6-1-2017

# Efficient redundancy in wired and wireless S2A architectures for NCS

Medhat Medhat Toubar

The American University in Cairo

School of Sciences and Engineering

# EFFICIENT REDUNDANCY IN WIRED AND WIRELESS S2A ARCHITECTURES FOR NCS

A Thesis Submitted to

Electronics and Communication Engineering Department

*in partial fulfillment of the requirements for the degree of Master of Science*

by Medhat Medhat Hassan Mahmoud Toubar

Supervisor: Prof. Hassanein H. Amer

May 2017

# DEDICATION

*I would dedicate this thesis to my dear family, my beloved parents, my brother and my fiancé who have been always supporting me all over my life journey. I would also like to dedicate it to all my professors.*

*Mom Aida: I cannot imagine my life without you. You help and support me all the time and I owe you all the success in my life.*

*Dad Mahmoud: You are my role model and without your help and faith, I could not be the man I am today.*

*Dad Medhat: I know you feel my success. I know you are proud of me. Rest in peace*

*Brother Mohamed (Mido): Thank you Mido for all your help and support. I'm sure you will be a successful man.*

*Fiancé Seham (Tota): We have begun the long Master's degree journey together and now we are ending it also together. I would like to thank you for your help and patience. I'm counting days for your graduation and then our wedding. Love you*

*Prof. Hassanein Amer: Thanks a million for all your help and support since I joined the department in 2006. Really, without you I couldn't make it.*

*Moreover, I would like to thank all my friends for their help and support.*

*May Allah grant you all happiness and success in your life.*

# Abstract

This thesis focuses on the integration of wired and wireless nodes running on top of Gigabit Ethernet and WiFi respectively in Networked Control Systems. Such a networked control system investigated in this work consists of two wireless sensors, two wireless actuators, 14 wired sensors, two wired actuators and one wired supervisor. The architecture is based on Sensor-To-Actuator model. It is revealed through OMNeT++ simulations that the wired and wireless packet end-to-end delays in the developed model satisfy system requirements with no packet loss. Moreover, wired, wireless and mixed interferences are studied and quantified. The amount of interference that the model can withstand is determined. All results are subjected to a 95% confidence analysis. Additionally, the thesis focuses on reliability in the design of networked control systems that is becoming greatly important. Fault-tolerance is often used to increase system reliability. In this work, Triple Modular Redundancy (TMR) and Parallel Redundancy Protocol (PRP) are both applied to a Sensor-to-Actuator architecture with 16 sensors, four Actuators and one Supervisor. Two of the 16 sensors as well as two of the four actuators are wireless while the rest of the nodes are wired. It is first shown that this NCS succeeds in meeting all control system requirements (zero packet loss and bounded end-to-end delay). Reliability models are then developed to help designers choose the appropriate mix of fault-tolerant techniques in order to maximize lifetime while at the same time minimizing the extra cost due to the added redundancy.

# ACKNOWLEDGMENTS

Undoubtedly, I owe to my advisers, Prof. Hassanein Amer a great deal. He has greatly supported me under every possible aspect of this long and difficult journey. He has been continuous source of motivation, and inspiration. He has provided guidance, technical criticism, and valuable feedback.

I would also like to thank the other two members of my committee: Dr. Ramez Daoud for his generous assistance and support throughout my work, and Eng. Hassan Halawa who has provided valuable feedback during this work

I would like to acknowledge the graduate program directors: Dr. Ayman El Ezabi and Dr Karim Seddik.

# TABLE OF CONTENTS

# List of Tables

# List of Figures

# List of Abbreviations

| | |
|---|---|
| **AP** | **A**ccess **P**oint |
| **ARQ** | **A**utomatic **R**epeat **R**equest |
| **CAN** | **C**ontroller **A**rea **N**etwork |
| **FT** | **F**ault **T**olerance |
| **FTTE** | **F**lexible **T**riggered **E**thernet |
| **NCS** | **N**etworked **C**ontrol **S**ystems |
| **OSI** | **O**pen **S**ystems **I**nterconnection |
| **PRP** | **P**arallel **R**edundancy **P**rotocol |
| **RCT** | **R**edundancy **C**ontrol **T**railer |
| **RedBox** | **R**edundancy **B**ox |
| **S2A** | **S**ensor-to-**A**ctuator |
| **TCP** | **T**ransmission **C**ontrol **P**rotocol |
| **TMR** | **T**riple **M**odular **R**edundancy |
| **TTE** | **T**riggered **E**thernet |
| **UDP** | **U**ser **D**atagram **P**rotocol |
| **WLAN** | **W**ireless **L**ocal **A**rea **N**etwork |
| **WNCS** | **W**ireless **N**etwork **C**ontrol **S**ystem |

# Chapter 1

# Introduction

## 1.1 Background

There are two major types of networks: data communication networks and control networks. Industrial control has been increasingly moving towards the implementations of control systems. Thus, Networks are now being used in control systems to communicate the data instead of using classical point-to-point communication. In comparison to traditional control networks, point-to-point architecture requires more wiring and more maintenance. Such control networks carry a huge number of small control packets among many nodes, and these packets have to meet the real-time control system's delay constraints. The major difference between conventional data networks and control networks is that control networks must have the ability to support time-critical applications.

In Networked Control Systems (NCS) control and feedback signals are exchanged among the system's components in the form of information packages through a network. Sensor-to-Actuator (S2A) and In-Loop Control are two main types of NCS data flow schemes. An NCS workcell consists of four components which are smart sensor nodes, smart actuators, controllers and the network fabric. A smart node is simply one that has self diagnostic capabilities as well as network connectivity. The traffic that is transmitted through an NCS is fundamentally made of small packets with useful information in order to control the system. The information sequence starts from the sensor nodes which monitor the environment and transmit their readings to the controllers where the control action is computed. After that, control actions are transmitted to actuators that affect the

physical system. In an NCS, all nodes including sensors, actuators and controllers are linked over a network that can be either wired or wireless.

There are four factors that influence the utilization of the network bandwidth: the sampling rate, the size of the information sent, the number of nodes requiring synchronous operation and the protocol used. Either event-triggered or time-triggered (clock-driven) are two types of NCS systems. A time-triggered system is made of sensors and actuators with fixed sampling time where samples are captured at discrete time points. On the other side, an event-triggered system includes continuous sampling and an event that triggers the control process. The time taken by a packet to pass from a sensor to a controller and from a controller to an actuator respectively is considered to be the total end-to-end delay which consists of all types of propagation, encapsulation and de-capsulation and queuing delays.

## 1.2 Contribution of this Thesis

A direct sensor to actuator NCS architecture using WiFi (wireless) and Ethernet-based (wired) protocols is presented in this thesis. Wired, wireless and mixed interfering nodes are included. Much more, two different fault-tolerant techniques are used, which are Triple Modular Redundancy (TMR) that is applied on wired and wireless nodes and Parallel Redundancy Protocol (PRP) that is applied on wireless nodes only in S2A networked control systems. Finally, eight different fault tolerance scenarios with varied failure rates are analyzed to measure reliability with respect to cost in order to help the user with the choice of the appropriate fault tolerance combination.

## 1.1 Thesis Organization

Chapter 2 summarizes the literature review. First, the definition of networked control systems and its usage is illustrated. Then, transport layer protocols, such as UDP and TCP are studied. Moreover, a data flow analysis of both sensor-to-actuator model and in-loop model are presented. After that, Fast and Gigabit Ethernet in wired connections in addition to wireless connectivity are listed. Much more, reliability as well as fault tolerance techniques such as Triple Modular Redundancy and Parallel

Redundancy Protocol are highlighted. Finally, related work is shown followed by summary of the chapter.

In Chapter 3, a new direct sensor to actuator architecture is developed; it consists of 16 sensors, one supervisor and four actuators. Each sensor communicates with the suitable actuator directly without passing through a controller node. In other words, each actuator is integrated with its own controller. This proposed architecture used wireless and wired connections in shape of WiFi and Ethernet respectively. Wired, wireless and a mix of wired and wireless interfering nodes were excreted on the system developed. It was revealed via OMNeT++ simulations that this model succeeds to meet the required time constraints.

In Chapter 4, a new model is developed in which two fault-tolerance techniques are applied to the proposed system of 16 sensors, one supervisor and four actuators nodes. Thus, TMR at wireless and wired nodes plus PRP on wireless nodes are applied. Some of the nodes will run over Gigabit switched Ethernet, while the rest run wirelessly over Wi-Fi (IEEE 802.11g). The end-to-end delay and packet loss were observed by OMNeT++ simulations, and it is clear that this model succeeds to meet all the required time constraints with no packet loss. Finally, different fault tolerance scenarios are presented putting into consideration reliability and cost then the chapter is concluded.

Finally, Chapter 5 concludes the thesis in addition to mentioning the future work.

# Chapter 2

# Literature Review

This chapter starts by defining the concept of Networked Control Systems and its importance. Then, it is followed by explaining the usage and differences between the transport layer protocols. In addition, a data flow analysis of sensor-to-actuator model and in-loop model are listed. Fast and Gigabit Ethernet in wired connections as well as wireless connectivity are elaborated. Towards the end of this chapter, reliability and fault tolerance techniques such as Triple Modular Redundancy and Parallel Redundancy Protocol are presented. Finally, related work is shown followed by summary of the chapter.

## 2.1 Networked Control Systems

A control system is consisting of one or more devices used to regulate or manage the operation of other devices or systems. Along the years, several strategies of control systems –starting from the simple open-loop control to more complicated algorithms— appeared in literature. The concept of controlling a system remotely introduced by the merge of communication networks and control systems which, in turn, was the advent of what is called *networked control systems* (NCS) [Rachana, 2008] and [Hong, 1998]. NCS can be defined as a typical closed-loop system which shares the feedback link (through a real-time network) with other nodes outside the control system. NCS is characterized basically by the following information:

- Control Input
- Reference Input
- Plant Output

which are exchanged among the system components using a communication network. The control system components are:

- Sensors (S) to collect information
- Controllers (K) to take decision
- Actuators (A) to perform control action
- Communication network to enable exchange of information

Figure 2.1 shows the structure of networked control systems.

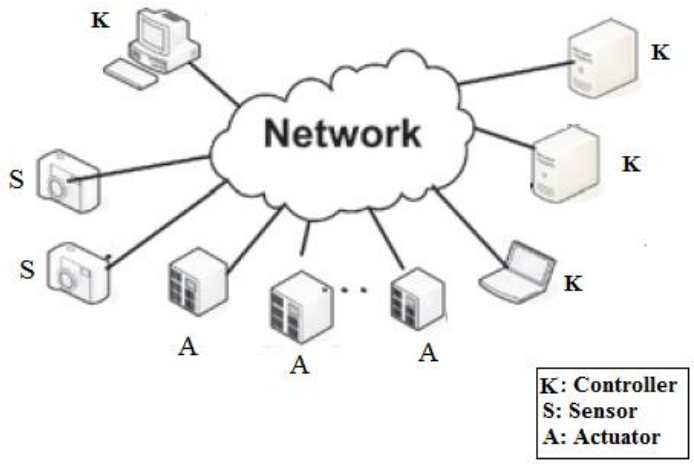



**Figure 2.1: Basic Structure of NCS**

The sequence of information starts from sensors to controllers. Then information is sent to actuators for instantaneous action on the physical system after the control action is taken from controllers [Lian, 2001], [Daoud, 2003] and [Pedreiras, 2002].

Building control systems based on communication networks has many advantages starting from the reduction of complexity of the system interconnection of its components, such as sensors, controllers, and actuators, to the efficiency of sharing data between system components and controllers. Furthermore, using communication networks in control systems allows spreading the information and making smart decisions over the large physical space and eliminates the unnecessary wiring. Moreover, adding more sensor, actuator, or controller will be efficient in terms of cost.

NCS are used widely in several applications such as remote diagnostics and troubleshooting, factory automation, hazardous environments, domestic robots, tele-robotics, tele-operation, automobiles and aircraft.

## 2.2 Transport Layer Protocols

There are four different factors which influence the utilization of the network bandwidth:

- The number of nodes requiring synchronous operation
- The sampling rate
- The size of the data sent
- Protocol used

The number of nodes requiring synchronous operation is the number of system's nodes requiring changing their states and output values at discrete instants of time, which are specified by the rising and falling edge of a free-running clock signal. Moreover, sampling rate is the number of samples per second taken from a continuous signal (analog) to make a discrete signal (discrete). The size of the data sent means the number of bytes of the packet's data exchanged by the system's nodes through the network. The protocols used are under the fourth layer of the OSI model (third in the TCP/IP model) which is the transport layer. The transport layer is responsible for the providing services such as multiplexing, flow control, reliability and connection-oriented data streaming. Two end-to-end protocols have been defined: the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) [Tanenbaum, 2002] and [Pariale, 2006].

The first protocol, TCP, assures the efficient use of bandwidth and no delay guarantees. It is a reliable connection-oriented protocol that allows a data to delivery without any error. It divides the data to segments and passes it to the internet layer, at the receiving end the segments is combined into output stream. TCP flow control mechanism ensures that fast transmission will not overflow the slow receiving end.

The other protocol, UDP, is simple end-to-end protocol. It is connectionless unreliable protocol, i.e., it is not an Automatic Repeat Request (ARQ) protocol. It is widely used for client-server type applications at which prompt delivery is more

important than accurate delivery. UDP is sometimes called *Unreliable* Datagram Protocol as it does not guarantee data delivery or correct ordering.

## 2.3 Data Flow Analysis

Data-flow analysis is a technique for gathering information about the possible set of values. There are two main NCS data flow arrangements, which are In-Loop Control [Nilsson, 1998] and Sensor-to-Actuator (S2A) [Marti, 2001] that are shown in figure 2.2.

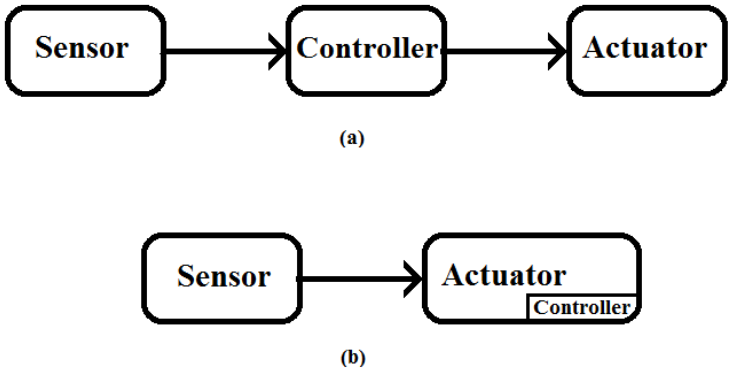



**Figure 2.2: Data-Flow Arrangement: a) In-Loop, b) S2A**

### 2.3.1 In-Loop Control System

In case of In-Loop Control shown in Figure 2.2.a, NCS is made of Sensors, Controllers and Actuators. The Sensors are used to sense information that is encapsulated after it has been converted, and then this information is sent to a main controller. The controller is used to process the collected data and an appropriate control action is computed, encapsulated and finally sent through the network to the actuators. At the actuators, de-capsulation is applied first; and then, the control action that is obtained from the network is excreted on the physical system. In the In-Loop model, the network is involved in the control loop twice: S→K and K→A.

### 2.3.2 Sensor-to-Actuator Control System

S2A is another type of NCS architecture in which the controller is embedded in each actuator [Marti, 2001] as shown in Figure 2.2.b. Unlike the two-hop connection in the In-Loop system, using only one-hop connection from the sensor to the actuator increases the total system performance. Based on an unmodified switched Ethernet, S2A architecture was studied in [Moustafa, 2014]. The study was based on 16-1-4 system at which 16 sensors, one supervisor and four actuators. The performance of the S2A model results in lower control packet end-to-end delays when it was compared with In-Loop control system in both Fast and Gigabit Ethernet cases. This is also true in case of mixed communication traffic including real-time control packets as well as non-real-time communication.

For a properly functioning control system, there are many protocols used with a deterministic behavior, such as DeviceNET and ControlNET [ODVA1, 2007] and [ODVA2, 2007]. Moreover, Controller Area Network (CAN), PROFIBUS and EtherNet/IP, a union between ControlNET and Ethernet, are used to implement many real-time applications [PROFIBUS, 2016], [Bradley, 2001], [ODVA3, 2016], [IEC1, 2014] and [IEC2, 2014].

## 2.4 Ethernet in NCS

Nowadays, Ethernet (IEEE 802.3) [IEEE, 2001] is considered to be a very important wired communication connection that is used in networked control systems because it has been proved that it is a very beneficial protocol. Maintenance cost and installation are reduced in industrial application due to the usage of such a protocol. Two approaches are used to enhance the performance of Ethernet communication in NCS which are either changing the packet format for real-time control words or giving them higher priorities [Lee, 2001] and [Pedreiras, 2002]. In addition, the Ethernet standard had been passed through a lot of modifications to be applied in real-time networks. Examples of these modifications are -Triggered Ethernet (TT Ethernet), Flexible Triggered Ethernet

(FTT Ethernet) and EtherNet/IP Time [Steinhammer, 2007] and [Ferreira, 2002]. Real-time and non-real-time traffic were merged without applying any changes to the Ethernet protocol [Daoud, 2003].

Several functions are enabled when using Ethernet in NCS which were not possible in the previous models. As a result of the machines network connection on the industrial floor is working on the top of Ethernet, they have the opportunity to be connected along with the management and engineering network connections on the management floor that will lead to diminishing the diagnostic and set up problems; thus a lot of functions can be added. This can be happening in online system fix-up and diagnostics by adjusting some parameters with no need to stop the operation of machine when it is working in normal operation mode. Moreover, integrating communication packets, such as downloading and uploading files during performing the traffic of real time control packets can be done easily. The network can tolerate all these tasks that are added to the communication load as overhead to the pure control data [Daoud, 2003].

### 2.4.1 Performance Evaluation of Fast and Gigabit Ethernet in NCS

Several studies were conducted to investigate the performance of both Fast and Gigabit Ethernet in networked control systems.

In [Daoud, 2003], Fast and Gigabit Ethernet were used in NCS. Both real-time and non-real time were integrated. Several loading cases were considered to test the effect of increasing network speed. Two models were built to compare the performance: one model run on Gigabit Ethernet and the other on Fast Ethernet. The first model is called heavy traffic system as it used 48 sensors and 4 actuators while the other model called light traffic system as it used 16 sensors and 4 actuators; both models have one controller. The system used sampling frequency of 1.4 KHz and deadline of 694$\mu$s meaning that the complete control action from sensor, passing through the controller and transmitted over the network to reach the actuator should be taken within 694$\mu$s. The simulation of this system is performed on OPNET. The simulation results show that the

Fast Ethernet failed to meet one of the timing constraint while the Gigabit Ethernet succeeded.

In [Skeie, 2002], a study was done by to test the fast Ethernet in power distribution systems. The result of this study shows that the fast Ethernet was satisfactory in respect to the time frame of the given application. The reason for such success is that the considered application has a relatively large time frame.

## 2.5 WiFi in NCS

Several ongoing applications in industry are majorly based on the wireless local area network (WLAN) connection. WLAN has several advantages over the hard-wired connections in terms of cost effectives, weight, and moving parts. Moreover, the manipulation of wireless connections on the production basis enables totally new definitions for planning and implementation. Although the current technical advances and acceptance of wireless solutions, it is also considered as a highly challenging and increasingly complicated type of connectivity.

Wireless connectivity for networked control systems is also available through wireless interface for sensors and actuators [Steigmann, 2006] and [ABB, 2009]. Unlike the hard-wired systems, the majority of nodes can be connected wirelessly to exchange control and data packets to form wireless networked control system (WNCS). Wireless connectivity has several advantages such as cost reduction, time saving and higher efficiency through removing the risk of cabling threats due to moving nodes [Boggia, 2009], [Pinheiro, 2009] and [Cena, 2009]. Wireless connectivity may include Bluetooth, WiFi [IEEE, 2012], and ZigBee; which are all operated on 2.4 GHz frequency [Refaat, 2010] and [Steigmann, 2006].

In NCS, nodes can run wirelessly over 802.11/g (WiFi) protocol which will add mobility option to the system. The 802.11/g protocol is chosen over the other wireless protocols, as it is the wireless extension of Ethernet.

## 2.6 Fault-Tolerance in NCS

The defect that takes a place in a part of the system is called fault. Networked control systems are developed through a number of phases. Defining specifications, design, prototyping, implementation and installation are the main five phases. Faults perceived in the form of error in the system operation, which in turn lead to failure and the system cannot deliver the required output. Figure 2.3 shows a simple example that illustrates the difference between fault, error and failure.

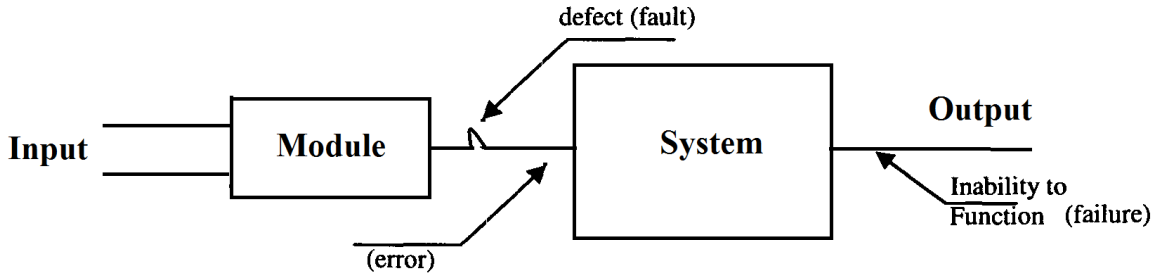



**Figure 2.3: The difference between fault, error, and failure.**

Several techniques can be used to deal with faults starting from fault avoidance through fault masking to fault tolerance (FT) [Abd-El-barr, 2006].

- Design review, testing, shielding are all fall in **Fault avoidance**, which are used to prevent faults from occurrence.
- **Fault masking** refers to the techniques used to prevent faults from introducing errors, e.g. error correcting codes, majority voting, etc.
- A **fault-tolerant** system is "*a system that continues to function correctly in the presence of hardware failures and/or software errors*". Fault detection, location, containment, and recovery are all attributes of typical fault-tolerant system.

Fault-tolerant computing is defined as the correct computing although some errors existence in the system. Essentially, having redundant functions or components in the system are considered to be properties of fault tolerance. An example of a redundant system is a notebook computer and a desktop computer having the same operating system and files. Because both computers have similar functionalities, any of the two computers

can tolerate part of hardware and software failure of the other computer. The advancement of the current digital systems can handle complex fault-tolerance approaches, some of which are as effective as they are sophisticated. Some of these techniques were designed for the analog systems, but digital systems permit implementation of these techniques to be speedier, better, and less expensive. There are other reasons that increase the demand of fault tolerance:

- Novice users
- Larger systems
- Increasing repair costs
- Harsher environments

There are several aspects used in comparing fault-tolerance techniques, such as:

- Reliability
- Availability
- Cost
- Weight
- Volume.

There are two types of redundancy which are hot redundancy and cold or standby redundancy. Hot redundancy means that in case of two systems work simultaneously at which one of them may fail first. On the other hand, in cold redundancy the system has it is own backup. When the backup system power down, it cannot fail until the online system fails, then it is switched on and takes over. Despite the fact it is more sophisticated to deal with synchronization, the cold redundancy is much more reliable than the hot redundancy, as it has less failure probability. Failure detection is considered to be a complex process; however, there is some simple scheme, such as voting system. In such system, a digital comparator (Voter) is used to compare the output of three systems working in parallel choosing the output that agrees with the majority output. In other words, the system succeeds if two or the three systems work properly [Abd-El-barr, 2006].

A more reliable voting system can be made by adding repairing technique for a failed system once a single failure happens. Most of the networks have many paths between nodes; therefore in case of a network path fails, the connection is maintained through other routes (redundancy) and the message is delivered. As a result of the above cases, the redundancy drawback is the presence of extra cost, weight, and volume. Another form of redundancy is to increase the transmission time rather than duplicating the network equipment. Signal can be transmitted two or three times to guard against undetected, corrupting transmission noise [Shooman, 2002].

### 2.6.1 Reliability

Along with performance, cost and the development time, the reliability of the system should always be included. In a given system, the expected number of failures per unit time is defined as a ***failure rate*** $\lambda$, while the probability that this system operates correctly over certain interval is called the ***Reliability*** $R(t)$[Shooman, 2002]. Both the failure rate and the reliability are related by

$$R(t) = e^{-\lambda t}$$

As an example, consider a system of 50 personal computers which are operating for 1000 hours. While testing, two of those computers failed. Therefore, the probability of failure is given by

$$P_f = \frac{2}{50} = 0.04.$$

Clearly, the probability of success is

$$1 - P_f = 0.96$$

which can be calculated by

$$\lambda = \frac{2}{50\text{x}1000} = 4\text{x}10^{-5}$$

$$R(t) = e^{-\lambda t} = e^{-4\text{x}10^{-5}\text{x}1000} = 0.96$$

which agrees with the previous computation.

For a system with $n$ components, all of the components determine the reliability of the overall system. Thus, the system reliability is given by:

$$R_{sys}(t) = \{R(t)\}^n = (e^{-\lambda t})^n = e^{-n\lambda t}$$

As an example, consider the case where a supercomputer of 400000 transistors for which the failure rate $\lambda = 4\text{x}10^{-9}$ failures per hour. Therefore, the reliability for 1000 hours is given by

$$R_{sys}(1{,}1000) = e^{-400000\text{x}4\text{x}10^{-9}\text{x}1000} = 0.2$$

## 2.6.2 Triple Modular Redundancy

The basic modular redundancy circuit is ***triple modular redundancy*** (TMR). The system shown in Figure 2.4 consists of three parallel circuits—A, B, and C—all having the same input. The voter is used to compare the outputs of the three circuits, which sides with the majority and gives the majority opinion as the system output [Shooman, 2002].

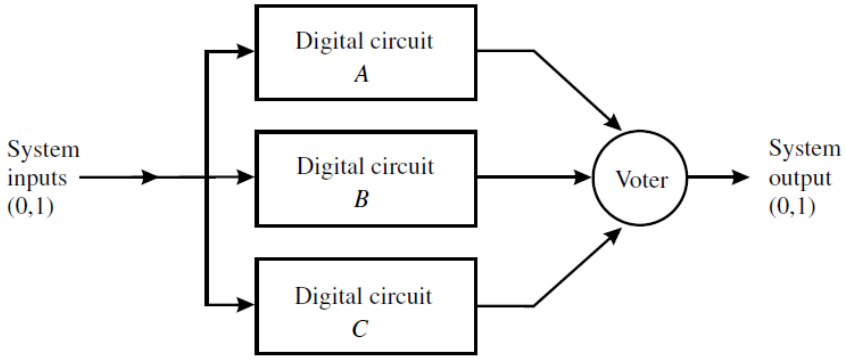



**Figure 2.4: Majority voter TMR (adapted from [Shooman, 2002]).**

The decision will be one of the following:

- Case 1: All three circuits are working correctly, all outputs agree; therefore the system output is correct.

- Case 2: One circuit is working incorrectly so that it has generated an incorrect output, the voter chooses the output of the other two good circuits as the system output; therefore the system output is correct.
- Case 3: Two circuits are working incorrectly, the voter agrees with the majority (the two that have failed); therefore the system output is incorrect.
- Case 4: Three circuits are working incorrectly, the system output is incorrect.

In the cases 1 and 2, where the voter does not fail, the system reliability is given by:

$$R = P(A.B.C) \qquad \text{(Case 1)}$$

$$R = P(A.B + A.C + B.C) \qquad \text{(Case 2)}$$

If all the circuits are identical and independent with $p$ probability of success, in terms of the binomial theorem, the last equation can be rewritten as

$$R = B(3:3) + B(2:3)$$

$$= \binom{3}{3} p^3 (1-p)^0 + \binom{3}{2} p^2 (1-p)^1$$

$$= 3p^2 - 2p^3 \ = p^2(3-2p)$$

This of course represents the reliability expression for a two-out-of-three system. For the systems with N-modular redundancy, the behavior can has different ways in practice. Consider, in more details, how the TMR system works. As mentioned previously, the TMR system functions properly if there are no system failures or one system failure.

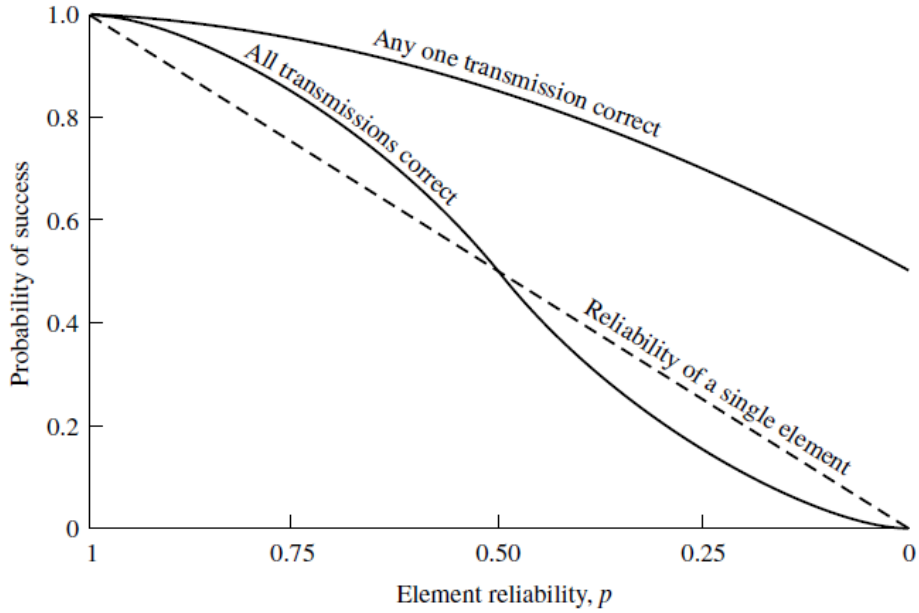



**Figure 2.5: Comparison of probability of successful transmission with the reliability. (adapted from [Shooman, 2002]).**

The reliability was previously expressed in terms of the probability of element success (*p*) that is formulated as

$$R = 3p^2 - 2p^3$$

With the failure rateλ, each component has a reliability of

$$p = e^{-\lambda t},$$

Then by substituting Eq. (4.10) into Eq. (4.9), Reliability becomes

$$R(t) = 3e^{-2\lambda t} + 2e^{-3\lambda t}$$

## 2.6.3 Parallel Redundancy Protocol WLAN

Fault-tolerance enables a system to keep working properly in case of the failure of any of its components. Fault-tolerant systems should have no single point of failure. Thus in order to achieve that, communication paths or redundant components are involved

The main idea of the diversity in communications technology is redundant transmission of data through different independent channels that only at which the probability of error is very small at the same time frame [Nilsson, 1998].

A Parallel Redundancy Protocol (PRP) network is made of two independent LANs (LAN X and LAN Y). Each PRP node consists of two Ethernet interfaces connected to one of the LANs and simultaneously transmits information through the two interfaces into both networks. This communication adds to each frame four octets Redundancy Control Trailer (RTC) having the same sequence numbers and increased for each next frame sent. The first received frame is accepted by the PRP receiver node, while the second is discarded. One of the duplicated frames constantly reaches the receiver provided that one of the two LANs is working. To take the advantage of the PRP redundancy capability, non-PRP nodes must be connected via a Redundancy Box (RedBox).

The PRP RedBox can be modeled as a post-detection selection combiner at the receiver [Lian, 2001], where the better signal out of the two branches is selected and processed, which is considered as the first arriving Ethernet packet. The second arriving packet is discarded. Therefore, this type of combiner is called “timing combiner”.

To create 1-out-of-2 system, transmitted traffic is duplicated and transferred over Ethernet level. Figure 2.6shows an example of two networks forming a parallel redundant network applying PRP as splitter and combiner.

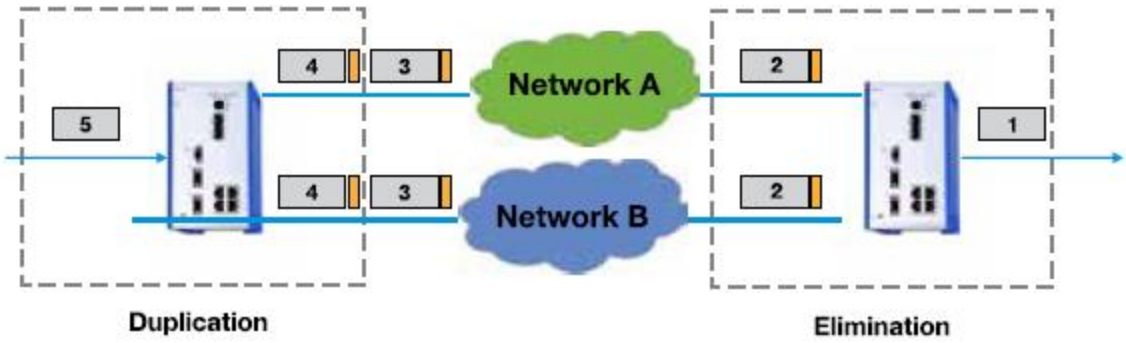



**Figure 2.6: PRP: two redundant paths are used simultaneously. (adapted from [Heer, 2015]).**

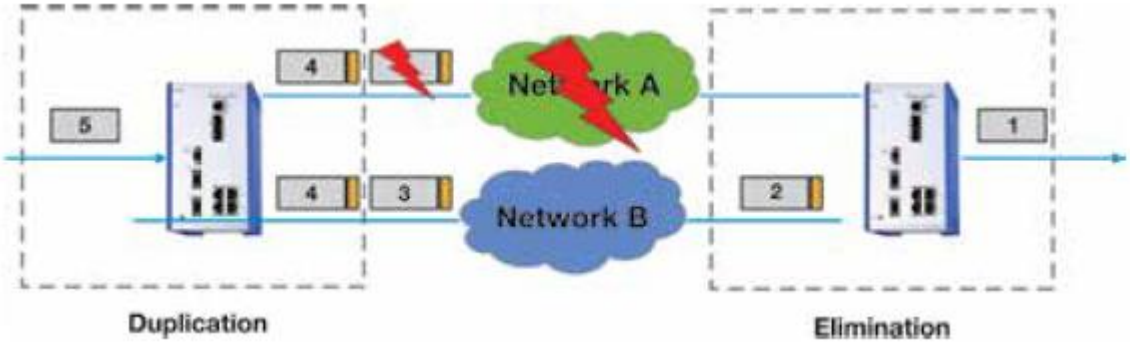



**Figure 2.7: PRP in case of a network failure. Packets from the second network path are used (adapted from [Heer, 2015]).**

PRP can also be used in a wireless environment, although the impact manifests itself in a completely different and even more beneficial way from the wired scenario. The reason is that the parallel redundancy can be used to compensate for the inherent small-scale disruptions (e.g. interference) in a wireless network as well as for total network disruptions. The effect of packet loss in two different wireless transmission paths (Figure2.8) can be eliminated when PRP transmits packets simultaneously.

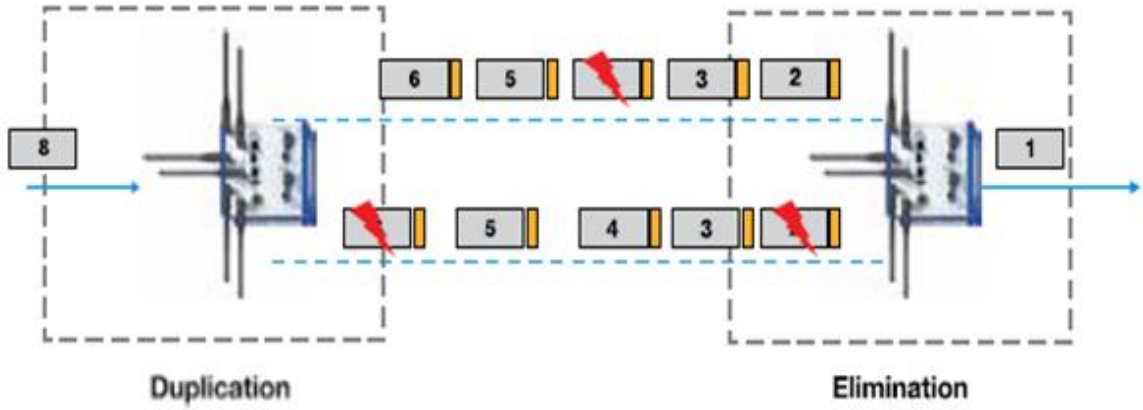
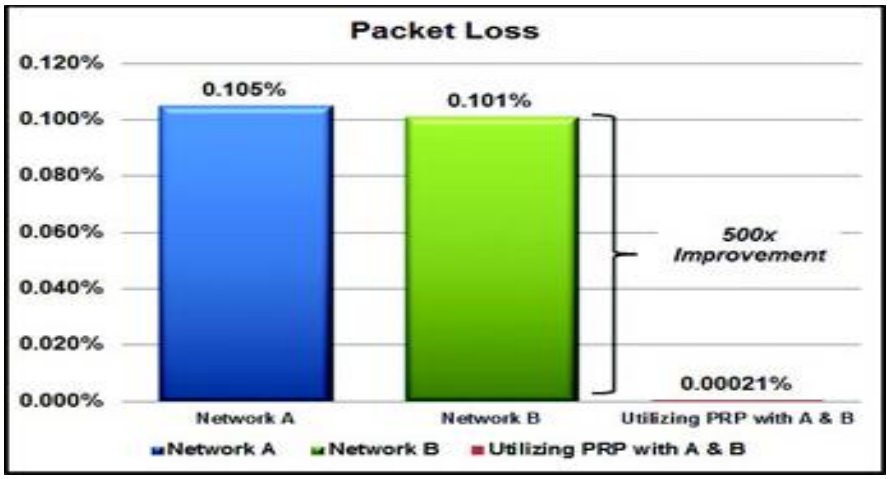


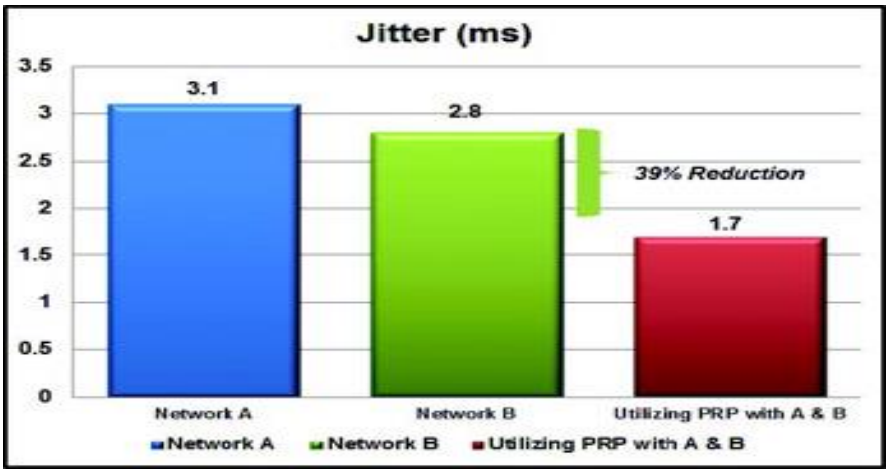

**Figure 2.8: PRP over two WLAN transmission paths: The redundant transmission compensates for packet losses and counterbalances load and interference-related transit time differences. (adapted from [Heer, 2015] ).**

If both paths failed in the same time, the transmission or reception error becomes visible. Thus, we can say that the systems apply PRP will never exhibit an error in the case of uncorrelated packet losses. Despite the way used by PRP is the same in wireless and wired connections via packet elimination and duplication, the effect accomplished for wireless is more dramatic. Several advantages of using the PRP switchover between two networks in wireless scenario over the wired case, as

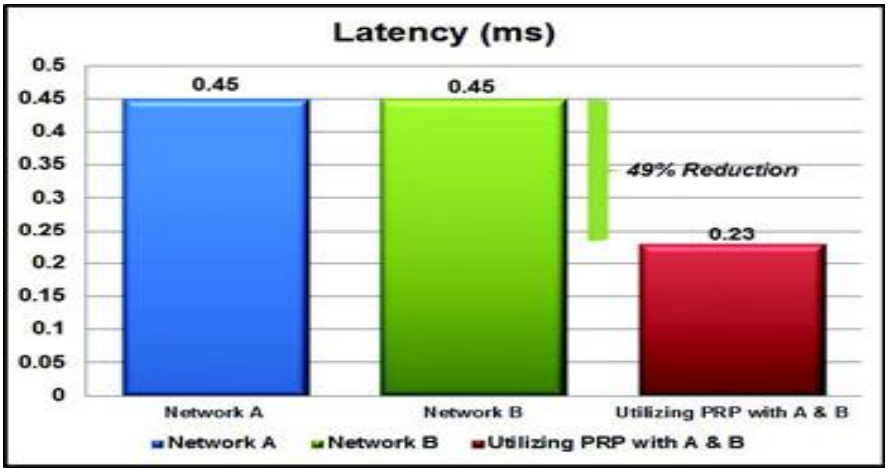
- Reliability is increased due to the compensation for packet losses if temporary disturbance happens, such as interference
- Jitter is reduced because of variation that only appears if both packets arrive late.
- Latency is decreased, as the faster of the two duplicated packets is always forwarded.




**(a)**




**(b)**




**(c)**

**Figure 2.9: Several advantages of using the PRP in wireless scenario over the wired case. (adapted from [Heer, 2015] ).**

## 2.7 Related Work

This section is a link between the terms and terminologies defined in the previous sections and the new work developed in the following chapters.

Recall again the S2A is a model of NCS architecture where the controller is included with the each actuator instead of being separate [Marti, 2001]. Using the S2A model improves overall system performance, as the control packets are transferred from the sensor nodes to the actuator nodes directly through one hop instead of two hops as in the case of the In-Loop model.

In [Moustafa, 2013], an S2A model was studied. The model was made of 16 sensors, one supervisor and four actuators; each of the sensors was connected directly to actuators through a switch, however each actuator included its own controller integrated in the same node.

On the other hand, in [Nilsson, 1998] In-Loop architecture was studied that composed of the same number of nodes with similar functionality apart from nonexistence of a supervisor; also, each packet was communicated from a sensor to a main controller for calculating the control action and processing before reaching the intended actuator (leading to more end-to-end delay). In the S2A model, all 16 sensors and the four actuators transmit every packet to the supervisor which is used to keep an eye on the actions and performance of the network. The Ethernet protocol was included in both S2A and the In-Loop models due to its comparatively low cost and direct integration with management floor functions.

In [Moustafa, 2013], OMNeT++ simulator was used to model the S2A architecture. Sensors, actuators and the supervisor were presented in shape of standard hosts. The sampling frequency was 1,440Hz [Daoud, 2003]; also, the control action was taken during a time frame of 694$\mu$s. The control packets were communicated through UDP [Boggia, 2009]. Fast Ethernet and Gigabit Ethernet performance were compared in cases of the S2A and the In-Loop models via OMNeT++ simulations. It was highlighted that the S2A architecture performance overweighed the In-Loop architecture, as the maximum end-to-end delay for both Gigabit Ethernet and Fast Ethernet was higher in the In-Loop model than the S2A model.

Another S2A NCS model was studied in [Boggia, 2009]. This system consisted of 16 sensors, four actuators and one supervisor; however, not all the nodes used the same protocol. On the one hand, two sensors and another two actuators communicated through an access point (AP) using 802.11/g protocol to allow for mobility in the system. On the other hand, 14 sensors, two actuators and one supervisor communicated their information using Gigabit Ethernet protocol via a switch. All the nodes sent their data to the supervisor to allow system to monitor the behavior and actions of the network. With a 10-byte load, control packets communicated are sent over the channel using UDP. This S2A model was simulated using OMNeT++ and was shown to meet the control system criteria for packet loss and delay constraints.

In [Abdel Reheem, 2012], the PRP was proposed using WiFi wireless connection in order to provide redundancy and to improve the overall performance of packet transmissions over the network. By introducing a Redundancy Box (RedBox), transmitted traffic is duplicated and transferred over two independent networks thus providing fault-tolerance against any failures that might occur in one of the underlying networks. Moreover, the first packet to arrive at the receiver is immediately processed by the system thus improving overall performance.

# Chapter 3

# Wired and Wireless S2A for NCS

In this chapter, a proposed system of 16 sensors, one supervisor and four actuators nodes is modeled and manipulated. The chapter starts by introducing the architecture of the proposed model followed by presenting a discussion of the interference exerted on wired and wireless nodes. After that, the simulation results of the implied model with a confidence analysis are illustrated. Finally, conclusions are given at the end of this chapter.

## 3.1 Model Architecture

In the architecture proposed in this work, both wired and wireless nodes will be utilized. In addition, the complete description of the S2A architecture is given. The system has 16 sensors, four actuators and one supervisor as based on the architecture described in [Moustafa, 2013]. However, some modifications have been added to this architecture to emphasis new feature of introducing mobility within the proposed workcell that is common to find such nodes in relevant new applications [ABB, 2009].

The mobility is introduced through making two of the 16 sensors as well as two of the four actuators are wirelessly connected. These four wireless-connected nodes (two sensors and two actuators) are communicated using the WiFi protocol. The remaining 14 sensors, two actuators and the supervisor are all wired and communicated on top of Ethernet. The wireless nodes use the 802.11/g protocol while the wired nodes communicate on top of Gigabit Ethernet. All nodes communicate with a supervisor. The area of the workcell that simulates the above mentioned architecture in OMNet++ simulator is $9m^2$ (3m×3m).

## 3.2 System Communication

According to the model in [Moustafa, 2013], sensors, actuators and the supervisor communicate directly through a switch and an access point without an intermediate controller, that is why this model is called S2A. However, each of the actuators includes its own controller; which decreases the number of hops from two to one, and consequently decreases the end-to-end delay. Figure 1 shows a snapshot from OMNeT++ describing the node placement for the studied model.

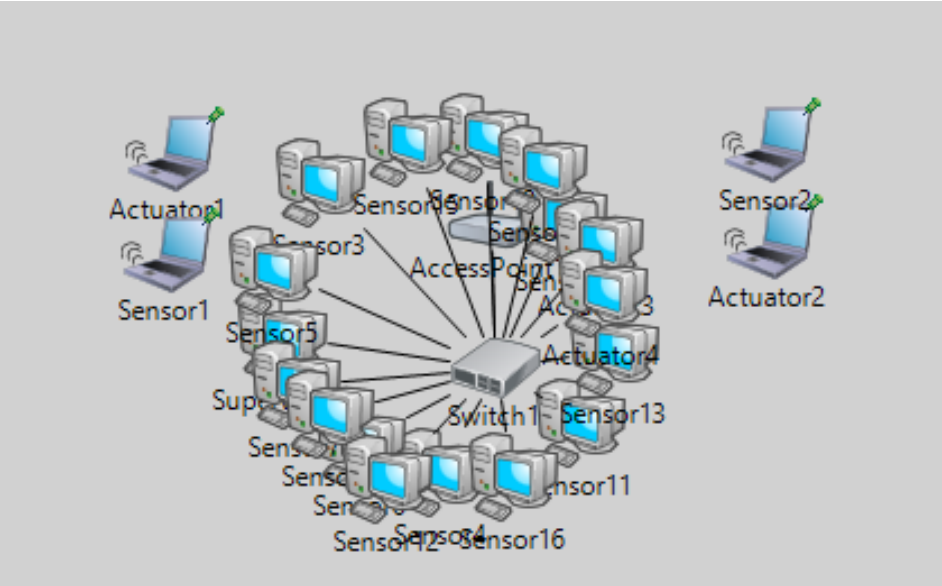



**Figure 3.1: A layout of the proposed model**

As described in [Moustafa, 2013], all sensors send their information to all four actuators. In contrary to the model developed in this work, the two wireless sensors only send their information to the two wireless actuators. This is because, in schemes involving motion, these nodes are responsible for the position control of the system. Accordingly, no information from these nodes is important for the wired control. The 14 wired sensors transmit their information to all the four actuators (wired and wireless). All

20 nodes (16 sensors and four actuators) transmit their information to the supervisor. Control packets are transmitted on-top-of UDP with a10-byte fixed load.

## 3.3 Maximum Delay Deadlines

The presence of two different protocols, which are IEEE 802.11/g (2.4GHz) and Gigabit Ethernet, leads to two different end-to-end packet delay times. The maximum delay time of the wireless nodes is 36ms (40ms minus 10 percent (4ms) safety factor) as in [Abdel Reheem, 2012]. On the other hand, the maximum delay of the wired deadline is 694$\mu$s as in [Daoud, 2003]. Therefore, the delay scheme in this architecture is described as follow:

- 36ms is the maximum delay time for:
    1. The packets that are sent from the two wired sensors to the two wireless actuators.
    2. The packets that are sent from the wireless sensors to the supervisor.
    3. Part of the data from the wired sensors to the supervisor.

- 694$\mu$s is the maximum delay time for:
    1. The packets that are sent from the wired sensors to the wired actuators
    2. Some of the packets from the wired sensors to the supervisor.

Table 3.1 summarizes the packet delay between the nodes in the implied architecture.

**Table 3.1: Maximum Delay Deadline Among Wireless Sensors, Wired Sensors, Wireless Actuators, Wired Actuators and The Supervisor**

| From | To | Maximum Delay Deadline |
|---|---|---|
| Wired Sensors | Wired Actuators | 694$\mu$s |
| Wired Sensors | Wireless Actuators | 36ms |

| From | To | Maximum Delay Deadline |
|---|---|---|
| Wireless Sensors | Wireless Actuators | 36ms |
| Wireless Sensors | Supervisor | 36ms |
| Wireless Actuators | Supervisor | 36ms |
| Wired Actuators | Supervisor | 694µs |
| Wired Sensors | Supervisor | 36ms |
| Wired Sensors | Supervisor | 694µs |

## 3.4 Gigabit Ethernet

As expected, using Gigabit Ethernet is much better in performance than Fast Ethernet which matches what is presented in [Daoud, 2003] and [Moustafa, 2013]. In the same line, under extra loading conditions, some systems fail with Fast Ethernet but operate correctly with Gigabit Ethernet. Therefore, in the proposed architecture in this work, Gigabit Ethernet is used rather than Fast Ethernet. In turn, all wired sensors, actuators and the supervisor are communicated with each other through Gigabit Ethernet of 1Gbps.

## 3.5 Wired and Wireless Nodes Position

In the delay calculations, the nearer the wireless nodes to the AP, the lower the delay can be achieved. Therefore, it is important to consider the position of the wireless sensors and wireless actuators with respect to the AP. In the proposed S2A architecture, the two wireless sensor nodes are put vertically on the left of the AP, while the other two actuator nodes are put vertically on the right of the AP. The AP is located at the center of the work cell. Figure 3.2 highlights the position of the wireless nodes.

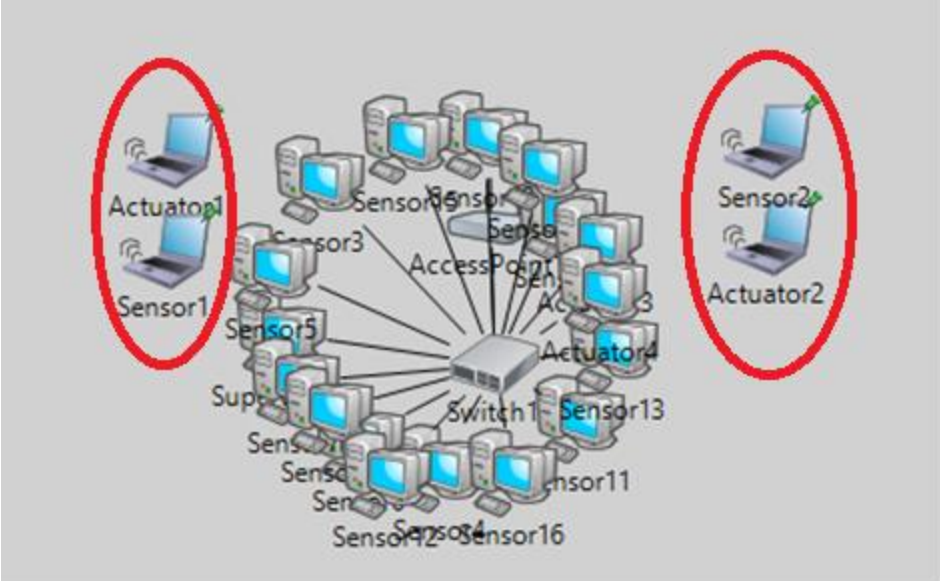



**Figure 3.2: Wireless nodes layout**

In figure 3.2 two wireless sensor nodes are put vertically on the left of the AP, while the other two actuator nodes are put vertically on the right of the AP.

## 3.6 Interfering nodes Study

In the proposed architecture, it is important to study the effect of interference on system performance. Three different types of interference are applied and discussed in the following subsections. The first interfering nodes type is applied on wired nodes, while the second one is applied on wireless nodes and finally a mixture of wired and wireless interfering nodes are exerted on the entire system.

### 3.6.1 Wired Interfering node

In the proposed architecture, an external interfering_1 node is added in order to send packets to the supervisor to emulate the wired interference. The packets are sent to the Supervisor through Gigabit Ethernet wire via the switch and then the supervisor replies by sending packets back to the interfering_1 node. The packets are sent on-top-of TCP. This interference is considered to be the source of non-real-time traffic application in the system. A typical example is data transfer between a senior engineer and the machine supervisor during normal machine operation. The maximum communicated

payload is the highest number of bytes that can be sent without violating the delay deadline (694$\mu$s).

### 3.6.2 Wireless Interfering nodes

In the proposed architecture, an external interfering_2 and interfering_3 nodes are added in order to emulate the wireless interference. These nodes are used to study the performance of the wireless sensors and actuators. Interfering_2 node sends packets to Interfering_3 node using 802.11/g via the system AP used to communicate the real-time traffic; and Interfering_3 replies by sending packets back to Interfering_2. The packets are sent on-top-of TCP. Interfering_2 node position is 0.75m below the border of the workcell, while Interfering_3 is located 0.75m above the border of the workcell. The maximum payload is the highest number of bytes that can be sent without violating the delay deadline (36ms).Figure 3.3 illustrates the nodes position.

### 3.6.3 Mixture of Wired and Wireless Interfering nodes

Using mixed interference technique helps to investigate the maximum load the system can tolerate in the presence of all other nodes without packet loss or breaking system delay requirements. All Interfering_1, Interfering_2, Interfering_3 nodes and Supervisor, mentioned in the previous subsections, exert interference on the proposed architecture simultaneously. All the settings in wired interference and wireless interference: the protocols, position of the nodes, wire length and delay deadlines are still the same as when they were applied on the proposed system separately.

## 3.7 Simulation results

In this section, OMNeT++ simulations are used to test the end-to-end delay of the proposed model. All simulation results are subjected to a 95% confidence analysis. However, four different simulations are done each with its own results. These four results are for the proposed system without interference, wired interference, wireless interference and both wired and wireless interference. The maximum end-to-end delay benchmarks are 36ms for the wireless nodes and 694$\mu$s for the wired nodes as mentioned before.

The nodes used for end-to-end delays are measurement are:

- Actuator 1 (A1)
- Wireless Actuator 2 (A2)
- Wired Actuator 3 (A3)
- Wired Actuator 4 (A4)
- Supervisor.

The Supervisor has two ports to receive packets with different sampling periods:

- S1 with 36ms
- S2with 694$\mu$s

The wireless and wired interference consists of the following nodes:

- Interfering_1 (wireless)
- Interfering_2 (wireless)
- Interfering_3 (wired)
- Third port of the supervisor (wired)

### 3.7.1 Without Interfering Nodes Simulation Results

For the proposed system without interfering nodes, the maximum end-to-end delay is 21.74ms at A2 node that meets the maximum delay deadline of the wireless nodes is 36ms as in [Abdel Reheem, 2012]. Table 3.2 shows the delay at each node.

**TABLE 3.2: End-To-End Delay Without Interfering nodes**

| A1 | A2 | A3 | A4 | S1 | S2 |
|---|---|---|---|---|---|
| 21.18 (ms) | 21.74 (ms) | 12.13 (µs) | 12.80 (µs) | 18.80 (ms) | 20.90 (µs) |

Figure 3.3 shows the entire nodes placement using without interfering nodes case in the OMNeT++ simulator

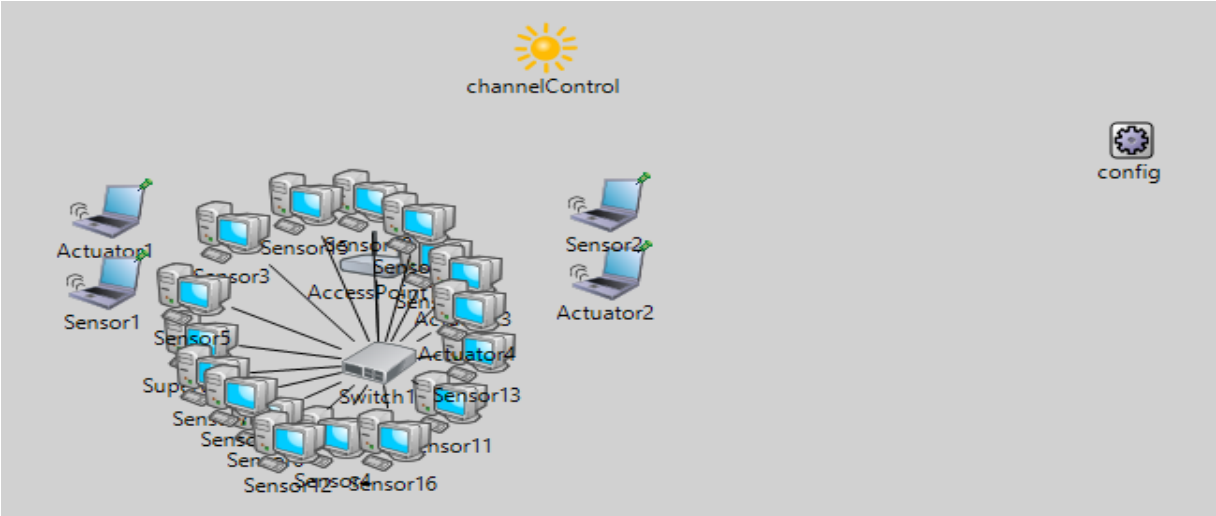



**Figure 3.3: Simulated topology of without interfering nodes**

Figure 3.4 shows an example of one of the wired actuator's delay in without interfering nodes case using OMNeT++ simulator

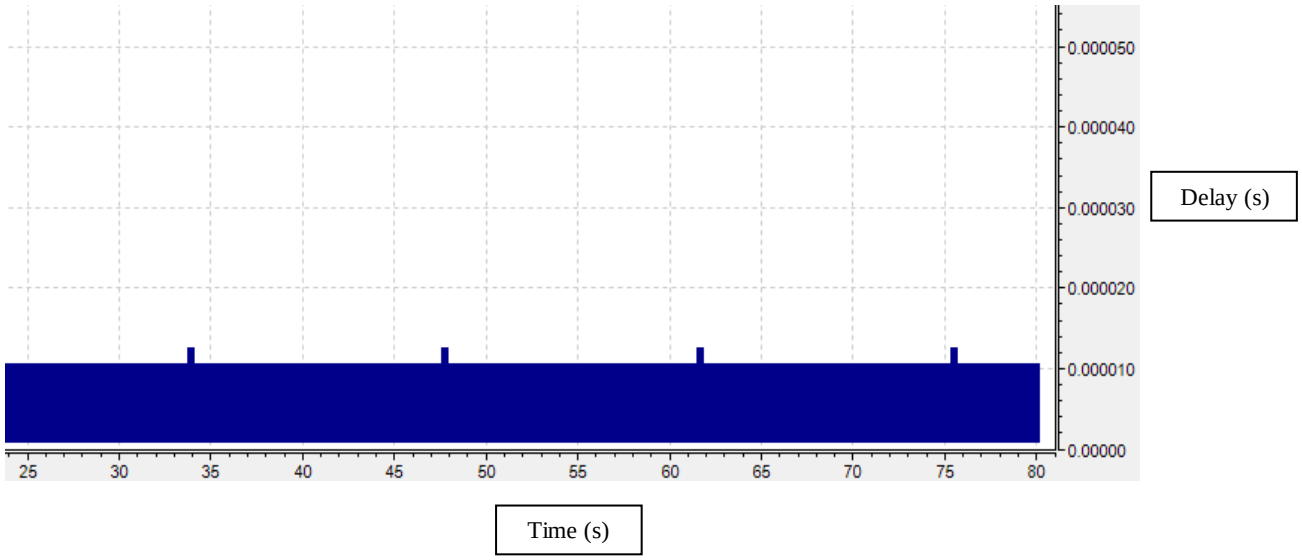



**Figure 3.4: Sample results for end-to-end delay of without interfering nodes on wired actuator**

Figure 3.4 reveals that the maximum delay of the wired actuator is 12.80 µs which means that it does not exceed the maximum end-to-end constrains that is 694 µs.

Figure 3.5 shows an example of one of the wireless actuator's delay in without interfering nodes case using OMNeT++ simulator

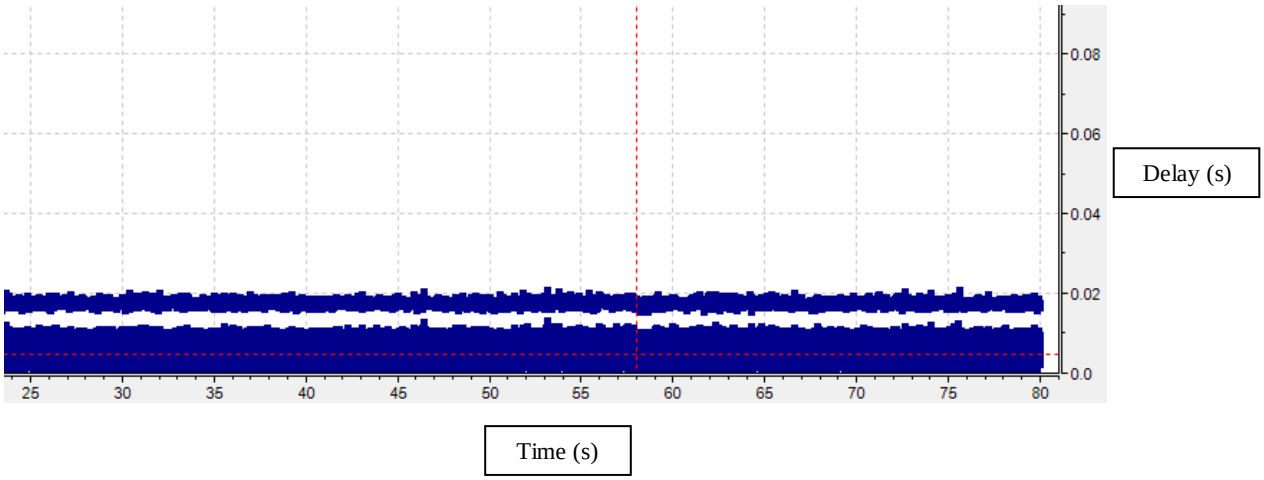



**Figure 3.5: Sample results for end-to-end delay of without interfering nodes on wireless actuator**

Figure 3.5 reveals that the maximum delay of the wireless actuator is 21.18 ms which means that it does not exceed the maximum end-to-end constrains that is 36 ms.

Figure 3.6 shows an example of supervisor's port 1 (wireless) delay in without interfering nodes case using OMNeT++ simulator

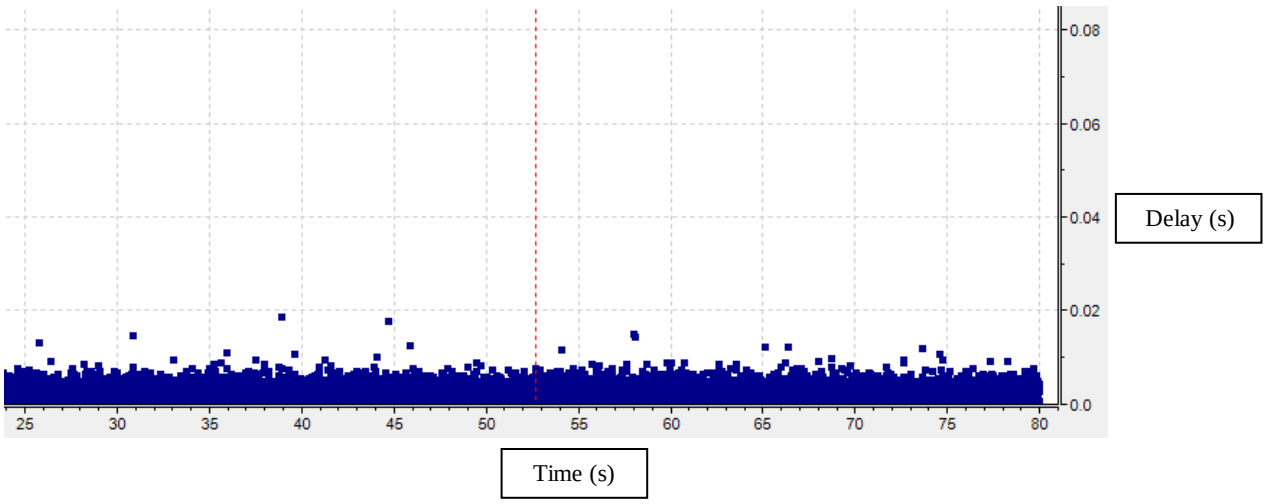



**Figure 3.6: Sample results for end-to-end delay of without interfering nodes on supervisor port 1**

Figure 3.6 reveals that the maximum delay of the first port of the supervisor is 18.80 ms which means that it does not exceed the maximum end-to-end constrains that is 36 ms.

Figure 3.7 shows an example of supervisor's port 2 (wired) delay in without interfering nodes case using OMNeT++ simulator

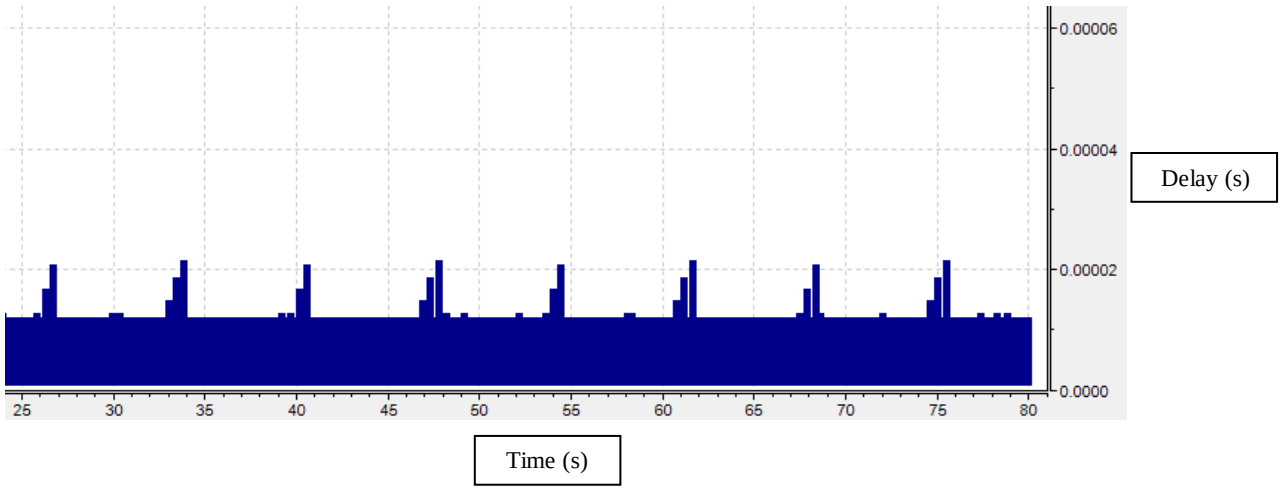



**Figure 3.7: Sample results for end-to-end delay of without interfering nodes on supervisor port 2**

Figure 3.7 reveals that the maximum delay of the supervisor's second port is 20.90 µs which means that it does not exceed the maximum end-to-end constrains that is 694 µs.

### 3.7.2 With-Interfering Nodes Simulation Results

Concerning the proposed system with wireless interfering nodes, the maximum wireless interference can be exerted within the time constrains is 12.8KB with 34.54 ms end-to-end delay at A2 node. Starting from 12.9KB wireless interference, the maximum delay deadline is higher than the 36ms deadline at node A2 with a 36.19ms delay at A2 node that break the maximum delay deadline of the wireless nodes is 36ms (as in [Abdel Reheem, 2012]). Table 3.3 illustrates the end-to-end delay with 12.8KB and 12.9KB wireless interfering nodes.

**TABLE 3.3: End-To-End Delay with Wireless Interference**

| **Wireless Interfering Nodes** | **A1** | **A2** | **A3** | **A4** | **S1** | **S2** |
|---|---|---|---|---|---|---|
| 12.8KB | 32.90 (ms) | 34.54 (ms) | 12.13 (µs) | 12.80 (µs) | 19.02 (ms) | 20.90 (µs) |
| 12.9KB | 34.28 (ms) | 36.19 (ms) | 12.13 (µs) | 12.80 (µs) | 19.88 (ms) | 20.90 (µs) |

Simulation results with wired interference show that the effect is minimal. Henceforth, the wired interference will be consisting of a 1MB per second. Results are tabulated in Table 3.4. Figure 3.3 illustrates sample OMNet++ results for one wired actuator subjected to wired interference. This is the result for one seed out of 33 seed used to build the confidence interval. The x-axis shows the simulation time while the y-axis is the measured delay.

**TABLE 3.4: End-To-End Delay with Wired Interference**

| **Wired interfering node** | **A1** | **A2** | **A3** | **A4** | **S1** | **S2** |
|---|---|---|---|---|---|---|
| 1MB | 20.99 (ms) | 21.59 (ms) | 12.13 (µs) | 12.80 (µs) | 18.44 (ms) | 20.90 (µs) |
| 5MB | 20.99 (ms) | 21.59 (ms) | 12.13 (µs) | 12.80 (µs) | 18.84 (ms) | 20.90 (µs) |

Figure 3.8 shows the entire nodes placement using wired interfering node case in the OMNeT++ simulator

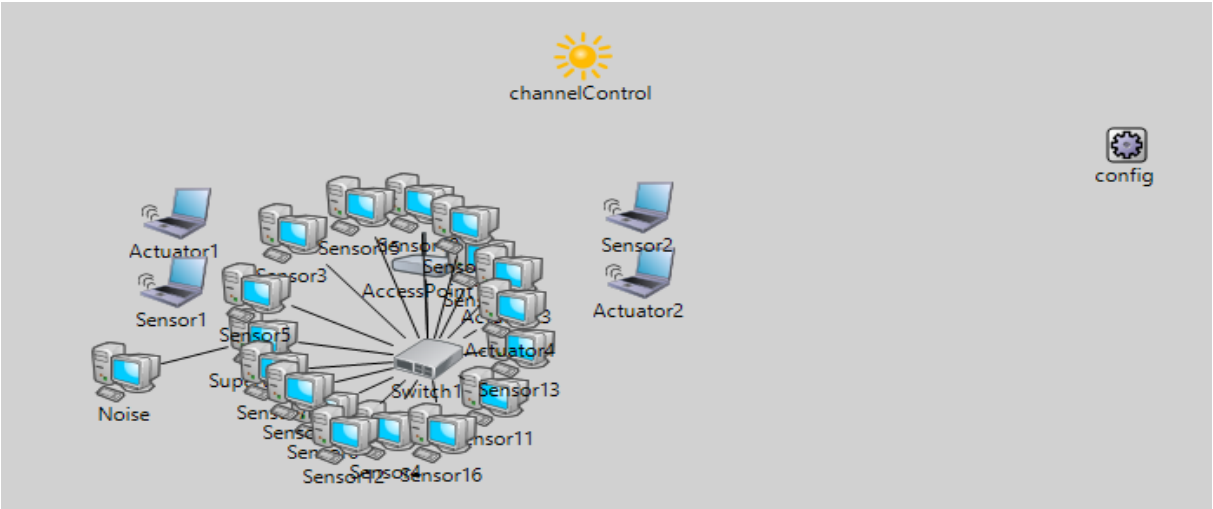



**Figure 3.8: Simulated topology of wired interfering node**

Figure 3.9 shows an example of one of the wired actuator's delay in wired interfering node case using OMNeT++ simulator

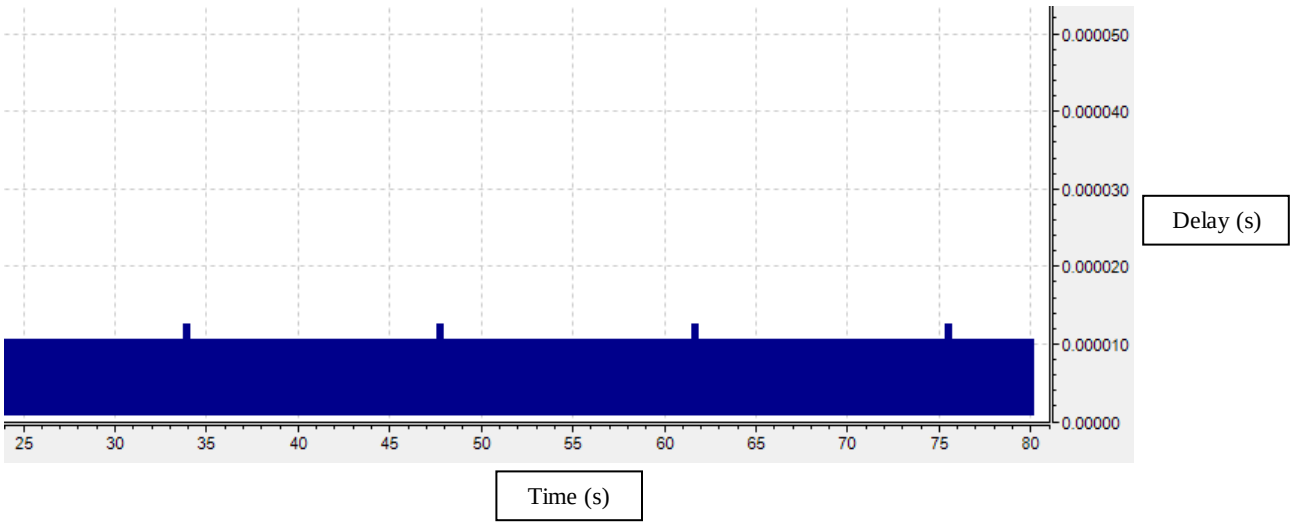



**Figure 3.9: Sample results for end-to-end delay of wired interfering node on wired actuator**

Figure 3.9 reveals that the maximum delay of the wired actuator is 12.80 µs which means that it does not exceed the maximum end-to-end constrains that is 694 µs.

Figure 3.10 shows an example of one of the wireless actuator's delay in interfering nodes case using OMNeT++ simulator

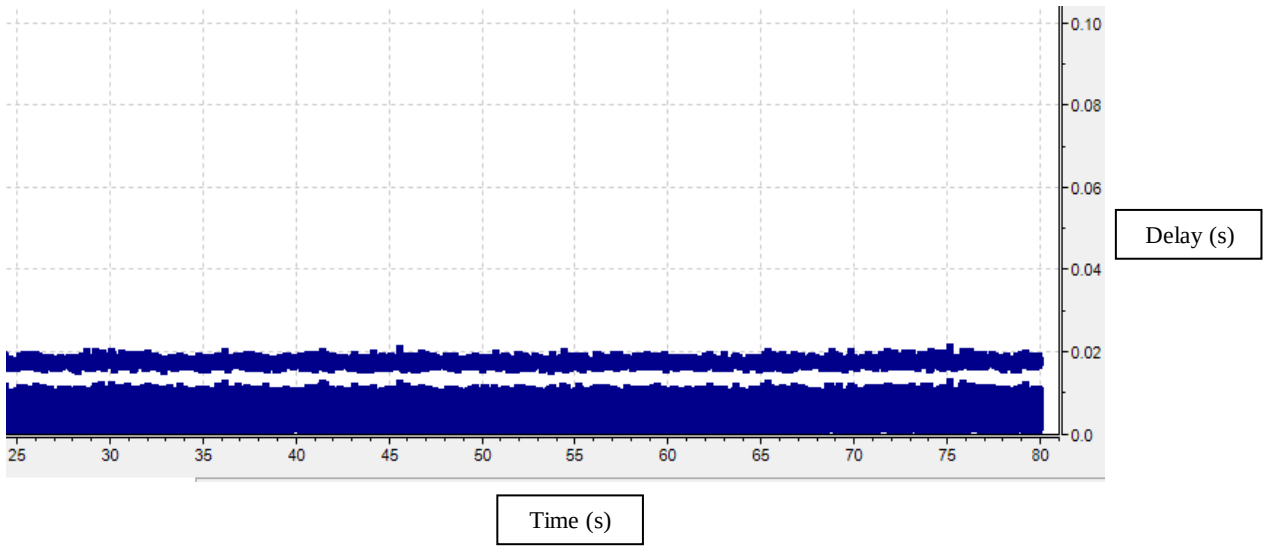



**Figure 3.10: Sample results for end-to-end delay of wired interfering node on wireless actuator**

Figure 3.10 reveals that the maximum delay of the wireless actuator is 21.74 ms which means that it does not exceed the maximum end-to-end constrains that is 36 ms.

Figure 3.11 shows an example of supervisor's port 1 (wired) delay in wired interfering node case using OMNeT++ simulator

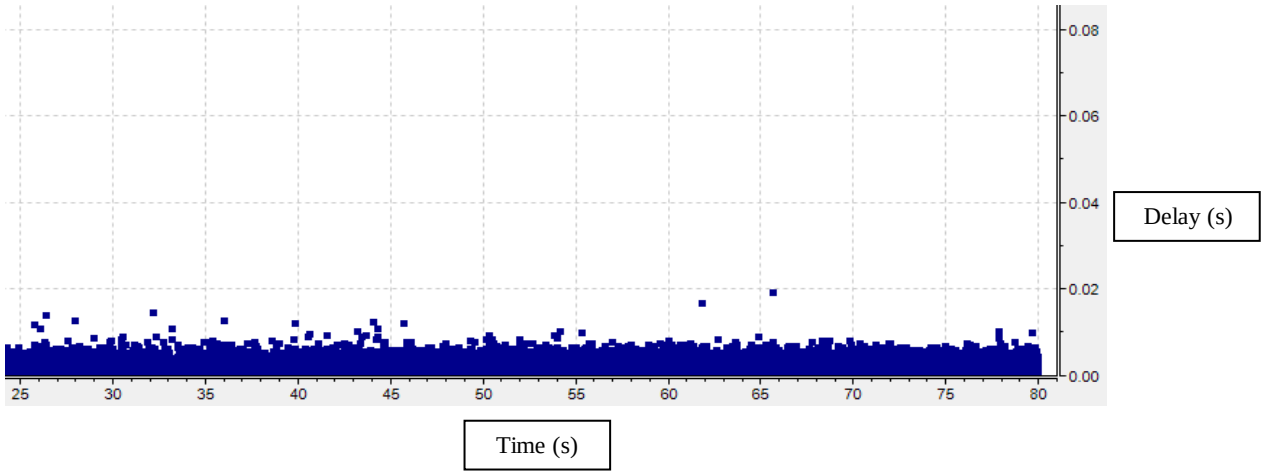



**Figure 3.11: Sample results for end-to-end delay of wired interfering node on Supervisor Port 1**

Figure 3.11 reveals that the maximum delay of the first port of the supervisor is 18.8 ms which means that it does not exceed the maximum end-to-end constrains that is 36 ms.

Figure 3.12 shows an example of supervisor's port 2 (wired) delay in wired interfering node case using OMNeT++ simulator

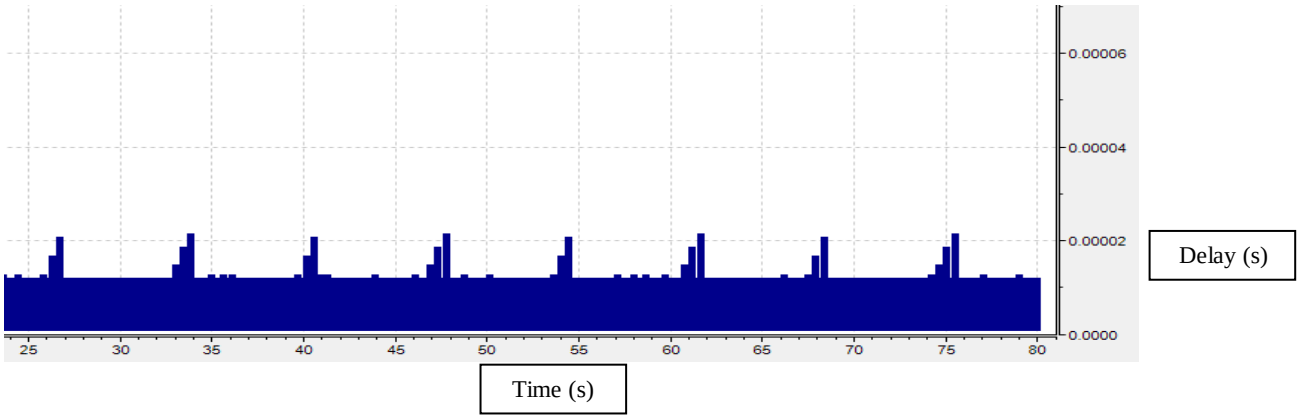



**Figure 3.12: Sample results for end-to-end delay of wired interfering node on Supervisor Port 2**

Figure 3.12 reveals that the maximum delay of the supervisor's second port is 20.90 µs which means that it does not exceed the maximum end-to-end constrains that is 694 µs.

Finally, a mixture of wireless and wired interfering nodes is applied to the system, more specifically 12.7KB wireless interference and 1MB wired interference. The maximum wireless end-to-end delay was 35.35ms as shown in Table V. Starting from 12.8KB for wireless interfering nodes and 1MB for wired interfering node, the end-to-end delay exceeds the 36ms deadline with a maximum delay of 36.01ms.

Table 3.5 shows the end-to-end delays with 12.7KB plus 1MB, 12.8KB plus 1MB and 12.9KB plus 1MB.

**TABLE 3.5: End-To-End Delay with Mixed Interfering Nodes**

| **Interfering (Wireless) & (Wired) Nodes** | **A1** | **A2** | **A3** | **A4** | **S1** | **S2** |
|---|---|---|---|---|---|---|
| 12.7KB & 1MB | 34.26 (ms) | 35.43 (ms) | 12.13 (µs) | 12.80 (µs) | 19.72 (ms) | 20.90 (µs) |
| 12.8KB & 1MB | 33.52 (ms) | 36.01 (ms) | 12.13 (µs) | 12.80 (µs) | 20.07 (ms) | 20.90 (µs) |
| 12.9KB & 1MB | 34.09 (ms) | 37.58 (ms) | 12.13 (µs) | 12.80 (µs) | 20.92 (ms) | 20.90 (µs) |

The results prevail that the wireless plus wired interfering node mixture will be slightly worse than wireless or wired separately. This is because the wired and wireless delays are added in series for some of the traffic. For example, if we have the maximum file size, with delay almost equal to the deadline, in the wireless side then the added

interfering node in the wired size will increase the delay slightly leading to a total delay greater than the deadline.

Figure 3.13 shows the entire nodes placement using mixed interfering nodes case in the OMNeT++ simulator

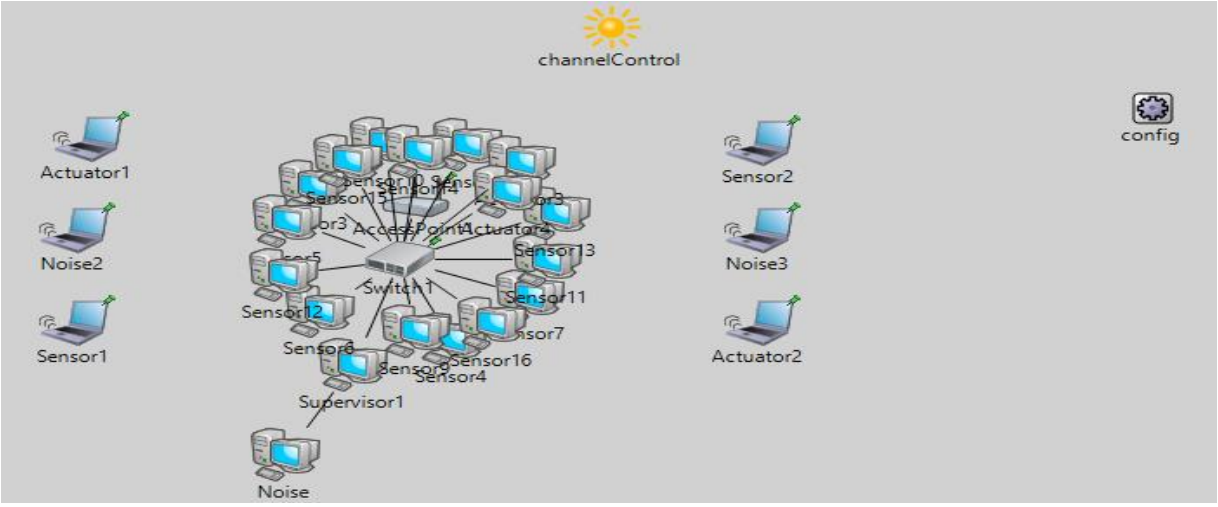



**Figure 3.13: Simulated topology of mixed interfering nodes**

Figure 3.14 shows an example of one of the wired actuator's delay in mixed interfering nodes case using OMNeT++ simulator

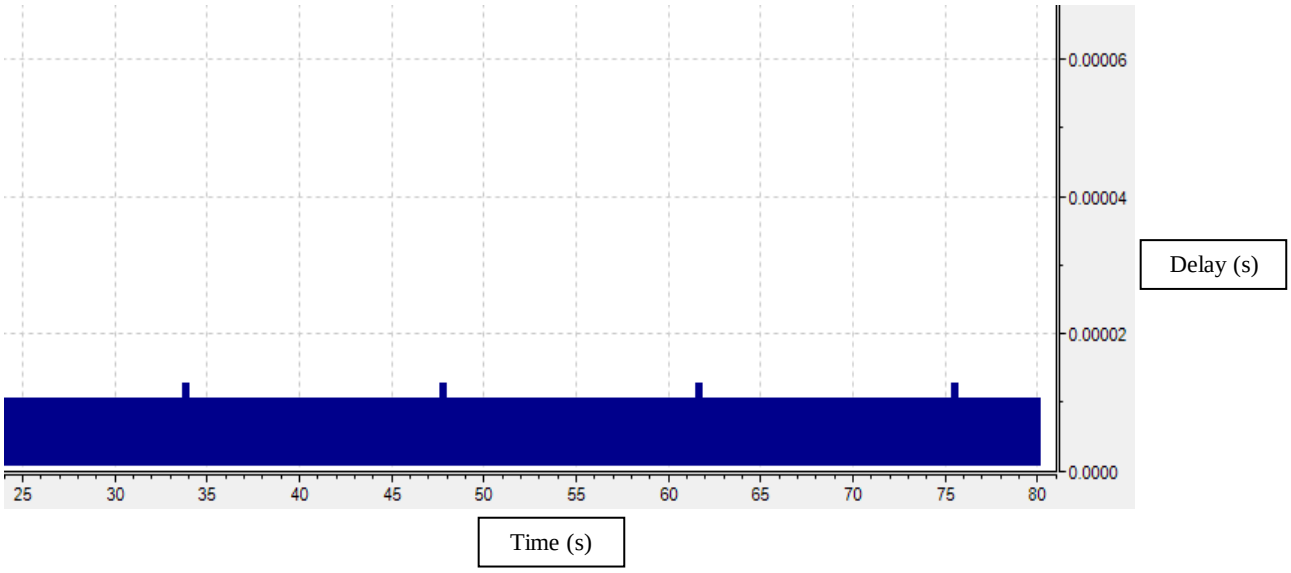



**Figure 3.14: Sample results for end-to-end delay of mixed interfering nodes on wired actuator**

Figure 3.14 reveals that the maximum delay of the wired actuator is 12.80 µs which means that it does not exceed the maximum end-to-end constrains that is 694 µs.

Figure 3.15 shows an example of one of the wireless actuator's delay in mixed interfering nodes case using OMNeT++ simulator

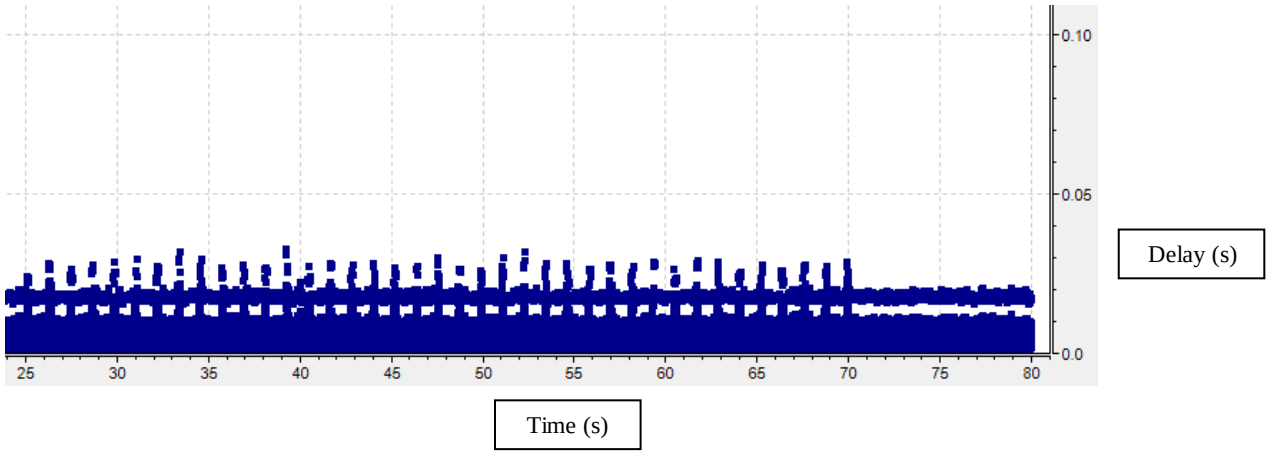



**Figure 3.15: Sample results for end-to-end delay of mixed interfering nodes on wireless actuator**

Figure 3.15 reveals that the maximum delay of the wireless actuator is 35.43 ms which means that it does not exceed the maximum end-to-end constrains that is 36 ms.

Figure 3.16 shows an example of supervisor's port 1 (wired) delay in mixed interfering nodes case using OMNeT++ simulator

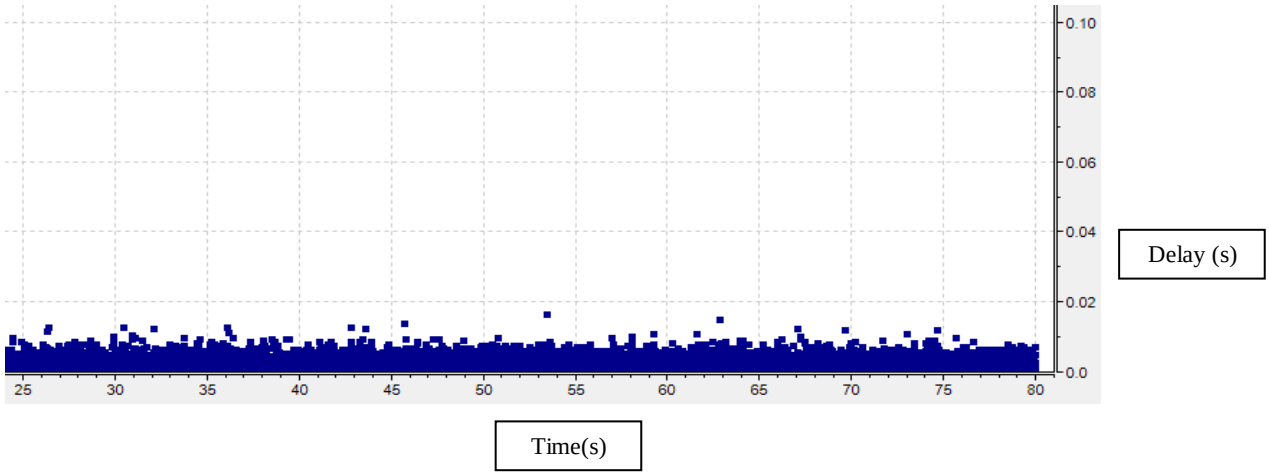



**Figure 3.16: Sample results for end-to-end delay of mixed interfering nodes on supervisor port 1**

Figure 3.16 reveals that the maximum delay of the first supervisor port is 19.72 ms which means that it does not exceed the maximum end-to-end constrains that is 36 ms.

Figure 3.17 shows an example of supervisor's port 2 (wired) delay in mixed interfering nodes case using OMNeT++ simulator

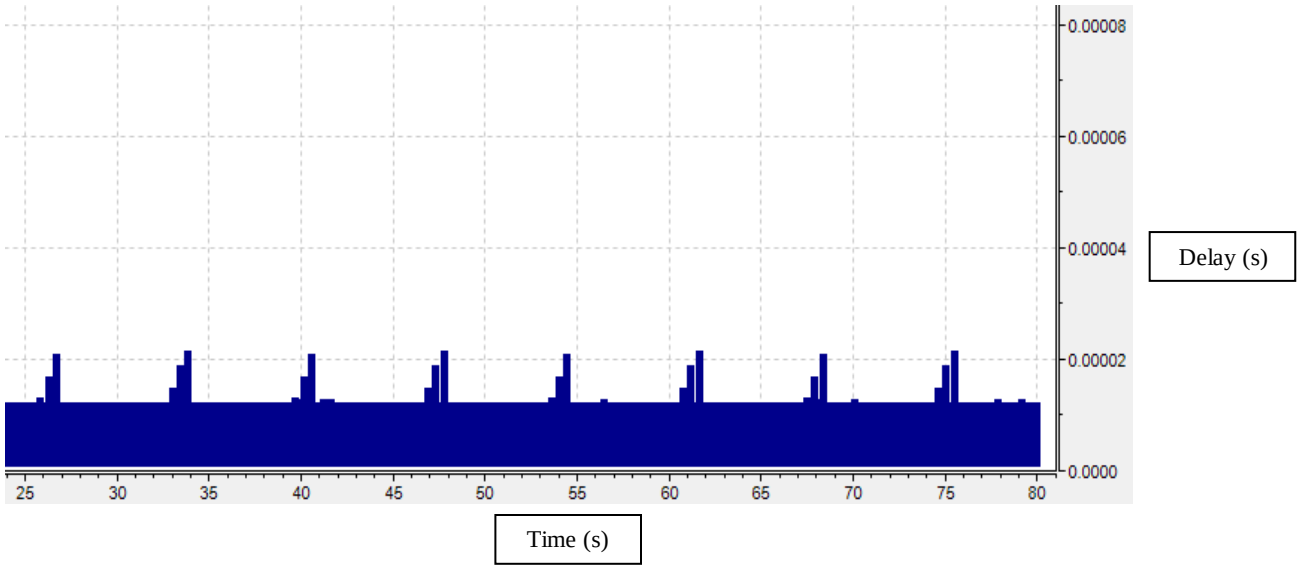



**Figure 3.17: Sample results for end-to-end delay of mixed interfering nodes on supervisor port 2**

Figure 3.17 reveals that the maximum delay of the supervisor's second port is 20.90 µs which means that it does not exceed the maximum end-to-end constrains that is 694 µs.

It is important to note that the delays shown in all tables represent the maximum of the confidence interval.

## 3.8 Summary

Certain applications of NCSs may require both wired and wireless sensors/actuators in the same workcell. In this chapter, such an NCS is investigated. It consists of 16 sensors, four actuators and one supervisory node. The architecture is the S2A architecture. Two of the 16 sensors as well as two of the four actuators are wireless. The remaining 14 sensors, two actuators and the supervisor are all wired. The wireless

nodes communicate using the 802.11/g protocol while the wired nodes communicate on top of unmodified Ethernet.

It is shown via OMNeT++ simulations that this NCS does not suffer any packet loss and that end-to-end delays satisfy the 36ms wireless deadline as well as the 694µs wired deadline. In order to further study the robustness of this NCS, it is subjected to both wired and wireless interfering nodes. It is observed that the NCS can withstand up to 12.8KB wireless interference and 12.7KB plus 1MB mixed interference. All end-to-end delays incorporate all types of propagation, encapsulation, de-capsulation and queuing delays. Furthermore, all results are based on a 95% confidence analysis.

# Chapter 4

# Redundancy for Sensor-to-Actuator Networked Control Systems

In this chapter, a model is developed in which different fault-tolerance methods are incorporated into the proposed system of 16 sensors, one supervisor and four actuators nodes. TMR at wireless and wired nodes plus PRP on wireless nodes are expected to increase reliability. Part of the proposed model will run over Gigabit switched Ethernet, while the other part will run over WiFi (IEEE 802.11g). The end-to-end delay and packet loss will be observed by OMNeT++ simulations. The chapter starts by presenting a discussing the proposed model followed by presenting different reliability scenarios; finally, the chapter is concluded.

## 4.1 Improved Architecture Analysis

In this section, an optimized S2A model including TMR and PRP fault tolerance is fully described using a work space area $9m^2$ (3m x 3m) as in [OMNeT, 2016].

### 4.1.1 Model Description

The proposed architecture is composed of 48 sensors, four actuators and one supervisor. TMR technique is implemented by triplicating the number of the wired nodes from 14 sensors to a total of 42 sensors at the node level, while the two wired actuators remain unchanged. Regarding the two wireless sensors, TMR is applied at the level of the sensor element, while the PRP at the level of the network interface. Figure 4.1 shows overview of the proposed model. In the model developed in this chapter, the two wireless sensors send information only to the two wireless actuator nodes using PRP WLAN over

two independent channels contradicting the data transmission scheme studied in [Proenza, 2012] in which all wireless sensors communicated with all wireless actuators. On the other hand, each of the 42 wired sensors transmit data to each of the two wired actuators through the switch. The reason that the wireless nodes do not communicate with the wired actuator nodes is that the wireless nodes do not generate data that may impact the control of the wired nodes. All the 48 nodes (44 sensors and four actuators) send their data to the supervisor. With a 10-byte fixed load, control packets are communicated on-top-of UDP.

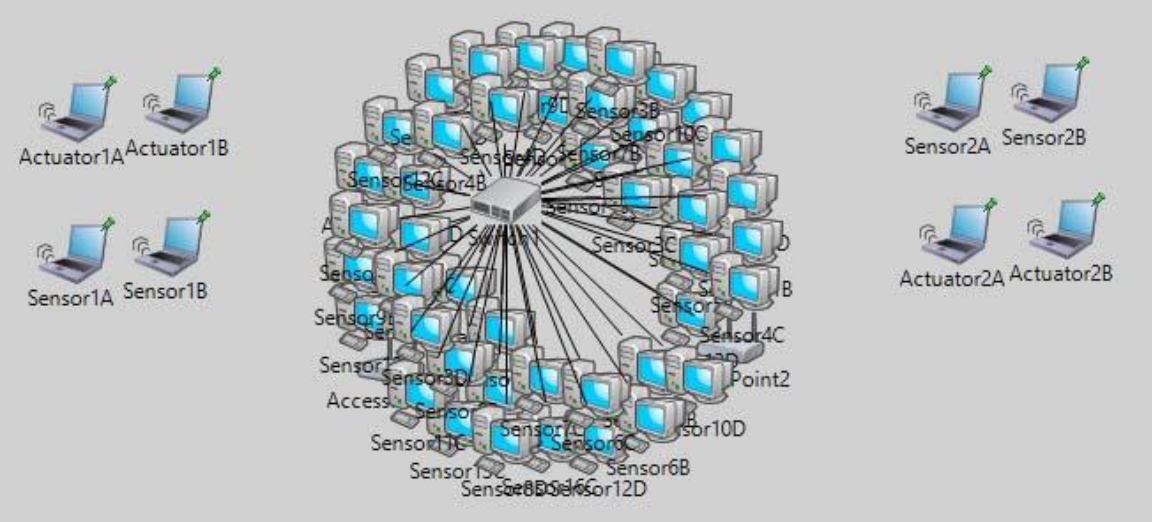



**Figure 4.1: Simulated model**

## 4.1.2 Gigabit Ethernet

Gigabit Ethernet should be used instead of Fast Ethernet, for the system described as before, to satisfy that the control system has a criterion of no packet loss and zero over-delayed packets, as shown in OMNeT++ simulations.

## 4.1.3 Nodes Position

Unlike the wired nodes, the delay of the wireless nodes is in fact significantly affected by the position of AP; the propagation delay decreases when the wireless nodes

are closer to the AP [Proenza, 2012]. In this simulation, position of the system nodes is as follows:

- The AP is positioned at the center of the workspace.
- Two of the sensors are positioned vertically on the left of AP.
- Two of the actuators are positioned vertically on the left of AP.
- The rest are on the right close to the edge of the workcell.

### 4.1.4 Maximum Delay Deadlines

For the reason of using both IEEE 802.11/g (2.4GHzWiFi) and Gigabit Ethernet and there are two different packet delay constraints: the deadline constraint for the wireless nodes is 36ms [Toubar, 2015] whereas the maximum delay deadline for the wired nodes is 694µs [Daoud, 2003]. The maximum delay deadlines for all nodes are summarized in Table 4.1.

**TABLE 4.1: Maximum Delay Deadline Among Wireless Sensors, Wired Sensors, Wireless Actuators, Wired Actuators and The Supervisor [16].**

| From | To | Maximum Delay Deadline |
|---|---|---|
| Wired Sensors | Wired Actuators | 694µs |
| Wired Sensors | Wireless Actuators | 36ms |
| Wireless Sensors | Wireless Actuators | 36ms |
| Wireless Sensors | Supervisor | 36ms |
| Wireless Actuators | Supervisor | 36ms |
| Wired Actuators | Supervisor | 694µs |
| Wired Sensors | Supervisor | 36ms |
| Wired Sensors | Supervisor | 694µs |

### 4.1.5 Simulation Results

The proposed architecture, described in subsection 4.1.1, was simulated using the OMNeT++ network modeler [OMNeT, 2016]. The sensor, actuator and supervisor nodes were modeled using standard hosts as described in the previous subsection. It was shown that the proposed fault-tolerant architecture meets all the required control constraints including zero packet loss without violating any end-to-delay deadline for both wired and wireless transmissions (as summarized in Table 4.1).

The results observed concerning the maximum the end-to-end delay from OMNeT++ simulations of the proposed network after a 95% confidence analysis is summarized in Table 4.2. From these results, it can be seen that the proposed architecture meets the maximum end-to-end delay deadlines specified in Table 4.1.

It should be noted that from Table 4.2:

- The TMR-based wired sensor nodes (42 sensors) are divided into three Groups B, C and D (3 groups of 14 wired sensors each) communicating directly with the two wired actuators.
- The PRP-WLAN based wireless nodes (2 sensors and 2 actuators) communicate using dual wireless network interface cards and are labeled as Group A.
- The end-to-end delays at the supervisor node are split to distinguish between traffic from wired/wireless nodes belonging to the various groups: S[0] is for traffic from the wireless sensors in Group A, S[1] is from the wired sensors in Group B, S[2] is from the wired sensors in Group C, S[3] from the wired sensors in Group D, S[4] is from the wired actuators and S[5] is from the wireless actuators.

**TABLE 4.2: End-to-End delay**

| Node | Delay | Connectivity |
|---|---|---|
| A1 $G_A$ | 7.61 ms | Wireless |
| A1 $G_B$ | 7.81 ms | |
| A2 $G_A$ | 8.49 ms | |
| A2 $G_B$ | 8.49 ms | |
| A3 $G_B$ | 27.58 µs | Wired |
| A3 $G_C$ | 28.25 µs | |
| A3 $G_D$ | 28.90 µs | |
| A4 $G_B$ | 28.25 µs | |
| A4 $G_C$ | 28.93 µs | |
| A4 $G_D$ | 29.60 µs | |
| S[0] | 8.78 ms | Wireless |
| S[1] | 29.66 µs | Wired |
| S[2] | 30.33 µs | |
| S[3] | 31.00 µs | |
| S[4] | 2.08 µs | |
| S[5] | 9.07 ms | Wireless |

## 4.2 Reliability Calculations

In the proposed architecture, TMR on wired sensor nodes, TMR on wireless sensor elements and PRP are three different FT techniques are used, however, these three techniques do not have to be applied together. Depending on the reliability requirements and the cost constraints, the user of the proposed model can use any of the eight ($2^3$) scenarios available. The eight different scenarios are analyzed in order to help the user with the choice of the appropriate FT combination. The combination of the fault tolerance techniques used in the proposed model are encoded by three digits as: the first digit in the name of each scenario indicates whether TMR on wired sensor nodes is used (1) or not (0), the second digit indicates whether TMR on wireless sensor elements is used (1) or

not (0) while the third is for PRP. For example, scenario 011 means that TMR on the wireless sensor elements as well as PRP are applied while TMR on the wired sensor nodes is not. The eight FT scenarios include reliability on six components of the proposed system which are:

- R1: Reliability of a wired sensor node.
- R2: Reliability of a wireless sensor node.
- R3: Reliability of the RedBox.
- R4: Reliability of the RedBox Antenna (after the RedBox and before the wireless communication channel)
- R5: Reliability of the voter for wireless sensors
- R6: Reliability of the transmitter connected to the wireless voter.

The reliability equations are calculated by tracing the path of the packets through the system's components. For example, in scenario 000 concerning the wired part, the reliability is powered by 14 which is the number of wired sensors without triplicating as there is no TMR asserted on the wired sensors, while regarding the wireless part, the equation of reliability on wireless sensors and the transmitter connected to the wireless voter is powered by two, as there is no TMR applied too. R3 and R4 are not used in this equation as no PRP is applied. Also, R5 is also not used, as there no TMR on the wireless sensors, so there is no need for the voter. Table 4.3 highlights these eight FT scenarios and the corresponding reliability, *R*, equations.

**TABLE 4.3: Different FT Scenarios (Eight Scenarios)**

| Scenario | FT Type | Equation |
|---|---|---|
| **000** | None | $(R_1)^{14} \times [R_2 \times R_6]^2$ |
| **001** | PRP | $[R_2 \times R_3 \times (2R_4 - R_4^2)]^2 \times R_1^{14}$ |
| **010** | TMR Wireless | $[(3R_2^2 - 2R_2^3) \times R_5 \times R_6]^2 \times R_1^{14}$ |
| **011** | TMR Wireless + PRP | $[(3R_2^2 - 2R_2^3) \times R_5 \times R_3 \times (2R_4 - R_4^2)]^2 \times R_1^{14}$ |
| **100** | TMR Wired | $(3R_1^2 - 2R_1^3)^{14} \times [R_2 \times R_6]^2$ |
| **101** | TMR Wired + PRP | $(3R_1^2 - 2R_1^3)^{14} \times [R_2 \times R_3 \times (2R_4 - R_4^2)]^2$ |
| **110** | TMR Wired + TMR Wireless | $[(3R_2^2 - 2R_2^3) \times R_5 \times R_6]^2 \times (3R_1^2 - 2R_1^3)^{14}$ |
| **111** | TMR Wired + TMR Wireless+PRP | $(3R_1^2 - 2R_1^3)^{14} \times [(3R_2^2 - 2R_2^3) \times R_5 \times R_3 \times (2R_4 - R_4^2)]^2$ |

In Table 4.3, the reliability is calculated by$R_i = e^{-\lambda_i t}$ at which the failure rate (per month), λ, is considered to be constant and the time, *t*, is distributed exponentially. The different failure rates are defined as:

- λ1 is the failure rate of a wired sensor node
- λ2 is the failure rate of a wireless sensor node
- λ3 is the failure rate of the RedBox
- λ4 is the failure rate of the RedBox Antenna (after the RedBox and before the wireless communication channel)

- $\lambda 5$ is the failure rate of the voter for wireless sensors
- $\lambda 6$ is the failure rate of the transmitter connected to the wireless voter.

Next, several case studies will be presented to illustrate the use of the equations in Table 4.3 as a design tool. For each case study, a set of different values is assumed for the six failure rates to investigate their effect on reliability and then plotting them using MATLAB. Moreover, the several below cases will help the user to choose the lowest scenario in term of cost, which happens when two or more cases has nearly the same reliability value, but one of them uses less reliability techniques that leads to less components used that means lower cost. The failure rates for the first case are shown in Table 4.4:

**TABLE 4.4: Failure Rates for the First case**

| **Failure rate** | $\lambda 1$ | $\lambda 2$ | $\lambda 3$ | $\lambda 4$ | $\lambda 5$ | $\lambda 6$ |
|---|---|---|---|---|---|---|
| | 1/6 | | | | | |

Figure 4.2 shows the change in reliabilities over time for the eight scenarios prevailing the order of the scenarios from lowest reliability to highest is (000, 010, 001 and 011), then (100,101,110 and 111).

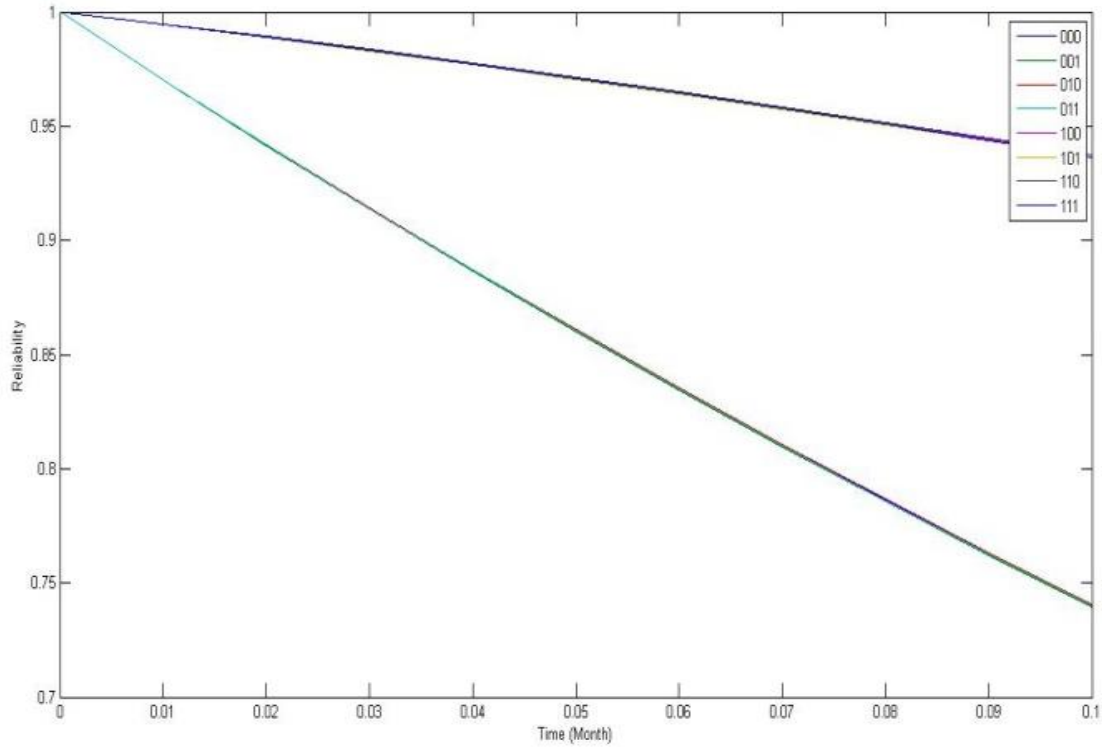



**Figure 4.2: Reliability analysis of the First Case.**

Therefore, the user will prefer 100 over {101, 110 and 111}, as it has nearly the same reliability, but with reduced cost, as TMR on wired sensor nodes is only used. Moreover, it will become clear that cases that involve TMR on the wired sensor nodes have the highest reliability; the reason for that is that the $R_1^{14}$ term dominates every scenario as can be seen from the reliability equations in Table 4.3.

Based on the previous case, another example is generated by increasing the values of λ1 and λ2 from 1/6 to 1 (as in Table 4.5) to examine system reliability when the failure rates of the wired and the wireless sensor elements are changed.

**TABLE 4.5: Failure Rates for the Second Case**

| **Failure rate** | λ1 | λ2 | λ3 | λ4 | λ5 | λ6 |
|---|---|---|---|---|---|---|
| | 1 | | 1/6 | | | |

Figure 4.3 shows that the highest reliability belongs to scenarios (110 and 111) where (000 and 001), (010 and 011) then (100 and 101) have the lowest reliability, respectively. The first observation here is that TMR for the wired nodes has a strong effect on reliability as mentioned above; moreover, TMR on the wireless sensor elements increases reliability. However, since scenarios 110 and 111 have almost the same reliability, it would be preferable not to implement PRP for cost effectiveness.

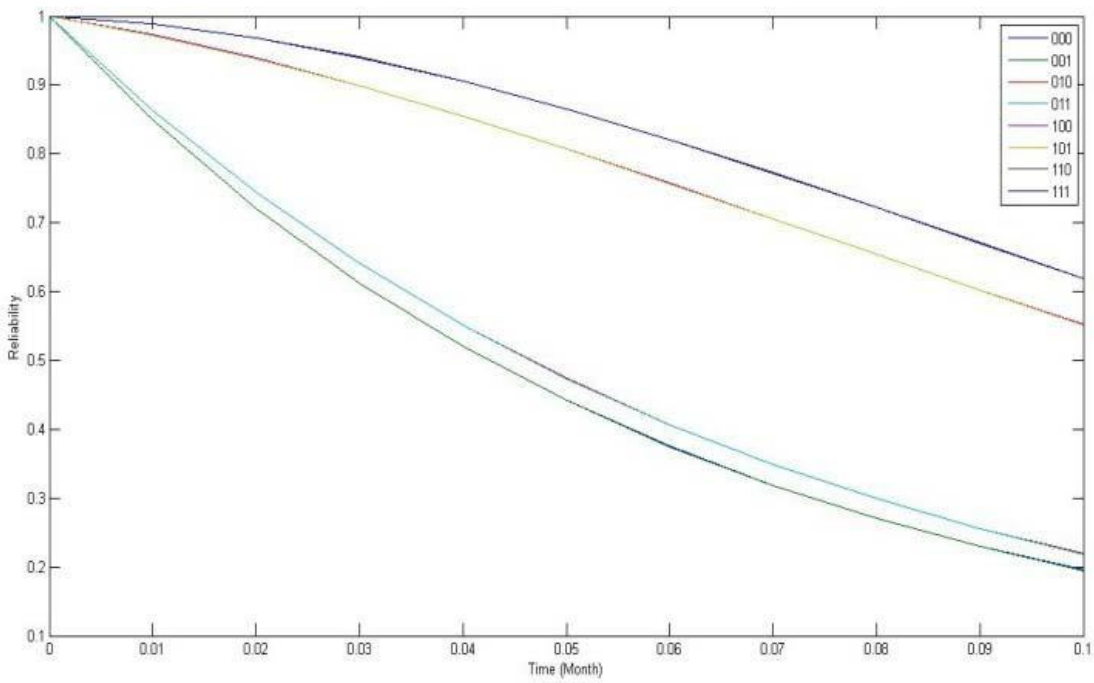



**Figure 4.3: Reliability analysis of the Second Case.**

In the third case study, the value of λ3 is changed from 1/6 to 1/24 with respect to the previous case as in Table 4.6.

**TABLE 4.6: Failure Rates for the Third Case**

| **Failure rate** | λ1 | λ2 | λ3 | λ4 | λ5 | λ6 |
|---|---|---|---|---|---|---|
| | 1 | | 1/24 | 1/6 | | |

Figure 4.3 indicates that a more reliable RedBox in the PRP FT scheme will not notably change the ranking of the eight scenarios (from a reliability point of view); however, any scenario with PRP is slightly more reliable than the same scenario without PRP.

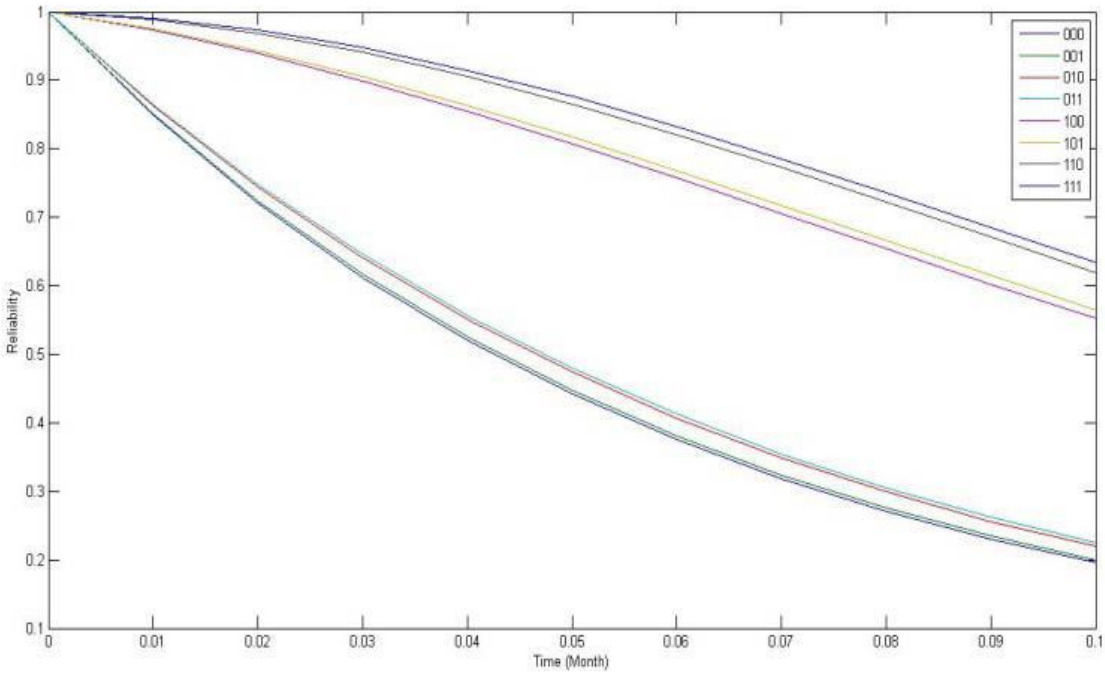



**Figure 4.4: Reliability analysis of the Third Case.**

The rate of the RedBox may not justify the extra cost of this equipment. In the three previous case studies, PRP did not significantly affect system reliability. Using the same failure rates as in the second case study, only λ1 is changed to 1/3 instead of 1. The failure rates are in Table 4.7.

**TABLE 4.7: Failure Rates for the Fourth case**

| **Failure rate** | λ1 | λ2 | λ3 | λ4 | λ5 | λ6 |
|---|---|---|---|---|---|---|
| | 1/3 | 1 | 1/6 | | | |

Figure 4.5 shows that PRP does make a difference in the reliability calculations. The order of the scenarios in this case is the highest reliability (110 and 111) whereas (000 and 001), (010 and 100) then (011 and 101) are the least reliability, respectively. Note that, the 101 scenario has a higher reliability than 100, i.e., the addition of PRP as a FT scheme does increase reliability.

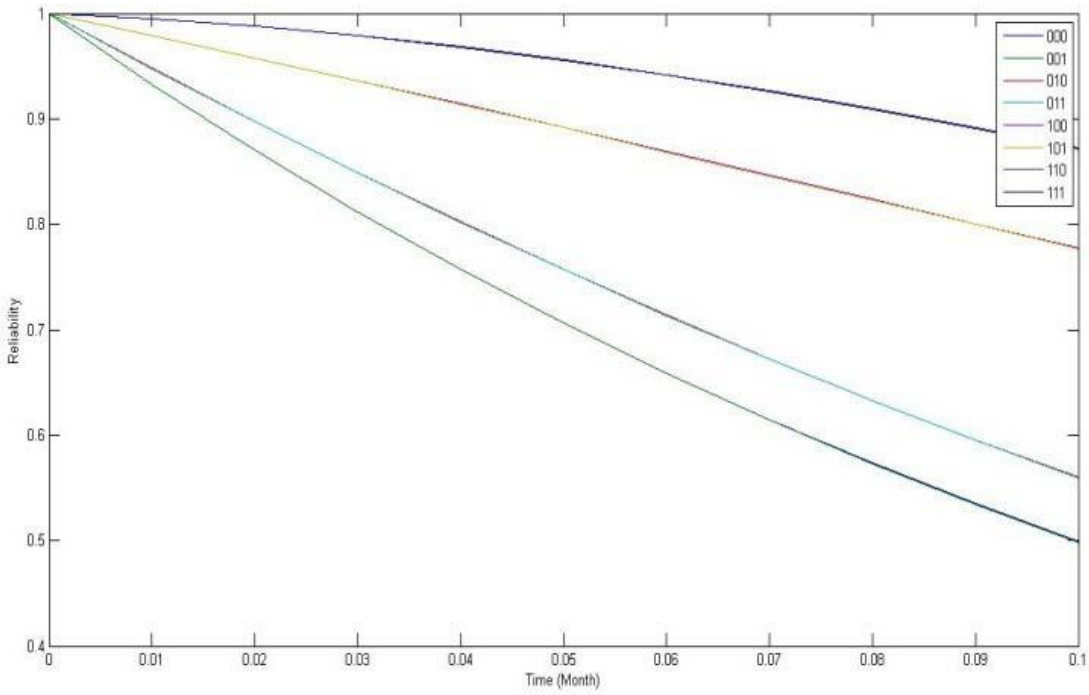



**Figure 4.5: Reliability analysis of the Fourth Case.**

As another example, the failure rates λ1 and λ2 are reversed to be 1 and 1/3, respectively, in comparison to the previous example, while the other rates are kept unchanged as presented in Table 4.8.

**TABLE 4.8: Failure Rates for the Fifth Case**

| **Failure rate** | λ1 | λ2 | λ3 | λ4 | λ5 | λ6 |
|---|---|---|---|---|---|---|
| | 1 | 1/3 | 1/6 | | | |

Accordingly, in Fig. 4.6, it is observed that there are nearly just two reliability values. Scenarios (100,101,110 and 111) have a higher reliability than scenarios (000, 001, 010 and 0110).

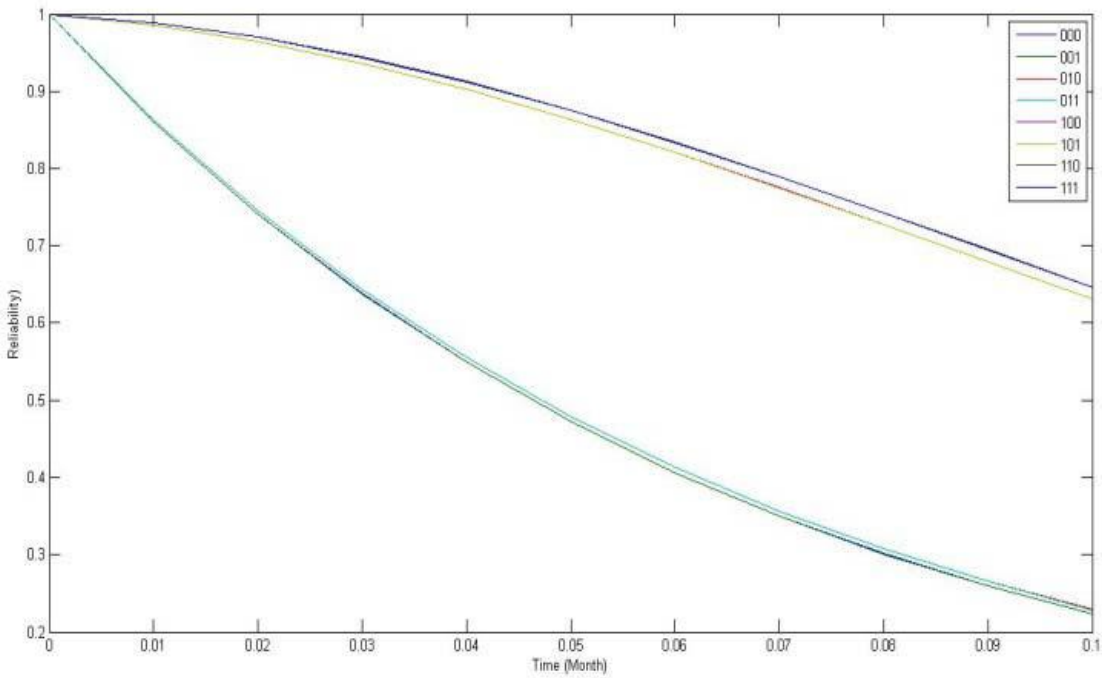



**Figure 4.6: Reliability analysis of the Fifth Case.**

Here, the decision would be to choose scenario 100 as the system reliability will not increase by the addition of TMR to the wireless sensor elements and PRP.

The failure rates λ1 and λ2 are decreased to be 0.1 and ½ (in comparison to Figure 4.4), respectively, in order minimize their dominance over the model, while the other rates are kept unchanged as presented in Table 4.9.

**TABLE 4.9: Failure Rates for the Sixth Case**

| **Failure rate** | λ1 | λ2 | λ3 | λ4 | λ5 | λ6 |
|---|---|---|---|---|---|---|
| | 0.1 | ½ | 1/24 | 1/6 | | |

As for this case, as shown in Figure 4.7, less dominancy of λ1 and λ2 led to nearly equal graphical distance between each type; however, it's a little higher between 100 and 011.

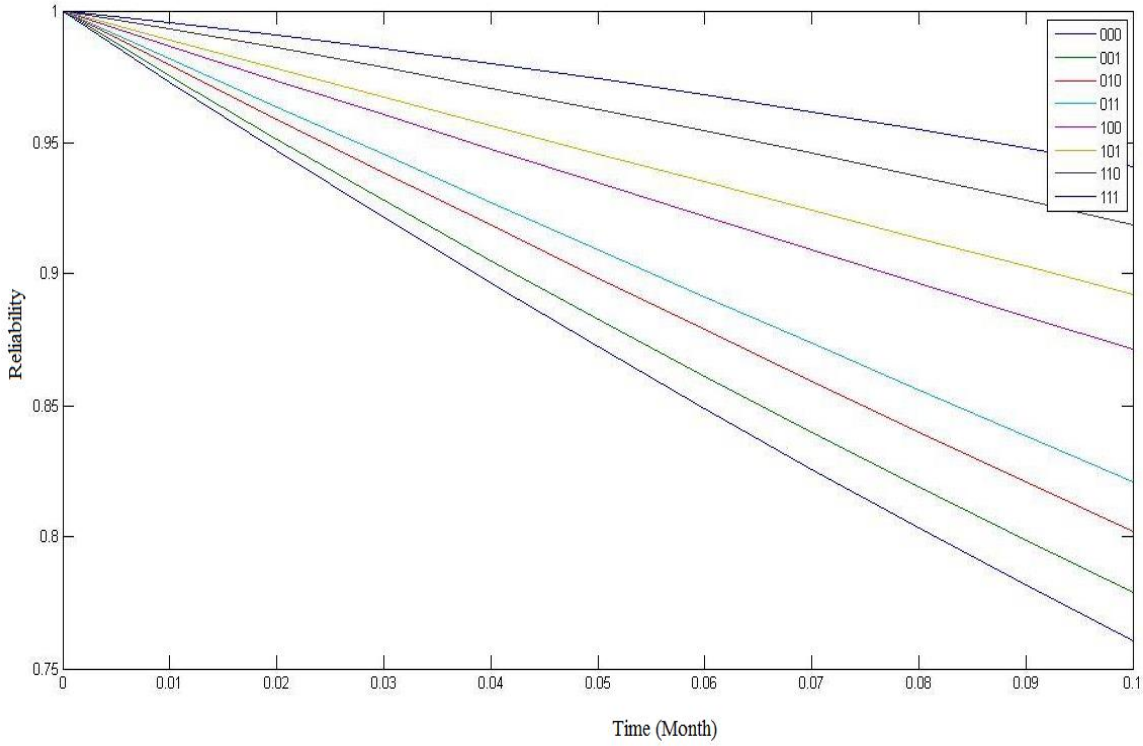



**Figure 4.7: Reliability analysis of the Sixth Case.**

The Failure rates in this example are the same as the previous case except for λ1=1 instead of 0.1 to examine the order of reliability when increasing the value of λ1. There is a huge reliability gap among the types with no wired reliability (001,000,011,010) and the types with wired reliability (101,100,111,110) due to λ1 dominancy (High value of λ1) as shown in Figure 4.8.

**TABLE 4.10: Failure Rates for the Seventh Case**

| **Failure rate** | λ1 | λ2 | λ3 | λ4 | λ5 | λ6 |
|---|---|---|---|---|---|---|
| | 1 | ½ | 1/24 | 1/6 | | |

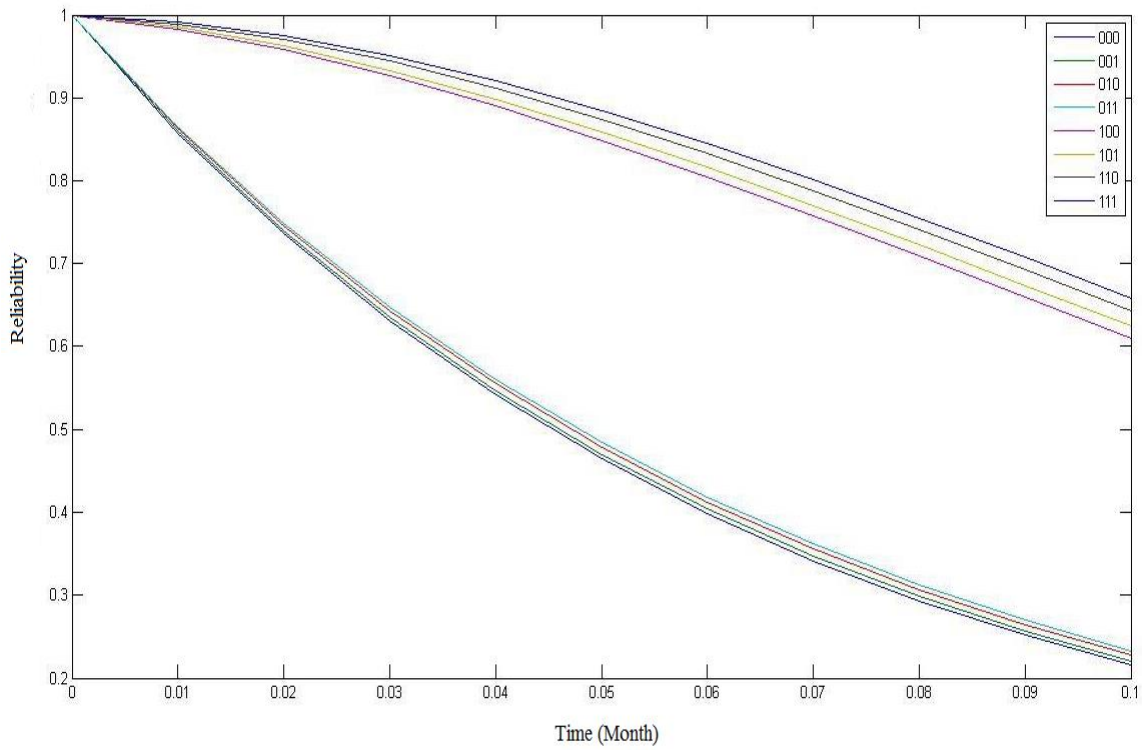



**Figure 4.8: Reliability analysis of the Seventh Case.**

In respect to the sixth case, in order to check the effect of increasing λ2 in comparison to the other failure rates in this example, the value of λ2 is increased to 1 instead of 0.5, while the other failure rates are kept unchanged. It is observed that as long the scenario has more fault tolerance techniques, the value of reliability also increases. Thus in figure 4.9 it is shown that 000 & 001 are near to each other with the lowest reliability, then the reliability increases at 010, 100, 011,101 and finally 110 & 111 with the highest reliability.

**TABLE 4.11: Failure Rates for the Eighth Case**

| **Failure rate** | λ1 | λ2 | λ3 | λ4 | λ5 | λ6 |
|---|---|---|---|---|---|---|
| | 0.1 | 1 | 1/24 | 1/6 | | |

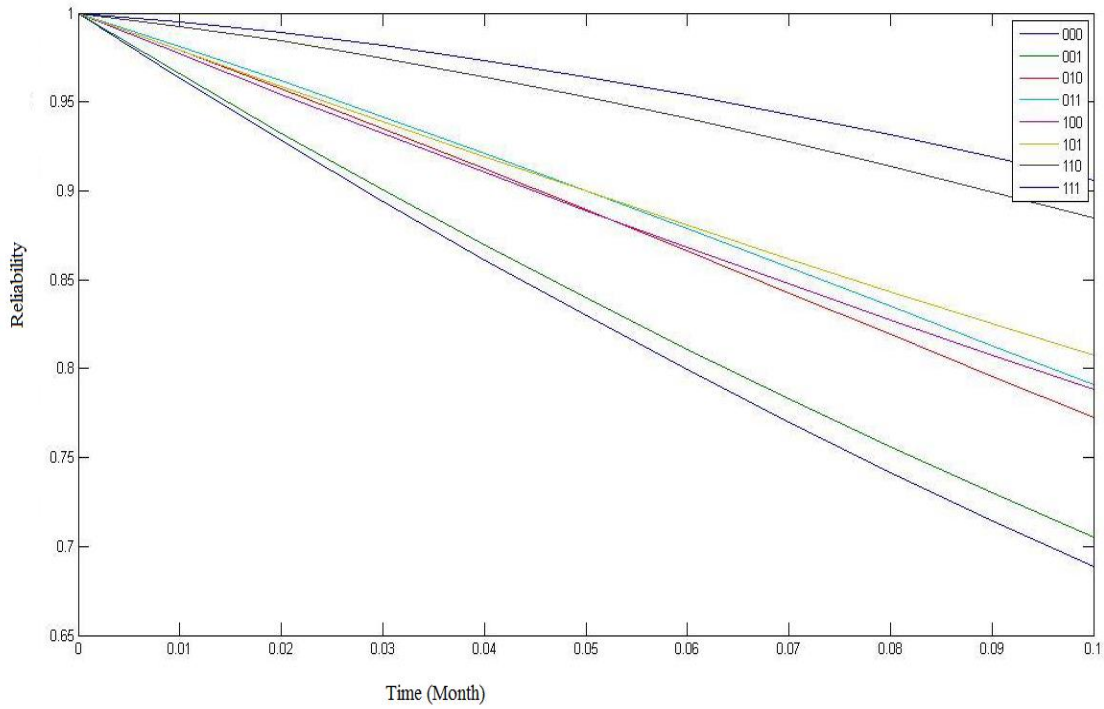



**Figure 4.9: Reliability analysis of the Eighth Case.**

The entire failure rates at following example are the same as the sixth case except for λ1 that is increased to 1 instead of 0.5 to equalize the effect of λ2. All the scenarios order is the same as sixth case except for 011 that is swapped with 100due to λ1 dominancy. Figure 4.10 reveals that Any Reliability type that each two consecutive reliability types have nearly the same value (000& 001), (010 & 011), (100 &101), (110 & 111). Moreover, there is a big reliability gap among the types with no TMR on wired sensors (001,000,011,010) and the types with TMR on wired sensors (101,100,111,110).

**TABLE 4.12: Failure Rates for the Ninth Case**

| **Failure rate** | λ1 | λ2 | λ3 | λ4 | λ5 | λ6 |
|---|---|---|---|---|---|---|
| | 1 | 1 | 1/24 | 1/6 | | |

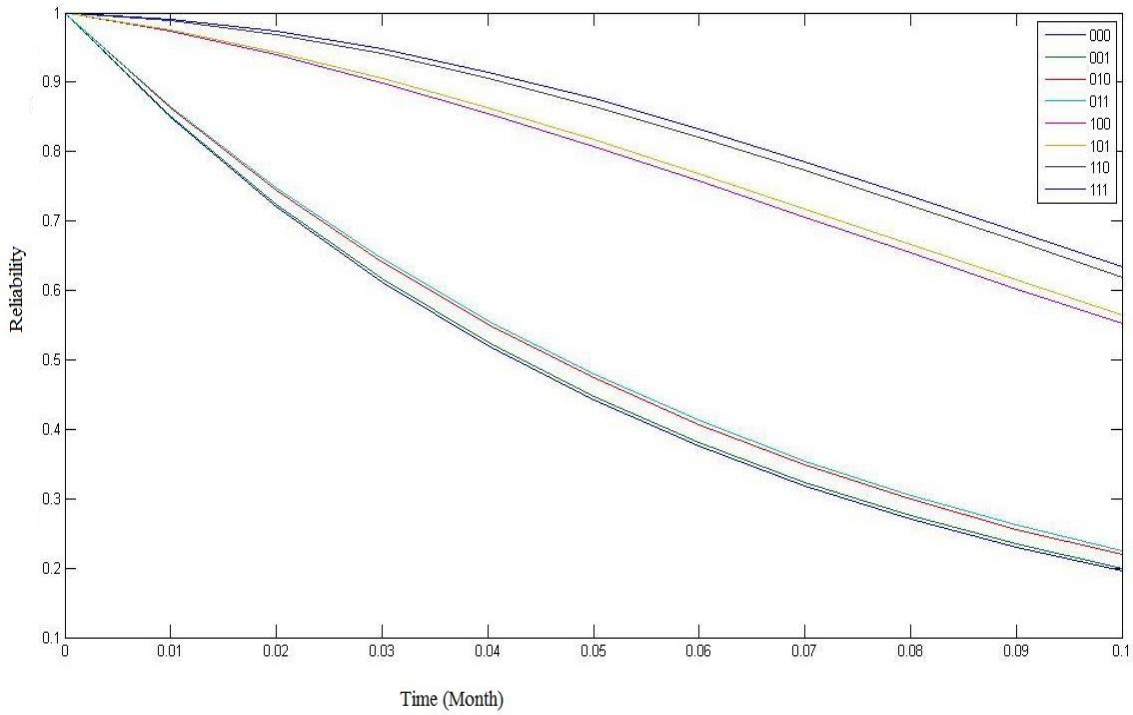



**Figure 4.10: Reliability analysis of the Ninth Case.**

The tenth case in our analysis is different than the last case of that λ3 is increased to 0.5. This higher λ3 leads to less PRP reliability. Therefore, 001 becomes less than (000), while (111) less than (110). It is observed from figure 4.11 that there is a big reliability gap among the types with no wired reliability (001,000,011,010) & the types with wired reliability (101,100,111,110) due to λ1 dominancy.

**TABLE 4.13: Failure Rates for the Tenth Case**

| **Failure rate** | λ1 | λ2 | λ3 | λ4 | λ5 | λ6 |
|---|---|---|---|---|---|---|
| | 1 | 1 | 1/2 | 1/6 | | |

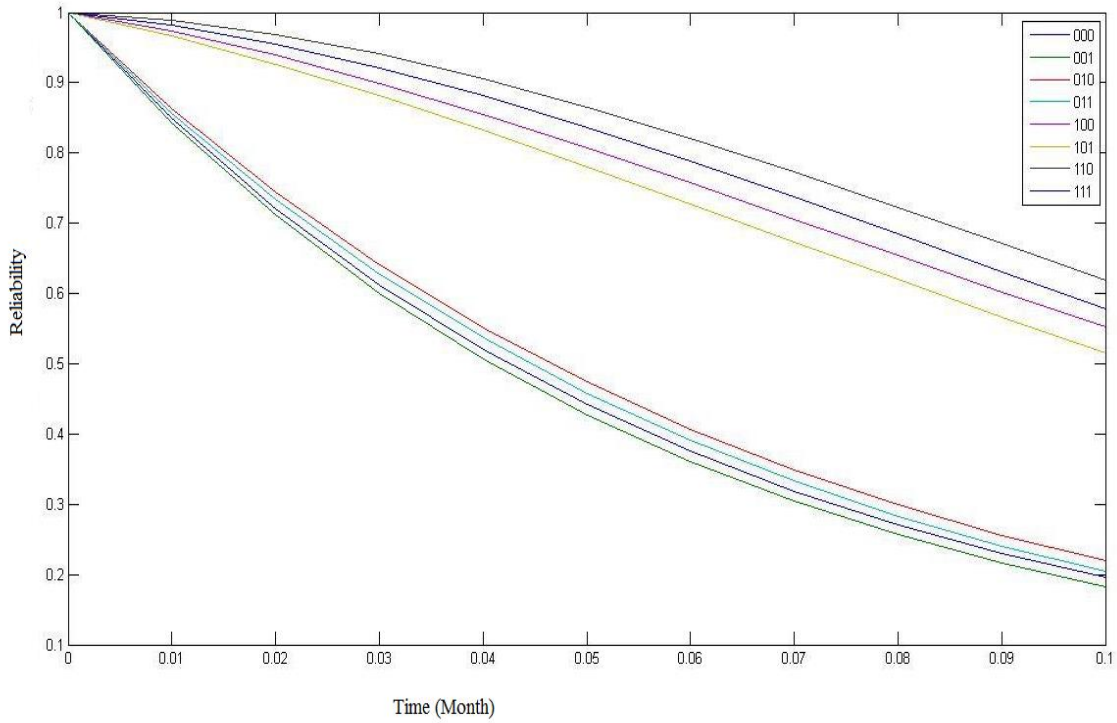



**Figure 4.11: Reliability analysis of the Tenth Case.**

## 4.3 Summary

In this Chapter, architecture of four sensors, four actuators and a supervisor is modeled. Also, fault tolerance in wired and wireless S2A NCS using Gigabit Ethernet and Wi-Fi protocols is applied. The number of nodes is adjusted to enable TMR on wired and wireless nodes and PRP on wireless nodes FT. The simulation results, using the OMNeT++ simulator, show that the system meets the control constraints with no packet loss or over-delayed packets. Eight fault-tolerance scenarios are studied to make it easier for the user to choose the most efficient fault-tolerance technique; taking cost into consideration. The studied cases show that the failure rates of the wired nodes are the most dominant for the overall system reliability, as it contains the largest nodes number in comparison to other nodes. Moreover, the scenarios 100 (Wired TMR only) or 110 (Wired TMR and Wireless TMR) are the most efficient and the best options for a user from a reliability versus cost point of view.

# Chapter 5

# Conclusions and Future Work

A new NCS (S2A) model is developed. It is composed of 16 sensors, one Supervisor and four actuators (16-1-4). Two of the 16 sensors in addition to two of the four actuators are wirelessly linked. The rest of the 14 sensors, the supervisor and two actuators are all wired connected. The wireless nodes communicate on-top of the 802.11/g (WiFi) protocol. On the other hand, the wired nodes communicate via Gigabit Ethernet.

It is revealed through OMNeT++ simulations that the investigated model does not suffer any packet loss. Also, the end-to-end delays meet the 36ms deadline at wireless nodes and 694µs deadline at wired nodes. In order to check the robustness of the system, it is subjected to wireless, wired and mixed interferences. It is observed that the developed NCS model is able to withstand up to 12.8KB wireless interference while 12.7KB plus 1MB mixed interference.

Queuing, propagation, encapsulation and de-capsulation delays are taken into consideration to calculate the end-to-end delays. All results are subjected to a 95% confidence analysis.

Work in this area could be extended to calculate the end-to-end delay and packet loss in case of applying motion on the wireless nodes instead of being stationary.

Moreover, the idea of applying fault tolerance on wired and wireless S2A NCS using Gigabit Ethernet and WiFi protocols are tackled. A new model is developed that consists of 16 sensors, 4 actuators and a supervisor. The number of wired sensors is

triplicated, while the number of wireless nodes is doubled to enable TMR on wired and wireless nodes and PRP on wireless nodes FT.

Using the OMNeT++ simulator, the system was shown to meet the system control constraints with no packet loss or over-delayed packets. To make it easier for the user to choose the most efficient fault-tolerance technique, taking cost into consideration, case studies for eight fault-tolerance scenarios are studied. Thus, by just substituting the values of the failure rates per one month for the different components into the presented equations, a graph can be plotted to compare the reliability of the system while employing different combinations of the studied fault tolerance techniques. The studied cases show that the failure rates of the wired nodes are the most dominant for the overall system reliability, as it contains the largest nodes number in comparison to other nodes.

Work in this area could be extended by taking into account the power consumption of the NCS system and its tradeoff with performance.

# References List

[ABB, 2009] ABB, "Technical Description WISA Wireless Interface for Sensors and Actuators: Planning Installation and Commissioning Guidelines," March 2009.

[Abdel Reheem, 2012] E.E. Abdel Reheem, Y.I. El Faramawy, H.H. Halawa, M.A. Ibrahim, A. Elhamy, T.K. Refaat, R.M. Daoud and H.H. Amer, "On the Effect of Interference on Wi-Fi-Based Wireless Networked Control Systems", Proceedings of the IEEE, IET International Symposium on Communication Systems, Networks and Digital Signal Processing CSNDSP, Poznan, Poland, July 2012.

[Abd-El-barr, 2006] Mostafa Abd-El-barr, Design And Analysis of Reliable And Fault-tolerant Computer Systems, Imperial College Press, London, UK, 2006

[Boggia, 2009] G. Boggia, P. Camarda, V. Divittorio and L.A. Grieco, "A simulation-based performance evaluation of Wireless Networked Control Systems", Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation ETFA, Mallorca, Spain, September 2009.

[Bradley, 2001] EtherNet/IP Performance and Application Guide (2001), Allen-Bradley, Rockwell Automation, Application Solution.

[Cena, 2009] G. Cena, A. Valenzano, C. Zunin and, L Seno "Evaluation of Real-Time Communication Performance in QoS-Enabled Infrastructure WLANs," Proceedings of the 14th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Mallorca-Spain, September 2009.

[Daoud, 2003] R.M. Daoud, H.M. ElSayed, H.H. Amer and S.Z. Eid, “Performance of Fast and Gigabit Ethernet in Networked Control Systems,” Proceedings of the IEEE Midwest Symposium on Circuits and Systems MWSCAS, Cairo, Egypt, December 2003, Vol. 1, pp. 505-508.

[Ferreira, 2002] J. Ferreira, P. Pedreiras, L. Almeida, and J. Fonseca, “Achieving Fault-Tolerance in FTTCAN,” Proceedings of the IEEE International workshop on Factory Communication Systems WFCS, Vasteras, Sweden, August 2002.

[Heer, 2015] Heer T (2015) Parallel Redundancy Protocol Notably Improves Industrial Wireless Reliability (White Paper). [Online document] available http://belden.picturepark.com/Website/Download.aspx?DownloadToken=ee314318-b4e3-4efa-a739-b2e9c69f0a7e&Purpose=AssetManager&mime-type=application/pdf

[Hong, 1998] Hong Y (1998) Networked Control Systems. [Online document] available http://www.enme.umd.edu/ice lab/ncs/ncs.html

[IEC1, 2014] IEC 61784-1 (2014), available at: www.iec.ch.

[IEC2, 2014] IEC 61784-2 (2014), available at: www.iec.ch

[IEEE, 2001] IEEE Standard for Ethernet," in IEEE Std 802.3-2015 (Revision of IEEE Std 802.3-2012) , vol., no., pp.1-4017, March 4 2016

[IEEE, 2012] IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," in IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007) , vol., no., pp.1-2793, March 29 2012.

[Lee, 2001] S.H.Lee and K. H.Cho. “Congestion Control of High-Speed Gigabit-Ethernet Networks for Industrial Applications,” Proceedings of the IEEE ISIE, Pusan, Korea, June 2001, pp. 270-275.

[Lian, 2001] F.L. Lian, J.R. Moyne, and D.M. Tilbury, “Networked control systems toolkit: A simulation package for analysis and design of control systems with network communication,” Tech. Rep., UM-ME-01-04, July 2001.

[Marti, 2001] P. Marti, J.M. Fuertes, and G. Fohler, “An integrated approach to realtime distributed control systems over fieldbuses,” Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation ETFA, Antipes/Juan les Pins, France, October 2001.

[Moustafa, 2013] E. Moustafa, H. Halawa, R. Daoud and H. Amer, “Sensor Actuator Ethernet-Based Networked Control Systems,” Proceedings of the International IEEE Conference on Sciences & Computer Engineering STA, Sousse, Tunisia, December 2013, pp.530-534.

[Moustafa, 2014] Eslam Moustafa, Hassan Halawa, Ramez Daoud and Hassanein Amer, “Evaluating the Performance of Fault-Tolerant S2A vs. In-Loop Controller Models for Ethernet-Based NCS”, Intelligent Control and Automation, Scientific Research Publishing, May 2014, Vol. 5, No. 2, pp. 81-90

[Nilsson, 1998] J. Nilsson, "Real-Time control systems with delays," PhD thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, 1998.

[ODVA1, 2007] ODVA, Volume 1: CIP Common (2007). http://www.odva.org/10_2/03_events/03_ethernet-homepage.htm.

[ODVA2, 2007] ODVA, Volume 2: EtherNet/IP Adaptation on CIP (). http://www.odva.org/10_2/03_events/03_ethernet-homepage.htm

[ODVA3, 2016] Official Site for Control Net. http://odva.org/default.aspx?tabid=244.

[OMNeT, 2016] Official Site For OMNeT++: www.omnetpp.org/

[Pariale, 2006] L. Parziale, et al., “TCP/IP Tutorial and Technical Overview,” IBM: Internal Technical and Support Organization, 2006.

[Pedreiras, 2002] P. Pedreiras, L. Almeida, and P. Gai, "The FTT-Ethernet protocol: merging flexibility, timeliness and efficiency," Proceedings of the IEEE Euromicro Conference on Real-Time Systems ECRTS, Vienna, Austria, June 2002.

[Pinheiro, 2009] M. Pinheiro, S. Sampaio, P. Souto and F. Vasques, "A DHT-based approach for Path Selection and Message Forwarding in IEEE 802.11s Industrial Wireless Mesh Networks," Proceedings of the 14th IEEE International Conference on Emerging Technologies and FactoryAutomation, ETFA, Mallorca-Spain, September 2009.

[Proenza, 2012] J. Proenza, M. Barranco, G. Rodriguez-Navas, D. Gessner, F. Guardiola, and L. Almeida, "The design of the CANbids architecture," Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation ETFA, Krakow, Poland, September 2012.

[PROFIBUS,2016] Official Site For PROFIBUS and PROFINET. http://www.profibus.com

[Rachana, 2008] Rachana A. Gupta and Mo-Yuen Chow, Overview of Networked Control Systems, Springer 2008.

[Refaat, 2010] Tarek K. Refaat, Ramez M. Daoud, Hassanein H. Amer and Esraa A. Makled, "WiFi Implementation of Wireless Networked Control Systems", Proceedings of the Seventh International Conference on Networked Sensing Systems INSS, Kassel, Germany, June 2010, pp. 145-148.

[Shooman, 2002] M.L. Shooman, Reliability of Computer Systems and Networks: Fault Tolerance, Analysis and Design, Wiley, New York, 2002.

[Skeie, 2002] T. Skeie, S. Johannessen, and C. Brunner, "Ethernet in substation automation," IEEEControl Syst., Vol. 22, No. 3, 2002.

[Steigmann, 2006] R. Steigmann and J. Endresen, "Introduction to WISA: WISA – Wireless Interface for Sensors and Actuators," White paper, ABB, July 2006.

[Steinhammer, 2007] K. Steinhammer and A. Ademaj, “Hardware implementation of the Time-Triggered Ethernet controller,” Embedded System Design: Topics, Techniques and Trends, Vol. 231.Springer Boston, 2007.

[Tanenbaum, 2002] Andrew Tanenbaum. 2002. Computer Networks (4th ed.). Prentice Hall Professional Technical Reference.

# Appendix: MATLAB Codes of Reliability Analysis Cases

## Case 1

```
t= 0:0.01:0.1;
a = ((exp(-0.167*t).^14) .* ((exp(-0.167*t)) .* (exp(-0.167*t))) .^2);
b = ((exp(-0.167*t).^14) .* ((exp(-0.167*t) .* exp(-0.167*t)) .* ((2*exp(-0.167*t)) - (exp(-
0.33*t)))) .^2);
c = ((exp(-0.167*t).^14) .* ((exp(-0.167*t) .* exp(-0.167*t)) .* ((3*exp(-0.33*t)) - (2*exp(-
0.5*t)))) .^2);
d = ((exp(-0.167*t).^14) .* (((3*exp(-0.33*t)) - (2*exp(-0.5*t))) .* ((2*exp(-0.167*t)) - (exp(-
0.33*t))) .* (exp(-0.167*t)) .* (exp(-0.167*t))) .^2);
e = ((((exp(-0.167*t)) .* (exp(-0.167*t))).^2) .* ((3*exp(-0.33*t)) - (2*exp(-0.5*t))).^14);
f = ((((3*exp(-0.33*t)) - (2*exp(-0.5*t))).^14) .* (((2*exp(-0.167*t)) - (exp(-0.33*t))) .*
(exp(-0.167*t)) .* (exp(-0.167*t))) .^2);
g = ((((3*exp(-0.33*t)) - (2*exp(-0.5*t))).^14) .* (((3*exp(-0.33*t)) - (2*exp(-0.5*t))) .*
(exp(-0.167*t)) .* (exp(-0.167*t)))  .^2);
h = ((((3*exp(-0.33*t)) - (2*exp(-0.5*t))).^14) .* (((3*exp(-0.33*t)) - (2*exp(-0.5*t))) .*
(exp(-0.167*t)) .* (exp(-0.167*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))))  .^2);
```

## Case 2

```
t= 0:0.01:0.1;
a = ((exp(-1*t).^14) .* ((exp(-1*t)) .* (exp(-0.167*t))) .^2);
b = ((exp(-1*t).^14) .* ((exp(-1*t) .* exp(-0.167*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))))
.^2);
c = ((exp(-1*t).^14) .* ((exp(-0.167*t) .* exp(-0.167*t)) .* ((3*exp(-2*t)) - (2*exp(-3*t))))
.^2);
d = ((exp(-1*t).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* ((2*exp(-0.167*t)) - (exp(-0.33*t)))
.* (exp(-0.167*t)) .* (exp(-0.167*t))) .^2);
e = ((((exp(-0.167*t)) .* (exp(-1*t))).^2) .* ((3*exp(-2*t)) - (2*exp(-3*t))).^14);
f = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((2*exp(-0.167*t)) - (exp(-0.33*t))) .* (exp(-
1*t)) .* (exp(-0.167*t))) .^2);
g = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* (exp(-
0.167*t)) .* (exp(-0.167*t)))  .^2);
h = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* (exp(-
0.167*t)) .* (exp(-0.167*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))))  .^2);
```

# Case 3

```
t= 0:0.01:0.1;
a = ((exp(-1*t).^14) .* ((exp(-1*t)) .* (exp(-0.167*t))) .^2);
b = ((exp(-1*t).^14) .* ((exp(-1*t) .* exp(-0.042*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))))
.^2);
c = ((exp(-1*t).^14) .* ((exp(-0.167*t) .* exp(-0.167*t)) .* ((3*exp(-2*t)) - (2*exp(-3*t))))
.^2);
d = ((exp(-1*t).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* ((2*exp(-0.167*t)) - (exp(-0.33*t)))
.* (exp(-0.042*t)) .* (exp(-0.167*t))) .^2);
e = ((((exp(-0.167*t)) .* (exp(-1*t))).^2) .* ((3*exp(-2*t)) - (2*exp(-3*t))).^14);
f = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((2*exp(-0.167*t)) - (exp(-0.33*t))) .* (exp(-
1*t)) .* (exp(-0.042*t))) .^2);
g = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* (exp(-
0.167*t)) .* (exp(-0.167*t)))  .^2);
h = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* (exp(-
0.042*t)) .* (exp(-0.167*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))))  .^2);
```

# Case 4

```
t= 0:0.01:0.1;
a = ((exp(-0.33*t).^14) .* ((exp(-1*t)) .* (exp(-0.167*t))) .^2);
b = ((exp(-0.33*t).^14) .* ((exp(-1*t) .* exp(-0.167*t)) .* ((2*exp(-0.167*t)) - (exp(-
0.33*t)))) .^2);
c = ((exp(-0.33*t).^14) .* ((exp(-0.167*t) .* exp(-0.167*t)) .* ((3*exp(-2*t)) - (2*exp(-3*t))))
.^2);
d = ((exp(-0.33*t).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* ((2*exp(-0.167*t)) - (exp(-
0.33*t))) .* (exp(-0.167*t)) .* (exp(-0.167*t))) .^2);
e = ((((exp(-0.167*t)) .* (exp(-1*t))).^2) .* ((3*exp(-0.67*t)) - (2*exp(-1*t))).^14);
f = ((((3*exp(-0.67*t)) - (2*exp(-1*t))).^14) .* (((2*exp(-0.167*t)) - (exp(-0.33*t))) .* (exp(-
1*t)) .* (exp(-0.167*t))) .^2);
g = ((((3*exp(-0.67*t)) - (2*exp(-1*t))).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* (exp(-
0.167*t)) .* (exp(-0.167*t)))  .^2);
h = ((((3*exp(-0.67*t)) - (2*exp(-1*t))).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* (exp(-
0.167*t)) .* (exp(-0.167*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))))  .^2);
```

# Case 5

```
t= 0:0.01:0.1;
a = ((exp(-1*t).^14) .* ((exp(-0.33*t)) .* (exp(-0.167*t))) .^2);
b = ((exp(-1*t).^14) .* ((exp(-0.33*t) .* exp(-0.167*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t)))) .^2);
c = ((exp(-1*t).^14) .* ((exp(-0.167*t) .* exp(-0.167*t)) .* ((3*exp(-0.67*t)) - (2*exp(-1*t)))) .^2);
d = ((exp(-1*t).^14) .* (((3*exp(-0.67*t)) - (2*exp(-1*t))) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))) .* (exp(-0.167*t)) .* (exp(-0.167*t))) .^2);
e = ((((exp(-0.167*t)) .* (exp(-0.33*t))).^2) .* ((3*exp(-2*t)) - (2*exp(-3*t))).^14);
f = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((2*exp(-0.167*t)) - (exp(-0.33*t))) .* (exp(-0.33*t)) .* (exp(-0.167*t))) .^2);
g = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((3*exp(-0.67*t)) - (2*exp(-1*t))) .* (exp(-0.167*t)) .* (exp(-0.167*t))) .^2);
h = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((3*exp(-0.67*t)) - (2*exp(-1*t))) .* (exp(-0.167*t)) .* (exp(-0.167*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t)))) .^2);
```

# Case 6

```
t= 0:0.01:0.1;
a = ((exp(-0.1*t).^14) .* ((exp(-0.5*t)) .* (exp(-0.167*t))) .^2);
b = ((exp(-0.1*t).^14) .* ((exp(-0.5*t) .* exp(-0.042*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t)))) .^2);
c = ((exp(-0.1*t).^14) .* ((exp(-0.167*t) .* exp(-0.167*t)) .* ((3*exp(-1*t)) - (2*exp(-1.5*t)))) .^2);
d = ((exp(-0.1*t).^14) .* (((3*exp(-1*t)) - (2*exp(-1.5*t))) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))) .* (exp(-0.167*t)) .* (exp(-0.042*t))) .^2);
e = ((((exp(-0.167*t)) .* (exp(-0.5*t))).^2) .* ((3*exp(-0.2*t)) - (2*exp(-0.3*t))).^14);
f = ((((3*exp(-0.2*t)) - (2*exp(-0.3*t))).^14) .* (((2*exp(-0.167*t)) - (exp(-0.33*t))) .* (exp(-0.5*t)) .* (exp(-0.042*t))) .^2);
g = ((((3*exp(-0.2*t)) - (2*exp(-0.3*t))).^14) .* (((3*exp(-1*t)) - (2*exp(-1.5*t))) .* (exp(-0.167*t)) .* (exp(-0.167*t))) .^2);
h = ((((3*exp(-0.2*t)) - (2*exp(-0.3*t))).^14) .* (((3*exp(-1*t)) - (2*exp(-1.5*t))) .* (exp(-0.167*t)) .* (exp(-0.042*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t)))) .^2);
```

# Case 7

```
 t= 0:0.01:0.1;
a = ((exp(-1*t).^14) .* ((exp(-0.5*t)) .* (exp(-0.167*t))) .^2);
b = ((exp(-1*t).^14) .* ((exp(-0.5*t) .* exp(-0.042*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))))
.^2);
c = ((exp(-1*t).^14) .* ((exp(-0.167*t) .* exp(-0.167*t)) .* ((3*exp(-1*t)) - (2*exp(-1.5*t))))
.^2);
d = ((exp(-1*t).^14) .* (((3*exp(-1*t)) - (2*exp(-1.5*t))) .* ((2*exp(-0.167*t)) - (exp(-
0.33*t))) .* (exp(-0.167*t)) .* (exp(-0.042*t))) .^2);
e = ((((exp(-0.167*t)) .* (exp(-0.5*t))).^2) .* ((3*exp(-2*t)) - (2*exp(-3*t))).^14);
f = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((2*exp(-0.167*t)) - (exp(-0.33*t))) .* (exp(-
0.5*t)) .* (exp(-0.042*t))) .^2);
g = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((3*exp(-1*t)) - (2*exp(-1.5*t))) .* (exp(-
0.167*t)) .* (exp(-0.167*t)))  .^2);
h = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((3*exp(-1*t)) - (2*exp(-1.5*t))) .* (exp(-
0.167*t)) .* (exp(-0.042*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))))  .^2);
```

# Case 8

```
t= 0:0.01:0.1;
a = ((exp(-0.1*t).^14) .* ((exp(-1*t)) .* (exp(-0.167*t))) .^2);
b = ((exp(-0.1*t).^14) .* ((exp(-1*t) .* exp(-0.042*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))))
.^2);
c = ((exp(-0.1*t).^14) .* ((exp(-0.167*t) .* exp(-0.167*t)) .* ((3*exp(-2*t)) - (2*exp(-3*t))))
.^2);
d = ((exp(-0.1*t).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* ((2*exp(-0.167*t)) - (exp(-
0.33*t))) .* (exp(-0.167*t)) .* (exp(-0.042*t))) .^2);
e = ((((exp(-0.167*t)) .* (exp(-1*t))).^2) .* ((3*exp(-0.2*t)) - (2*exp(-0.3*t))).^14);
f = ((((3*exp(-0.2*t)) - (2*exp(-0.3*t))).^14) .* (((2*exp(-0.167*t)) - (exp(-0.33*t))) .* (exp(-
1*t)) .* (exp(-0.042*t))) .^2);
g = ((((3*exp(-0.2*t)) - (2*exp(-0.3*t))).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* (exp(-
0.167*t)) .* (exp(-0.167*t)))  .^2);
h = ((((3*exp(-0.2*t)) - (2*exp(-0.3*t))).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* (exp(-
0.167*t)) .* (exp(-0.042*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))))  .^2);
```

# Case 9

```
t= 0:0.01:0.1;
a = ((exp(-1*t).^14) .* ((exp(-1*t)) .* (exp(-0.167*t))) .^2);
b = ((exp(-1*t).^14) .* ((exp(-1*t) .* exp(-0.042*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))))
.^2);
c = ((exp(-1*t).^14) .* ((exp(-0.167*t) .* exp(-0.167*t)) .* ((3*exp(-2*t)) - (2*exp(-3*t))))
.^2);
d = ((exp(-1*t).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* ((2*exp(-0.167*t)) - (exp(-0.33*t)))
.* (exp(-0.167*t)) .* (exp(-0.042*t))) .^2);
e = ((((exp(-0.167*t)) .* (exp(-1*t))).^2) .* ((3*exp(-2*t)) - (2*exp(-3*t))).^14);
f = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((2*exp(-0.167*t)) - (exp(-0.33*t))) .* (exp(-
1*t)) .* (exp(-0.042*t))) .^2);
g = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* (exp(-
0.167*t)) .* (exp(-0.167*t)))  .^2);
h = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* (exp(-
0.167*t)) .* (exp(-0.042*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))))  .^2);
```

# Case 10

```
t= 0:0.01:0.1;
a = ((exp(-1*t).^14) .* ((exp(-1*t)) .* (exp(-0.167*t))) .^2);
b = ((exp(-1*t).^14) .* ((exp(-1*t) .* exp(-0.5*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t)))) .^2);
c = ((exp(-1*t).^14) .* ((exp(-0.167*t) .* exp(-0.167*t)) .* ((3*exp(-2*t)) - (2*exp(-3*t))))
.^2);
d = ((exp(-1*t).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* ((2*exp(-0.167*t)) - (exp(-0.33*t)))
.* (exp(-0.167*t)) .* (exp(-0.5*t))) .^2);
e = ((((exp(-0.167*t)) .* (exp(-1*t))).^2) .* ((3*exp(-2*t)) - (2*exp(-3*t))).^14);
f = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((2*exp(-0.167*t)) - (exp(-0.33*t))) .* (exp(-
1*t)) .* (exp(-0.5*t))) .^2);
g = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* (exp(-
0.167*t)) .* (exp(-0.167*t)))  .^2);
h = ((((3*exp(-2*t)) - (2*exp(-3*t))).^14) .* (((3*exp(-2*t)) - (2*exp(-3*t))) .* (exp(-
0.167*t)) .* (exp(-0.5*t)) .* ((2*exp(-0.167*t)) - (exp(-0.33*t))))  .^2);
```

# Published and Accepted For Publication Papers Out Of the Thesis Work

- M. Toubar, H. Halawa, R. Daoud and H. Amer, "Wired & Wireless S2A for NCS - A Case Study", Proceedings of the 16th International conference on Sciences and Techniques of Automatic control STA'2015, Monastir, Tunisia, 2015

- M. Toubar, H. Halawa, R. Daoud and H. Amer, “Sensor/Channel Redundancy for Sensor-to-Actuator Networked Control Systems ", accepted for publication in the proceedings of International Siberian Conference on Control and Communications SIBCON 2017, Astana, Kazakhstan, 2017.