BLEKINGE INSTITUTE OF TECHNOLOGY

Developer Documentation

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 1.1
Publication Date: 2015/06/01

1. GLOSSARY AND ABBREVATION:

RTT - Round Trip Time
The total time taken for a data unit to reach destination from source and the acknowledgement from destination to reach the source.

SST - Socket Setup Time
The time taken for the three-way handshake to be executed.

ACK - Acknowledgement
The packet indicating the acknowledgement number of the packet received.

SEQ - Sequence Number
The number assigned to the data packet, which is being sent to destination.

MySQL - Structured Query Language
Used for storing and managing data in relational database management system.

PHP - Hypertext Preprocessor
It is a server side scripting language for creating dynamic Web pages.

HTML - Hypertext Markup Language
This protocol defines how messages are transmitted and formatted and interaction with the web pages.

RRD tool - Round Robin Database Tool
The RRD tool handles time series data which includes RRD data for graphical representation of the retrieved or stored data.

TCP – Transfer Control Protocol
A Standard way in which the communication between two systems on a network takes place.

2. UNDERSTANDING THE SOFTWARE DEVELOPMENT:

The steps during the software development can be understood by understanding the architectural design of the software. The tool is divided into 4 modules namely frontend, database, backend and RESTful API. These modules combined to give the whole software.

Frontend Module:

The Frontend module of the tool contains the developed WEBGUI of the tool. In this module, the user will be logging on to the tool through the frontend and running it. The process of analyzing the packets starts after the user enters the consumer details on the dashboard and clicks on the submit button on the dashboard of the tool. The user will be selecting the metric (RTT, Socket setup time and data rate per stream) of his choice on the frontend and the evaluated data will be displayed on the WEBGUI when requested. The graphical format of the data is displayed on the WEBGUI when user clicks on the "Statistics" button. The user will be defining the threshold levels on the WEBGUI and also provides the e-mail address where the generated notifications are sent. A URL is also created on the webpage which connects the tool with the RESTful API.

Database Module:

The Database module of the tool will be storing all of the data generated during the processing of the tool. The inputs (consumer details, threshold level, mail address etc.,) given on the WEBGUI are stored in the database. The data from the backend module is stored in the MySQL data. The data in MySQL database is updated every minute hence the old data is aggregated and the aggregated value is stored in the RRD and the raw data is flushed out. This happens as a continuous process. The RRD will save the aggregate values of the raw data with reference to time. The graphs of these aggregate values with respect to time are formed. Database will provide data to third party when requested through RESTful API.

Backend Module:

Backend module of the tool will be capturing and analyzing of the TCP packets is performed. The backend module will take consumer details from the database and captures TCP packets at the consumer of the DPMI. Then the captured packets are analyzed and required parameters (RTT, Socket setup time and data rate per stream) are evaluated and stored in the database.

RESTful API:

RESTful API module of the tool will be creating a interface to connect with the third party database. A URL is created which is given to the third party. The third party will open the URL where the data in JSON format will be observed. in the frontend module of the tool through which the user of the tool will be able to any third party. The RESTful API is used to export the data to third party. The data in the database is converted into JSON format and then displayed on a new webpage.

3. STEPS INVOLVED IN SOFTWARE DEVELOPMENT:

The retrieval of data was done by using *capdump* command from *libcap_utils*. T-shark is used to filter out the required packets and to extract the required fields.
These are then stored in a text file. The entire code is written in PERL. There were three parameters that were supposed to be calculated namely, RTT, SST and Data Rate per Stream. The matching of source-destination pair and ACK and SEQ numbers corroborating the mentioned formula used for packet matching.

$$ACK = SEQ + Length + 1$$

RTT

The timestamps of the matched SYN and SYN-ACK packets are observed and their difference will give RTT in seconds.

Socket Setup Time

Previously matched SYN-ACK packet (in RTT calculation) is matched with ACK packet. Then the timestamps difference between SYN and ACK packets gives socket setup time in seconds.

Data rate per Stream

The captured TCP packets are stored in *pcap* file. A terminal command *capinfos* is given to the file with the captured packets. Then the unique TCP streams with their details like Source IP: source port, destination IP: destination port, data transfer in bytes and its duration in seconds is observed. The data transfer in bytes divided by duration in seconds give data rate per stream.

The processed data is stored in the MySQL database where the information gets updated and is constantly displayed in the frontend. The RRD tool is used for plotting the graphs.

PHP and HTML scripting languages have been used for developing the Frontend of the tool.

4. LANGUAGES USED FOR THE DEVELOPMENT OF THE TOOL:

a) MySQL
b) PERL
c) PHP
d) HTML

5. SYSTEM COMPONENTS:

a. DPMI
b. Traffic monitoring via servers

6. EXTERNAL MODULES:

a. List::MoreUtils
b. Data::Dumper
c. DBI
d. Libcap_utils
e. DBD::MySQL
f. RRD::simple

g. Net::SNMP
h. Mail::Sender
i. Experimental
j. Net::SSH::perl
k. Net::SCP::Expect

7. PSEUDO CODE FOR RETRIEVING DATA AND CALCULATING RTT:

a. Capture commands
b. Packet filters - In (SYN packets), Out (SYN-ACK | ACK packets)
c. Extract required fields from packet data and store them in text files
d. Match In-Out packets for RTT and SST
e. Calculate RTT and SST
f. Calculate Data Rate per Stream
g. Save calculated metrics in the database
h. Flush packets after one hour from the database and save the aggregated data for one hour instead
i. Read the values from the database and the plot the graphs.
j. Send the data to third party through RESTful API by URL created.

8. BRIEF DESCRIPTION of RESTful API:

Application Program Interface (API) specifies how the software components must interact within the system.

RESTful API (Representational state transfer application programming interface) is used to connect the tool with any third party. A URL is created for the tool which is provided to the third party for accessing the data in the tool. By clicking on the URL the data in JSON format is displayed on the webpage.

URL: http://localhost/Project2/Web/restful.php/?metrics=rrt

URL: http://localhost/Project2/Web/restful1.php/?data=data_rate