Design and Implementation of a Safety Communication Network in Railways with Intelligent Fault Diagnosis

César Mataix, Pedro Martín, Francisco J. Rodríguez, María J. Manzano,
Javier Pozo, Patricio G. Donato
Departamento de Electrónica. Universidad de Alcalá
Campus Universitario, s/n. 28805 Alcalá de Henares. Madrid
SPAIN

Abstract – **This paper presents a network that connects various safety sensors located on level crossings and in stations. These sensors are used to detect obstacles on the railway line and proximity between trains. The information is centralised in the Operations and Control Centre. The network has been designed in sections, each of which consists of a dual bus structure, with the particular feature that if one of the buses fails, the packets are routed to the other. Fault detection on the network is performed using intelligent diagnostic techniques, applying the IEEE 1232-2002 standard. By examining the result of the diagnosis, it is possible to ascertain the optimal route from each sensor to the OCC. Monitoring is performed using active network techniques. The diagnostic system sends packets containing code that is executed at each node.**

I. INTRODUCTION

The Department of Electronics of the University of Alcalá, in co-operation with the state-owned rail operator RENFE (*Red Nacional de Ferrocarriles Españoles*) and the firm Logitel, is working on a research project financed by the Ministry of Science and Technology. This project, titled TELEVÍA (*Control Integral de la Circulación y Seguridad en Líneas Ferroviarias* - Integral Control of Traffic and Safety on Railway Lines), takes an integrated approach to the various problems related to automated control and safety for rail traffic on lines with low-to-medium traffic density.

Part of this project focuses on telecontrol and telemonitoring, the objective being to monitor the status of a series of systems installed in stations or their environment (axle detector, signalling, level crossing control, presence of obstacles on the track, etc.) [1]. By using sensors located on level crossings and in stations, it is possible to detect critical safety situations [2] that could have grave consequences, such as, for example, obstructed level crossings, people crossing the track at prohibited points, excessive train axle temperature, etc. Depending on the circumstance that needs to be detected, the sensors may be ultrasonic sensors, infrared sensors, machine vision sensors, axle detection and temperature measurement sensors.

Each of these has to be connected to the Operations and Control Center (OCC), which is located at a rail terminal and is where the information is centralised. When a hazard situation is detected by a particular sensor, a warning is sent to the OCC, where appropriate measures will be taken [3].

Given that the number of sensors that may exist along a route covering hundreds of kilometers is likely to be high, the problem arises of establishing communication in a practical, safe and reliable manner [4]. This paper presents a new safety-communication network that reliably interconnects the various safety sensors and the OCC. It describes a Wide Area Network that incorporates intelligent diagnosis to detect faults using active network techniques and optimal routing of the packets to the OCC.

II. BACKGROUND

Previous works have been written on communication networks applied to the rail environment, but these have tended to focus on monitoring the energy system and SCADA systems. Communication between the remote terminal units (RTU) and the control center is established in [5] via a dual fiber optic ring and duplicated servers, the aim being to increase availability and reliability, but the work does not incorporate any elements to diagnose the status of the communication network. The RTUs are connected to front-ends, which perform the communication protocol adaptation and historical data storage tasks. Among the future works suggested, it highlights the possibility of including intelligent diagnosis, as well as improvements to facilitate maintenance. In [6], a monitoring system for a level crossing is designed. Access to the variables measured is facilitated either via an HTTP server incorporated in the remote system or via a local terminal located on the level crossing itself. The remote system is equipped with Telnet and FTP servers, which makes it possible to carry out maintenance operations, enabling, for example, the software version to be upgraded. Some of the drawbacks are that data analysis is performed off-line by an operator and the system is not provided with redundancy of any kind. In [7], a remote system able to detect the presence of rocks on the track using acoustic and infrared sensors is designed. The data capture and processing tasks, which require shorter processing times, are programmed in C++, whilst visualisation of the results is achieved via an applet that is downloaded to the client's browser from an incorporated HTTP server. An additional telephone line is included to monitor the status of the same. In [8], trends in railway energy management using distributed systems based on independent dual bus LANs and TCP/IP are presented. These operate in client-server mode to provide a high-availability rapid response in real time, as well as reducing network traffic. The possibility of checking the proper operation of the communications system is not included in this paper either.

III. NETWORK ARCHITECTURE

The monitoring network has been designed on a modular structure divided into sections (each section existing between two gateways), thereby facilitating maintenance and

0-7803-7937-3/03/$17.00 ©2003 IEEE

implementation. Fig. 1 shows one of the sections that make up the network. In addition, the network is organised into three hierarchical levels - sensor level, intermediate level and control level.

The *sensor level* is composed of the various sensors, the safety node (SN) and a LonWorks fieldbus that interconnects them, covering a distance of many kilometers. When the sensors detect a hazard situation, they generate an alarm packet that is sent to the safety node. The format of the packet depends on the type of sensor that has generated it, but it generally contains the sensor's unique identifier, the date and time and the alarm identification code. This, at the same time, stores the event in an historical file and transmits the packet to the intermediate level over a TCP/IP network. The machine vision sensors are able to provide, on demand, the sequence of images prior and subsequent to the moment at which the alarm was produced, which will be visualised in the OCC.

The *intermediate level* consists of the intermediate modules (IM), the gateways (G) and the dual communication bus. Each section can cover distances of up to 80 Km. Intermediate modules are located in the stations along the route and in the electric power substations. The safety nodes connected to bus *ghg i* send the alarm packets to the intermediate module connected to its own bus. In the case of the nodes connected to bus *gg i*, the packets are sent to the nearest gateway, which will resend the packet via bus *ghg i*. In the case that the intermediate module of the section is not available, the packets are sent, via the gateways, to the nearest adjacent intermediate module, and if this is also unavailable, to the next one until an operative intermediate module is found. A particular characteristic of the proposed architecture is that the two buses are not independent, as is usually the case, making it possible to route the packets to one or the other in the gateways, depending on the level of congestion or availability. This, along with the feature of being able to send alarm packets to any intermediate module, increases the reliability of the system and guarantees its operation, even in degraded mode. Although the intermediate modules are not assigned the control task, which is reserved for the OCC, it would be possible to place the system in a standby state from them if none of the OCCs were available. For this purpose, they are equipped with a screen that shows the status of their section and they store all of the alarm information in their local database (LDB).

The *control level* is composed of the OCCs, a bus for communications between the various intermediate modules and the OCCs, and a high-speed bus that enables database replication in real time. The route contains two OCCs, although only one of them will be in operational mode (able to perform actions), whilst the other will be in monitoring mode and will not be able to perform any actions. These concentrate the alarm warnings generated by all of the sensors along the route. Using acoustic and visual signals, these will display any possible alarms detected to the operator. The high-speed networks will be used to make periodic replicas of the OCCs databases (PDB), so that at any moment either of them will be able to take control of the network, if the situation so requires.

IV. INTELLIGENT FAULT DETECTION

The critical safety network designed, made up of the sensors, the communication system and the Operations and Control Centers, enables safety in a rail environment to be enhanced, but the possible operational faults that may be produced in the same, such as bus failure, out-of-order nodes or intermediate modules, also need to be taken into account. It is thus vital to establish a fault detection system that is able to ascertain whether a certain element is out-of-order and, if so, take appropriate measures to ensure that it affects the operation of the communication system as little as possible.

As it is a distributed system covering hundreds of kilometers, buses and digital systems monitoring should be performed using the network infrastructure and the TCP/IP protocol. By sending probe packets to each digital system and receiving the response, it will be possible to ascertain if these are operational. Moreover, if some packets do not reach their destination, it will be possible to deduce the existence of a fault in one of the sections of the bus.

In a preliminary implementation the results of the test were sent to the OCC, where they were displayed on screen, but it was still necessary for the operator to analyse them. It was then observed that it would be useful to add a result analysis system that would back up the operator's decision-making process. Thus, an intelligent diagnostic system has been included for this purpose.

Intelligent diagnostic systems enable the problem to be identified by analysing the symptoms observed, acting in a way similar to a human expert. The decision element (reasoner) can be based on any of the artificial intelligence techniques, such as neural networks, expert systems, induced learning systems, decision trees, etc. Said system was implemented in compliance with the AI-ESTATE [9][10][11][12] standard that makes the decision system independent of the test system, and at the same time uses standard data and knowledge models.

A. Application of the Diagnostic Standard on the Network Architecture

Fault detection in the communication network designed is performed using an intelligent diagnostic system conform to the AI-ESTATE standard. An additional computer has been connected on the control level of the network that implements the architecture shown in Fig. 3. The *Application Executive* (AE) sends a diagnostic execution request at 5-seconds intervals to the *Reasoning Component* (RC), invoking the services of the standard that have been implemented. This loads the network model from the database and sends the test vector to use to the *Test System Component* (TSC). This vector contains the list of elements and buses that should be probed, identifying them by their IP address. Testing of each of the network elements is performed by executing a test task that checks the execution status of the operation tasks and returns the result to TSC.

In order to increase the maintainability of the system, the code of the test task is sent over the network every time that it is executed, employing active network techniques.

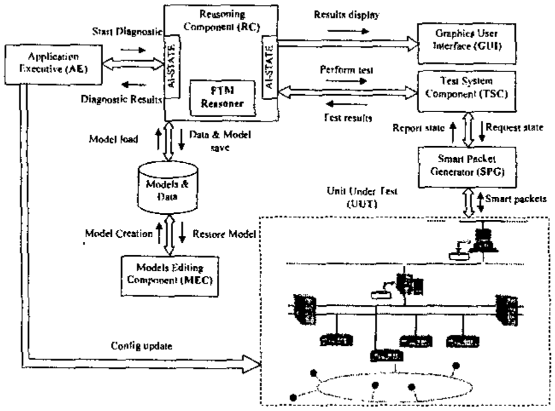

Fig.3 Implementation of the Fault Detection System

This enables the functionality of the test to be modified remotely, depending on the network model loaded or on the configuration. The test packets are routed through the gateways, in the same way as the alarm, image and configuration packets. Based on the test vector, the *Smart Packet Generator* (SPG) sends three active packets over each of the sections of the network - the first to the safety nodes, the second to the two gateways and the third to the intermediate module-. Each packet contains the IP addresses of the nodes in which the test task should be executed. The result of execution of this task is returned to the SPG. The information returned would vary depending on the element in which it has been executed. Thus, for example, the result of a gateway test is composed of the execution status of the operation task, the integrity of the four connected buses, the current routing table and the time that the bus probe packets have taken to traverse them.

The results of the test are returned to the RC, where they are evaluated using a fault tree model, ascertaining the status of each element, as well as identifying possible faults in the communication buses. The diagnostic is returned to the AE and is displayed in the *Graphics User Interface* (GUI).

The AE determines the optimal route from each node to the intermediate module for the alarm packets, this being the fastest route to an intermediate module. Based on the optimal routes, the new routing tables are calculated for each gateway and these are sent to the same. Therefore, although there may be a fault in one of the buses, it will still be possible to route the alarm packets to the intermediate module, avoiding the faulty bus. At the same time, the network status information is sent to the OCCs, where network monitoring is performed.

V. RESULTS OBTAINED

In order to make the test, a 10 Mbps Ethernet network with Pentium PC has been designed. The computers execute Suse Linux 8,1 Professional and all the applications have been programmed completely in Java. Real-time operating system is not used since the Ethernet network is no deterministic and the applications are oriented to network. The OCC executes Microsoft Windows 2000 Server. The Java version is JSDK 1.4.1. In Fig. 4 the test network topology can be observed.

Table I. Delay Measurements Taken in the Test

Type	Origin	Destination	Delay (min-max)
Integrity packets	SN-02-01	G-02-03	2-18 ms
Alarm packet	G-02-03	IM-02	3-10 ms
Integrity packets 10 Kb	G-02-03	IM-02	3-15 ms
Alarm packet	SN-02-02	IM-02	10-30 ms
Alarm packet	SN-02-02	OCC-01	12-33 ms
Alarm packet	SN-02-01	G-02-03	15-32 ms
Historical packet 10Kb	G-02-03	IM-02	20-32 ms
Image packet 64 Kb	G-02-03	IM-02	50-80 ms
Historical packet 10Kb	SN-02-01	G-02-03	82-250 ms
Image packet 64 Kb	SN-02-01	G-02-03	150-250 ms
Image packet 64 Kb	SN-02-02	IM-02	205-360 ms

As no sensors were available, their operation was simulated from the safety node itself, which enabled operation in various scenarios to be tested.

The network implementation has been made in two phases. In first, the operation tasks of each element have been programmed. These are: alarm packet repeat, image packet repeat, historical report, clock synchronization, automatic configuration of the network, graphic user interfaces and integrity of the network.

Network model is generated automatically from the integrity packets. The process takes about 2 sec. It's updated whenever a change in the network topology happens.

Fig. 5 presents the intermediate module GUI. It's appraised the connection with gateways and two safety nodes; one of them connected to the ghg 2 bus and the other to the gg 2 bus.

In order to test the response of the communication network, several response time measurement tests were performed. The minimum and maximum values of some of the measurements may be observed in Table I.

The safety nodes send three types of operation packets - alarm, historical and image-. The alarm packets originate from one of the connected sensors and have a size of 140 bytes. The historical packets contain the list of alarms that have been produced in the sensors connected to this node, whilst the image packets are sent by the machine vision sensors and contain an image of the scene in JPEG format. In order to check the status of each element, integrity packets were sent, which enabled the operational status of each of them to be checked. The integrity information for each section was stored in the gateways, from where it could

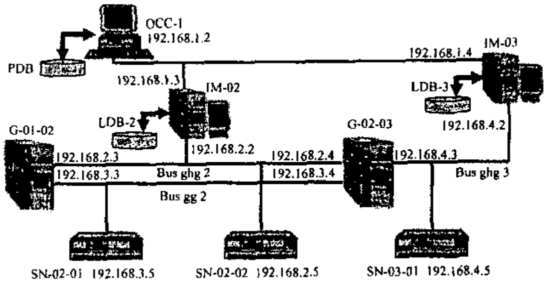

Fig.4 Communication Network Employed in the Trials

be queried by the intermediate modules.

The OCC Application is programmed in Java. This application supplies the operator with three types of information:

- Network topology: It shows the intermediate modules connected, and clicking on each one of them a new window appears with its complete section representation.
- Alarm history: containing a list of the alarms produced, indicating the date, time, sensor generating the alarm, etc. The alarms originating from a machine vision sensor will also display the image sequence.
- Integrity table: showing the status of the network and representing the information supplied by the diagnostic system. The operator will be able to observe whether any faults exist and to take appropriate measures.

In the second phase, a Java API for the AI-ESTATE standard is being programmed. This contains all the entities and services of models CEM, DCM, FTM and EDIM. An application has been made using this API. The application loads the network model and executes the diagnosis of the network. Currently, fine-tuning of the diagnostic system is underway, and implementation of the services necessary for the models employed in the to reasoner is being concluded.

VI. CONCLUSIONS

This paper proposes a fault-tolerant communication network of safety sensors on railway lines. The architecture, which has been divided into sections and structured into three hierarchical levels, along with automatic configuration of the elements, facilitates extension and maintenance. The dual bus structure that links the safety nodes and intermediate modules, along with the gateways, enables correct operation to continue, even though faults may exist in the network, thereby increasing system availability and reliability. The intelligent diagnostic system, in addition to detecting faults, makes it possible to update the routing tables for the gateways, thus ensuring that the packets reach an intermediate module quickly. The use of the AI-ESTATE standard makes the test system independent of the reasoner, which facilitates independent maintenance and updating of these elements. The response times in alarm retransmission are bounded because the safety packets have a higher priority level, even when a safety node is sending an image sequence.

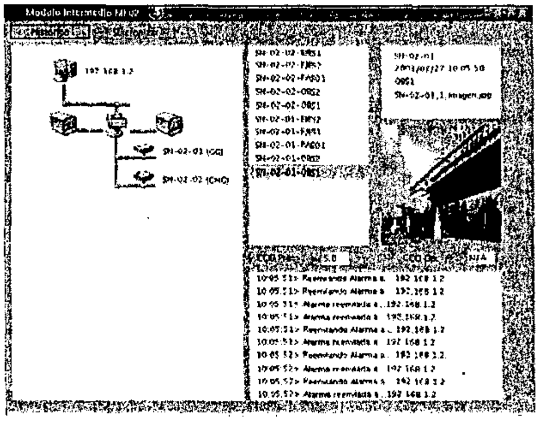

Fig.5 Intermediate Module Graphic User Interface

VII. ACKNOWLEDGEMENTS

The paper has been produced as part of the TELEVÍA project funded by the Ministry of Science and Technology of Spain (Reference COO1999-AX049).

VIII. REFERENCES

[1] P. Martín, C. Mataix, F.J. Rodríguez, "Topología de Red para Supervisión de la Seguridad en Líneas Ferroviarias". *SAAEI'2002 Seminario Anual de Automática Electrónica Industrial e Instrumentación*, 2002, pp. 469-472.

[2] N. Storey, *Safety-Critical Computer Systems*. Prentice-Hall. 1996.

[3] C. Mataix, P. Martín, F.J. Rodríguez, E. Santiso, J.A. Jiménez, "Aplicación de Java™ en Tiempo Real a la Telesupervisión Reconfigurable en Entornos Ferroviarios". *TELEC'2002*, 2002.

[4] K.P. Birman, *Building Secure and Reliable Network Applications*. Department of Computer Science. Cornell University. 1995.

[5] J. Brunton, G. Digby, A. Doherty, "Network Management System Architectures for a Railway Environment". *IEE Colloqium on Network management System Architecture*. 1996.

[6] F.B. Zhou, M.D. Duta, M.P. Henry, "Remote Condition Monitoring for Railway Point Machine". *Proceedings of the 2002 ASME/IEEE Joint Rail Conference*. Washington DC. 2002.

[7] L.F. Myers, M. Lovette, C.C. Kilgus, J.A. Giannini, D. C. Swanson, "A Java-Based Information System for Wayside Sensing and Control". *Proceedings of the 1998 ASME/IEEE Joint Rail Road Conference*. 1998.

[8] T.E. Dy-Liacco, "Modern Control Centers and Computer Networking". *IEEE Computer Application in Power*, nº 10, 1994, pp. 17-22.

[9] IEEE Std 1232-1995, *IEEE Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE): Overview and Architecture*, Piscataway, NJ: IEEE Standard Press, 1995.

[10] IEEE Std 1232.1-1997, *IEEE Trail-Use Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE): Data and Knowledge Specification*, Piscataway, NJ: IEEE Standard Press, 1997.

[11] IEEE Std 1232.2-1998, *IEEE Trial-Use Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE): Service Specification*, Piscataway, NJ: IEEE Standard Press, 1998.

[12] IEEE Std 1232-2002, IEEE Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE), Piscataway, NJ: IEEE Standard Press, 2002.