Analyzing Transport Layer Protocols to Create an API to Satisfy Timed Response in Dynamic Requirements of Wireless Networks

Dhammika H. De Silva[#1], M.M. Ahamed Ishak[#2], G.C.M De Silva[#3], A.V.S. Suranjith[#4], H. M. H. S. Herath[#5]

Faculty of Computing Sri Lanka Institute of Information Technology

[1]dhammika.d@sliit.lk, [2]ishak.live@gmail.com, [3]chathurashan89@gmail.com,
[4]shashikasuranjith@gmail.com, [5]harindrah@gmail.com

***Keywords*— API, Transport Layer Protocols**

In modern systems it is very vital to have an efficient data transmission mechanism. This becomes more critical factor when it is involved with a system like a way finding application for vision impaired people or a medical system that notifies the doctor's smartphone when the patient's heart rate goes down. The size of the data that is used frequently by the mentioned systems is relatively small. Additionally it might take more time for connection establishment despite of the requirement of urgency. Thus this kind of data transmission is not ideal for such systems. Therefore this diversity of data transmission makes a special category of systems that needs a reliable, efficient data transfer mechanism since requirements of those systems such as low latency, reliability are not being fully satisfied by the current Transport Layer Protocol (TLP) selection mechanism.

As a solution this research investigates a mechanism to select the optimal Transport Layer Protocol by analysing the behaviour data as shown in the table I. The output of this research is an Application Programming Interface (API) that is capable of analysing the characteristics of the incoming data and redirecting the data to the optimal transport layer protocol. In order to make the decision to select the best protocol, there is a data source that has been collected for various data types for different performance metrics.

This research mainly focuses on Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Stream Control Transmission Protocol (SCTP) and Datagram Congestion Control Protocol (DCCP). The research is done based on the requirement of way finding application uses for vision impaired people. The characteristic of data types in way finding system is illustrated in the Table I.

To analyse the performance of four protocols, TCP, UDP, SCTP and DCCP were investigated using the Network Simulator 2 (NS2) tool. As the performance matrices; Packet Loss, Average End-to-End Delay and Average Throughput were used for these simulations. For the simulation tests, a wireless mesh network of 2, 3, 5 and 10 nodes were used. The link speed between each node was set to 10Mb/s. Packet size was also changed in order to test the performance for different packet sizes. The simulation for each scenario was run for 100seconds. The data collected in these simulations is used by the API when selecting the optimal protocol.

TABLE I
BEHAVIOUR OF DATA TRANSFER

Data	Size	Frequency	Reliability
GPS - Processed	Very Small	High	No
Wi-Fi Processed	Very Small	High	No
Obstacle Fixed	Medium	Triggered	Yes
Obstacle –non stationary	Small	Urgent	Yes
Map Data	Large	Triggered	Yes
User Location	Very Small	High	No
Sensor Data Processed	Very Small	High	No
Instruction	Small	Urgent	Yes

As a prototype the research team has implemented a protocol selection model using the data source. For this implementation socket programming and shell programming has been used. In order to represent the four protocols, four socket programs for TCP, UDP, SCTP and DCCP was used. The collected data is stored in a XML file and a shell script contains the business logic. For this model pre defined static data has been used. The data was passed to the shell script and it analysed the characteristics of that static data. Then the shell script retrieved the necessary data from the data source and after evaluation, the incoming data is redirected to the relevant socket program.

Based on this prototype, the API was implemented and as the technology Python scripting language was used. The main reason to select python is that the final solution can be implemented as a library thus makes it reusable. Therefore other systems can use this library for data transmission.

As the future work this research will focus on testing the python library in a real world environment and getting the values for throughput, end-to end delay, packet loss and compare them with simulation values. Based on the comparisons the library should be optimized accordingly to get better performance.