.8: ATP_8: Threshold notifications as traps

ATP_9: Data accessibility through RESTful API

Test	ATP_9
Purpose	Data accessibility through RESTful API
Requirements	REQ_USRF_5, REQ_USRF_6
Environment	The link for the connection to the RESTful API
Operation	• A URL will be given to third party. • The third party will open the URL in the browser. • The third party will also have an option on opening the URL through terminal command: ***curl <URL>***
Expected Result	• The exported data will be seen on the webpage in JSON format (opened in browser). • The exported data will be displayed in the terminal. (opened in terminal)
Result	pass
Comment	

Table 3.9: ATP_9: Data accessibility through RESTful API

7. References:

[1] Patrik Arlos, Markus Fiedler, and Arne A. Nilsson. *A Distributed Passive Measurement Infrastructure*, In Passive and Active Measurement Workshop (PAM05), US, 2005.

[2] Ian Sommerville, *SOFTWARE ENGINEERING*, 9th ed. Pearson Publications, 2011