MS-Bullet Train System Control Protocol

Murtaza Ahmad Shahid

Abstract– **In this era of technology we can control many digital devices by some specific network protocol. Like digital robots, army equipments, office or home machinery, news broadcasting, digital locks and alarms, aerospace etc. We have developed a protocol for MS-Bullet Train System control, which allows to interact different dimensions and functionality of bullet train, like start engine, track control, speed changer etc. This protocol can set up connection between MS-Bullet Train and its control station.**

Keywords– **MS-Bullet Train, TCP/IP, Transport Layer, Application layer and Physical Layer**

I. INTRODUCTION

Bullet Trains are constructed to take passenger from one place to another on the earth. Bullet Train is a system of high-speed railway lines in railway field. Highly trained drivers are selected to control the bullet trains. The average speed of a bullet train is 262km/h or 164 mph which is the fastest scheduled train service in the world. As drivers inside the trains control it manually. Sometimes due to lack of information about tracks and trains timing prospective given to drivers cause accident, but at control stations we can see the whole map of railway lines, timing of trains, trains information running on different or track info in digital form on board, so if we control trains from control stations then it will helpful to avoid accident. So, we can control trains thought this protocol from control stations by passing commands that instruct the train computer which task you should perform now, and when.

Comparison of bullet train with other vehicles

Most high speed Bullet trains use electricity passing from overhead wires. Today's electricity batteries are not in such size to give full power to train and still fit on the train. Generators use diesel engines eject harmful material in the environment, so they are never consider as environment friendly, also the weight of engines and capacity to store the diesel fuel, aside with fire safety, would cause to generate other problems. Another option is using a shoe on the footer of train to take electricity from a third rail. Due to use of electricity to run its engine the CO_2 Emission rate of bullet train with respect to other Transport is least.

II. LITERATURE REVIEW

In Bullet train NIC burst behavior [1] depends upon the Application, Transport, OS, NIC, Disk drive, which control the traffic of packet of payload on transport layer. Packet Traffic control is an important area of networking, because this have effect on packet delay time or some time packet lose. We can control packet lose and use network protocol to control the train.

This era belongs to technology of controlling your daily interacting things remotely. You can easily control the functionality of the office and resident machinery by just telling them what to do. You can simply control the opening and closing of gate of house. Different network protocols are used to do these operations [2].

Aerospace can be controlled through commands. But there is a problem with fast moving vehicles and mostly with flying object is unknown disturbance and uncertainty with respect to their trajectory path. A trajectory linearization control (TLC) based on radial basis function neural network (RBFNN) [3]. Radial basis function neural network application on flight control system is presented. The effect of disturbances, instability and uncertainties is reduced by RBFNN. Stability analysis Theorems are useful to stable the flying or fast moving vehicles like bullet trains to avoid the insatiability.

Different Digital locks and alarms can control through remote. Mostly in security offices use such types of locks and alarms which can be turn on or turn off remotely. (Z-wave) wireless connection mostly used. For this type of control use certain protocols [4].

Now a days, to full fill the needs of people industries use fast methods to achieve their goals. In automobile companies, manufacturing of machinery can be done by robots. Light weight robots are very useful to perform different task in efficient manners. Robots are programmed to do certain task, these robots can control by specific protocol, through protocol different commands are sent to receiver side of robots, and robots perform that specific task [5].

Army radars and their related equipment mostly control remotely now. Such as under the water network communication with submarines, automatic cannons etc. can be control remotely by protocol [6]. In this type of protocols commands sends in encoded form and some time in spectrum form. There is a sender and a receiver side.

Transfer rate of packet control for high speed network is an important field. Using Virtual Clock algorithm, the transfer rate of packet switching can be control. For this Simulation Model can be used. This is very helpful for all protocol used

Murtaza Ahmad Shahid is a Research Scholar (Computer Network Theory and Practical), at Department of Computer Science and Engineering, University of Engineering & Technology, Lahore, Pakistan, (Email: murtaza.ahmad.csstd@gmail.com)

packet switching network [7].

III. BULLET TRAIN CONTROL PROTOCOL

We have developed transport layer protocol that takes commands in text form on application layer where command convert into respective Sigma of payload then we have attached sigma separator of application layer and put these all values in the payload of the packet of the transport layer where a socket number with respect to functional command type by the controller put in the head of packet, also sigma separator of transport layer attach with it and then send it to the network layer. After passing through all layers the packet send to last layer that is physical layer which convert packet into the voltage pulses.

These pulses are basically signals that receive by the train computer. Where invert system is installed that convert these pulses into packets. And then passing through layers this receiving packet moves up to receiving side transport layer. This take the socket number from packet and Sigma of payload and then computer of bullet train act according to the command. We hope you might get an idea of how this works; here is the pseudo code of the transport layer of controller side for communication network between control station and Bullet train. Before pseudo code we have some defined Directories (replica and socket number):

```
 /*sigma for payload={m,s}
   Sigma Description:
   m m m m =START
   m m m s =Door Close
   m m s m =RUN
   m m s s =Increase Speed
   m s m m =Decrease Speed
   m s m s =Stop
   m s s m =door_open
   m s s s =auto gear
   s m m m =change track
   s m m s
   s m s m
   s m s s
   s s m m
   s s m s
   s s s m
   s s s s
    SIGMA OF APPLEYAR=3000;
  SiGMA OF TRANSPOT=3001;

   int socket_no={2001,.............,2009};*/
```

Explanation of pseudo code:

We have an object of a class application “obj1”which is able to contact with “get_command” function in application class that have head pointer name as “FIRST” of the link list name as “Train” that I have developed to takes the commands from controller convert it into respective Sigma of Payload and put it into the nodes of link list “Train” and attach Sigma separator of application layer and transfer it to the transport layer.

Now we have an object of a class transport “obj2” which is able to contact with “Attach” function in transport class which takes the coming nodes of link list “Train” from the Application layer class.

The “*ptr” is the local pointer of the link list “Train” in Attach function of the transport class. This assign the all nodes of Link list “Train” of head “First”. This is taken for managing the nodes of Link list “Train”.

In the “Attach” function I have add two nodes at head place of link list “Train” first one node contain sigma separator of transport layer and second one contains the socket number with respect to the functional command. Add all nodes coming from above application layer. Then also attach a node at the end of link list “Train” that contains the Sigma of transport layer. This Sigma separator is taken so that we can easily understand that a new packet is started. Duplicate of this package is store in “replica” that is object of “Train” link list. And then this package is send to next layers.

pseudo code:

```
  Attach()
     {
       Train * ptr=FIRST;
       FIRST=NULL;
       FIRST=new Train;
       FIRST->no=sockets[number];
       FIRST->next=ptr;
       ptr=FIRST;
       FIRST=NULL;
       FIRST=new Train;
       FIRST->no=SiGMA_FOR_TRANSPOT;
       FIRST->next=ptr;

       Train * add=FIRST;
                   while(add>next!=NULL)
                   {
                     add=add->next;
                   }


                 add->next=new Train;
                 add=add->next;
                 add->no=SiGMA_FOR_TRANSPOT;
                 add->next=NULL;
               replica=FIRST;
```

Now, we are going to explain the receiving side code. Here we have “Detach” function that is detaching the first two nodes. Second one contains the socket number we are storing it into the “number” integer variable. Here we have a string “compare” that stores the bytes of Sigma of payload of that command given by the controller. After that we detach last node of packet or link list “Train” that contain Sigma of transport layer.

After that we are comparing the “compare” sting with the defined replica of command and bullet train perform action according to command, and showing replica and socket number at application layer.

pseudo code:

```
void Detach()
{
   number=FIRST->next->no;
  cout<<"DATA is "<<FIRST->next->nex>no<<endl;
  FIRST=FIRST->next->next->next;
  Train * ptr=FIRST;

  for(int data=0;data<4;data++)
  {
    compare+=ptr->value;
    cout<<"DATA is "<<ptr->value<<endl;
    ptr=ptr->next;
  }
  cout<<"DATA is "<<ptr->no<<endl;
  delete ptr;
if(compare=="mmmm")
      {
        cout<<"START THE TRAIN ENGIN"<<endl;
      }
  else if(compare=="mmms")
    {
      cout<<"Close The TRAIN's Doors"<<endl;
    }
  else if(compare=="mmsm")
    {
        cout<<"RUN THE TRAIN "<<endl;
    }
  else if(compare=="mmss")
    {
        cout<<"Increase THE TRAIN Speed"<<endl;
    }
  else if(compare=="msmm")
    {
          cout<<"Decrease THE TRAIN Speed"<<endl;
    }
  else if(compare=="msms")
    {
      cout<<"Stop THE TRAIN "<<endl;
    }
  else if(compare=="mssm")
    {
          cout<<"Open THE TRAIN's Door"<<endl;
    }
  else if(compare=="msss")
    {
          cout<<"Gear Of Train Has Changed!"<<endl;
    }
  else if(compare=="smmm")
    {
          cout<<"Track Of Train Has Changed!"<<endl;
      }
}
```

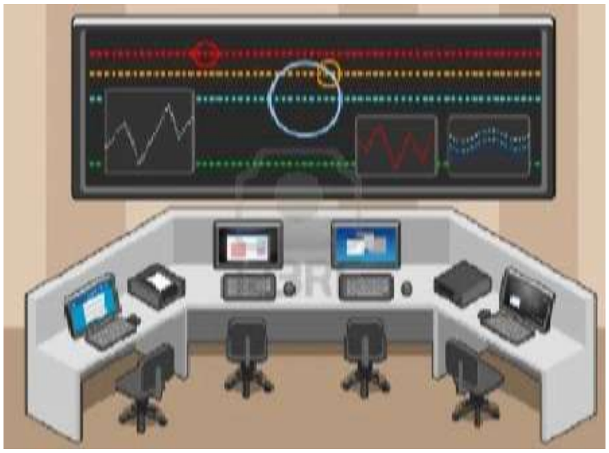

Fig. 1: Control Center

Fig. 2: Bullet train computer

IV. HOW IT WORKS

The explained protocol works as when the control center is connected to the Bullet train computer. Bullet train computer and control center can connect through (Ethernet connection) wires or by WiFi (wireless connection) spread on the track. When such link is establish between control center (shown in Fig. 1) and the Bullet train computer (shown in Fig. 2) then we can start the communication of command between the bullet train and control center. This whole system can be seen in (Fig. 3).

Let consider we are going to enter a command “start”. After entering the “start” the engine of bullet train start and ready to run on the track. After entering “door close” the all doors of train shut. So that no one can enter inside or go outside. After entering “run” the train runs on the track. Similarly other commands perform specific task.

When we enter a specific command then the replica in sigma of payload of that command sends to the transport layer, from Application layer. At Application layer as we have told earlier sigma separator of application layer attaches at place of head and tail.

This Sigma of separator is attached so that we can easily determine that the packet is belonging from which layer and all nodes are received from this packet or not.

After that transport layer attaches a socket number with respect to the command entered by the controller to this packet and two Sigma separators one at the head place and one at the tail place. Then this packet is passed to next layers which perform their functionality and then send packet to next layer.

After passing through all layers the packet reaches to physical layer that convert the packet payload into binary form which are easy to send in voltage pulses on the wire or in air with a specific frequency, this packet is sent in bit form or in for of pluses of voltage. TCP/IP reference model is used for communication of these commands on network.

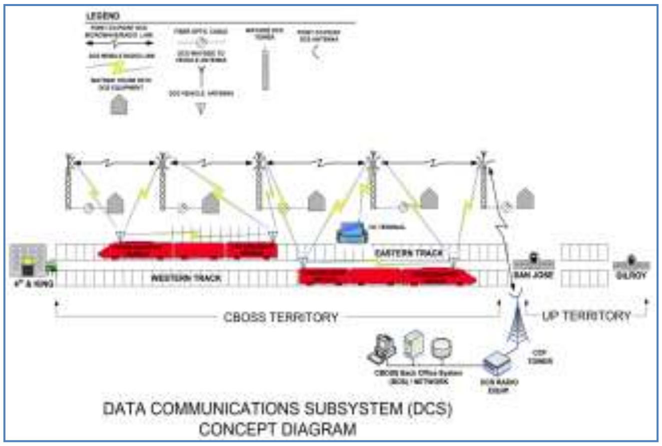

Fig. 3: DCS

Sending of command from control center to bullet train is done by the controller computer and these packets mostly send on wire network which is covered the whole track of bullet train.

The Sender side can see the location of bullet train on control board. We can track the location of bullet train by GPS tracking or by communication network that is connected to office system as shown in Fig. 4.

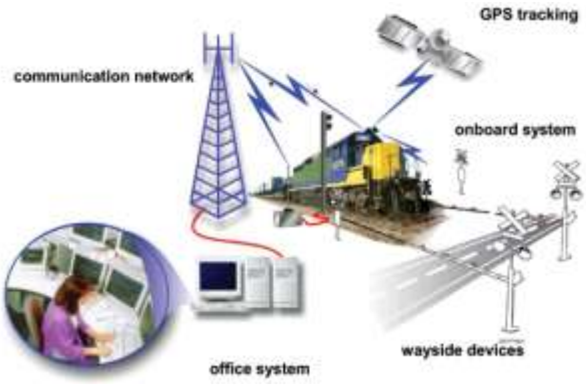

Fig. 4: Tracking

When computer of bullet train invoke with the packet reached up to the train through wires. Computer converts the binary form of packet into respective sigma of payload.

Then that packet sends to the other layers and after passing through all layers the packet reaches to receiving side transport layer.

Transport layer detach the first two nodes and last node of packet, first and last node contain the sigma separator of transport layer. The second one node contains the socket number of specific function.

The sigma of payload from that packet store into string and that string compare with the series of sigma of payload. After comparison, if results of comparison are positive. Then the specific action performs on the train.

The result or replica of command and relative socket number shows on the screen of bullet train computer through the application layer, and this all commands during the journey of train shown on the computer of bullet train.

This sigma of payload with respective to specific command can store in the Hard Disk of bullet train computer. So that in any case of misshape we can see the controlling problems and decision of making command by the controller, this hard disk performs functionality as black box in airplanes.

V. WHAT CAN BE DONE

In this given protocol, we have designed the application layer and Transport layer. But working on transport layer is major goal. We have entered some basic command to control the train. But we can add more functionality like: Lights on or off, electricity power control which is passing through wire that is used by engine of Train, Feed the destination details in train computer through this protocol.

We can convert this train as an automatic robot that can get all commands at a time and then perform all work. There is no need to control at each and every station or movement.

It means that at starting time feed all information in bullet train brain computer that at which station it will stop and what is the opening time of door or closing.

We can also add auto collision detection feature in the trains. This will be helpful to reduce the accident rate of Bullet trains (Fig. 5).

In this feature we can add such functionality that if there is another train on the same track then at a reasonable distance the other train brain computer get a signal and vice versa, and the speed of both train decrease up to stop the trains.

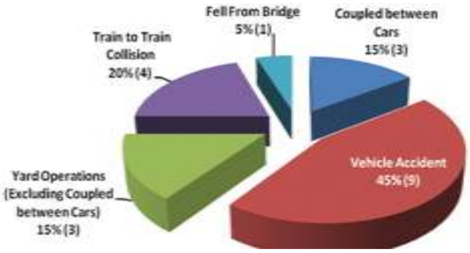

Fig. 5: Chart of accident rate of vehicles

A signal sends to control center. That tells the Bullet train's brain computer what to do with this protocol that we have developed. Like: change track.

We can also add speed and stability feature. The functionality of this feature is that if there is some possibility of instability of train then the train's brain computer sent a signal packet to control center to decrease the speed of train. Then by using this above protocol controller can decrease the speed of train.

VI. CONCLUSION

In above protocol, transport layer is an important one. As we are controlling bullet train by protocol that sends the packet from control center to bullet train computer that contain important replica of command typed by the controller, this packet gets sigma of payload and socket number related to command from transport layer, and perform respective action on receiving side transport layer. Bullet train is fast travelling transport so, this protocol can help to make ease for controllers, and enhance the quality of service of bullet trains. Controller simply enters commands to control the speed, track, door functionality, gear control of bullet train and the train performs given task.

REFERENCES

[1]. Kapoor, Rishi, Alex C. Snoeren, Geoffrey M. Voelker, and George Porter. "Bullet Trains: A study of NIC burst behavior at microsecond timescales." In *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, pp. 133-138. ACM, 2013.

[2]. Dixon, Colin, Ratul Mahajan, Sharad Agarwal, A. Brush, Bongshin Lee, Stefan Saroiu, and Victor Bahl. "An operating system for the home." In *Proc. NSDI*. 2012.

[3]. Yali, Xue, and Jiang Changsheng. "Trajectory linearization control of an aerospace vehicle based on RBF neural network." *Journal of Systems Engineering and Electronics* 19, no. 4 (2008): 799-805.

[4]. Fouladi, Behrang, and Sahand Ghanoun. "Security Evaluation of the Z-Wave Wireless Protocol." *Black hat USA* 24 (2013).

[5]. Albu-Schäffer, Alin, Sami Haddadin, Ch Ott, Andreas Stemmer, Thomas Wimböck, and Gerd Hirzinger. "The DLR lightweight robot: design and control concepts for robots in human environments." *Industrial Robot: An International Journal* 34, no. 5 (2007): 376-385.

[6]. Heidemann, John, Milica Stojanovic, and Michele Zorzi. "Underwater sensor networks: applications, advances and challenges." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 370, no. 1958 (2012): 158-175.

[7]. Zhang, Lixia, "Virtual clock: A new traffic control algorithm for packet switching networks." In *ACM SIGCOMM Computer Communication Review*, vol. 20, no. 4, pp. 19-29. ACM, 1990.