Centralized Traffic Controlling System for Sri Lanka Railways

A.L.A.T.D. Ambegoda, W.T.S. De Silva, K.T. Hemachandra, T.N. Samarasinghe, and A.T.L.K. Samarasinghe
Department of Electronics and Telecommunications Engineering
University of Moratuwa, Sri Lanka
tsameera1@gmail.com

***Abstract*— This paper discusses the design of a centralized control and monitoring system for railway traffic that can be implemented on top of the existing control logic (interlocking system) of Sri Lanka Railways. The overall solution comprises of a central server system and databases with redundancy, Relay House Hardware Unit (with embedded Ethernet and a custom communication protocol for microcontrollers) to interface with the relays of the interlocking system, Dispatchers Controlling Interface for remote monitoring and controlling, and a TCP/IP network which interconnects all the above mentioned components of the total system.**

The current centralized traffic control (CTC) system in Sri Lanka, located at Maradana, was implemented in the year 1985. This has manually operated switch boards, a relay based operation (separate relays for controls and indications) having every switch hardwired to the relays which are located along side the railway tracks via a copper medium.

The main features of the new system are flexibility, reliability, fault isolation and robustness. The product adds flexibility to the user interface and automated report generation increases the efficiency of administrative purposes. User friendliness is a major concern that has been addressed in the development of the solution so highest priority was given to the users feedback and requirements. On top of this the user interfaces were designed to suit the existing railway signaling conventions to minimize the resistance to change. Static redundancy is provided at the heart of the system which is the server in the hot standby mode. These features minimize the probability of system failure hence increases the reliability of the system. Since this system uses TCP/IP as the main communication protocol, any type of physical media can be used for the communication network.

The new solution was successfully tested in the Test Track in Dematagoda with the presence of senior officials of Sri Lanka railways and proved its potential for implementation.

I. Introduction

Centralized Traffic Control (CTC) has been a widely used technology all over the world to remotely control and monitor railway traffic. The current system in Sri Lanka, located at Maradana, was implemented in the year 1985.

This has manually operated switch boards, with every switch hardwired to relays which are located along side the railway tracks, via a copper medium. The train dispatcher (controller) uses these switches for controlling and a relay based operation (separate relays for controls and indications) implements the controlling action.

The main drawbacks of the current system are the lack of reliability and flexibility. The copper medium is vulnerable to lightning and flexibility issues arise as altering the switch panel is extremely difficult in case any change is implemented at a station, such as an addition of a new platform. The communication is done using a hexadecimal message format with a letter used for every track, thus giving an upper bound of five to the maximum number of tracks per station. The maintenance processes are time consuming and tedious resulting in high down time. Many components are discontinued products of the respective suppliers so the authorities are compelled to go towards custom manufacturing which is very expensive. All administration work is done through manually kept reports which can be easily automated for increased efficiency. These major drawbacks and bottlenecks were addressed successfully in this project.

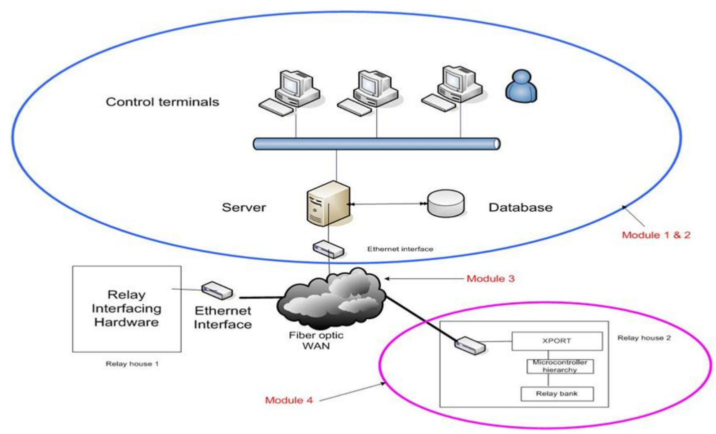

Fig. 1. The overall architecture of the system

II. Methodology

The basic architecture of the whole project can be identified as 4 main modules (Figure 1).

1) Dispatcher's Control Interface for the dispatchers to give commands as well as to see the indications.
2) Centralized Server System to handle the communication between the relays and dispatchers.
3) The communication network
4) Relay house hardware interfacing unit

In this particular solution train dispatchers are given PCs that are connected to a Local Area Network at the control station. At the relay houses there are interfacing equipment to trigger the relays with the appropriate 24 DC voltage. A Server placed at the control room handles most of the operations. It is responsible in keeping logs and communicating with the controller and the relay houses. Controls are given using the dispatchers control interface (DCI) which triggers a database query, and sends the control signal encapsulated inside a TCP/IP packet to the server, through a socket. The server forwards it to the correct relay house through an optical fiber ring, and in the relay house the Ethernet module decodes and transfers the packet to a serial bit stream. This bit stream is fed to a hierarchy of micro controllers which drives the relevant relay. Ultimately the relay drives the respective motor which is able to implement the control action.

After the operation is completed another relay, called an indication relay will be triggered to notify the user that the action is done. The status of the indications is polled by another hierarchy of microcontrollers and sent to the server through the Ethernet module. The server checks the bit stream, keeps any required information for report generation and forwards it to the respective dispatchers control interface (DCI). The DCI software updates the interface according to the incoming bit stream.

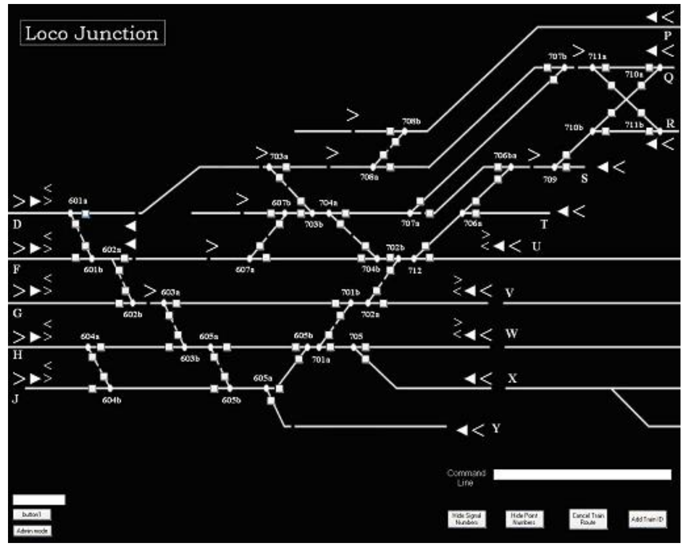

Fig. 2. Dispatcher Control Interface

A. Dispatcher's Control Interface

The first step in designing the DCI was getting a thorough understanding on customer requirements and railway signaling. After gaining this knowledge track layouts of each station (Fort, Maradana and Loco junction - the highest traffic dense areas in Sri Lanka) were analyzed to create a list of commands and indications. The track layouts were also used to design the GUI. As this was the main user interaction point special attention was given in designing the GUI. The design procedure was divided into several stages which each required the approval of the railway officials for completion. This made sure that the final GUI was optimized for the requirements of SLR. Firstly all the required functions and facilities available in the current system were implemented in the new solution to minimize resistance to change. This also included the controls and indications for each station which were designed using the layouts.

The new solution was further enhanced by some additional features which were provided to improve the user friendliness and the efficiency of the whole controlling process. These features included menus to add new indications and change track layouts, thus eliminating the flexibility issue of the current system. Menus for easy data management were provided with the ability to access the clients database and automatic report generation increased the efficiency of administration. Automation of the controlling process was addressed through time triggered routing where the controller could pre schedule the necessary control actions. The necessary security, in terms of the DCI, was provided through different password protected user accounts. Figure 2 gives a screenshot of the Loco Junction DCI.

B. The Central Server System

The centralized server is the heart of the system that handles all the communication operations and keeps the system up and running. The server is responsible for many non-trivial tasks inevitable for the system, out of which, most prominent being providing the connectivity to the relay houses and dispatcher terminals. Server also has to keep track of the records for administrative purposes and generate alarms at system failure. Due to the gravity and complexity of its operation, a special care was taken in designing the server to improve the reliability.

Firstly, the server should provide a smooth communication interface with the other nodes of the system. On the one hand the server keeps communicating with the relay houses placed alongside the railroad, and updates the dispatcher terminals located at the central office on the other. There are two major types of signals thus being transmitted; namely the controls and indications. The server keeps relay houses polling in order to get the indication signals and the relay house interface equipment responds to the server periodically, updating the status of the relays. Indication signals are forwarded by the server only to the relevant relay house using its ip address. Similar to the latter, there is asynchronous transmission of signals between the dispatcher terminals and the server, directing messages only to the relevant ip address. Synchronizing the server in these operations was a formidable task that was achieved using modularity, which will be discussed later in the section.

In the current system, there was a lot of clerical work that kept the dispatchers busy most of the time. The dispatchers have to record the time of the commands given in a sheet of paper provided to them, and these will be used for administrative purposes, especially to track at a failure. The recorded data can be both inaccurate and inadequate. But with the computerization, there was the possibility of automating most of these tasks. In order to achieve this, all the signals that pass through the server are recorded in a database. Out of these records, informative reports can be generated that will be helpful for administrative purposes. For an example, in a

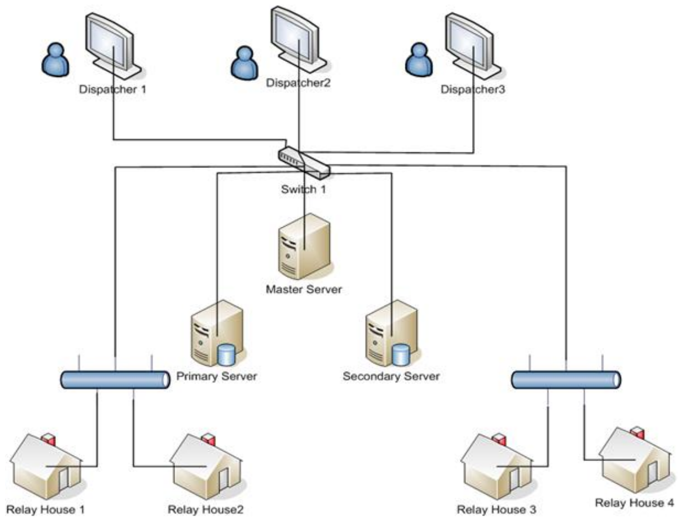

Fig. 3. Server Architecture

particular station arrival, and departure times of each train in a certain day or for a particular train the type of commands given and their time within two certain stations can be readily generated by the interface provided at the server.

The other major task of the server is to handle critical situations where some of the nodes of the system can fail. There can be temporary failures of the system that can correct by itself after some time and permanent failures that keep down unless someone does something about it. This system has the capability of reconfiguring it automatically at temporary failures. This is very important when several network connections are involved where any one of them can fail at a loose connection. When the connection is up again the system automatically reconfigures itself by listening to the relevant port number and connecting to it. Therefore, no human involvement is necessary at a temporary failure like a loose connection in the network equipment. On the other hand, some parts of the system can stop functioning permanently. Either the cards at the relay house or even the relay house itself may stop responding due to some malfunctioning. In these circumstances, the server generates an alarm and notifies the users about this, so that he can take some action. The server has the capability to point failures to the card level-i.e. server says which card at which relay house is at failure so that only that one needs to be replaced.

Some design techniques were devised to improve the reliability of the server, since a server failure can never be tolerated. [1] The approach was mainly two fold; firstly, the system was modularized both in hardware and software; secondly, redundancy was used to improve the reliability. Modularity, is a widely used technique to achieve fault isolation [2] [3]. The server was designed in several modules so that even if a single module fails it does not propagate from module to module bringing the whole system down. This was also helpful in achieving the synchronization in the communication process, because the processes of listening, receiving, data-processing and sending could be made independent of one another. Principles of object oriented programming and multiple threads of execution were used extensively in modularization as well as to increase the throughput of the system. To abstract the complexity of communication special classes were created to handle and abstract the process of receiving and sending data to and from different nodes of the system. The objects created out of these classes will simplify the operation of communication to the main module. At a failure any node, an error is reported to the main module and the thread of communication is restarted so that it is capable of reconfiguring when the connection is up again. This property made our system resilient to the temporary TCP/IP connection failures that could occur. The other method that was utilized in improving reliability was by providing redundancy. The server is replicated to provide hot standby redundancy. This is known as static redundancy, which keeps both the servers alive all the time and data go through both of them simultaneously, thereby keeping the databases attached to each server updated. In addition to these primary and secondary servers, there is a control master server which receives results from both. The objective of this master server controller is to detect and initiate the switch server operation at a primary server failure. The figure 3 depicts the system architecture with redundant servers in place.

C. The Communication Network

The main goal of this network is to provide reliable data transfer between the relay houses and the control office. The proposed network is a fiber optic ring which eliminates the hazard of lightening. However, as standard Ethernet interfaces are used at the two ends and as TCP/IP is used as the main communication protocol, the final user has the added flexibility of using the most optimized and preferred medium for the network by using media converters. Also standard security solutions can be implemented easily because of the usage of a common protocol.

D. The Relay House Hardware Unit

The main goal of hardware design is to develop a reliable and efficient controlling and monitoring unit which is compatible with the existing railway infrastructure. Therefore the hardware unit had to perform two major tasks. Firstly it had to interface the control signals carried through the TCP/IP based communication network to the existing control relay bank and then, transmit the status of the indication relays to the control room in the form of a bit stream. The modular approach which is used in the solution divided the hardware into three main modules.

1) TCP/IP module - communication and serial to TCP/IP conversion
2) Microcontroller network - process the serial bit streams
3) Isolation circuits - protect the microcontroller network from surge currents in the relay system.

An Ethernet module with a large operation temperature range, high data rate and low cost were selected and it was configured to automatically switch to the active server out of the two, at the control room. The microcontroller network, illustrated in figure 3, is basically responsible for two tasks which are decoding the serial bit stream received from the control room to drive the specific relay that is responsible for that control operation and extracting the status of indication relays, encoding it as a bit stream and sending it to the control room. For this purpose a microcontroller network which uses master / slave configuration for data communication within the network was developed. Slave microcontrollers interact with the relay system while master microcontrollers control the slaves using different types of commands. The slaves were made homogeneous and, the master and each of the slaves were implemented on separate Printed Circuit Boards (PCBs) to achieve modularity and the ease of maintenance with minimum down time.

A customized communication protocol was designed for efficient and reliable data communication within the microcontroller network. This protocol uses the RS 232 interface of microcontrollers for data communication. It includes time outs and acknowledgements for data retransmissions and uses majority vote logic when polling the status of the relays, to nullify bouncing. The customized protocol increases the operational reliability, and brings out extended features such as identification of system faults at the card level and notifying the user. Message formats, message lengths and delays were defined for optimum performance within the system after extensive testing.

Opto-coupler circuits were used to achieve electrical isolation between the two systems and the robustness of the circuits were increased such that it can support up to 3 times the maximum ratings specified in the relay specifications through proper current amplification

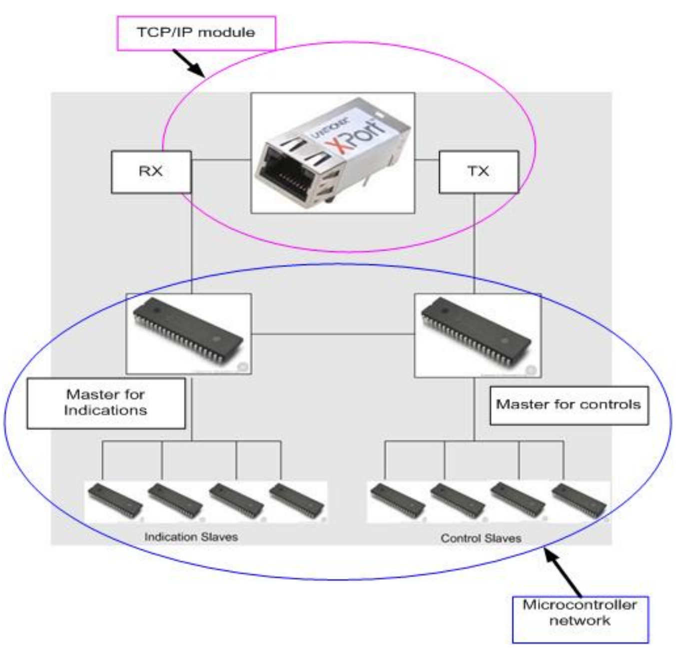

Fig. 4. The architecture of the hardware unit

III. Conclusion

At the initial design stage simulation techniques were used to test the accuracy of the design. Relay houses were simulated in software and full functionality of the DCI and the central server system was tested on the simulated relayhouses. On the second stage of system designing, prototyping of hardware circuitry was done and tested for accuracy and reliability with the DCI and the srever system. Optimization of communication protocols was done at this stage. After observing the interaction of network delays and the hardware circuitry, some communication protocols were modified to make the system more reliable. [4] [5] The prototyped hardware units were then tested with exact relays in the CTC system at railway workshop Dematagoda.

After further modifications on current handling capacity and current amplification factors, a more reliable hardware unit was designed for the relay house. Once again they were tested with working relays in the test track at Dematagoda. After successful testing, the actual implementation on Printed Circuit Boards (PCB) was done. After the implementation on PCBs, hardware were tested again with the working relays to make sure the final implementation is error free and reliable.

The final project demonstration was done with the presence of the General Manager Railways and the academic staff of the Department of Electronic and Telecommunication Engineering, University of Moratuwa, on 29th May 2008 at Dematagoda.

IV. Further Improvements

As described, this system runs on the existing control logic infrastructure, which watches over all the commands given by the centralized system, ensuring error free operation. It can be cited as an advantage of the new system, having the ability to be directly plugged on to the current control system, which has a proven track record. The current system has a purely relay based operation. However, due to reasons like lack of room for expansion and difficulty in replacing faulty parts, it is important to develop a microprocessor based interlocking system. The main objective of the interlocking system is to provide safety at all levels, starting with the prevention of catastrophic accidents. Fool proof operation and safety assurance should be achieved using Fault Tree Analysis and diversified redundancy schemes. Moreover, error reporting and logging should be in place at the interlocking system for ease of maintenance and fault identification. The microprocessor based interlocking system should be able to provide room for expansion. i.e. the logic system should have provisions for the addition of new signaling devices.

V. Acknowledgment

We would like to extend our gratitude to the General Manager and other officials of the Department of Railways Sri Lanka for the support they rendered throughout the process and the academic and non-academic staff of the Department of Electronic and Telecommunication Engineering, University

of Moratuwa, who extended their fullest support to make this project a success.

References

[1] S.-T. Levi and A. K. Agrawala, *Fault tolerant system design*. New York, NY, USA: McGraw-Hill, Inc., 1993.

[2] D. P. R. Saltzer J. H. and D. D. Clark, "End-to-end arguments in system design," *ACM transactions on Computer Systems*, vol. 2, no. 4, pp. 277–288, Nov. 1984.

[3] R. W. Scheifler and J. Gettys, "The x window system," *ACM Transactions on Graphics*, vol. 5, no. 2, pp. 79–109, 1986.

[4] P. Kubalik, P. Fiser, and H. Kubatova, "Fault tolerant system design method based on self-checking circuits," in *IOLTS '06: Proceedings of the 12th IEEE International Symposium on On-Line Testing*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 185–186.

[5] N. Lawson, "Fault tolerant system design."