A Project Report

on

**“Web based Real-Time ballistic and cruise missile Data Analysis and Visualization System”**

Submitted by

**Anurag Yadav**

cs2016021021

A close up of a sign

Description automatically generated

**Department of Computer Science & Engineering**

**Madan Mohan Malaviya University of Technology, Gorakhpur**

**(U.P.) India**

Under the Guidance of

**Mr. SANDIPAN SARKAR**

Scientist-E, Central Data Processing



**INTEGRATED TEST RANGE, CHANDIPUR**

**DEFENCE REASEARCH AND DEVELOPMENT ORGANISATION**

**INTEGRATED TEST RANGE, CHANDIPUR**

**DEFENCE REASEARCH AND DEVELOPMENT ORGANISATION**



**CERTIFICATE**

This is to certify that Project report on **“Web based Real-Time Data Analysis and Visualization System”** In Partial Fulfillment of Requirement of Industrial Training at INTEGRATED TEST RANGE, CHANDIPUR, ODISHA prepared by **ANURAG YADAV**, undergraduate student of the Department of Computer Science and Engineering, Madan Mohan Malaviya University of Technology, Gorakhpur, during the session 15th May-15th July 2019, is done under my guidance. This project has been successfully delivered and examined. It was found fit and approved for submission.

**Mr. SANDIPAN SARKAR**

Scientist-E, CDP  
ITR, Chandipur, DRDO

**ACKNOWLEDGEMENT**

I take this opportunity with much pleasure to thank all the people who have helped me through the course of my great journey towards producing this project report. I sincerely thank my project guide Mr. Sandipan Sarkar, Scientist “E”, for his guidance, help and motivation. Apart from the subject of my project work, I learnt a lot from him, which will be very useful in the later stages of my life.

I would also like to thank the following people

Dr. B K Das, Scientist “H” & Outstanding Scientist, Director, ITR

Mr. H K Ratha, Scientist “G”, Additional Director, ITR

Mr. R Barua, Scientist “F”, GD (CDP)

Mr. C R Ojha, Scientist “F”, GD (HR, PL & RM)

Mr. R K Patra, Scientist “F”, GD (PMA)

For providing me, this great opportunity, of being a trainee in this esteemed organisation.

I would like to acknowledge the support and encouragement of my friends. Without their help this training journey would not have been a reality.

Finally, this project would not have been possible without the confidence, endurance and support of my family. They have always been a source of inspiration and encouragement. I thank my parents, whose love and support has brought me this far.

**ANURAG YADAV**

**cs2016021021  
Department of Computer Science and Engineering  
Madan Mohan Malaviya University of Technology, Gorakhpur U.P**.

**Contents**

1. **About the Organization**
2. DRDO
3. ITR
4. CDP
5. **Abstract**
6. **System Decoupled Architecture**
7. UDP Server
8. Middle UDP to TCP Server
9. End Client Server
10. **Sending data through UDP Server**
11. **Receiving and Re-transmission of data by Middle Server**
12. **Receiving data to the End Client Server**
13. **Visualization of data at End Client Server**
14. Initialization of Chart.js Object
15. Visualizationof data
    1. Visualization of nominal data
    2. Visualization of real-time data
16. **Conclusion**
17. **Bibliography**
18. **About the Organization**
19. **DRDO**

|  |
| --- |
| **Defense Research & Development Organization (DRDO)** works under Department of Defense Research and Development of Ministry of Defense. DRDO dedicatedly working towards enhancing self-reliance in Defense Systems and undertakes design & development leading to production of world class weapon systems and equipment in accordance with the expressed needs and the qualitative requirements laid down by the three services.  DRDO is working in various areas of military technology which include aeronautics, armaments, combat vehicles, electronics, instrumentation engineering systems, missiles, materials, naval systems, advanced computing, simulation and life sciences. DRDO while striving to meet the Cutting-edge weapons technology requirements provides ample spinoff benefits to the society at large thereby contributing to the nation building.  **Vision**   Make India prosperous by establishing world-class science and technology base and provide our Defense Services decisive edge by equipping them with internationally competitive systems and solutions.   **Mission**   * Design develop and lead to production state-of-the-art sensors, weapon systems, platforms and allied equipment for our Defense Services. * Provide technological solutions to the Defense Services to optimize combat effectiveness and to promote well-being of the troops. * Develop infrastructure and committed quality manpower and build strong technology base. |
| **Core Competence**   * Dept. of Defense Research and Development (R&D) is working for indigenous development of weapons, sensors & platforms required by the three wings of the Armed Forces. To fulfill this mandate, Dept. of Defense Research and Development (R&D), is closely working with academic institutions, Research and Development (R&D) Centers and production agencies of Science and Technology (S&T) Ministries/Depts. in Public & Civil Sector including Defense Public Sector Undertakings & Ordnance Factories. |

DRDO has a dynamic training and development policy which is executed through the Continuing Educational Programs (CEP) for all cadre personnel viz DRDS, DRTC, Admin & Allied. At the entry level in DRDS, the newly recruited scientists undergo a 16 weeks Induction Course at Institute of Armament Technology (IAT), Pune. Under the Research and Training (R&T) scheme the scientists are sponsored for ME/M Tech programs at IITs/IISc and reputed universities. The fees are also reimbursed by the DRDO where scientists undergo PhD program. In addition to this, the Organization through its two premier Institutes namely Institute of Technology Management (ITM) and Institute of Armament Technology (IAT) deemed university offer courses for scientists and Armed Forces in the area of Technology Management, R&D Management and Armament. Recently, a training center at Jodhpur has been established to meet the training needs for Admin & Allied cadre. In order to attract the futuristic talent, DRDO has Junior Research Fellow (JRF), Senior Research Fellow (SRF) and Research Associate (RA) schemes for young & dynamic personnel & interested in Defense Research and Development.

1. **ITR**

Integrated Test Range (ITR) Chandipur provide safe and reliable launch facilities as well as over the Project specified data for the performance evaluation of rocket, missiles, flight, vehicles and airborne weapon systems.

Starting in 1982 as a Project under Integrated Guided Missile Development program (IGMDP), it has been gradated to perfection over the years and has reached the status of world class Test Range Ballistic missile; Agni AE 01 on 22 May, 1989 from this Test Range has put ITR a prominent place in the world Map. Since then, ITR has tested more than 1000 missions of national importance.

For this purpose, several instruction systems such as Central Computer Data & Processing, Radar System, Telemetry System, Electro Optics Tracking system, Campus Area Network, Communication System, Central Timing & Tele command System, Power System, Target System, Flight Safety System, Environmental Safety, Workshop.

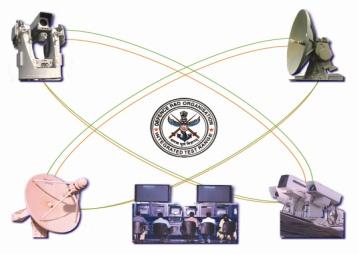
**Vision**

To be trusted integrated Test facility, delivering reliable and quality service to all customers.

**Mission**

* To achieve excellence in service quality and reliability.
* To attain global best practices and become a world class Test Range.
* To be a technology driven and efficient organization.
* To encourage ideas, talent and value systems.
* To work with vigor, dedication and innovation with total customer satisfaction as the ultimate goal
* To promote work culture that fosters individual Growth, Team spirit and creativity to overcome challenges and attain goals.
* To contribute towards community development and nation Building.

1. **CDP**



**Role of Computer Data & Processing division**

* Designed the data analysis look angle computer.
* Tracking instruments track coverage analysis.
* Range network monitoring and link checks.
* Multi sensor data processing in real time.
* Range instrument calibration and validation bias correction.
* Designed mission data CDM generation and disruption to tracking instrument.
* CDM data transmission to tracking instruments in real time including demanded CDM data.
* Target selection in multi target scenario.
* Data logging in real-time.
* Displaying mission specific information like trajectory data, telemetry health data etc.
* Permission simulation of mission scenario using HILS and design data.
* PFA (Post Flight Analysis)
* Presentation to PFA committee and processed data delivery to projects.

1. **Abstract**

Realtime data analysis and visualization involves the analysis of data that is being received through the UDP socket and then visualizing that analyzed data in graphical format for better understanding.

This system architecture has three servers, one end server for sending data through UDP socket, one middle server for receiving data through UDP socket and retransmit again that data through TCP socket for End Client Server.

Requirement of the middle UDP to TCP server is because we want to system data visualization to be done at fast rate that can be achieve only if we receive data through fast UDP socket. But Web Browser data transmission is done by the HTTP and TCP socket is implemented on top of HTTP, that why browsers support only TCP socket not the UDP socket. That’s why we use a middle UDP to TCP Server.

This system leverages the functionality of Chart.js JavaScript library and Node.js’s backend Express server, all the server has implemented on Express server and End Client server front-end also implemented on Express server.

Visualization is done first for nominal data and then for real-time data for better accuracy check. Also, real-time data get visualized at every 100milisecond to the DOM, that overcome the limitation of HTML5 that support at least 1second to get update the DOM.

1. **System Decoupled Architecture**

System involves three servers, one UDP server, one middle server and one end client server.

1. **UDP Server**

UDP server is the main server which is responsible for sending data through the UDP socket. Further that data is consumed by the middle TCP server. Here UDP server is used to send data through socket because one of the attractive features of UDP is that since it does not need to retransmit lost packets nor does it do any connection setup, sending data incurs less delay.

This lower delay makes UDP an appealing choice for delay-sensitive applications like audio and video.

Second, multicast applications are built on top of UDP since they have to do point to multipoint. Using TCP for multicast applications would be hard since now the sender would have to keep track of retransmissions/sending for multiple receivers.

Is this system UDP server is used to send data at higher rate to do the visualization at more high speed for better understanding to users.

1. **TCP Server**

A middle TCP server is used in this system that consumes the data send by the UDP server.

The requirement of the TCP server in this system is that, this is web-based system and web does not support the UDP socket in browser document. This is the main reason to choose a middle TCP server.

The main purpose of this TCP server is that is consume the data from UDP server whenever it gets data and it retransmit the data to End client server through the TCP socket, that is it works like UDP to TCP socket convertor.

1. **End Client Server**

End client server is the third server that is used in this system. End client server is used to connect the end users for the visualization of data. End client server involves the both front-end and back-end. Front-end part is used for visualization of data and back-end part used to receive data through the middle TCP server that is used to convert UDP socket to TCP socket.

End client server receives the data though TCP socket that is connected to End client server and send that data to front-end part where data get visualized.

1. **Sending data through UDP server**

Sending data through UDP server is done by UDP socket. UDP socket is used to send the data.

In the UDP server there are below socket event to send the response –

* 1. **connection**

Connection event is fired when a new TCP socket client get connected.

io.on(“connection”, (socket)=>{

})

* 1. **disconnect**

Disconnect event is fired when an any connected client disconnected.

io.on(“disconnect”, (socket)=>{

})

* 1. **getNominalData**

This user defined event is fired by the middle TCP server that has also implemented UDP and TCP server both. Middle server will fire this event to send nominal data that will be further sent to end-client-server.

io.on(“getNominalData”, (socket)=>{

})

* 1. **getRealTimeData**

This user defined event is fired by the middle TCP server that has also implemented UDP and TCP server both. Middle server will fire this event to send real-time data that will be further sent to end-client-server.

io.on(“getRealTimeData”, (socket)=>{

})

* 1. **error**

This event get fired when any error occurred in the socket.

io.on(“error”, (socket)=>{

})

1. **Receiving and Sending data by Middle server**

Receiving data through Middle server is done by UDP socket. UDP socket is used to receive the data from UDP server through UDP socket and then send it to end client server through TCP socket.

In the TCP server there are below socket event to send and receive events –

1. **connection**

Connection event is fired when a new end client socket gets connected.

clientSocket.on(“connection”, (socket)=>{

io.sockets.emit(“nominalData”, nominalData)

})

Whenever an end client socket gets connected middle server emits the nominal data that has been received by the main UDP server.

1. **disconnect**

Disconnect event is fired when any connected end client disconnected.

clientSocket.on(“disconnect”, (socket)=>{

})

1. **serverNominalData**

This is used defined event that is received by the middle server. This event is used to get nominal data from UDP server.

clientSocket.on(“serverNominalData”, (socket)=>{

})

1. **serverRealTimeData**

This is user defined event that is received by the middle server. This event is used to get real-time data from UDP server.

clientSocket.on(“serverRealTimeData”, (socket)=>{

})

1. **error**

This event gets fired when any error occurred in the socket.

clientSocket.on(“error”, (socket)=>{

})

1. **Receiving data to the End Client server**

Receiving data at the end client server is done by the TCP socket that receives data from the middle TCP server all the nominal and real-time data will be received to this end client server. After data get received data will be visualized to the frontend of this end client server.

Below are the events that is implemented at the end client server to receive data from middle server-

1. **connection**

Connection event is fired when a new end client socket gets connected.

clientSocket.on(“connection”, (socket)=>{

})

Whenever an end client socket gets connected middle this event gets fired.

1. **disconnect**

Disconnect event is fired when any connected end client disconnected.

clientSocket.on(“disconnect”, (socket)=>{

})

1. **nominalData**

This is user defined event that is received by the end client server. This event is used to get nominal data from middle TCP server.

clientSocket.on(“nominalData”, (socket)=>{

})

1. **realTimeData**

This is user defined event that is received by the end client server. This event is used to get real-time data from middle TCP server.

clientSocket.on(“realTimeData”, (socket)=>{

})

1. **error**

This event gets fired when any error occurred in the socket.

clientSocket.on(“error”, (socket)=>{

})

1. **Visualization of data at End Client Server**

Visualization of data is done at the end client server using lightweight chart.js library. Visualization of chart is done by first plotting the nominal data and then real-time data.

1. **Initialization of Chart.js Object**

Initialization of chart.js object is done by making config object and then passing to the chart.js constructor.

**Config Object-**

chartContext = document.getElementById(“chart”).getContext(“2d”)

chartOptions = {

layout:{},

responsive: true,

scales: {

xAxes:[],

yAxes:[]

}

}

chartConfig = {

type: “scatter”,

data: {

labels: Array<string>,

datasets: Array<any>,

options: chartOptions

}

}

chartObject = new Chart.Scatter(chartContext, chartConfig)

1. **Visualization of Data**
2. **Visualization of nominal Data**

Visualization of nominal Data is done by passing all the data points object of the nominal dataset to the chart.js object datasets.

Passing the point data object of the nominal dataset is done repeatedly pushing point data object one by one to the nominal dataset of the chart.js object.

while(dataPoint){

firstConfig.data.datasets[1].data.push({x,y})

}

Then calling update to update the chart.js object and get render chart again for only the update data point not for the whole chart.js object to the DOM in the browser.

firstChart.update()

1. **Visualization of real-time Data**

Visualization of the real-time data is done by passing all the data points object to the real-time dataset to the chart.js object datasets.

Passing the data point object is done at real-time whenever it is received by the end client server by the middle server through the TCP socket at any specified time quantum.

while(dataPoint){

firstConfig.data.datasets[0].data.push({x,y})

}

Then calling update to update the chart.js object and get render chart again for only the update data point not for the whole chart.js object to the DOM in the browser.

secondChart.update()

1. **Conclusion**

This system has implemented three servers one for sending the data through UDP socket, one middle server that have defined two sockets one is UDP and another is TCP socket. Middle server is being used as a UDP socket to TCP socket and last server is end client sever to visualize the data at end users front-end. UDP socket is used to send data at fast rate to visualize at end-client server more accurately and at better performance level.

Normal HTML5 takes 1second to update DOM object, but this system overcome that limitation using chart.js and JavaScript and plotting data points smoothly in DOM, in window object smoothly at every 100ms.

Also, this system has decoupled the architecture in three servers as mentioned above, UDP server, middle UDP to TCP Server and End Client Server for scalability and better user interface.

1. **Bibliography**

* <https://www.chartjs.org/>
* <https://stackoverflow.com/>
* <https://angular.io/>
* <https://nodejs.org/>
* https://github.com/



**INTEGRATED TEST RANGE, CHANDIPUR, ODISHA**

**DEFENCE RESEARCH AND DEVELOPMENT ORGANISATION**

**BONAFIDE CERTIFICATE**

This is to certify that this project report entitled **“Web based Real-Time Data Analysis and Visualization System”** submitted to “INTEGRATED TEST RANGE, CHANDIPUR, ODISHA, DEFENCE RESEARCH & DEVELOPMENT ORGANISATION”, is a bona fide record of work done by **Anurag Yadav**  pursuing B. Tech in Department of Computer Science and Engineering at Madan Mohan Malaviya University of Technology Gorakhpur, U.P. under the supervision of **Mr. Sandipan Sarkar,** Scientist “E”, CENTRAL DATA PROCESSING division from **15th May to 15rd July, 2019.** During this period, he worked under my guidance and supervision. He has successfully completed the above project assigned by me. During his training he was sincere and showed keen interest during his project, thus completing it in stipulated time.

**SANDIPAN SARKAR**

Scientist-E

(Central Data Processing) ITR