Fault-Tolerant Communication System to Improve Safety in Railway Environments

César Mataix, Pedro Martín, Francisco Javier Rodríguez, María José Manzano, and Javier Pozo

Departamento de Electrónica, Universidad de Alcalá, Campus Universitario, s/n,
28805 Alcalá de Henares, Madrid, Spain
Tel.: +34 91 885 65 50. Fax: +34 91 885 65 91
{mataix, martin, fjrs}@depeca.uah.es

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.

1 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]. At the present time, in most of the railway lines the intermediate stations monitors the state of the systems located near them. It doesn't exist any central system that allows to obtain a global vision of the state of the railway line. The exception is the high-speed lines. Also, the employment of telecontrol systems is scarce. 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

S. Anderson et al. (Eds.): SAFECOMP 2003, LNCS 2788, pp. 263–274, 2003.

temperature, etc. Depending on the circumstance that needs to be detected, the sensors may be one of the following types:

- Ultrasonic sensors: based on multiple transducers and relatively high emission powers, these may be used to warn of the presence of obstacles in the monitored zones.
- Infrared sensors: by emitting structured light and using CCD sensors, these are able to detect the presence of obstacles, even in very poor light conditions.
- Machine vision sensors: intelligent analysis of images in outside environments makes it possible to ascertain whether or not the track is free of obstacles.
- Axle detection and temperature measurement sensors: installed alongside the track, these detect the presence of a train and the number of axles of the same. Moreover, these measure the temperature of all of the axles, brake discs and wheels of any train that passes over them.

Each of these has to be connected to the Operations and Control Centre (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 kilometres is likely to be high, the problem arises of establishing communication in a practical, safe and reliable manner [4]. This paper presents a new fault-tolerant communication system 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. Section 2 describes previous works on the railway environment. Section 3 describes the hardware architecture for the communication network. Section 4 presents the intelligent diagnosis system used for fault detection. Section 5 presents the results and the conclusions of the paper.

2 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 centre is established in [5] via a dual fibre 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 communications 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. In [9], the monitoring network for the CERN energy system is described. It is designed around a hybrid architecture that combines a centralised system (SCADA) and a distributed one to facilitate maintainability, extendibility, modularity and configurability. In [10], a communications system is designed for automatic traffic control (ATC). This is a distributed system connected using sections of two independent fibre optic rings. The need to add a reliable fault detection system that activates back-up equipment is commented on.

3 Network Architecture

The communication network has been designed on a modular structure divided into sections (each section existing between two gateways), thereby facilitating maintenance and 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 double fieldbus that interconnects them, covering a distance of many kilometres. 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. The 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.

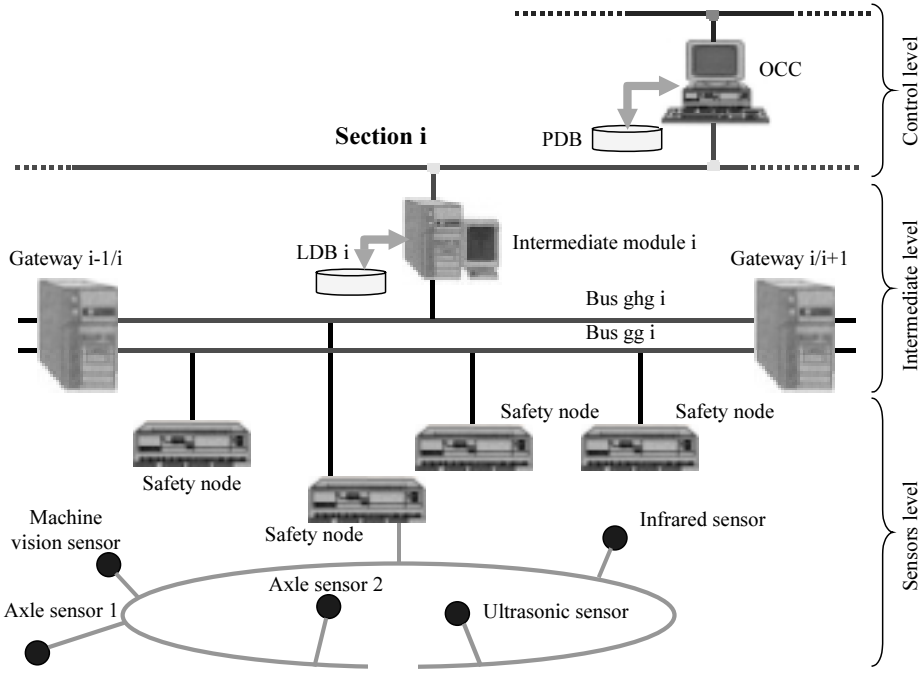

Fig. 1. Network Architecture

Packet routing in the gateways is based on the data supplied by the intelligent diagnostic system. By periodically sending monitoring packets, it is possible to ascertain the operational status of each intermediate module and gateway, as well as the delay of each of the possible routes to the nearest intermediate module. Analysis of this information will determine the routing tables to send to each gateway, thereby achieving faster response times.

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.

To facilitate network configuration and maintenance, each of the buses on the route has been assigned an IP address on a different network and the nodes connected to the same will have an address on this same network. Configuration of the nodes connected to the buses is performed automatically by multicast. Multicast packets are sent periodically over each bus. The nodes respond and identify themselves, and in this way the network configuration is known at all times.

As it is a distributed system, with a local time in each of the digital systems, the problem of clock synchronisation arises. This is dealt with by using an NTP server located on the control level [11].

4 Intelligent Fault Detection

The critical safety network designed, made up of the sensors, the communication system and the Operations and Control Centres, 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, etc., 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 kilometres, it seems obvious that bus 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 of the communication system, the network probe was implemented as a static task in each of the safety nodes, gateways and intermediate modules. The results of the probe 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-STATE standard that makes the decision system independent of the test system, and at the same time uses standard data and knowledge models.

4.1 The AI-STATE Standard

In November 2002, the definitive version of the IEEE 1232-2002 standard was published, titled AI-STATE (Artificial Intelligence and Service Tie to All Test Environments) [12], with the aim of providing a reference for the development of artificial intelligence applications in diagnostic systems. This standard unifies and extends a series that was started in 1995 with the release of the first standard, IEEE 1232-1995 [13], which defines the architecture. This was continued with the IEEE 1232.1-1997 standard [14], which defined the data and knowledge models, and IEEE 1232.2-1998 standard [15], which defined the software services for the diagnostic system.

AI-STATE defines a methodology to develop interoperable diagnostic systems, based on open architecture, that can easily include decision systems based on various artificial intelligence techniques and that generate reusable software. The architecture of a diagnostic system compatible with the standard is shown in Fig. 2.

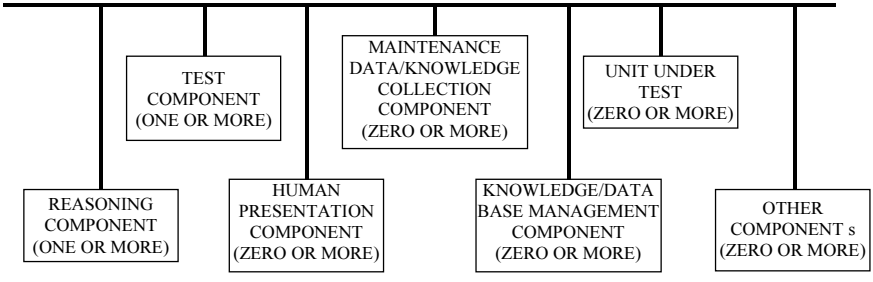

Fig. 2. AI-STATE Architecture

An AI-STATE application may be made up of any combination of these components, but at least one decision system and one test system must always exist, the other components being optional [16][17]. The reasoner will include the test sequence generators, the maintenance data analysers, the intelligent interfaces and the test programs. Each of the components may be found in different computers connected over a network, or all of them may be executed on the same computer.

The standard defines four types of model for use in diagnostic systems - common element model, fault tree model, diagnostic inference model, enhanced diagnostic inference model and dynamic context model. All of these are defined using the EXPRESS language [18]. The common element model (CEM) defines the basic information entities. The other three models represent data and knowledge specific to

the application, taking as a basis the entities defined in the CEM. The last of these makes it possible to perform model management and operate the reasoner during the diagnostic process.

4.2 Application of the Diagnostic Standard on the Network Architecture

Fault detection in the communication system designed is performed using an intelligent diagnostic system conform with the AI-STATE standard. An additional computer has been connected on the control level of the network that implements the architecture shown in Fig. 3.

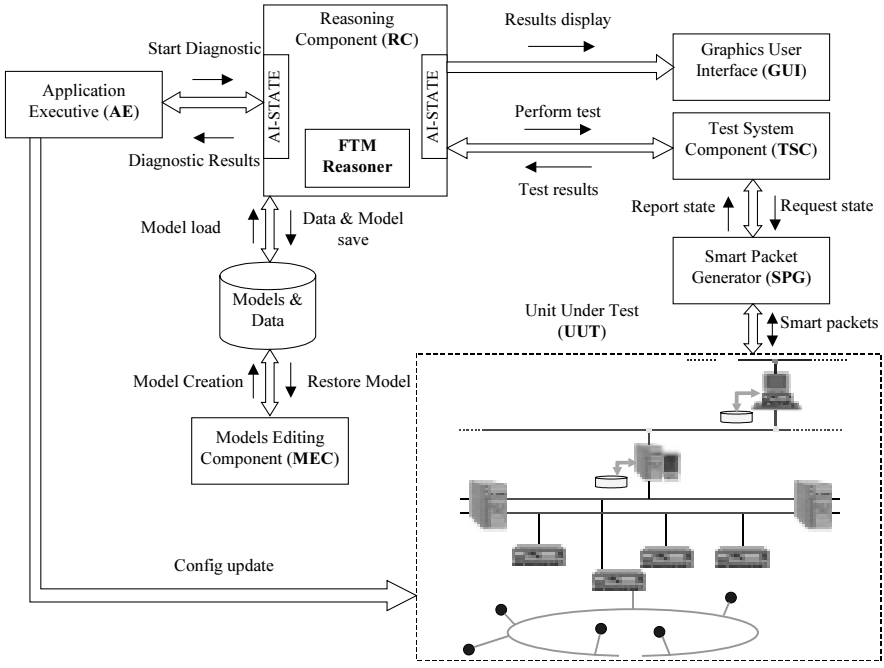

Fig. 3. Implementation of the fault detection system

The *Application Executive* (AE) sends a diagnostic execution request at 5-second 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 elements of the communication system is performed by executing a test task that checks the execution status of the operation tasks and returns the result of the test to the 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 [19][20][21]. 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 task test should be executed. The result of execution of this task is returned to the SPG. The information returned will 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 result of the diagnosis 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 by the operator.

5 Results Obtained

Initially, a prototype of the network was implemented in the laboratory without intelligent diagnosis. The safety nodes, intermediate modules and gateways were implemented in VxWorks 5.4 on a Pentium IV. The test task code was stored statically in each element. These tasks, as well as the operation tasks, were programmed in C. A test network was designed in a laboratory using an ethernet for the data network and setting up a section made up of two gateways, two safety nodes, two intermediate modules and an OCC (Fig. 4).

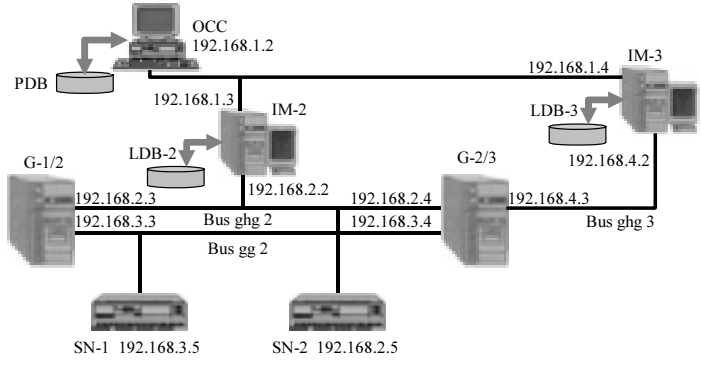

Fig. 4. Communication system employed in the trials

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

One of the drawbacks presented by the initial development was that the status of the buses was not known *a priori* and, therefore, it was not known when the packets were not reaching the OCC. In this case, a certain amount of time was allowed for the packet acknowledgement to be received and, if this did not occur, the packet was sent via a different route, with the subsequent delay in the alarm's arrival.

In order to test the response of the communications network, several response time measurement tests were performed. Some of the results may be observed in Table 1.

Table 1. Delay measurements taken in the initial tests

ORIGIN	DESTINATION	TYPE	DELAY
SN-2	IM-2	Alarm packet	170 ms
SN-2	IM-2	Historical packet 10Kb	1300 ms
SN-2	IM-2	Image packet 64 Kb	1900 ms
SN-2	IM-2	Alarm packet	340 ms
SN-2	IM-3	Historical packet 10Kb	2600 ms
SN-2	IM-3	Image packet 64 Kb	3800 ms
SN-1	G-2/3	Alarm packet	170 ms
SN-1	G-2/3	Historical packet 10Kb	1300 ms
SN-1	G-2/3	Image packet 64 Kb	1900 ms
SN-1	G-2/3	Integrity packets	170 ms
G-2/3	IM-2	Alarm packet	170 ms
G-2/3	IM-2	Historical packet 10Kb	1300 ms
G-2/3	IM-2	Image packet 64 Kb	1900 ms
G-2/3	IM-2	Integrity packets 10 Kb	1300 ms
IM-3	G-2/3	Integrity packets	170 ms

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 256 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 be queried by the intermediate modules.

The OCC was composed of a PC running the Microsoft Windows™ 2000 server operating system and an application written in Java. The application supplies the operator with three types of information:

– 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 (Fig. 5).

– 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.
– Routing table: enabling the routing tables for the gateways to be observed. Their function is designed to facilitate maintenance.

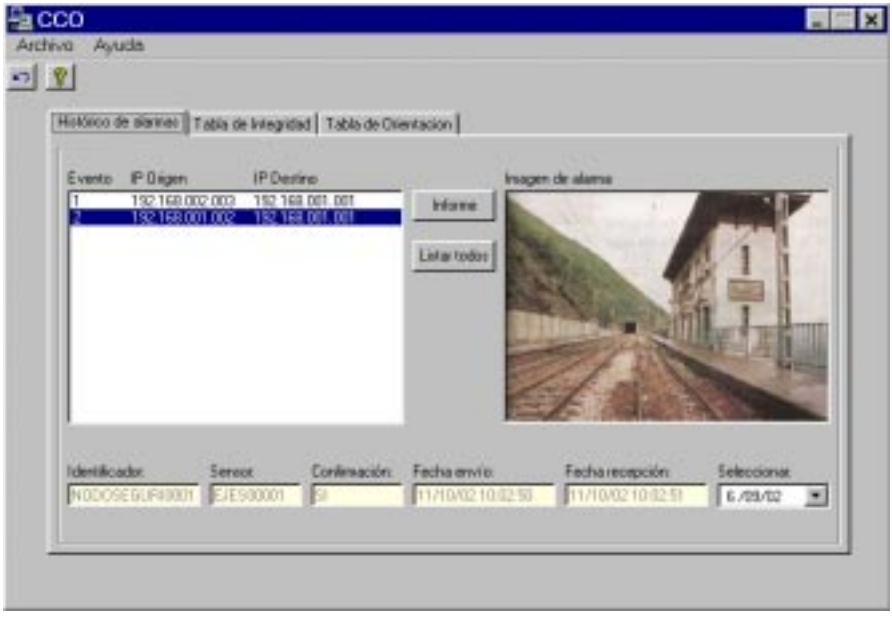

Fig. 5. OCC application window. Alarm history showing the information originating from a obstacle detection sensor in a station

The fault detection system implemented in this preliminary version (based on the integrity packets) was basic. In order to increase the reliability of the system, intelligent diagnosis was added, applying the AI-STATE standard. It was necessary to replace the operating system with Linux, as the execution environments for active networks, required for the test tasks, are implemented on this operating system. By coding the operation tasks on the new operating system, similar response times to those shown in Table 1 were obtained. Currently, fine-tuning of the prototype of the diagnostic system is underway, and implementation of the services necessary for the models employed in the reasoner is being concluded. Once concluded, the safety case wil will be made.

6 Conclusions

This paper proposes a fault-tolerant communication system for a 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 as quickly as possible. The use of the AI-STATE 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.

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).

References

1 Martín, P., Mataix, C., Rodríguez, F.J.: Topología de Red para Supervisión de la Seguridad en Líneas Ferroviarias. SAAEI'2002 Seminario Anual de Automática Electrónica Industrial e Instrumentación. (2002) 469–472
2 Storey, N.: Safety-Critical Computer Systems. Prentice-Hall (1996)
3 Mataix, C., Martín, P., Rodríguez, F.J., Santiso, E., Jiménez, J.A.: Aplicación de Java™ en Tiempo Real a la Telesupervisión Reconfigurable en Entornos Ferroviarios. TELEC'2002 (2002)
4 Birman, K. P.: Building Secure and Reliable Network Applications. Department of Computer Science. Cornell University (1995)
5 Brunton, J., Digby, G., Doherty, A.: Network Management System Architectures for a Railway Environment. IEE Colloqium on Network Management System Architecture. 1996.
6 Zhou, F.B., Duta, M.D., Henry, M.P.: Remote Condition Monitoring for Railway Point Machine. Proceedings of the 2002 ASME/IEEE Joint Rail Conference. Washington DC. (2002)
7 Myers, L.F., Lovette, M., Kilgus, C.C., Giannini, J.A., Swanson, D.C.: A Java-Based Information System for Wayside Sensing and Control. Proceedings of the 1998 ASME/IEEE Joint Rail Road Conference. (1998)
8 Dy-Liacco, T.E.: Modern Control Centers and Computer Networking. IEEE Computer Application in Power, nº 10. (1994) 17–22
9 Roldan, M.C.S.-C., Alonso-Betanzos, A., Arias-Rodriguez, J.E.: Developing an electrical distribution monitoring system. IEEE Computer Applications in Power, Vol: 10 Issue: 1. (1997) 36–41
10 Matsumoto, M., Kitamura, S., Sato, M.: High assurance technologies for autonomous decentralized train control system. High Assurance Systems Engineering, Sixth IEEE International Symposium on (2001) 220–227
11 Mills, D.L.: Network Time Protocol (Version 3): Specification, Implementation and Analysis. Network Working Group Report RFC-1305. University of Delaware. (1992)

12 IEEE Std 1232–1995: IEEE Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-STATE): Overview and Architecture, Piscataway, NJ: IEEE Standard Press (1995)
13 IEEE Std 1232.1–1997: IEEE Trail-Use Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-STATE): Data and Knowledge Specification, Piscataway, NJ: IEEE Standard Press (1997)
14 IEEE Std 1232.2–1998: IEEE Trial-Use Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-STATE): Service Specification, Piscataway, NJ: IEEE Standard Press (1998)
15 IEEE Std 1232–2002: IEEE Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-STATE), Piscataway, NJ: IEEE Standard Press (2002)
16 Sheppard, J., Kaufman, M.: AI-ESTATE-the next generation. AUTOTESTCON '99. IEEE Systems Readiness Technology Conference. (1999) 11–18
17 Sheppard, J., Kaufman, M.: IEEE test and diagnostics standards. Digital Avionics Systems Conferences, 2000. Proceedings. DASC. The 19th , Volume: 2 (2000) 6B1/1 - 6B1/8
18 ISO 10303-11: Industrial Automatic Systems – Product Data Representation and Exchange Part 11: EXPRESS Language Reference Manual. (1992)
19 Tennenhouse, D.L., Wetherall, D.J.: Towards an Active Network Architecture. Proceedings of the DARPA Active Networks Conference and Exposition (DANCE'02) (2002) 2–15
20 Calvert, K.L., Bhattacharjee, S., Zegura, E., Sterbenz, J.: Directions in Active Networks. IEEE Communications Magazine , Vol 36 Issue 10, (1998) 72–78
21 Branden, R., Lindell, B., Berson, S. Faber, T.: The ASP EE: An Active Network Execution Environment. Proceedings of the DARPA Active Networks Conference and Exposition (DANCE'02) (2002) 238–254