THE DEVELOPMENT OF A TEST-BED FOR PERFORMANCE MEASUREMENT OF ETHERNET BASED FIELDBUSES

C. A. Germond * G. J. Gibbon **

* *Postgraduate student at the School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg, South Africa.*
** *Associate professor at the School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg, South Africa.*

Abstract: With the advances in microprocessors and computer networks, fieldbuses are becoming evermore popular. This paper describes the design aspects of a test-bed that allows the performance of Ethernet based fieldbuses to be tested. The test-bed introduces impairments to a fieldbus at the network level. These impairments and the test-bed implementation are outlined in the paper. The results from testing the test-bed, on both a simulated and physical application, are also presented. These results demonstrate the test-bed's ability to successfully add impairments to Ethernet based fieldbuses.

Keywords: Smart Transducer, Fieldbus, Ethernet, Test-bed, Impairments

1. INTRODUCTION

This paper outlines the design and implementation of a test-bed that allows the performance of Ethernet based fieldbuses to be gauged. Before looking at the test-bed implementation and results from testing its functionality, an overview of fieldbuses and a look at Ethernet as a fieldbus are presented.

1.1 Fieldbuses

Advances in computer networks and microprocessors have yielded a special group of transducers, known as *smart transducers*. Smart transducers provide functionality over and above that of dumb transducers (Schaevitz Sensors, 2001) & (IEEE 1451.2, 1997*a*), which typically simplify the integration into a networked environment (IEEE 1451.2, 1997*b*).

The networking of smart transducers has become a very economical and attractive solution for a wide range of measurement and control applications (Lee, 1999). These networks are known as *industrial networks* or *fieldbuses*.

Fieldbuses offer a number of benefits over traditional analogue systems. These include: reduced wiring costs and complexity (Sink, 2001*a*); increased system modularity (Sink, 2001*a*); potential for powerful diagnostics (Sink, 2001*b*); device self-configuration (Sink, 2001*a*); and the ability for effective future expansion (Pinto, 2000).

On the down side however, fieldbuses also have a number of disadvantages. These disadvantages are far less documented than the easily marketable advantages and include: limited interoperability between fieldbuses (Warrior, 1998); a lack of standardisation (Lee, 1999); and a lack of exposure and expertise.

1.2 *Ethernet as a Fieldbus*

Ethernet is the least expensive, high speed LAN alternative currently available (Gilbert, 1995). It is based on the IEEE 802.3 standard and has gained much support within computing sectors since its initial development (Antonakos, 2002). But is Ethernet suitable as fieldbus, where timing can often be critical?

Ethernet is a non-deterministic protocol (Danner, 2001). This means the delays on the network due to congestion and other factors are not able to be exactly determined, but rather estimated in terms of probabilities. Ethernet cannot ensure a message will arrive at its destination within a desired time period and is hence described as a best-effort network (Hanrahan, 2001). Non-deterministic protocols are particularly unsuitable for industrial applications, especially control systems, where often consistent sample times are required.

This would imply that Ethernet is unsuitable for use as an industrial network. The benefits offered by Ethernet, however, have led to many vendors developing Ethernet based industrial networks. These vendors justify the use of Ethernet with the following strategy (Moxa Technologies, 2002) & (SMS, Inc., 1999). All fieldbuses (and any communication protocol for that matter) have a certain propagation delay. With any best-effort network, such as Ethernet, an increase in traffic increases the propagation delay, due to collisions and congestion. This means if the traffic is kept low enough, the propagation delay will remain below some acceptable level and the network may be seen as deterministic.

Currently, the more popular fieldbuses implemented over TCP/IP and Ethernet are: EtherNet/IP; PROFInet; Modbus/TCP; and Foundation Fieldbus HSE (Sink, 2001*b*). Ethernet has become attractive as a fieldbus due to the cost effectiveness and availability of equipment as well as the fact that it has successfully established itself in many corporate networks.

1.3 *Test-bed outline*

The purpose of the test-bed, outlined in this paper, is to provide a tool that allows the performance of Ethernet based fieldbuses to be gauged. This is performed by introducing impairments at the network level. In short these impairments are network and application related. The test-bed is not an attempt to solve protocol specific problems, but rather to investigate the effect these problems have on a specific network and its total performance.

The test-bed has two main possible applications. The first is as a network performance analyser. The test-bed allows the user to visually see the effect of different impairments on their fieldbus. This allows any necessary changes to be made to the fieldbus to ensure a desired performance level. This could include changing a network device, the network configuration or a controller design depending on the effect added impairments have on the network. And secondly, the test-bed could be used as a teaching aid. Using a suitable application, students would clearly be able to see how different impairments effect measurement and control systems.

2. IMPAIRMENTS

This section briefly describes impairments that can effect the performance of fieldbuses. These impairments are grouped into three sections, namely packet delaying, packet manipulation and application incurred. The test-bed attempts to simulate the effect of these impairments, as will be discussed in section 3.1.

2.1 *Packet delaying*

The packet delaying impairments effect the time a single packet takes to reach its destination.

2.1.1. Latency Latency is the time it takes a data packet to travel from one designated end point to another (Bennett, 1995). Ideally, data is be transmitted instantly between the points, i.e.: there is no delay between the time it is transmitted and the time it is received. In practice however there are propagation, transmission, en route, processing and storage delays that all contribute to the latency.

2.1.2. Jitter Jitter is defined as the deviation from the ideal timing of an event (Fibre-Channel, 1999). In terms of a network link, it is the random changes in time it takes a packet to travel from one designated end point to another. Jitter would typically be caused by other network traffic, which is continuously changing.

2.2 *Packet manipulation*

The packet manipulation impairments are concerned with the packet themselves and not their contents.

2.2.1. Packet Dropping Occasionally the situation arises where a packet is dropped from the stream. For example, on a busy network a switch's input buffer could overflow causing packets to be lost. Another example would be if an intermediate node (such as a bridge, router or gateway) is unable to process the packets quick enough, resulting in the packets being lost.

2.2.2. Packet Disordering Packet disordering can occur when there are paths to a destination with varying bandwidths. For example, packets of a TCP connection are arbitrarily transmitted partially over a low bandwidth terrestrial path and partially over a high bandwidth satellite path. The packets reached at the destination will not be in the same order as they were transmitted.

2.2.3. Packet Duplication Like packet dropping, occasionally packet duplication can occur. For example, when a link goes down after the receiver correctly receives a packet but before the transmitter receives the acknowledgement; the transmitter and receiver will each take responsibility for attempting to deliver the same packet once the link is restored (McKenzie, 1989). A duplicate packet will often not effect the network operation.

2.2.4. Network bit errors In digital systems, occasionally bits are read in error. These are known as bit errors. Possible causes of bit errors include: excessive signal noise and signal degradation (Halsall, 1996). Network bit errors are the result of network itself. Bit error rate (BER), is defined as the ratio of bits that have errors relative to the total number of bits received in a transmission.

2.3 Application incurred impairments

The application incurred impairments are on a different level to the packet delaying and manipulation impairments. They are concerned with the application itself, rather than the network. These impairments could for example be the result of variations in the output of a sensor in a control system.

2.3.1. Noise The noise in a system is caused by mainly by external sources (Bentley, 1995), that include mains pick-up (or hum), power circuit switching and electro-magnetic interference (EMI). Noise is responsible for degrading and often even swamping a signal beyond recognition.

2.3.2. Transducer degradation Wear and ageing are often responsible for changing the characteristics of a transducer (Bentley, 1995). These characteristics include the gain and bias. For example, with time, as a spring in a sensor becomes less stiff, the gain of the sensor decreases.

2.3.3. Transducer Fault/Failure When a transducer fails, its output will often reflect either extremity of its range. A faulty transducers output, on the other handle, will tend to be random.

2.3.4. Hysteresis For any given input to a system, the output may be different depending on whether the input is increasing or decreasing. The hysteresis of the system is defined as the difference of these outputs (Bentley, 1995). Hysteresis could, for example, be caused by mechanical play in the system.

2.3.5. Application bit errors Application bit errors are similar to the network bit errors expect they occur in the application equipment. For example a faulty A/D convertor could cause bit errors.

3. TEST-BED IMPLEMENTATION

3.1 Impairments

The test-bed provides two main levels of impairment generation. These are network related impairments and application related impairments. Both levels require the user to specify parameter values in order to simulate the impairments discussed in section 2.

3.1.1. Network related The network related impairments provide facility to simulate the packet manipulation and packet delaying impairments. These impairments allow the performance of the actual network to be measured. This could for example include acceptable levels of responsiveness or the robustness of the network's configuration. The test-bed is able to introduce these impairments to any Ethernet based fieldbus (or protocol), irrespective of the bus.

The network related impairment parameters are: fixed delays (μs) for latency simulation; a standard deviation for random delays (μs) for jitter simulation; fixed rate for packet dropping and duplication (every n^{th} packet) and the probabilities of a packet being randomly dropped or duplicated for packet dropping and duplication simulation; and a bit error rate for simulating network related bit errors.

3.1.2. Application related The application related impairments provide facility to simulate the application incurred impairments outlined in section 2. These impairments allow the performance of an application, implemented on a fieldbus, to be measured. This could for example include the robustness of a controller design. As this functionality requires the data of each packet to be modified, this level of impairment generation is protocol specific.

The application related impairment parameters are: a maximum signal level that provides a reference for the other parameters; the maximum noise and hysteresis levels as a percentage of the maximum signal; the change in bias as a percentage of the maximum signal; a multiplier for the change in gain; a boolean to select random output data simulating a faulty transducer; and a bit error rate for simulating the application bit errors.

3.2 Hardware

In order for a test-bed to introduce the network and application related impairments, discussed in section 3.1, it will need to be set up as in figure 1. The figure shows, the test-bed will typically split a network into two segments of interest. This set-up allows the resulting effect, of the impairments added to a network by the test-bed, to be seen. For example, *segment a* might contain a controller and *segment b* a sensor. The test-bed, by introducing the necessary impairments to the data transferred between the sensor and the controller, could allow the performance of that section of the network to be tested. The test-bed could then be moved to another region of the network to test between a different two nodes.

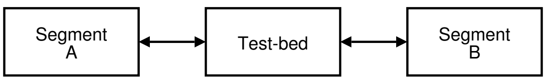

Fig. 1. Test-bed configuration

3.3 Software

Figure 2 shows the flow of the major aspects of the test-bed software. The test-bed is responsible for capturing packets from the interface cards connected to both network segments. Impairments are added to received packets, as selected by the user, and finally the packet is transmitted on the interface card opposite to the one it was received on.

If application related impairments are to be added, the packet's protocol is checked first. If the protocol is correct, the data is modified to simulate the selected impairments. The TCP header checksum is adjusted to correctly reflect the new data. The headers and data are then re-assembled and the new packet is passed onto the next stage.

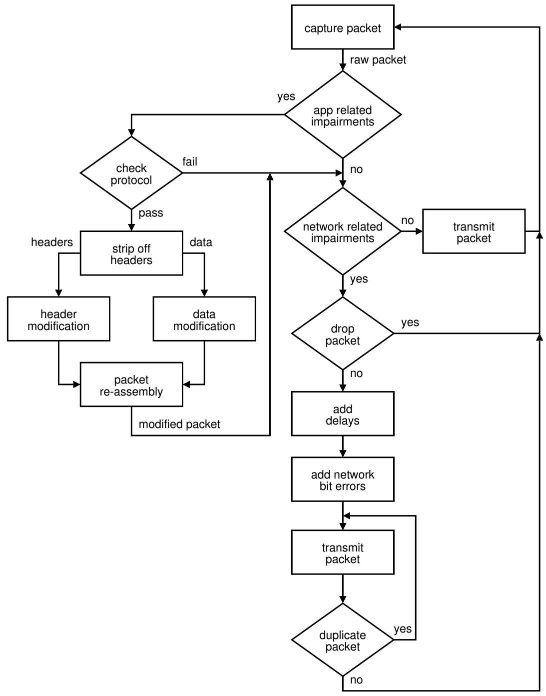

Fig. 2. High level test-bed software flow diagram

Were the protocol check to fail the packet is simply passed onto the next stage.

The next stage involves the generation of the network related impairments. If no impairments are selected the packet is transmitted and the test-bed returns to capture the next packet. The selected impairments are added in the order shown in the figure. If the packet is to be dropped, the test-bed simply returns to capturing the next packet. This stage ends off by transmitting the packet, re-transmitting the packet if required and finally returning to capture the next packet.

4. DEMONSTRATION OF THE TEST-BED

In order to demonstrate the operation of the test-bed, three sets of tests were performed. The first test demonstrates the ability of the test-bed to add network related impairments to any Ethernet based protocol. The second and third tests demonstrate both network and application related impairments using simulated and physical applications, respectively. Only first and third tests are discussed in this paper.

4.1 Testing of network related impairments

For this test, the test-bed is set-up as shown in figure 3. The test involves station 1 attempting to transmit a packet to station 2 (or station 3), which in turn sends a reply. The test-bed intercepts the packets and adds the selected network related impairments. The packets are generated using a well known program called *ping*. Although ping is not a fieldbus, it effectively demonstrates the test-bed's ability to introduce impairments on any Ethernet based protocol [1].

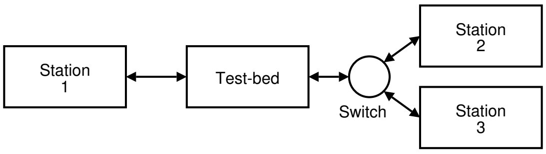

Fig. 3. Test-bed set-up

Table 1 shows the results from adding three network related impairments to 64 byte ping packets. For the packet dropping test 200 ping packets were transmitted and 1000 packets for the duplication test. The fixed delay results show an extra delay of about 3 *ms*. This is due to the time taken processing the packet by the test-bed and the transmission delay of the packet en route. *Note:* As selection of packets to be dropped or duplicated is a random process, an increase in the number of transmitted packets will cause the results to move closer to the selected values.

Table 1. Network related impairments

Selected delay (ms)	0	200	1000
Resulting delay (ms)	2.714	202.747	1003.231
Selected drop %	0	10	50
Packets dropped	0	21	98
Result drop %	0	10.5	49
Selected duplication %	0	10	50
Packets duplicated	0	99	524
Result duplication %	0	9.9	52.4

4.2 Testing of network and application related impairments

In order to test the application related impairments an application and fieldbus are required. Both the simulated and physical applications (discussed below) are implemented on a simple fieldbus, developed especially for the testing. This fieldbus provides a simple means of transporting data between nodes of a measurement or control network. It makes use of the encapsulation protocol defined by the EtherNet/IP Adaptation of CIP specification (described in (ODVA, 2001)) to wrap its own packets on a TCP/IP network. The encapsulation commands as well its methodology for session management, as outlined in the specification, are utilised. It should be noted, the fieldbus used here could have been EtherNet/IP; PROFInet; Modbus/TCP; or Foundation Fieldbus HSE.

4.2.1. Physical control application The physical application used for testing was a simple control system. This system comprises a tank being filled with water at an arbitrary rate; a pressure sensor attached to the tank that allows the tank level to be determined; and an outlet controlled by an actuator value. The system is shown in figure 4. The flow from the outlet is regulated so that the level in the tank remains constant.

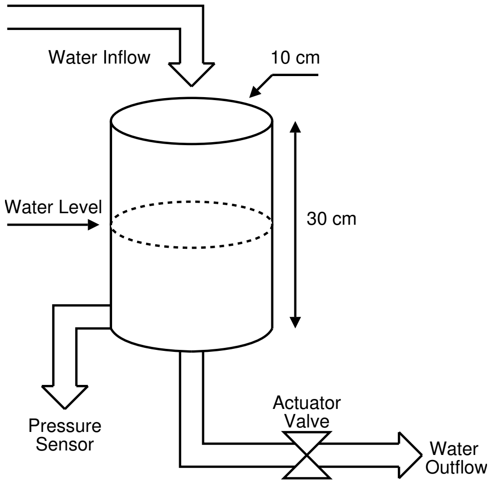

Fig. 4. Control system application

The test-bed was again set-up as in figure 3, where station 1 was the controller and stations 2 and 3 were the pressure sensor and valve respectively. In normal operation the water level oscillates about the controller set point, resulting in an average sensor reading of 3.12 V, a reading deviation of 0.05 V and an oscillation period of 17.39 s. As the water inflow is kept constant throughout the testing, these values provide a reference that enables the effect of impairments on the system to be quantified.

Table 2. Application related impairments

Selected bias	5%	20%	-5%
Average level (V)	2.87	2.11	3.36
Level deviation (V)	0.05	0.05	0.05
Period (s)	18.39	18.61	18.28
Selected noise	5%	10%	20%
Average level (V)	3.14	3.16	3.22
Level deviation (V)	0.07	0.07	0.11
Period (s)	24.87	33.22	43.19

Table 3. Network related impairments

Selected delay (ms)	100	250	1000
Average level (V)	3.14	3.16	3.14
Level deviation (V)	0.09	0.12	0.22
Period (s)	24.18	29.97	55.21
Selected drop chance	1%	5%	10%
Average level (V)	3.12	3.12	3.12
Level deviation (V)	0.06	0.07	0.10
Period (s)	19.06	20.07	23.37

Tables 2 and 3, list a portion of the application and network related impairments tested on the system. For each impairment, the effect on the

[1] Ping is based on Internet Control Message Protocol (ICMP), which is an extension to IP.

system is noted. For example, changing the selected bias, as shown in table 2, causes the average water level to change. Table 3 shows, as the packet delay increases, so to does both the level deviation and oscillation period. This would be expected, since the longer it takes the controller to adjust the valve, the more the water is able to overshoot the set point. In a similar manner the tested impairments, not listed here, also exhibit predictable effects on the system.

5. CONCLUSION

The results given in this paper are from only a portion of the tests performed to verify the test-bed operation. The tests performed on the test-bed demonstrate the ability to add network related impairments to any Ethernet based fieldbus, as well as application related impairments to fieldbuses implemented on the test-bed. Using the physical application, such as the one described in this paper, the test-bed is able to be used effectively as an educational tool.

5.1 Further development

In order to improve the test-bed's functionality as a fieldbus analyser, the test-bed stack needs to be extended to include the more popular Ethernet based fieldbuses, mentioned previously. Through testing, default values for the network related impairment parameters can be obtained. These values would allow the test-bed to accurately simulate particular network configurations. For example, if the delays and BER for a satellite link were obtained through testing, the test-bed could accurately simulate such a satellite link.

REFERENCES

Antonakos, J. (2002). *Ethernet Technology. Part 1: Frames, Collisions, And 10 Mbps LANs.* [Online] http://www.circuitcellar.com/chipcenter-pdfs/0302/c0302ts.pdf.

Bennett, G. (1995). *Designing TCP/IP Internetworks.* Van Nostrand Reinhold.

Bentley, J. P. (1995). *Principles of Measurement Systems.* 3rd ed.. Longman.

Danner, P. (2001). *Networked Industrial Environments.* Cisco Industrial Networks. [Online] http://ethernet.industrial-networking.com/articles/i06articlenetworked.asp.

Fibre-Channel (1999). *Methodologies for Jitter Specification. Working Draft. Rev 10.* Jitter Working Group Technical Report.

Gilbert, H. (1995). *Ethernet.* Yale University. [Online] http://www.yale.edu/pclt/COMM/ETHER.HTM.

Halsall, F. (1996). *Data Communications, Computer Networks and Open Systems.* 4th ed.. Addison-Wesley.

Hanrahan, H. E. (2001). *Integrated Digital Communications.* University of the Witwatersrand, Johannesburg.

IEEE 1451.1 (1997*a*). *The IEEE Standard for a Smart Transducer Interface for Sensors and Actuators. Network Capable Application Processor Information Model. IEEE Std 1451.1-1997.*

IEEE 1451.2 (1997*b*). *The IEEE Standard for a Smart Transducer Interface for Sensors and Actuators. Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats. IEEE Std 1451.2-1997.*

Lee, K. B. (1999). *A Synopsis of the IEEE P1451 - Standards for Smart Transducer Commnication.* National Institute of Standards and Technology.

McKenzie, A. (1989). *A Problem with the TCP Big Window Option. RFC 1110.* Networking Working Group.

Moxa Technologies (2002). *What Makes Ethernet Industrial?*. [Online] http://www.moxanet.com/support/white_paper/.

ODVA (2001). *EtherNet/IP Specification. Volume 1: CIP Common Specification. Release 1.0.* ControlNet International and Open DeviceNet Vendor Association (ODVA).

Pinto, J. (2000). *How Do I Catch the Fieldbus?*. [Online] http://www.jimpinto.com/writings/worldbus.html.

Sink, P. (2001*a*). *A Comprehensive Guide To Industrial Networks.* Part 1: Why Use An Embedded Network or Fieldbus, And What Are The Most Popular Standards?. [Online] http://www.sensorsmag.com/articles/0601/28/main.shtml.

Sink, P. (2001*b*). *Eight Open Networks and Industrial Ethernet. A Brief Guide to the Pros annd Cons for Users and OEMs.* Synergetic Micro Systems, Inc.

Schaevitz Sensors (2001). *Smart Transducers for Industrial Applications.* [Online] http://www.msiusa.com/schaevitz/news/smart.html.

Synergetic Micro Systems, Inc. (1999). *Industrial Ethernet.* [Online] http://www.industrialethernet.com/Whtpaper/whtpaper.pdf.

Warrior, J. (1998). *Open Systems: Reality Or Illusion?*. [Online] http://www.sensorsmag.com/articles/0998/da0998/main.shtml.