2004 IEEE Intelligent Transportation Systems Conference
Washington, D.C., USA, October 3-6, 2004

Modeling Advanced Traffic Signal Control Systems: A Communication Network Prototype

Eresh M. Suwal, *Student Member IEEE*, Troy R. Cuff, *Student Member IEEE*, Ben E. Hamlett, Brian K. Johnson, *Member, IEEE*, Richard Wall, *Senior Member, IEEE*, and Ahmed Abdel-Rahim

Abstract—This paper discusses the development of a microprocessor based hardware design as a solution to the need for automated ITS network configuration for differing modes of traffic controller operation. The design, called Controller Interconnection Network (CIN) was required to be scalable in number of traffic controllers and permit multimode operation. The design allows remote control of some operations via a web-based interface.

I. Introduction

THE evolution of Intelligent Transportation Systems (ITS) technologies promises new tools for integrated traffic signal systems, including centralized control systems and the provision for real-time adaptive control. Many states and metropolitan areas have already deployed integrated centralized signal control systems to assure optimal control. These often consist of a network of traffic controllers and detection and surveillance devices connected to a centralized control center through an extensive communications network. The developments in communication and computing technologies, however, have not been accompanied by corresponding developments in evaluation and decision-making tools that can take advantage of real-time data to produce more efficient control strategies. This is primarily due to the inability to model advanced control systems, such as centralized and closed-loop control systems, in a controlled lab environment. This paper describes the design of a device, the Controller Interconnection Network (CIN), which will enable engineers to model and simulate such systems in the lab.

U.S. Department of Transportation University Transportation Centers Grant DTRS98-G-0027 supported the work on this project. This work was performed through the National Institute of Advanced Transportation Technology (NIATT) and was undertaken as part of a senior level design course in the Department of Electrical and Computer Engineering at the University of Idaho.

B. K. Johnson is with the Dept. of Electrical and computer engineering at the University of Idaho, Moscow, ID 83844 USA (phone: 208-885-6902; e-mail: b.k.johnson@ieee.org).

R. Wall is also with the Dept. of Electrical and computer engineering at the University of Idaho, Moscow, ID 83844 USA.

A. Abdel-Rahim is with the Dept. of Civil Engineering at the University of Idaho, Moscow, ID 83844 USA.

E. Suwal and T. Cuff are graduate students in the Dept. of Electrical and computer engineering at the University of Idaho, Moscow, ID 83844 USA.

B. Hamlett is an engineer with Idaho Instruments, Inc. in Twin Falls Idaho, USA.

II. Background

Traffic signals and traffic controllers are devices used to control and regulate the flow of traffic at intersections. Optimal signal timing for the intersections are developed and tested through a variety of traffic optimization and simulation models that simulate the traffic behavior and emulate the possible actions of the traffic controller. With the advancement in traffic controller computing power and control logic, the issue of whether the simulation model controller "emulator" accurately replicates the actual performance of the field controller has cast considerable doubts on the output of the simulation models.

In a major improvement to the simulation process, with the help of Controller Interface Devices (CIDs) engineers can now simulate their timing plans with the signals in the simulation being controlled by an actual traffic controller [1,2,3]. This is called real-time hardware-in-the-loop simulation, and it produces more realistic results than a software emulator when run in real time does [2].

To take this a step further, engineers need to be able to simulate ITS networks of traffic controllers in a lab without manually disconnecting and reconnecting the cables every time the network configuration needs to be changed. The CINs would facilitate such simulations in the lab. A graphical user interface (GUI) gives the user a graphical depiction of the network, eliminating the need to trace and reconfigure cables manually.

III. Hardware-in-the-Loop Simulation

It is important to understand how the CIN will be used in conjunction with other hardware and software as a part of hardware-in-the-loop simulation.

In a typical simulation, software such as CORSIM, Federal Highway Administration's (FHWA) microscopic simulation model, simulates a real-world traffic network by moving individual vehicles across a combined surface street and freeway network using accepted vehicle and

0-7803-8500-4/04/$20.00 ©2004 IEEE

driver behavior models and simulating various traffic control devices. The software contains algorithms to both track vehicles through a prescribed highway network and to implement a coordinated actuated signal system [4].

Hardware-in-the-loop simulation (HILS) is different in that, instead of having CORSIM simulate controller features, the CORSIM traffic model only simulates the vehicle detector signals. The control strategy is run on an actual traffic controller that will be used in the field. A controller interface device (CID) provides the real-time linkage between CORSIM and the traffic signal controller [1,2,3].

The CID is a USB based device that relays detector information from the simulation software to the traffic controller, and phase information from the traffic controller to the simulation software as shown in Fig. 1. The CID makes hardware in the loop simulation possible.

IV. CIN Project Objectives

The main objective of the project was to enable the user to simulate three different traffic controller network configurations in a laboratory environment: isolated, master-based and closed-loop configurations. The CIN makes this possible by allowing the user to connect and disconnect traffic controllers from a network using a graphical interface in the software. This eliminates the need for manual connection or disconnection of the cables once all of the controllers are connected to the device. One CIN connects to up to 5 traffic controllers, plus a connection to a neighboring CIN. The GUI supports the interconnection of up to 4 CINs for a total of 20 connections to traffic controllers at simulated intersections.

A. Isolated Setting

In the isolated setting, the traffic controllers function independent of other controllers in the vicinity. There is no sharing of information [4]. This is the most basic type of configuration and is effective in small towns where the volume of traffic to be controlled is low. Fig. 2 shows a isolated network with 5 traffic controllers connected to a computer that can be used to run the traffic controller setting software. There is no communication between the traffic controllers.

B. Master-Based Setting

In this setting the controllers are networked through a master controller and the controllers work as a system [4]. The master controller itself does not control any intersection and only functions as a hub for the other "slave" controllers. It can use the shared information to reprogram the slave controllers to handle the traffic more efficiently.

Fig. 3 shows controllers in a master-based setting. As

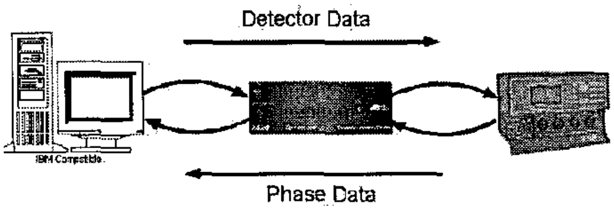

Fig. 1. Hardware-in-the-loop simulation with a CID

with isolated controllers, the computer can also be used in this system to program individual controllers. Notice that control signals now travel between the master controller and the slave controllers.

C. Closed-Loop Setting

The closed loop configuration is similar to the master-based setting, except that a computer controls the slave traffic controllers in a closed network instead of a master controller [4]. Depending on the number of controllers in the system, this might be a better option because it is easier to use and set up a computer than a controller. Plus a computer provides more flexibility and portability. Fig. 4 shows controllers in a closed-loop setting.

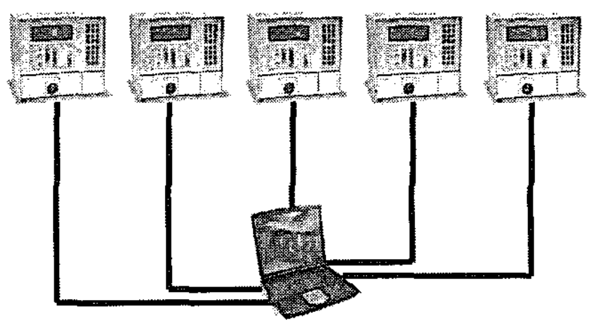

Fig. 2. Isolated Setting

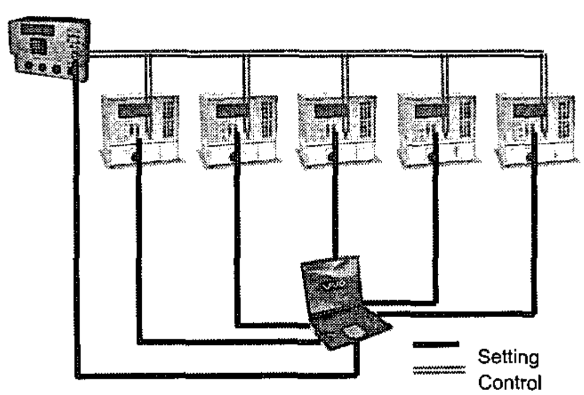

Fig. 3. Master-based setting (Master controls traffic controllers.)

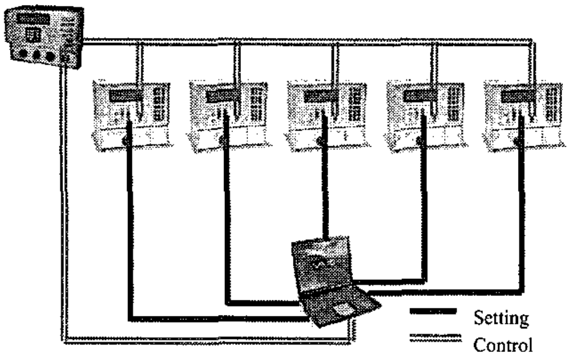

Fig. 4. Closed-loop setting (computer controls traffic controllers.)

V. APPROACH

To accomplish the given objectives, an embedded-controller approach was used for the hardware design. This method was chosen for its simplicity in design and ease for updating and change. A graphical user interface (GUI) displays the status of the network to the user and allows the user to make desired changes in network configuration. The GUI sends commands to the device (embedded controller) that makes the necessary changes in the hardware to set the configuration to the one desired by the user.

VI. IMPLEMENTATION

A. Hardware

Addressing the need to easily connect and disconnect controllers from the network was a key objective of the project. This was achieved using relay switches controlled by an embedded microcontroller. This microcontroller is controlled by Windows-based software using the RS232 communication scheme. It receives commands from the software, reacts to the commands (generally by opening or closing switches) and sends acknowledgements to the software.

Fig. 5 shows a conceptual diagram of the internal logic of the CIN. The internal network provides connectivity between the traffic controllers (TC) and the master controller (MC).

The external network extends this connectivity to traffic controllers and master controllers connected to other CIN devices. The setup lines provide connections for individual internal configuration of the controllers using the controller manufacturers proprietary setting software.

B. Software

A GUI was written that displays the current configuration of the network and allows the user to change it by selecting the buttons as shown in Fig. 6. The software supports up to four CINs simultaneously, detects them and updates the display automatically. Each CIN controls up to 5 traffic

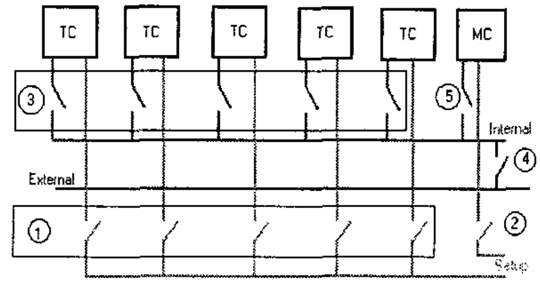

Fig. 5. Conceptual diagram of CIN logic

controllers, so the user can control up to twenty traffic controllers.

C. Embedded Controller

The embedded controller is the interface between the GUI and the hardware switches. The firmware code, written in C, resides in the memory of the microcontroller. The controller receives commands from the software via the serial interface and opens or closes switches to break or establish communication channels between traffic controllers. Each CIN has dipswitches that can be used to identify it. This number is stored in the firmware as the bank identification (BID) number and is used by the software to identify and communicate with a particular unit if more than one are connected. The use of BID allows the GUI to support multiple units.

VII. OPERATION

Initially, all the traffic controllers need to be connected to the CIN. There are two ports for each controller on the back of the CIN (ten total) that connect to two ports on the controller. The Setup port connects to a modem port on the controller. Communication via this channel can be used to set up the internal configurations of the controller and is independent of the network. The network port connects to the telemetry port on the controllers, which is used to network the controllers.

Three ports on the front of the CIN connect to three RS232 ports on the computer. The "box setup" port is used to control the network configuration through the CIN. The "MTC setup" and "TC setup" ports are used to set up the internal configuration of the master controller and the slave controllers respectively. The master controller has a separate port since it uses software different from the slave controllers and it also enables the user to set up a master and a slave at the same time if need be. There is only one setup port for the slave controllers because the software that is used to set up the controllers is the same for all, and the user can only set up one controller at a time. Two other ports on the front of the CIN are labeled "external bank connection." These ports can be used to connect the CIN to

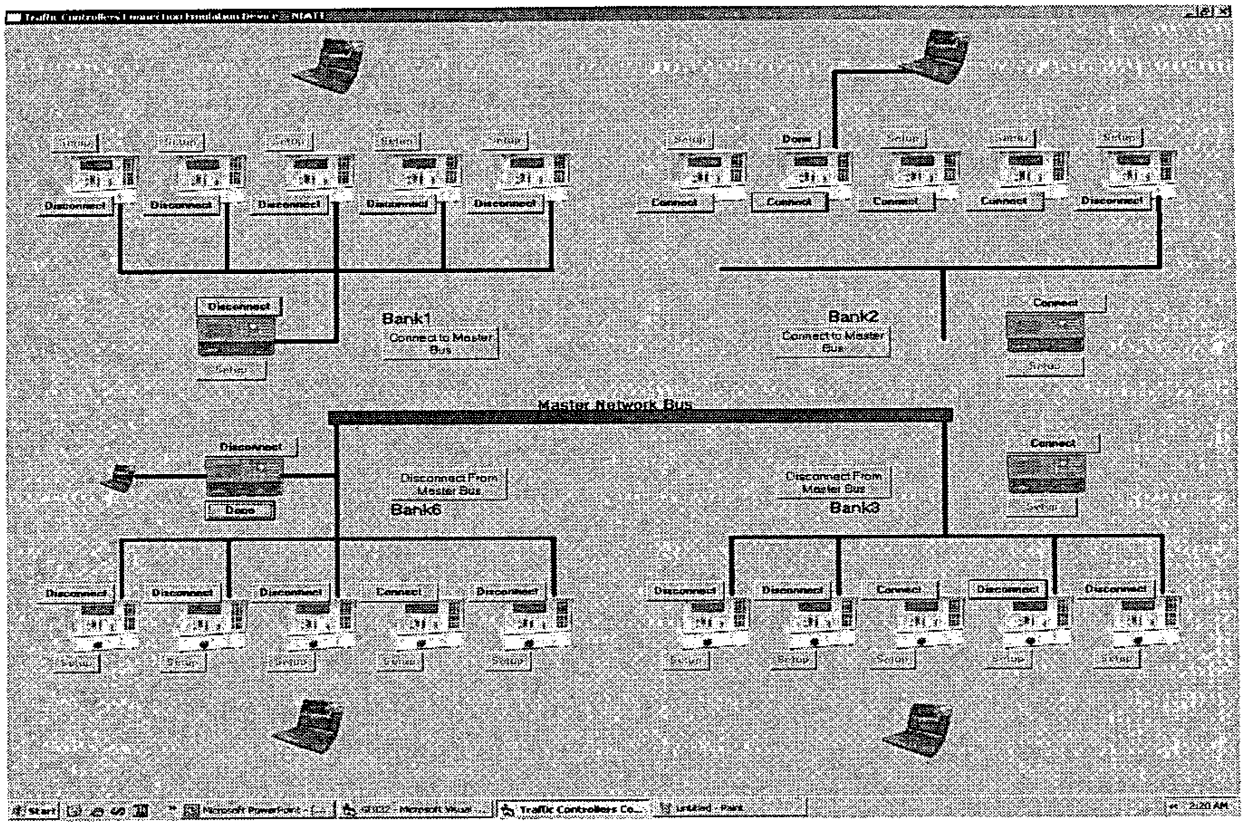

Fig. 6. GUI displays current configuration with four banks connected.

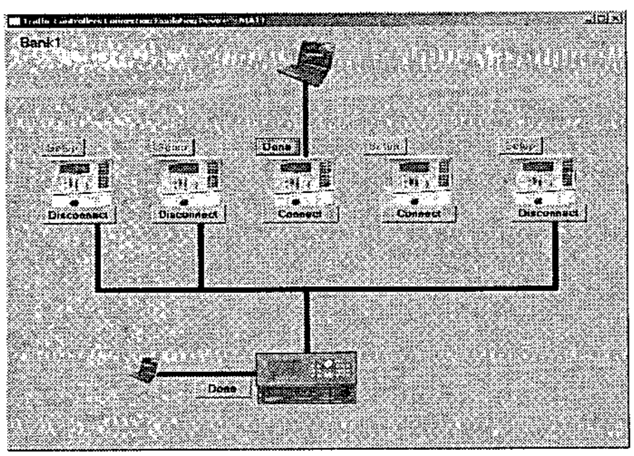

Fig. 7. GUI displays the configuration with only one bank connected.

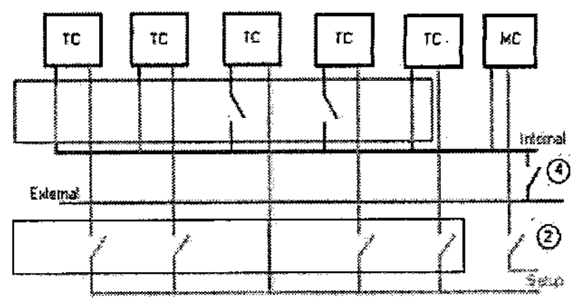

Fig. 8. Internal switch configuration for network in Fig. 7.

other CINs. Only one CIN is directly connected to the computer. This is the primary CIN. Up to three secondary CINs can be connected to the primary CIN, as shown in Fig. 6.

Once all the controllers are connected to the CINs and the CINs are connected to the computer, the CIN is ready for use. Before the GUI can detect the CINs, it has to know which serial port that the primary CIN is connected to. The port can be selected from a pull-down menu that shows the available ports. The GUI will display, depending on the number of CINs connected, an interface with four banks as in Fig. 6 or with just one as in Fig. 7.

As seen in Fig. 6 and Fig. 7, the buttons can be functionally categorized into four sets. The captions on the buttons depend on the current state of the corresponding connection. For example, the setup buttons connect the controller to the computer for internal controller configuration. When a controller is connected, there is a line connecting the controller and the computer and the button caption changes to "done" instead. It should also be noted that the "setup" buttons on the other controllers are

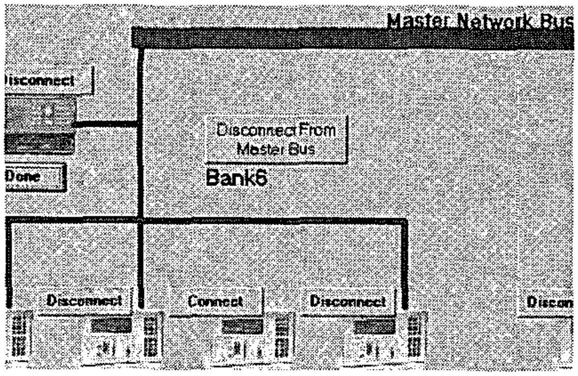

Fig. 9. Bank 6 is connected to the external master network bus; the caption on the button is hence "disconnect from master bus".

disabled since the setup software is only able communicate with one controller at a time. This does not apply to the master because the master controller uses a different port for setup, as discussed earlier.

Similarly, the "connect" button connects the controller to the internal network. If connected, a line appears that connects the controller to the master and the caption on the button changes to "disconnect." Fig. 7 can be compared to the internal state of the CIN in Fig. 8 that shows some switches closed to establish the required connections. Also, there are light emitting diodes (LED) on the front of the CIN that indicate the status of the connections.

If more than one bank is connected, as in Fig. 6., then the "master network bus" line also appears on the screen and the user is given the option of connecting a bank to the external "master network bus," as shown in Fig. 9. To change configurations, the user can use the onscreen buttons to connect or disconnect controllers from the network. The GUI also allows the user to save a configuration and load it at a future time, but requires that there be the same number of banks as when the configuration was saved. The BIDs need not be the same.

VIII. Results

The CIN device has been tested with traffic controllers and it has been verified that it successfully establishes and breaks communication channels between controllers. It was verified that the GUI and the LEDs on the CIN depict the correct status of the connections. It was also verified from the master controller's LCD that there was communication between the connected traffic controllers via the master. This was also done for controllers in two different CINs. Using controller setup software, the internal controller data was downloaded via the CIN, verifying that the setup connections were working.

It was tested and verified that the GUI supports multiple CINs and automatically detects CINs and changes in the BID. The GUI was tested with up to three CINs.

Tests involving more than five traffic controllers and more than one master controller have not be performed to date. However, the results obtained are very promising, and CINs will be tested with larger networks in the near future.

IX. Conclusion

This paper describes the development of the Controller Interconnection Network and how it can be used to simulate traffic controller networks in a lab environment without the need for manual connection and disconnection of cables every time the network configuration needs to be changed. The device makes process easy by graphically displaying the current configuration and status of the network, giving the user a picture of the entire network at a glance.

Acknowledgment

The authors would like to thank Ron Nelson and Zhen Li for their valuable suggestion and assistance, and Dr. Michael Kyte and Dr. Mike Dixon for their support and guidance.

References

[1] Bullock, D., and A. Catarella. "A Real-Time Simulation Environment for Evaluating Traffic Signal Systems". Paper presented at the *77th Annual Transportation. Research Board Meeting*. Washington D.C., January 1998.

[2] Bullock, D., B. Johnson, R. Wells, M. Kyte, and Z. Li, "Hardware-in-the-loop simulation," *Transportation Research Part C: Emerging Technologies*. Vol. 12, Issue 1, pp. 73-89, February 2004.

[3] Engelbrecht, R.J., K.N. Balke, S.P. Venglar, S.R. Sunkari. "Recent Applications of Hardware-in-the-Loop Traffic Simulation." *Compendium of Papers of the 70th Annual Meeting of the Institute of Transportation Engineers.* (CD-ROM), Institute of Transportation Engineers, Washington D.C., 2000.

[4] R. L. Gordon, R. A. Reiss, H. Haenel, E. R. Case, R. L. French, A. Mohaddes, R. Wolcott, *Traffic Control Systems Handbook*. FHWA-SA-95-032, February 1996, Ch 4.