5-31-2020

A novel fault tolerant scheme and reliability modeling for a wireless modular prosthetic limb

Mayar Mohamed Saeed
The American University in Cairo

THE AMERICAN UNIVERSITY IN CAIRO

الجامعة الأمريكية بالقاهرة

School of Sciences and Engineering

A Novel Fault Tolerant Scheme and Reliability Modeling for a Wireless Modular Prosthetic Limb

A Thesis Submitted to the

Electronics and Communications Engineering Department

In partial fulfillment of the requirements for

The degree of Master of Science

Submitted by: Mayar Mohamed Saeed

Under the supervision of

Prof. Hassanein Amer and Dr. Ramez Daoud

Submission Date: 05/May/2020

ACKNOWLEDGMENTS

At the beginning, I would like to thank my supervisors: Prof. Hassanein Amer and Dr. Ramez Daoud for their extraordinary support during my degree and throughout the courses and the thesis. Their endless support cannot be simply described in words.

And, I would like to acknowledge my appreciation for: Prof. Hany ElSayed, Prof. Moustafa Arafa, Dr. Ihab Adly, Eng. Gehad Alkady, Eng. Malak El-Salamouny and Eng. Manar Negm, for supporting me with their knowledge. And, thanks to Eng. Mohamed Salem, Eng. Mohamed Moustafa and Eng. Roshdi for facilitating my lab work by providing IT support.

Also, I would really like to thank my family and friends for the incredible emotional support they have been giving me which I really needed and for baring me during the whole degree and accepting that I was not available most of the time.

Publication out of this thesis

[Medhat 2020] M.M. Medhat, R.M. Daoud and H.H. Amer, "A Novel Fault-Tolerant Scheme for a Wireless Modular Prosthetic Limb", Proceedings of the IEEE International Conference on Factory Communication Systems WFCS, Porto, Portugal, April 2020.

Table of Content

List of Figures

List of Tables

List of Abbreviations

BMI– Brain-Machine Interface

DARPA– Defense Advanced Research Projects Agency

DOF– Degrees of Freedom

FT– Fault Tolerance

FTP –File Transfer Protocol

ISM– Industrial, Scientific and Medical

JHUAPL– Johns Hopkins University Applied Physics Laboratory

LMCs– Large Motor Controllers

MPL– Modular Prosthetic Limb

MTBF– Mean Time Between Failures

MTTF– Mean Time To Failures

MTTR– Mean Time To Repair

NCS – Networked Control System

SMCs– Small Motor Controllers

SPN– Stochastic Petri Nets

UDP–User Datagram Protocol

WBAN – Wireless Body Area Network

Abstract

This thesis presents a study for a wireless scheme for the Johns Hopkins modular prosthetic limb (MPL) as well as a demonstration for a novel fault tolerant scheme to further improve the arm's reliability. Currently, the innovative prosthetic limbs that depend on Network Control Systems are wired devices which have two main disadvantages; the wear and tear issue as well as the mobility limitation problem. Hence, the same function can be done by replacing the wired models by Wireless Body Area Networks (WBANs) in order to avoid the wear and tear and mobility issues. Furthermore, the prosthetic limbs are life-saving and real time medical devices which demand high reliability as failure may lead to harsh consequences. The reputable Modular Prosthetic Limb (MPL) that is developed by Johns Hopkins applied physics laboratory is revisited in this thesis. Using RIVERBED, the wireless scheme of the Johns Hopkins arm is studied as well as a fault-tolerant model for the same arm. All scenarios undergo interference analysis and a 95% confidence analysis. The simulation results have demonstrated that the end-to-end delays are below the system's deadlines and there is zero packet loss in all scenarios; thus, the system requirements are satisfied. Further, the reliability of the system was calculated by modelling several scenarios using SHARPE. It has been proven that a system that uses a supervisor with lower specifications will have a very close reliability values to the system that uses very powerful supervisor if it was repaired after the failure of the third controller. Finally, it was proved that the motor redundancy has significantly enhanced the reliability.

Chapter 1

INTRODUCTION

The latest improvements in wireless communications and integrated circuits allowed the fabrication of small sized, light weight and ultra-low power devices. Wireless Body Area Network (WBAN) is a technology used for physiological sensing. This technology is currently very essential as there is a huge demand for improving the medical devices which will lead to enhancement of the medical and health care applications. WBANs are made of sets of sensor nodes, which are used for sampling, processing and communicating with each other, as well as actuators and motors [Otto 2005]. These nodes are located on the human body, whether on the patients' outfits or embedded in their bodies [Otto 2005]. The sensors monitor the physiological signals and send them to a specialized medical server without disturbing the daily routine of the patient [Crosby 2012] or transmit the signals to actuators to perform certain actions. There are specific system requirements for WBANs. Firstly, the sensors should have specific sampling rate. The sampling rate should be between 10Hz to 100Hz depending on the application requirements [Kaur 2011]. Secondly, most of the WBANs are battery operated, so the system should be power efficient and the battery should have a long life time. Also, the implantable sensors should be self-powered from the environment [Kaur 2011]. Moreover, the nodes must be light in weight and small sized so they won't be remarkable. The issue is that a long life time battery conflicts the need for it to be light in weight and small in size [Kaur 2011]. The most common protocols used for WBANs operate in the Industrial, Scientific and Medical (ISM) band. Therefore, there is a problem of external interference which can be between two people who have WBANs if they are close, so WBANs should be implemented in a way that tolerate interference. WBANs can

make remarkable contribution in the area prosthetic arms. Prosthetics limbs are very essential and critical medical application because every year the number of amputees increase from 150,000 to 200,000. The upper limb amputation is about 30% and 60% of arm amputees are found between the age 21 to 64 years old, while 10% are patients under 21. 70% of the amputations are caused by a trauma or accidents [Toledo 2009]. Until now, the best solution for these cases is prosthetics limbs. Since the prosthesis is an artificial extension which substitutes a limb, it should be as natural as possible.

The Defense Advanced Research Projects Agency (DARPA) called out for advanced prosthetic designs to enhance the prosthetic arms. DARPA suggested that the prosthetic limb should look, feel, weight and be flawlessly integrated as if it were a natural limb [Burck 2011]. In addition, the arm should be controlled using the user's brain, throughout the neurons. The Johns Hopkins Model [Burck 2011] is one of the nearest designs to this target; therefore, it will be used as the base for this research. One can consider the Johns Hopkins model as a Networked Control System (NCS). A Networked Control System (NCS) [Skeie 2002, Lian 2001] can be defined as a set of sensor (S), controller (K) and actuator (A) nodes communicating over a network to complete a designated control scheme. In other words, there are several nodes intercommunicating to exchange sensor's data to a control node which applies actions on certain actuators. The problem with wired NCS is that the cables of the moving parts wear out with time. Therefore, wireless protocols used to connect what is called Wireless Network Control Systems (WNCSs) are getting more attention in recent years [Steigmann 2006]. The reason is that they solve issue mentioned previously with the wired network cables. In this research, a wireless architecture of the model described in [Johannes, 2011] will be simulated. The advantages of making a wireless arm instead of the wired model are that it will prevent the problem of the cables' wear and tear and increase the arm's degrees of freedom. Low Power Wifi will be used as the wireless communication protocol to

communicate the data between the sensors, controller and actuator nodes. This architecture could be recognized as a Wireless Sensor and Actor Network (WSAN) [Martin 2011]. This research focuses on the performance of the wireless architecture of Johns Hopkins modular prosthetic arm and proposes a novel fault tolerant scheme. Also, the model reliability is studied for several scenarios. Results are very satisfactory and will be explained in details in the rest of the document.

The rest of the thesis is divided as follows: Chapter 2 is the literature review and the previous work, chapter 3 presents the results of the Riverbed simulations that was used to study the fault free and the fault tolerant scheme. Chapter 4 presents the reliability results of several scenarios. And finally, the thesis is concluded in the last chapter.

Chapter 2

LITERATURE REVIEW

2.1 PROSTHETIC ARMS SURVEY:

The types of prosthetic arms available in the market are:

- Mechanical prosthetic arm
- Electrically Powered prosthetic arm
- Myoelectric Prosthetic Arm
- Brain-Machine Interface (BMI) based Prosthetic Arm.

The first type which is the mechanical Prosthetic arms operates by the power produced from the patient's movement while the electrically powered arms use batteries [Heger 1985].

Another available type of prosthetic arms is the myoelectric prosthetics. This type of arms moves as a result of the electric signals produced by the muscles. [Toledo 2009].

Last one is the Brain Machine Interface based prosthetic (BMI). These are man-made limbs that move as a result of converting the brain signals to electrical signals [Zhang 2011].

Several arms were made for the above mentioned types. The below prototypes are the most well-known :

- Boston Elbow
- Utah Arm
- BMI prosthetic of Osaka University
- Johns Hopkins modular prosthetic limb (MPL)

Boston Elbow, which is shown in figure 2.1, is a myoelectric limb, which moves by the signals produced by the user's movement of muscles [Toledo 2009].

The way it works is that the sensors, which are located on certain muscles, transmit the data to a microprocessor which converts it into orders to the electric motor so as to move joints [Heger 1985]. This elbow has only one DOF which is the elbow flexion [Toledo 2009]. This microprocessor is able to control other devices as well as elbow, such as shoulder actuators, hands and wrists rotator [Toledo 2009].

Figure 2.1 Boston Elbow [Toledo 2009]

The Utah Arm, which is shown in figure 2.2, is myoelectric limb as well but it is designed for the patients whose amputations are over the elbow. In 1981, the primary prototype (U1) has been produced, and it was the best myoelectric artificial limb at that time [Toledo 2009]. The Utah Arm 2 (U2) was delivered in 1987, and in 2004, the microchip innovation was included into the Utah Arm 3 (U3) and this gives a superior reaction to the arm as it permits the client to move the hand and arm toward any path. Moreover, Utah Arm 3 (U3) allows parallel movements of the arm, the hand, as well as the elbow rotation [Toledo 2009].

Figure 2. 2 Utah Arm [Toledo 2009]

Another common type of prosthetics in the field of research is Brain Machine Interface (BMI) prosthetic limbs. These type of prosthetic arms depend on the communication between the brain and the artificial limb, which is done by recognizing the brain electrical activity using EEG or ElectroCorticoGraphy (ECoG) [Zhang 2011] using certain electrodes as shown in figure 2.3. The major difference between both techniques is the location of the electrodes on the human's brain; the ECoG is more complicated since the electrodes are placed right on the exposed surface of the brain, while the EEG is a more simple technique where the electrodes are placed on the surface of the scalp without the need to expose the human's brain to record electrical activity. One of the common prosthetic models that are based on ECoG technique is made at Osaka University Medical Hospital in Japan. They predicted the type of movement based on the analysis of single trial ECoGs using a vector machine algorithm [Zhang 2011]. Consequently, the required neurophysiological features are extracted for the real-time voluntary control of the BMI based prosthetic limb such as releasing and grasping of objects.

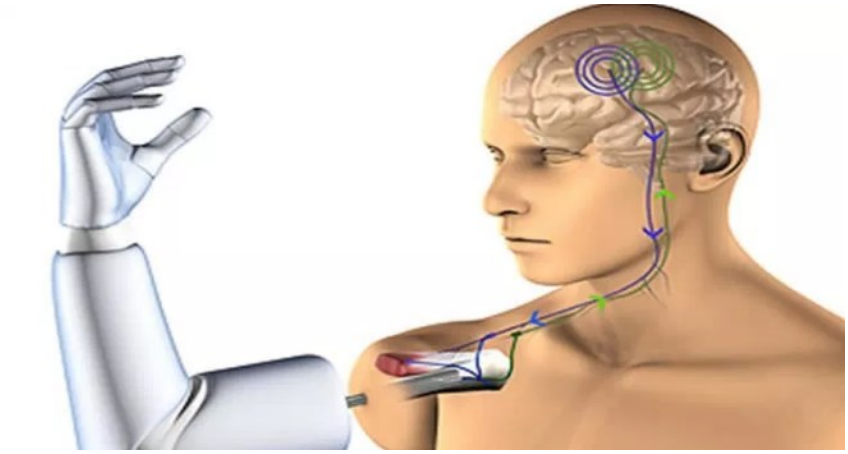

Figure 2. 3 Illustrative schematic for the BMI prosthetic Arm [Dadarlat 2014]

An important thing to keep in mind is finding the prosthetic limb prototype which is suitable for all amputation levels starting from the shoulder to the wrist, putting into consideration the size and weight which is closest to the real human arm. One of the models which serve the mentioned features is the Johns Hopkins model [Burck 2011]. The Johns Hopkins University Applied Physics Laboratory (APL), in 2005, started to develop one of the most innovative prosthetic arms in the world which is the Modular Prosthetic Limb Model (MPL) [Johannes 2011].

2.2 The Johns Hopkins (MPL) Model:

The process of developing Johns Hopkins (MPL) model can be divided into three stages. Two prototypes were created in phase 1. The first prototype had 7 of Freedom (DOF): elbow flexion/expansion, humeral rotation, shoulder flexion/ expansion, wrist rotation, wrist flexion/expansion and two actuated hand grasps [Johannes 2011]. The other Prototype, that includes the upper arm and the palm of the hand, has with total 26 joints and total 17 actuated degrees of freedom. Also, it was found that batteries are the most suitable power source [Johannes 2011].

For second stage, it was based on the second prototype and the focus was on the design, fabrication, and testing of the arm [Johannes 2011]. One of the major system design necessities for the MPL is to be modular [Burck 2011, Johannes 2011]. This modular design can serve more users with various amputation levels and decrease the complication of the fabrication [Burck 2011, Johannes 2011]. In terms of decreasing the system's complexity, Large Motor Controllers (LMCs) were made. These motors are located at several joints as the upper arm and the wrist [Johannes 2011]. The small motor controllers (SMCs) are located in the palm. The final phase of the MPL development is phase 3.In this stage, the emphasis was on improving the accuracy of reading the neural signals and giving a sensory feedback to the user. While comparing the MPL to other prostheses currently available in the market by performing a clinical evaluation, it was proved that the johns Hopkins arm has a greater number of simultaneous and sequential degrees of freedom [Perry 2018].

2.2 PREVIOUS WORKS

In [Saleh 2015], the authors simulated a wireless scheme for the MPL. They concentrated on simulating a simple model so as to study the performance of the arm with its nodes communicating wirelessly. The model was simulated using OMNeT++. The simulated model has total 7 degrees of freedom; the upper arm and the elbow have 4 degrees of freedom and the wrist has 3 degrees of freedom as shown in figure 2.4.

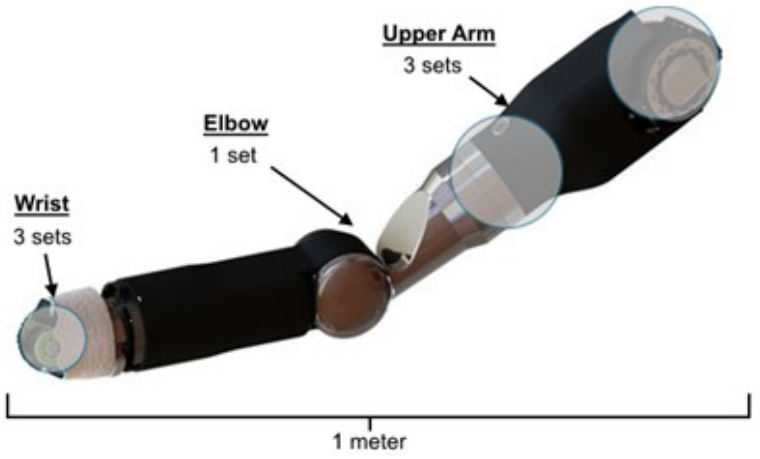

Figure 2. 4 set distribution on Johns Hopkins arm [Saleh 2015]

The four sets of the upper arm and the elbow consists of 4 sensors which are: Torque, Position, Current and Temperature beside the controller and the motor, while the sensors of each of the 3 sets of the wrist are: Torque, Position and Temperature. Zigbee is used for the wireless communication because one of the system requirements is low power consumption.

In the primary proposed scheme, the sensors in each set communicate with the actuator via a specific channel and the communication between the actuator and supervisor is done via another channel. Therefore, only 14 channels out of the 16 non-interfering channels of the Zigbee were utilized. However, this system has failed to meet the system deadlines. So, channel re-allocation was done. 14 channels were used for the communication between the supervisor and actuator, and 2 channels were used for the communication between the sensors and actuators; hence, all the all 16 non-interfering ZigBee channels were utilized.The results have met the deadlines as shown in table 2.1.

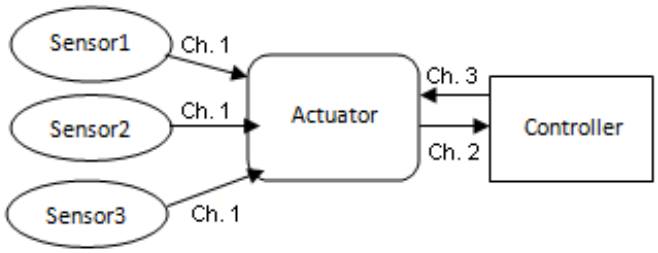

Figure 2. 5 Channel allocation [Saleh 2015]

Table 2. 1 95% Confidence Interval Results for the Proposed Architecture (in ms). Interference-Free Scenario

Set	S→A	A→K	K→A
1	[44.20;54.00]	[6.20;6.20]	[6.00;6.80]
2	[42.30;52.50]	[6.20;6.20]	[6.20;6.20]
3	[43.70;55.00]	[6.20;6.20]	[6.20;6.20]
4	[44.80;54.80]	[6.20;6.20]	[6.20;6.20]
5	[42.00;52.00]	[6.10;6.50]	[6.10;6.50]
6	[41.00;47.00]	[6.20;6.20]	[6.20;6.20]
7	[41.30;52.20]	[6.20;6.20]	[6.20;6.20]

Furthermore, 40 bytes payload was used as a source of interference. based on a 95% confidence analysis, the end-to-end delays and packet loss parameters have met the system's requirements, as shown in table 2.2.

Table 2. 2 95% Confidence Interval Results for the Proposed Architecture (in ms). Interference Scenario (40 bytes Payload)

Set	S→A	A→K	K→A
1	[43.30;51.70]	[8.80;12.90]	[8.30;12.30]
2	[43.70;52.80]	[9.20;13.40]	[8.80;12.50]
3	[46.60;56.60]	[9.70;13.60]	[9.20;13.20]
4	[49.20;57.30]	[11.00;15.50]	[9.00;12.90]
5	[42.50;51.50]	[9.20;12.90]	[7.60;11.20]
6	[41.60;50.00]	[8.20;12.50]	[8.50;12.00]
7	[43.50;51.20]	[7.90;11.60]	[9.00;13.90]

Chapter 3

Performance of the proposed wireless scheme

3.1 Proposed Model

This chapter presents the first contribution of this thesis. A novel scheme is proposed for the arm which has the advantage of being fault-tolerant, i.e., it will be proven that the arm can still operate correctly when one or more of its electronic components fails. First, the fault-free model will be described and studied. Next, the fault-tolerant model will be investigated where the arm will be shown to perform correctly after the failure of several of its controllers. Figure 3.1.1 shows the schematic model of data flow of all sets for the fault free scenario.

Recall that the Johns Hopkins MPL consists of 17 motors which control 26 joints (26 Degrees of Freedom) [McGee 2014]. The 17 motors are distributed as follows: the arm is controlled by 7 motors; the upper arm is controlled by 3 motors, the elbow is controlled by 1 motor, the wrist controlled by 3 motors. The hand is controlled by 10 motors; the thumb is controlled by 4 motors, each finger is controlled by 1 motor and finally the palm of the hand is controlled by 2 motors [McGee 2014]. Hence, the total number of motors is 17 motors; however, the scope of this research is only limited to the arm (7 motors only) not the palm of the hand.The total number of sensors used in the arm and included in this research is 39 sensors [McGee 2014, Ravitz 2013].

As shown in Figure 3.1.2, the arm's architecture can be described as several sets, for easier visualization. Each set consists of an actuator that controls a motor beside a number of sensors. Three of the sets are located at the upper arm, one set is located at the elbow and 3 sets are located at the wrist. Each set in the elbow or upper arm includes 6 sensors, while each set in the wrist includes 5 sensors. Table 3.1 shows the sets location in the arm and the types of sensors in each of them.

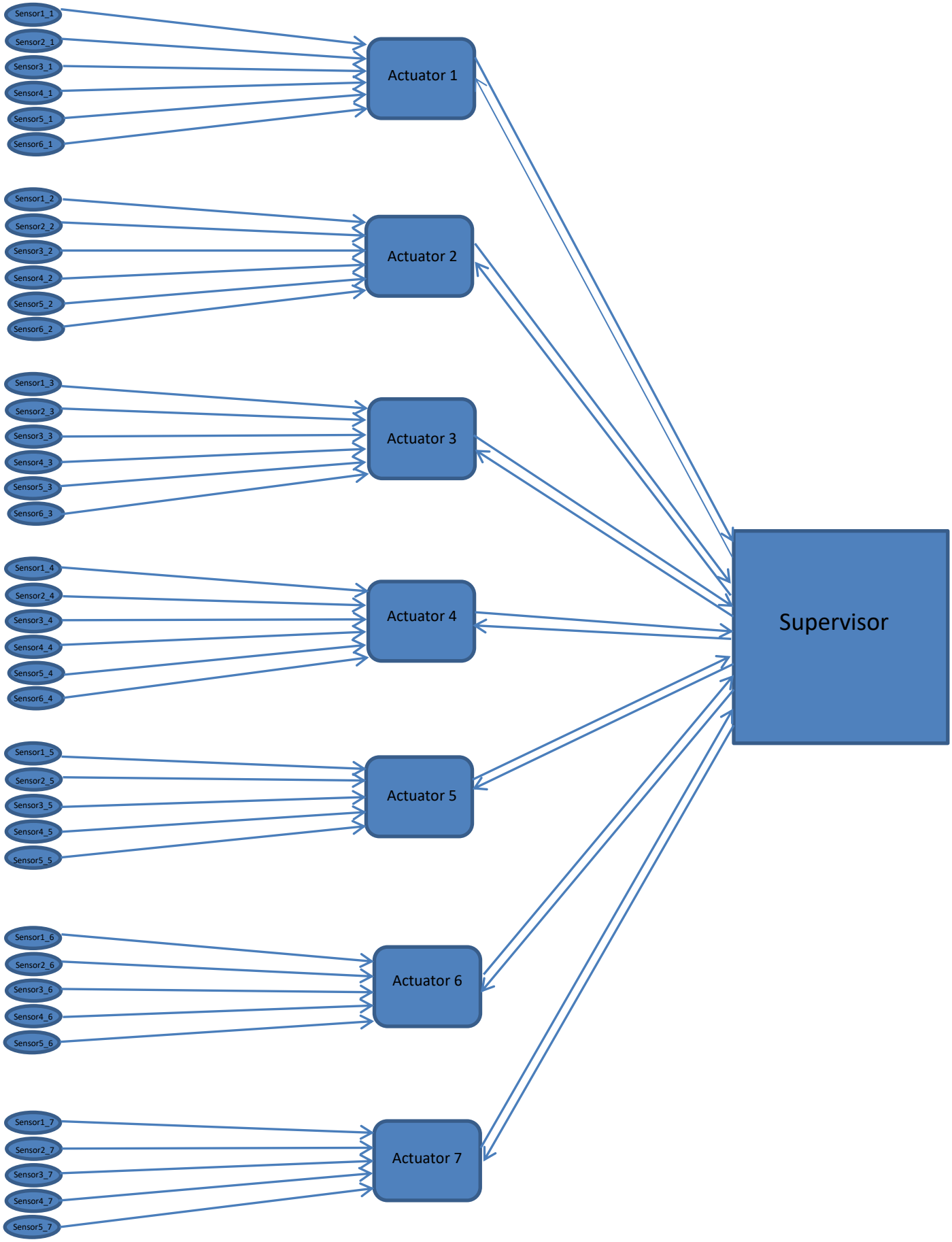

Figure 3.1. 1 Schematic model of data flow of all sets for the fault free scenario

Table 3. 1 SETS PROPERTIES

Set Position	Number of Sets	Number and types of sensors
Upper Arm	3	6 Sensors: Absolute Position Sensor, Torque Sensor, Joint temperature sensor, Incremental rotor position, Drive voltage, Upper arm drive current. Actuator: Motor
Elbow	1	6 Sensors: Absolute Position Sensor, Torque Sensor, Joint temperature sensor, Incremental rotor position, Drive voltage, Upper arm drive current. Actuator: Motor
Wrist	3	5 Sensors: Absolute Position Sensor, Torque Sensor, Joint temperature sensor, Incremental rotor position, Drive voltage. Actuator: Motor

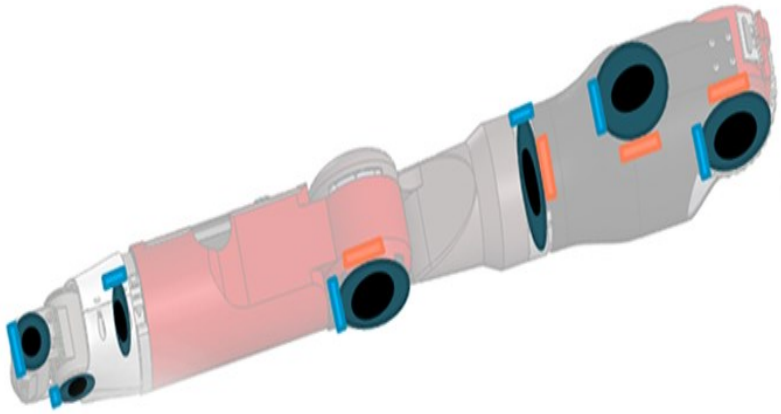

Figure 3.1. 2 Illustrative diagram for the motors and sensors distribution in the MPL [jhuapl 2019]

Regarding the communication rates, the upper arm, elbow and wrist sensors communicate with the actuators at 25Hz (40msec) while the actuators communicate with the supervisor at rate 50Hz (20msec) [Johannes 2011]. The reason that the frequency of the latter is twice the former is the watchdog signal between the actuator and the supervisor. The sensor payload is 1 byte. The supervisor serves as the access point.

The communication protocol is low power WiFi [Dekel 2019]. The reason for this choice is that the system is required to consume low power. Another reason is that it is necessary to have enough bandwidth since the arm consists of various nodes. Also, sufficient bandwidth will allow the model expansion by adding more sensors for example.

The bit rate is 65 Mbps. All nodes including the 39 sensors, 7 actuators and the supervisor communicate via channel 1. The transmit power is 1mW. Furthermore, UDP is used for data transmission between all the nodes because of the real-time constraints on the system [Boggia 2009].

As shown in figure 3.1.3, the communication scheme of the fault free model is a S2A (sensor to actuator) scheme. Therefore, in each set, the sensors (S) transmit the payload to the actuator (A) node. The controller (K) is part of the actuator node. The sensors' data are gathered at the actuator node and transmitted to the supervisor. In addition, Figure 3.1.4 shows part of the model (one set) on Riverbed.

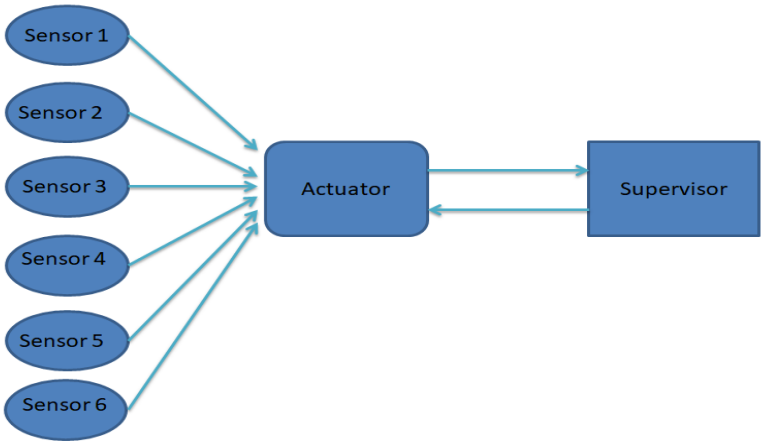

Figure 3.1. 3 Schematic model of data flow of one set for the fault free scenario

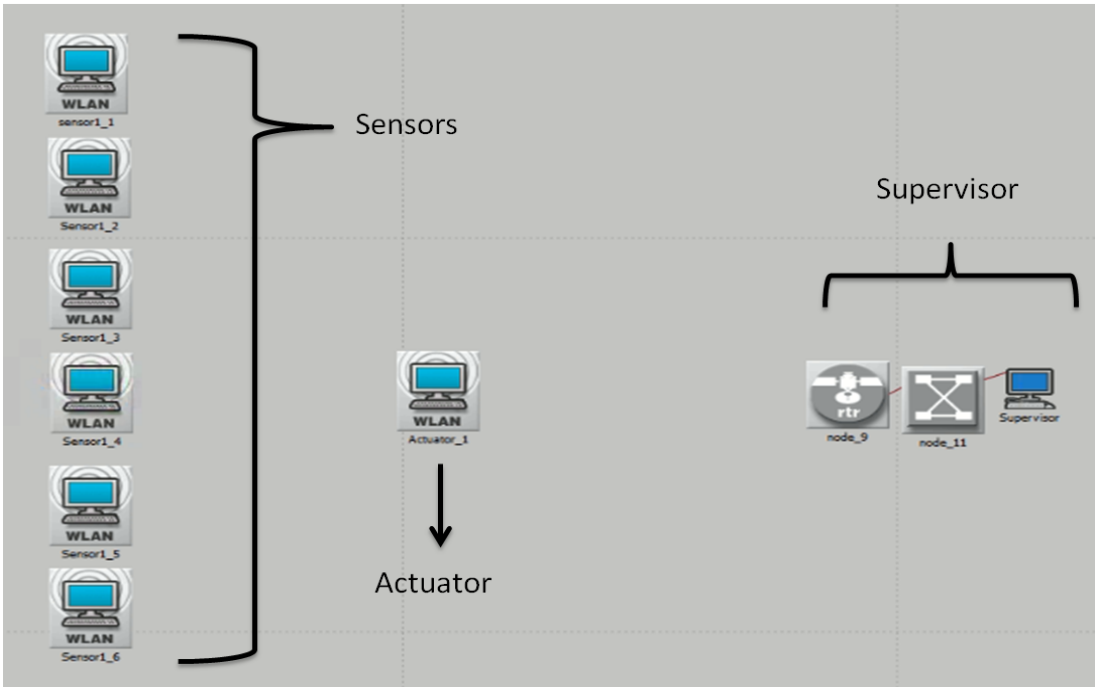

Figure 3.1. 4 Riverbed diagram for one set

The following is the simulation results of the fault free, interference free model and fault free, interfered model. All the results are based on a 95% confidence analysis. The system performance is measured by the packet delay and the packet loss.

3.1.1 Fault free, interference free model Results

In this subsection, the simulation results of the fault free, interference free model are described. Table 3.2 shows the packets' end-to-end delays of the supervisor and all actuators. As shown, all the delays are less than 0.4msec and the system required deadline is 20msec; therefore, the delay requirement is met. Figure 3.1.5 to Figure 3.1.12 show the end to end delays of all actuators and the supervisor for 33 runs [Daoud 2007]. The y-axis represents the packet end-to-end delay in seconds while the x-axis represents the simulation time. The different colors of the curves indicate the results of the 33 simulations.

Table 3. 2 95% Confidence intervals results for the fault free and interference free scenario

Packet end-to-end delay interval	Delay interval (msec)
Actuator 1	[0.276393; 0.263234]
Actuator 2	[0.321659; 0.308081]
Actuator 3	[0.274794; 0.260128]
Actuator 4	[0.262987; 0.255787]
Actuator 5	[0.303477; 0.276393]
Actuator 6	[0.297817; 0.277132]
Actuator 7	[0.343183; 0.288282]
Supervisor	[0.136719; 0.131542]

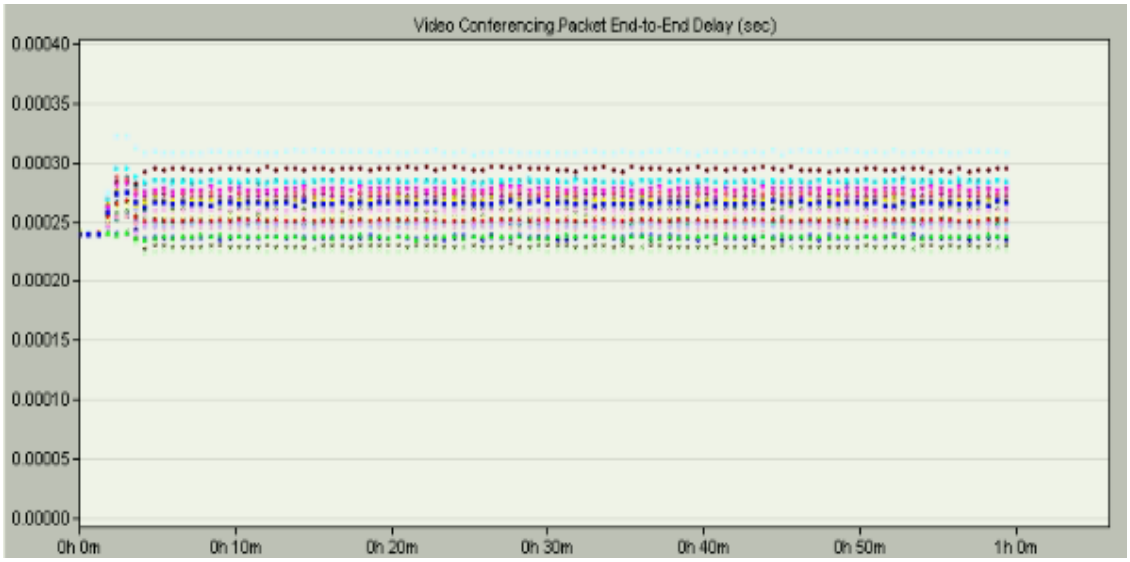

Figure 3.1. 5 Fault Free, Interference Free Model; Actuator 1 End-to-End delays (33 runs)

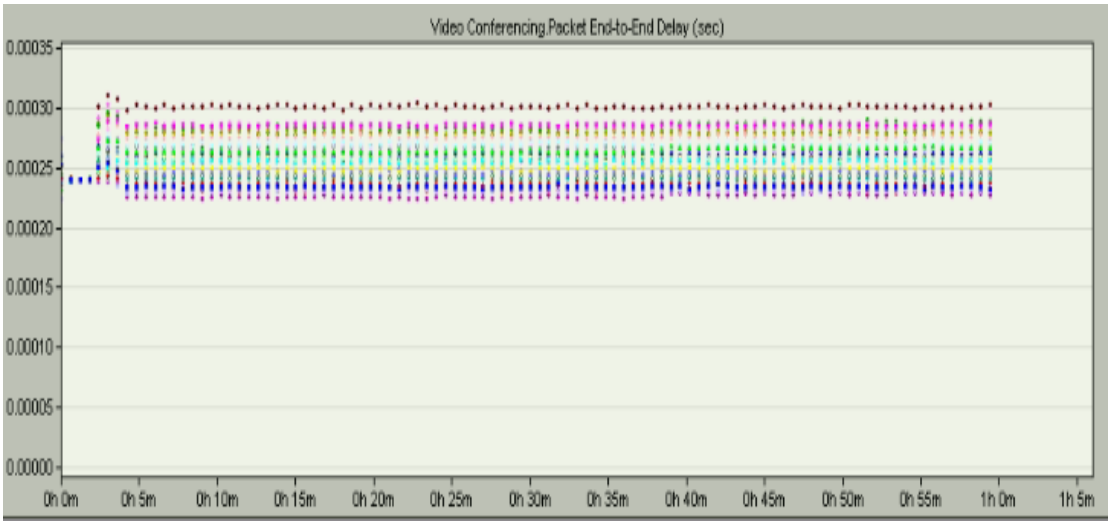

Figure 3.1. 6 Fault Free, Interference Free Model; Actuator 1 End-to-End delays (33 runs)

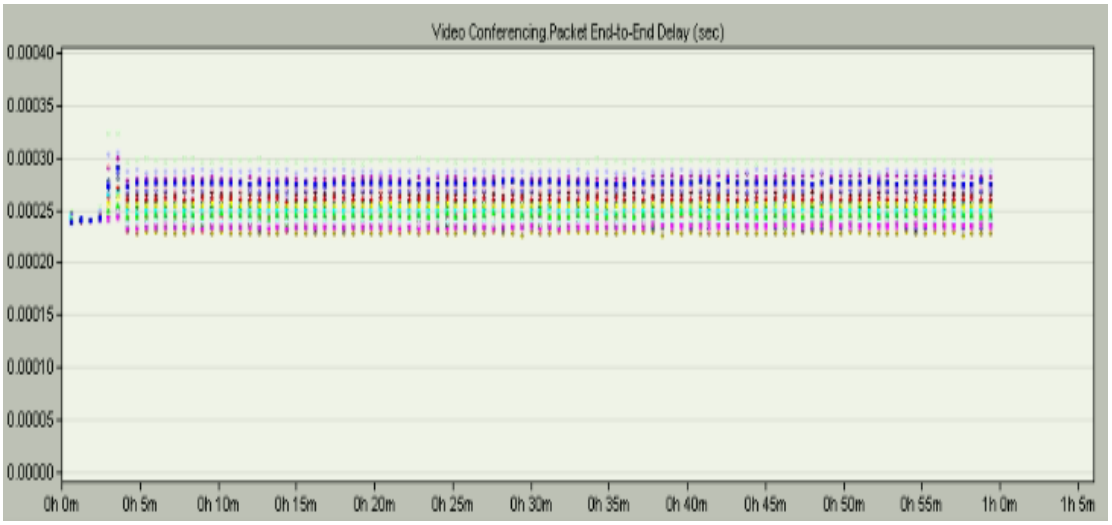

Figure 3.1. 7 Fault Free, Interference Free Model; Actuator 3 End-to-End delays (33 runs)

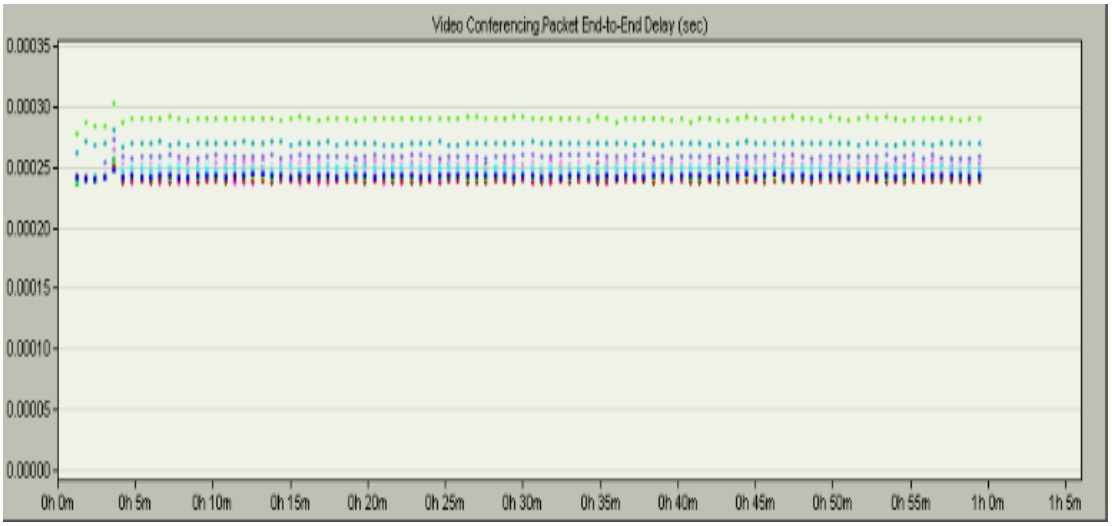

Figure 3.1. 8 Fault Free, Interference Free Model; Actuator 4 End-to-End delays (33 runs)

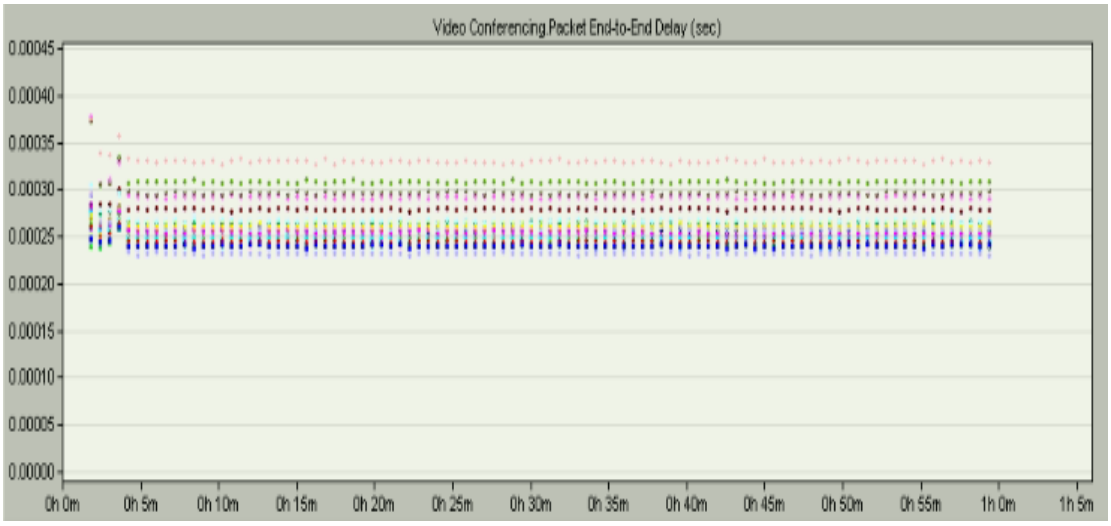

Figure 3.1. 9 Fault Free, Interference Free Model; Actuator 5 End-to-End delays (33 runs)

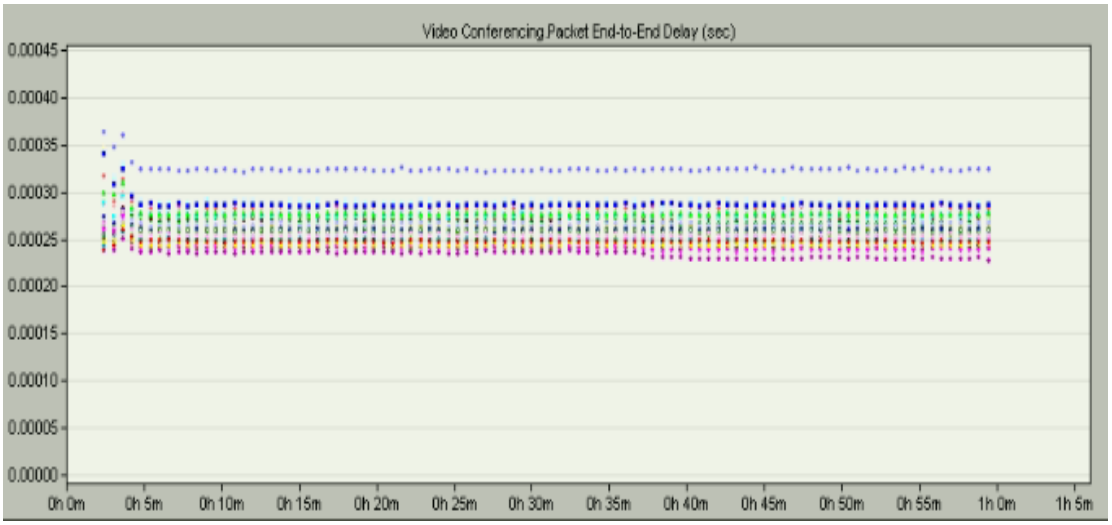

Figure 3.1. 10 Fault Free, Interference Free Model; Actuator 6 End-to-End delays (33 runs)

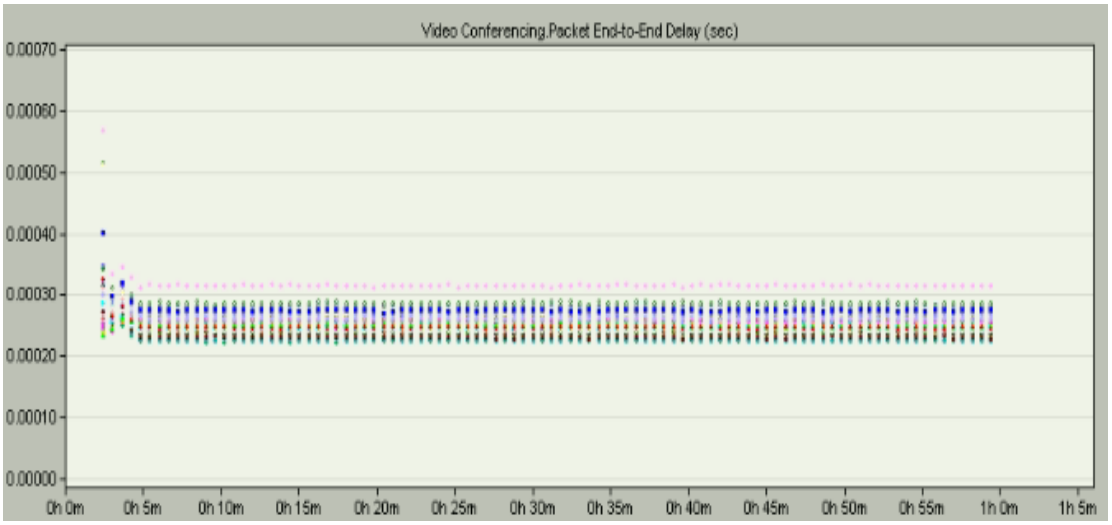

Figure 3.1. 11 Fault Free, Interference Free Model; Actuator 7 End-to-End delays (33 runs)

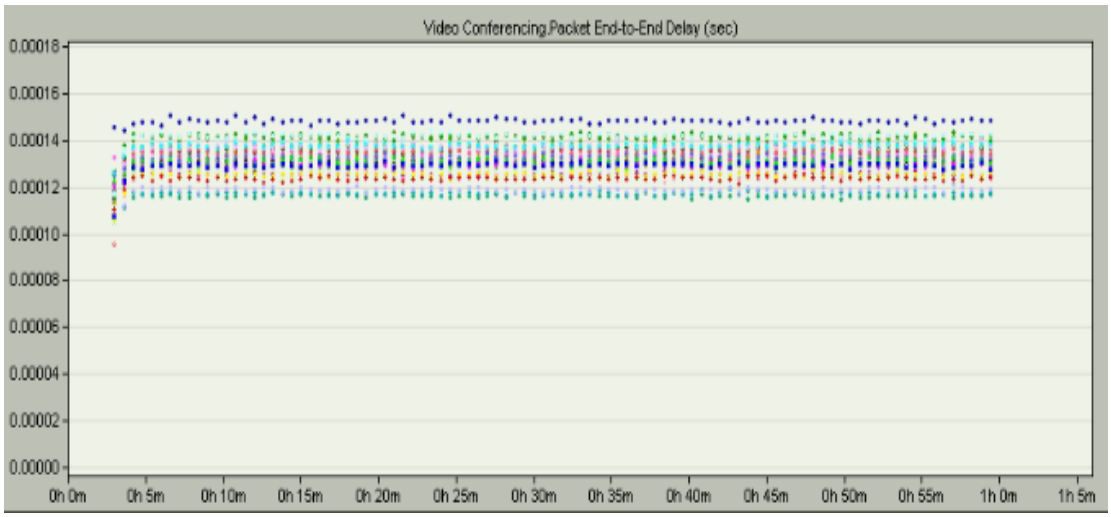

Figure 3.1. 12 Fault Free, Interference Free Model; Supervisor End-to-End delays (33 runs)

Further, Figure 3.1.13 to Figure 3.1.20 show the graphs of packets received at the 7 actuators and the supervisors. The x-axis represents the simulation time in minutes and the y-axis represents the received traffic in bytes/sec.

For simulating the outgoing traffic, the sensor outgoing stream inter-arrival time is 41msec, while the supervisor and the actuator outgoing stream inter-arrival time is 21msec.

Recall that sets 1,2,3 and 4 contains 6 sensors while sets 5,6 and 7 contains 5 sensors.

-The supervisor receives the data aggregated from all actuators is calculated as follows:

$\left(\frac{6bytes}{0.021sec}\right)$x 4 + $\left(\frac{5bytes}{0.021sec}\right)$x3

-In addition, the traffic received at actuators 1 → 4 is calculated as follows:

$\left(\frac{1byte}{0.041sec}\right)$ x6 + $\left(\frac{6bytes}{0.021sec}\right)$

In addition, the traffic received at actuators 5 → 7 is calculated as follows:

$\left(\frac{1byte}{0.041sec}\right)$ x5 + $\left(\frac{5bytes}{0.021sec}\right)$

Hence, each node will receive the below information:

Actuator 1>> 432 bytes/sec

Actuator 2>> 432 bytes/sec

Actuator 3>> 432 bytes/sec

Actuator 4>> 432 bytes/sec

Actuator 5>> 358 bytes/sec

Actuator 6>> 358 bytes/sec

Actuator 7>> 358 bytes/sec

Supervisor>> 1857 bytes/sec

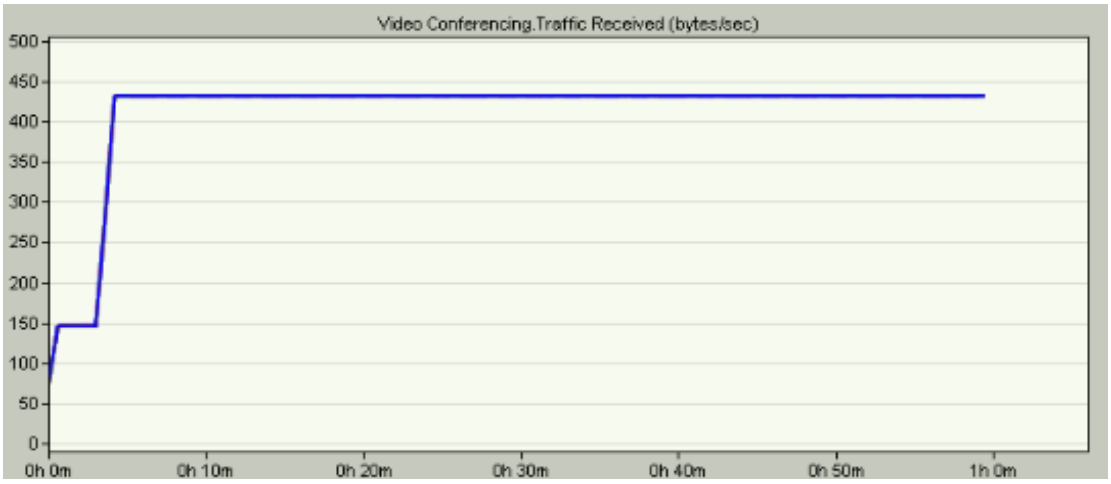

Figure 3.1. 13 Fault Free, Interference Free Model; traffic received at Actuator 1

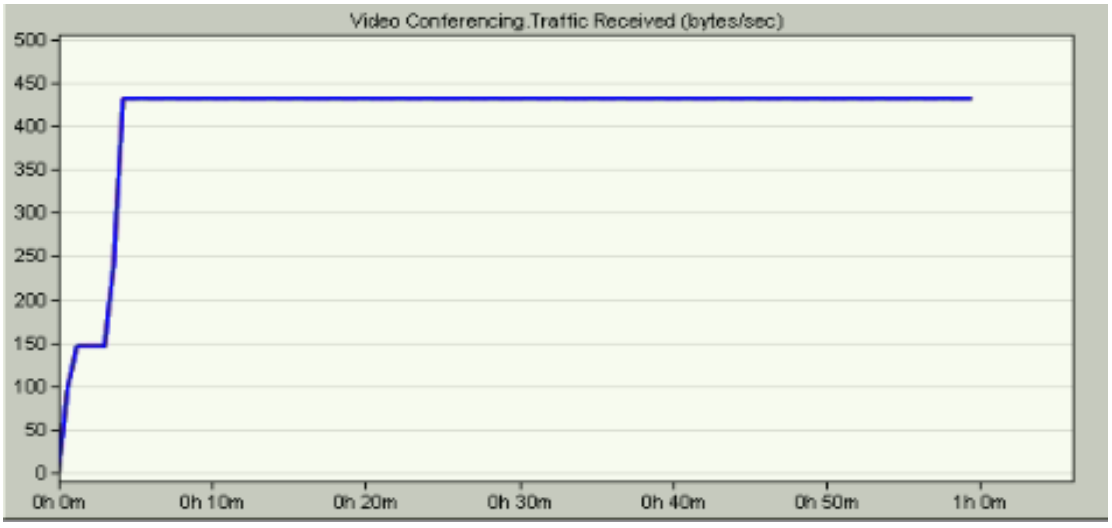

Figure 3.1. 14 Fault Free, Interference Free Model; traffic received at Actuator 2

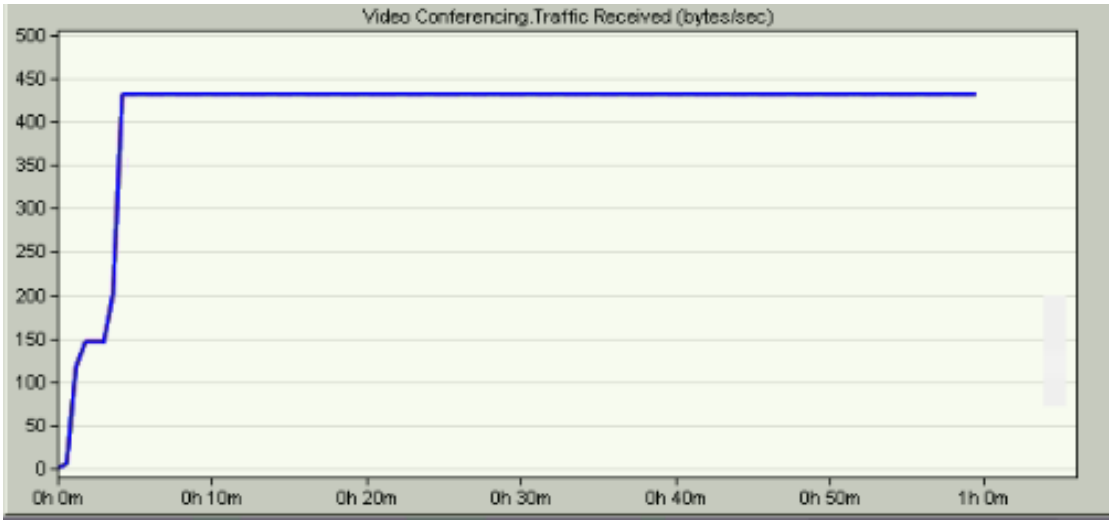

Figure 3.1. 15 Fault Free, Interference Free Model; traffic received at Actuator 3

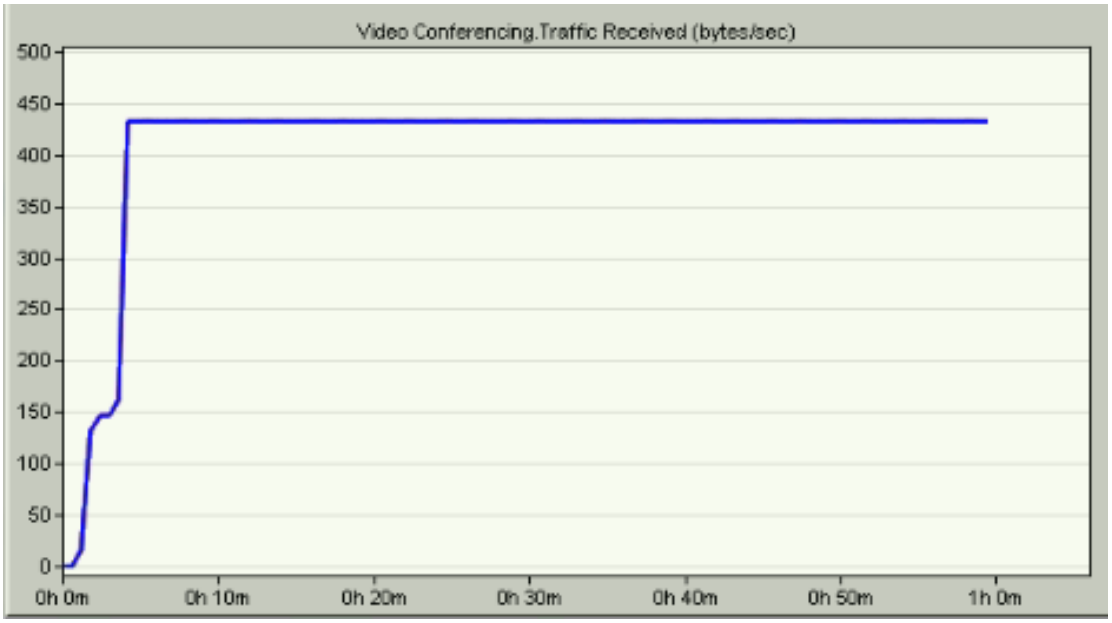

Figure 3.1. 16 Fault Free, Interference Free Model; traffic received at Actuator 4

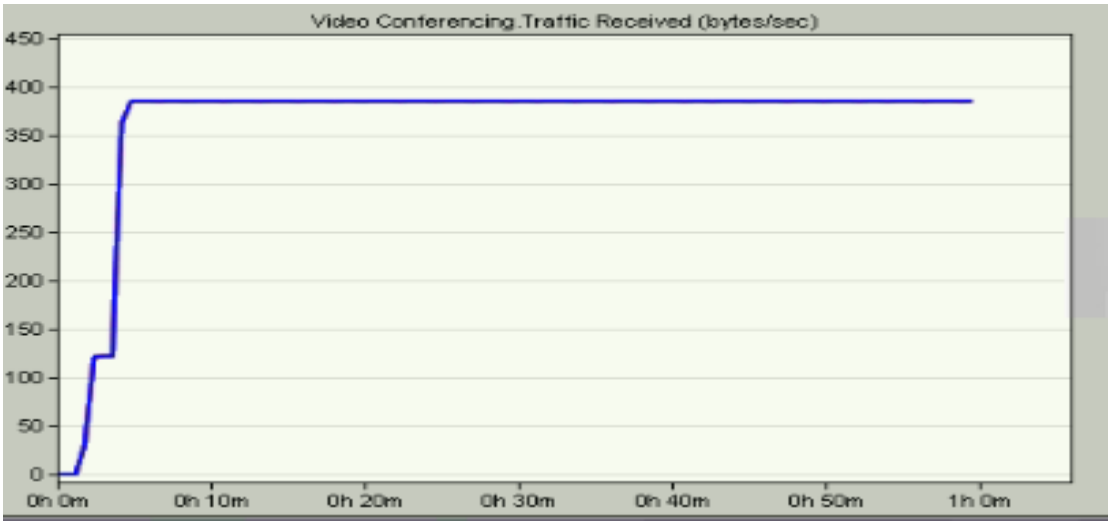

Figure 3.1. 17 Fault Free, Interference Free Model; traffic received at Actuator 5

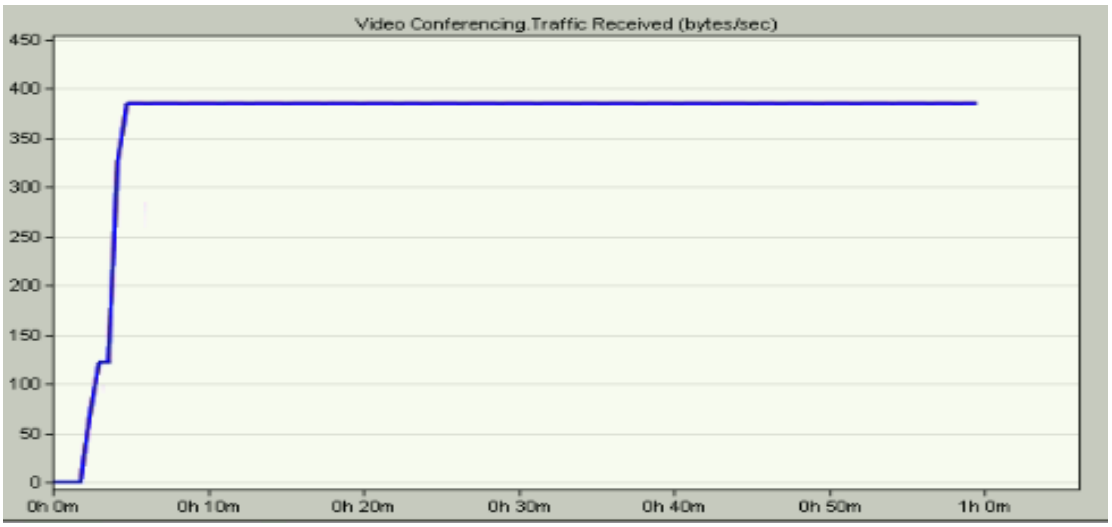

Figure 3.1. 18 Fault Free, Interference Free Model; traffic received at Actuator 6

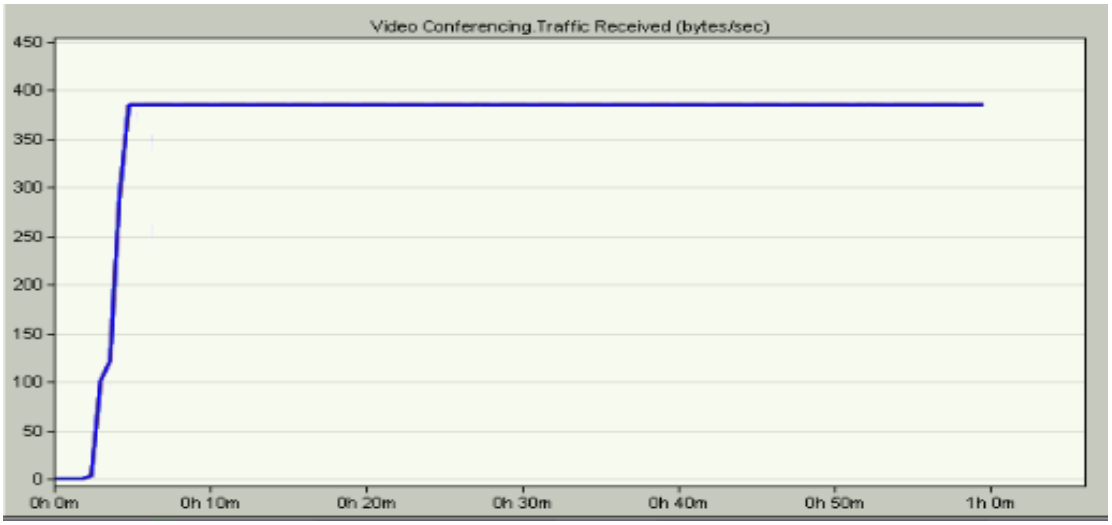

Figure 3.1. 19 Fault Free, Interference Free Model; traffic received at Actuator 7

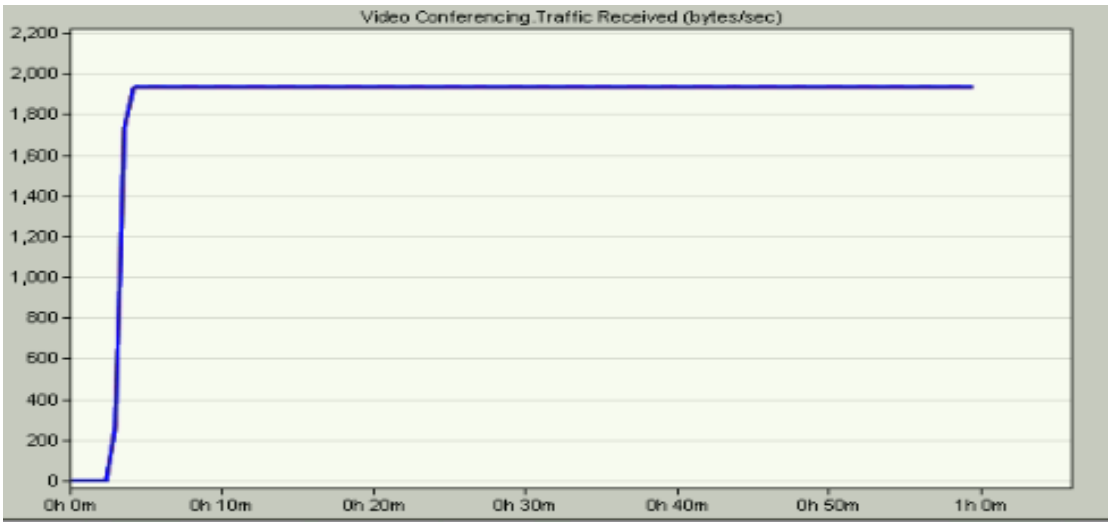

Figure 3.1. 20 Fault Free, Interference Free Model; traffic received at the Supervisor

3.1.2 Fault Free, with interference model results

To simulate the interference model on Riverbed, two external interferer nodes are positioned beside the arm nodes. Both nodes exchange packets via channel 1 and that is the same channel used for the communication of the rest of the arm nodes [El-Faramawy 2012]. They also have the same transmission power of 1mW. A file of size 1Gb is exchanged between the two interferer nodes. By trial and error, it was found that the file size which is less than 1 Gb does not affect the end-to-end delays of the rest of the nodes. Since there is no real time constraint on these two nodes, the protocol used for the file transfer is FTP.

Table 3.3 shows the packets' end-to-end delays of the supervisor and all actuators. As shown, all the delays are below 6msec. Although the delays of the interference model are higher than the delays of interference free model, they still meet the system deadlines. Even if the communication channels suffered interference by such large data, the model is still very robust.

In addition, Figure 3.1.21 to Figure 3.1.28 show the end to end delays of all actuators and the supervisor for 33 runs. The y-axis represents the packet end-to-end delay in seconds while the x-axis represents the simulation time. The different colors of the curves indicate the results of the different 33 simulations.

Table 3. 3 95% Confidence Intervals Results for the Fault Free with Interference Scenario

Packet end-to-end delay interval	Delay interval (msec)
Actuator 1	[4.75575; 4.71783]
Actuator 2	[4.83449; 4.80453]
Actuator 3	[4.89783; 4.85784]
Actuator 4	[4.76371; 4.73176]
Actuator 5	[4.91035; 4.86032]
Actuator 6	[4.92602; 4.89059]
Actuator 7	[4.93246; 4.87526]
Supervisor	[1.85942; 1.82018]

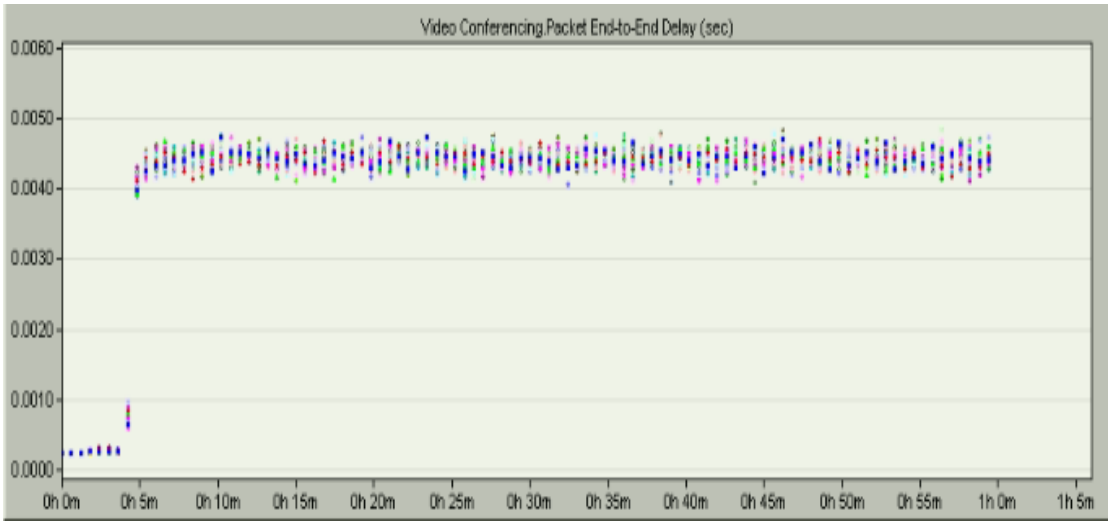

Figure 3.1. 21 Fault Free, Interference Model; Actuator 1 End-to-End delays (33 runs)

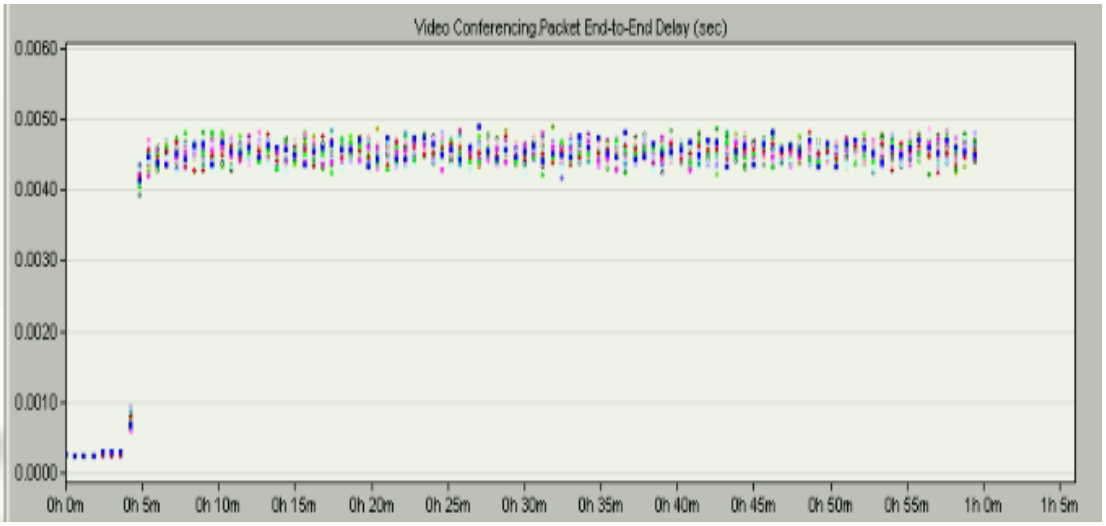

Figure 3.1. 22 Fault Free, Interference Model; Actuator 2 End-to-End delays (33 runs)

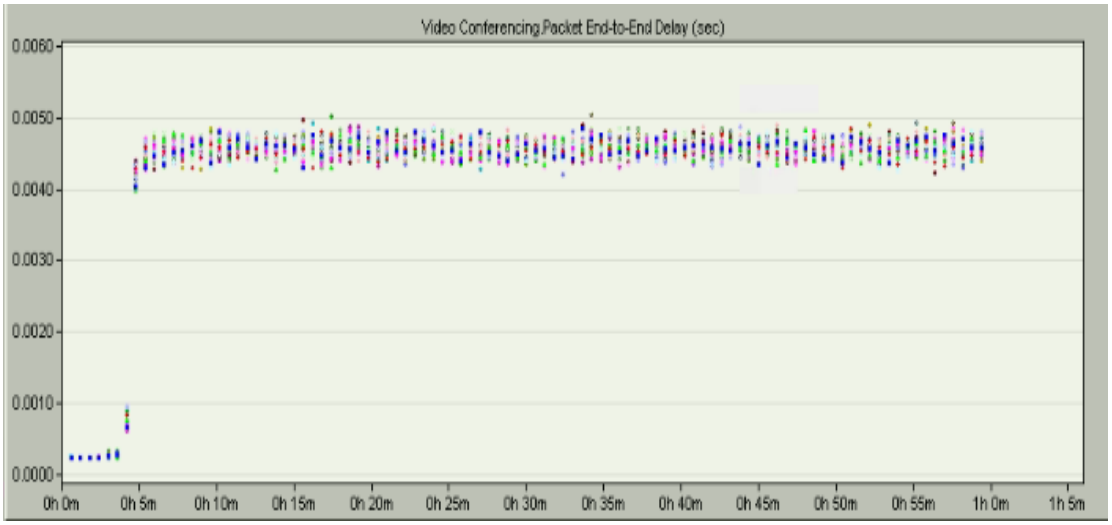

Figure 3.1. 23 Fault Free, Interference Model; Actuator 3 End-to-End delays (33 runs)

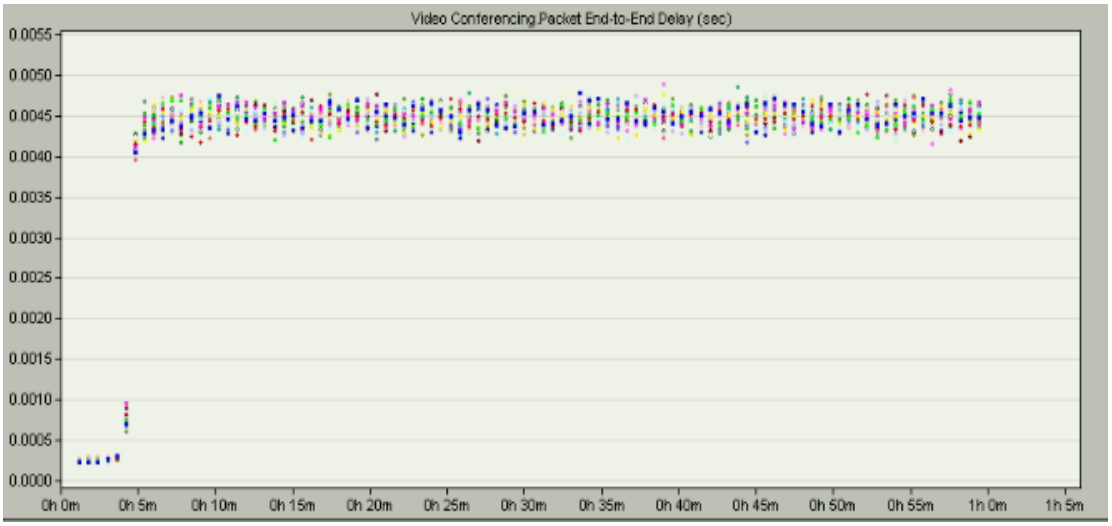

Figure 3.1. 24 Fault Free, Interference Model; Actuator 4 End-to-End delays (33 runs)

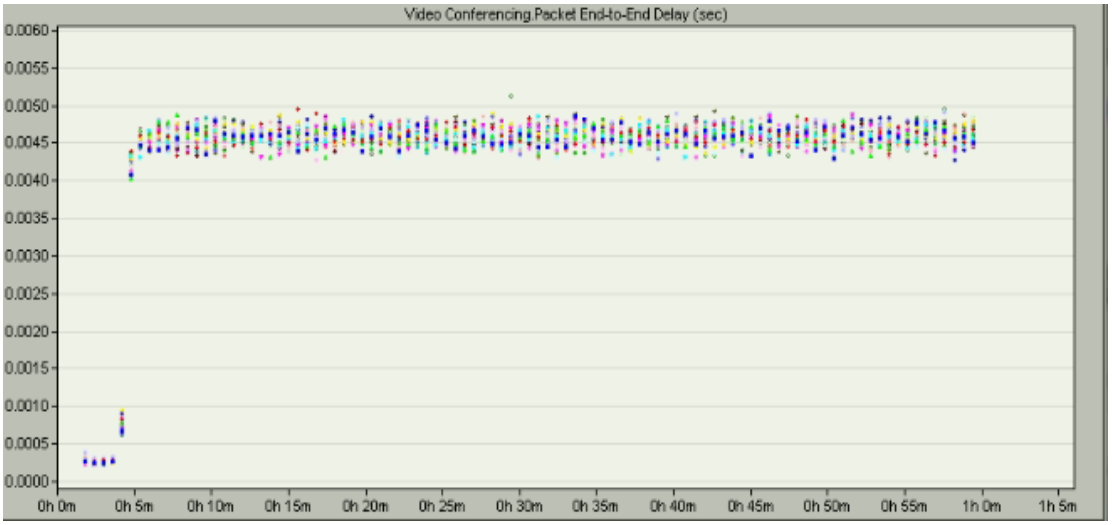

Figure 3.1. 25 Fault Free, Interference Model; Actuator 5 End-to-End delays (33 runs)

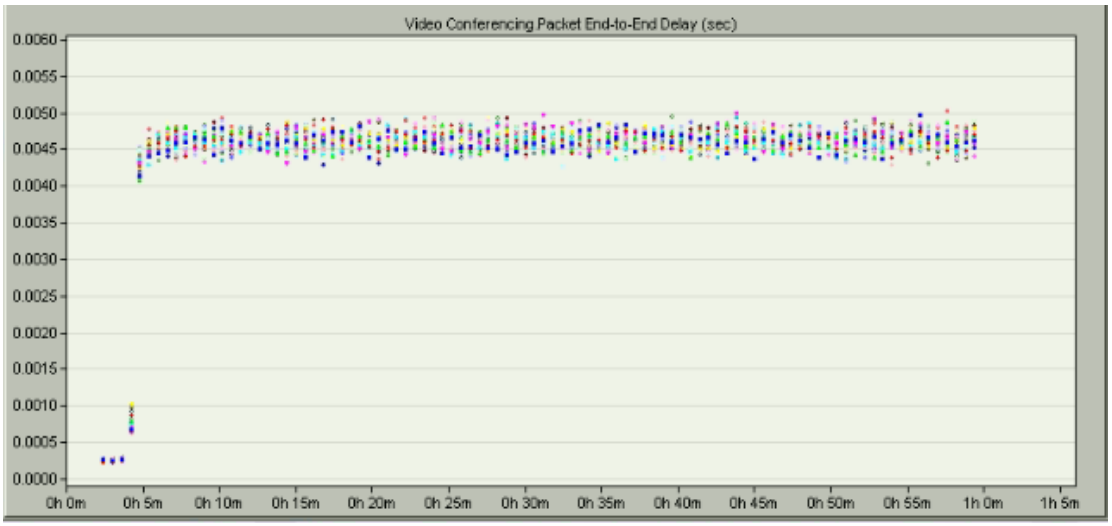

Figure 3.1. 26 Fault Free, Interference Model; Actuator 6 End-to-End delays (33 runs)

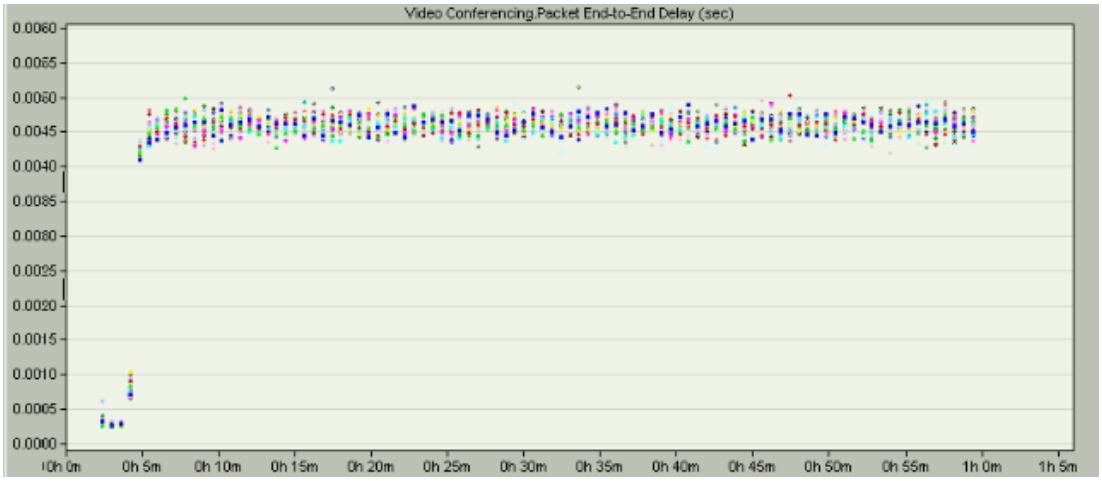

Figure 3.1. 27 Fault Free, Interference Model; Actuator 7 End-to-End delays (33 runs)

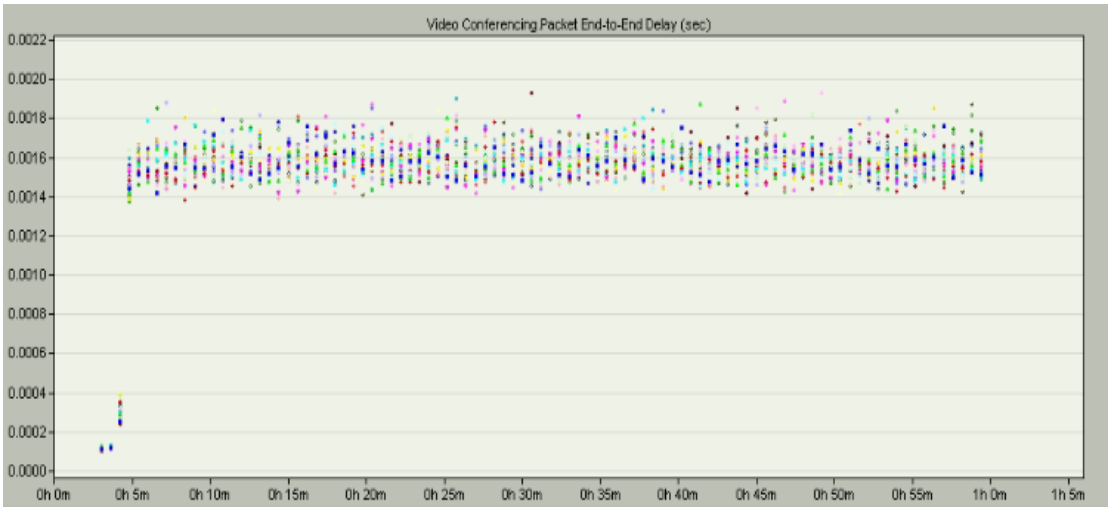

Figure 3.1. 28 Fault Free, Interference Model; Supervisor End-to-End delays (33 runs)

Further, Figure 3.1.29 to Figure 3.1.36 show the graphs of packets received at the 7 actuators and the supervisors. The x-axis represents the simulation time in minutes and the y-axis represents the received traffic in bytes/sec. The graphs show zero packet loss.

For simulating the outgoing traffic, the sensor outgoing stream inter-arrival time is 41msec, while the supervisor and the actuator outgoing stream inter-arrival time is 21msec.

Recall that sets 1,2,3 and 4 contains 6 sensors while sets 5,6 and 7 contains 5 sensors.

-The supervisor receives the data aggregated from all actuators is calculated as follows:

$\left(\frac{6bytes}{0.021sec}\right)$x 4 + $\left(\frac{5bytes}{0.021sec}\right)$x3

-In addition, the traffic received at actuators 1 → 4 is calculated as follows:

$\left(\frac{1byte}{0.041sec}\right)$ x6 + $\left(\frac{6bytes}{0.021sec}\right)$

In addition, the traffic received at actuators 5 → 7 is calculated as follows:

$\left(\frac{1byte}{0.041sec}\right)$ x5 + $\left(\frac{5bytes}{0.021sec}\right)$

Hence, each node will receive the below information:

Actuator 1>> 432 bytes/sec

Actuator 2>> 432 bytes/sec

Actuator 3>> 432 bytes/sec

Actuator 4>> 432 bytes/sec

Actuator 5>> 358 bytes/sec

Actuator 6>> 358 bytes/sec

Actuator 7>> 358 bytes/sec

Supervisor>> 1857 bytes/sec

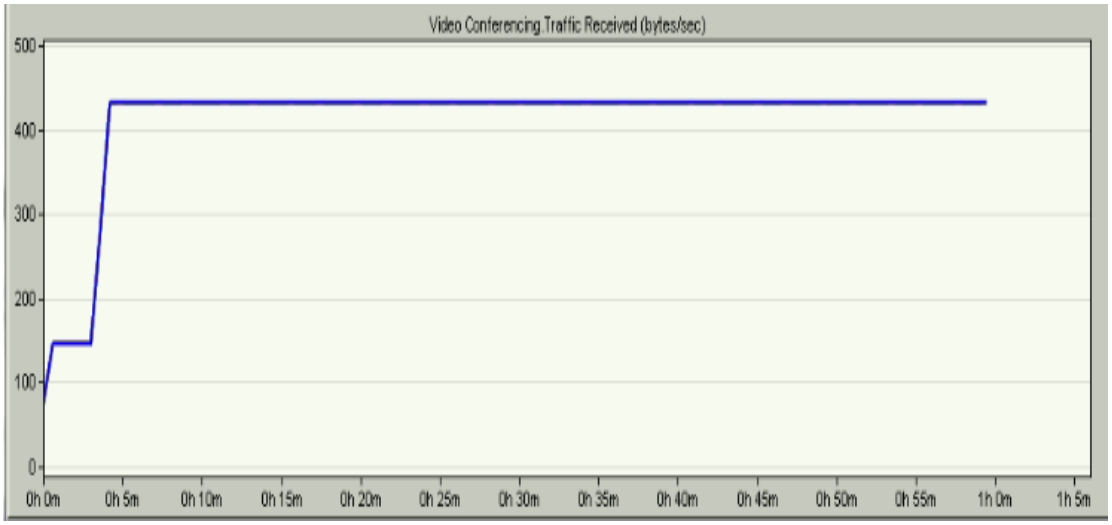

Figure 3.1. 29 Fault Free, Interference Model; traffic received at Actuator 1

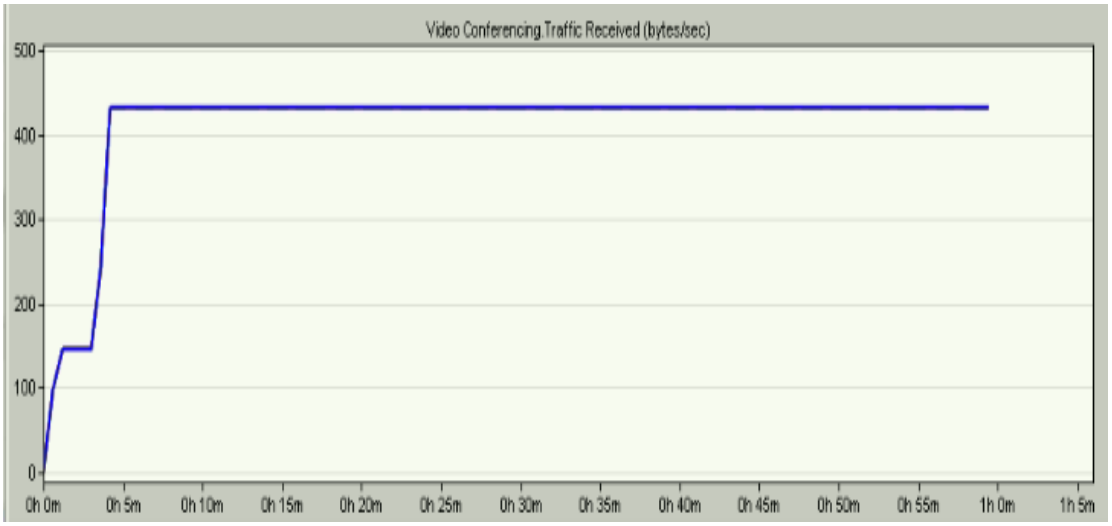

Figure 3.1. 30 Fault Free, Interference Model; traffic received at Actuator 2

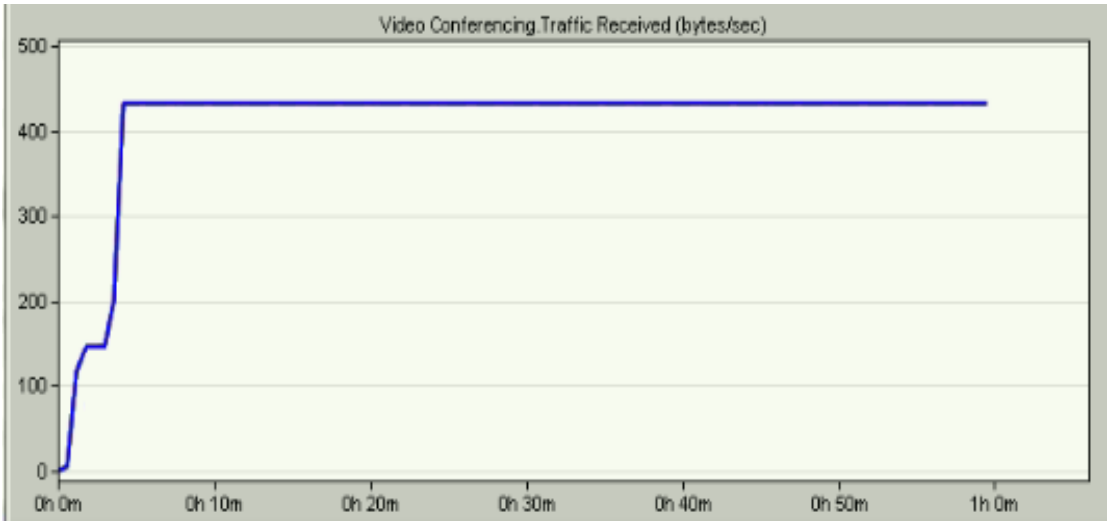

Figure 3.1. 31 Fault Free, Interference Model; traffic received at Actuator 3

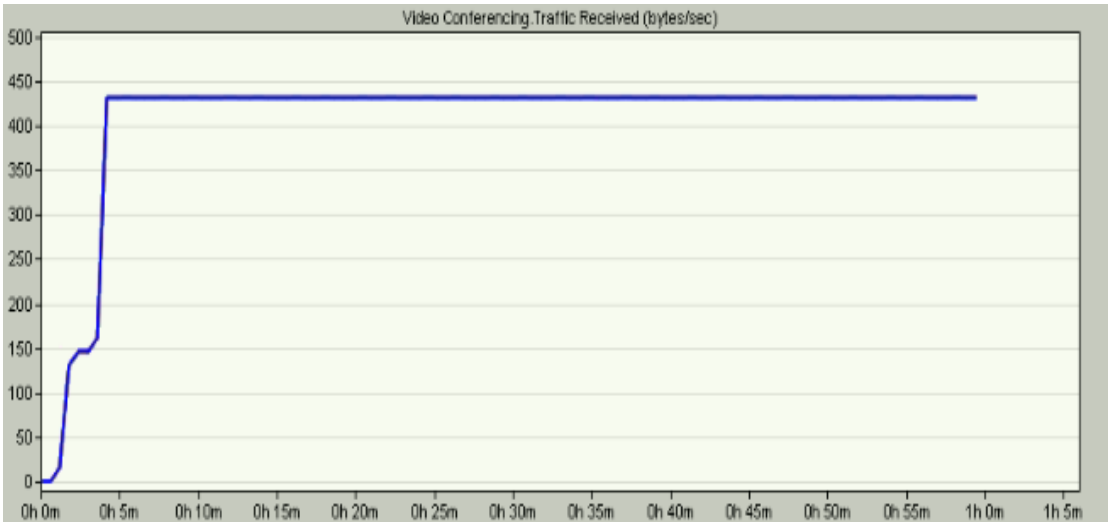

Figure 3.1. 32 Fault Free, Interference Model; traffic received at Actuator 4

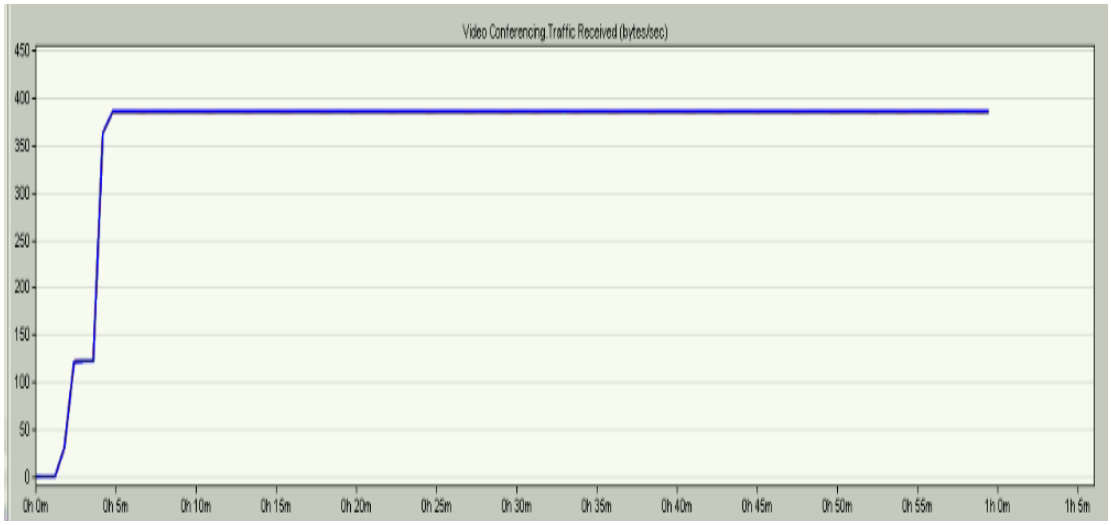

Figure 3.1. 33 Fault Free, Interference Model; traffic received at Actuator 5

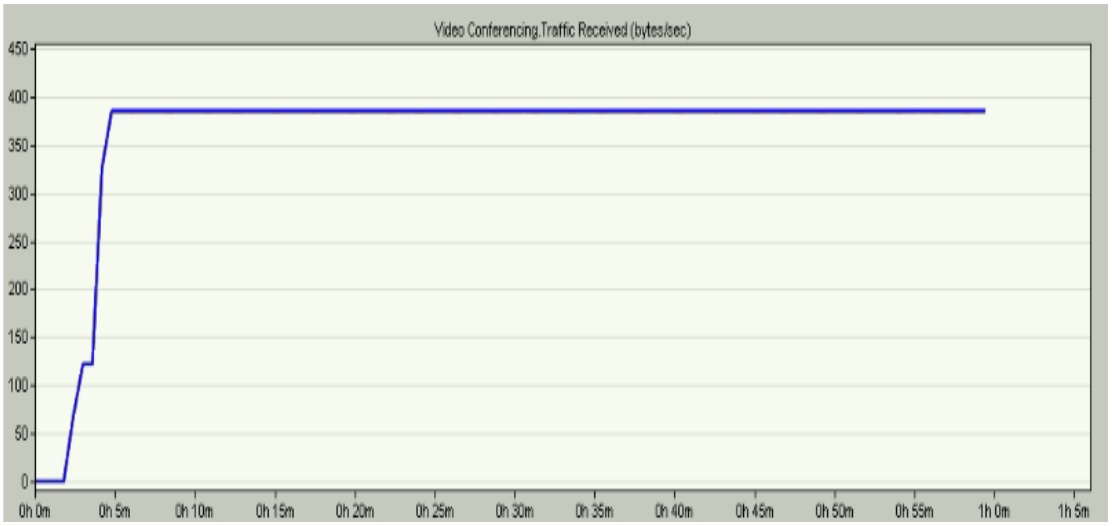

Figure 3.1. 34 Fault Free, Interference Model; traffic received at Actuator 6

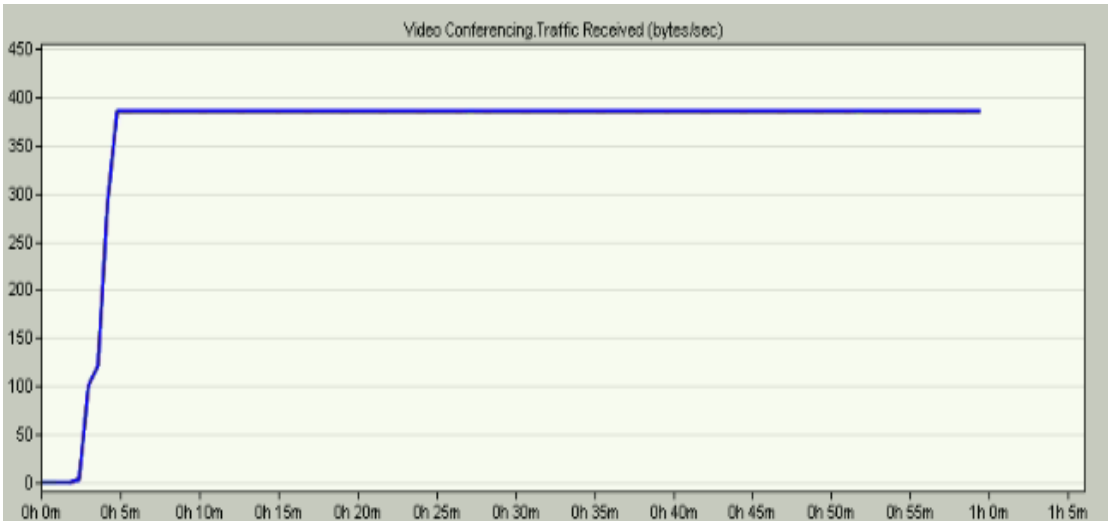

Figure 3.1. 35 Fault Free, Interference Model; traffic received at Actuator 7

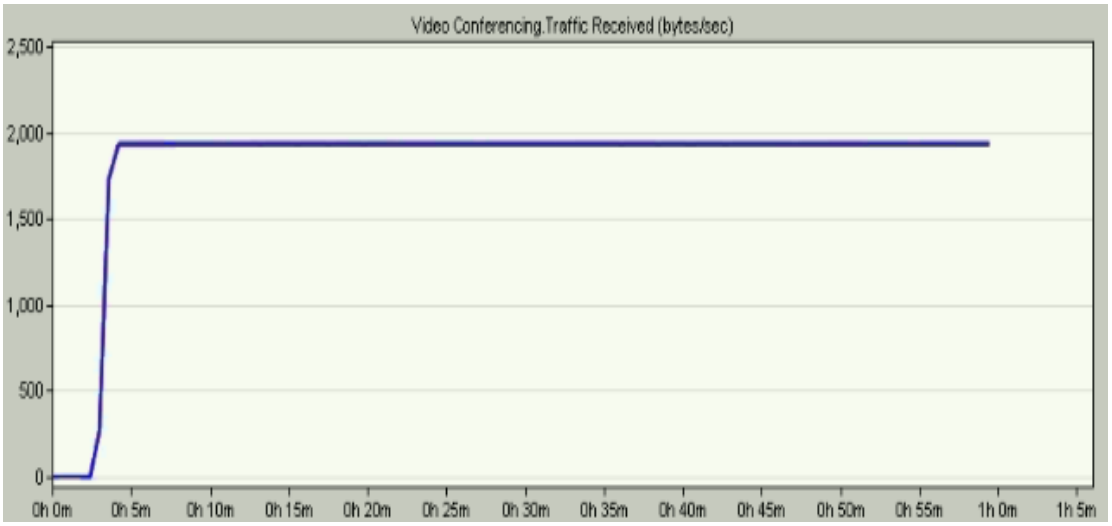

Figure 3.1. 36 Fault Free, Interference Model; traffic received at Supervisor

3.2 Fault-Tolerant Scenarios

Reliability is a definitely a matter of interest in networked control systems, particularly when it is related to medical applications such as WBANs. Accordingly, a fault-tolerant model is simulated to serve as an alternative scenario which the system will undertake when failure occurs. The on-board controller (which is part of the actuator node) is assumed to be faulty, so the sensors' payloads are transmitted to the arm's supervisor instead. This means that the faulty model is no longer based on the S2A communication scheme, so the sensor data has to be transmitted to the supervisor node where the control action is calculated, and sent from the supervisor to the actuator. A schematic model of data flow is shown in Figure 3.2.1.

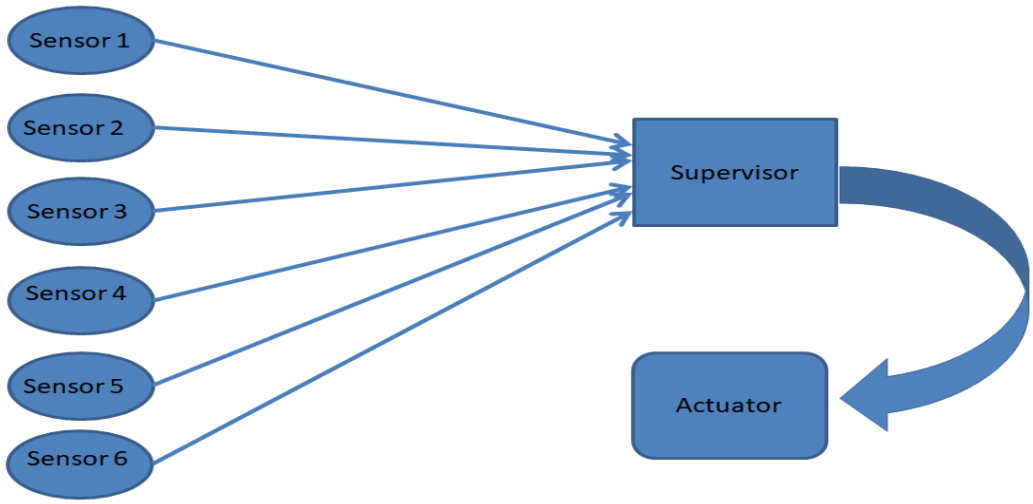

Figure 3.2. 1 Schematic model of data flow of one set for the faulty scenario

As mentioned previously, the used communication protocol is low power WiFi. All system nodes communicate via channel 1 at 1mW. The following two subsections present the Riverbed simulations results of the faulty model with and without interference. All the results are based on a 95% confidence analysis.

3.2.1 Fault-tolerant, interference free model results

The simulation results of the interference free model are demonstrated in this section. Table 3.4 shows the packets' end-to-end delays of the supervisor and all actuators. As shown, all the delays are less than 0.2msec; therefore, the delay requirement is met. Figure 3.2.2 to Figure 3.2.9 show the end to end delays of all actuators and the supervisor for 33 runs. The y-axis represents the packet end-to-end delay in seconds while the x-axis represents the simulation time. The different colors of the curves indicate the results of the 33 simulations.

Table 3. 4 95% Confidence intervals results for the faulty interference free scenario

Packet end-to-end delay interval	Delay interval (msec)
Actuator 1	[0.168905; 0.0578264]
Actuator 2	[0.121526; 0.0999862]
Actuator 3	[0.121881; 0.102323]
Actuator 4	[0.125128; 0.104354]
Actuator 5	[0.122000; 0.102423]
Actuator 6	[0.123000; 0.102183]
Actuator 7	[0.118000; 0.0975873]
Supervisor	[0.118000; 0.0975873]

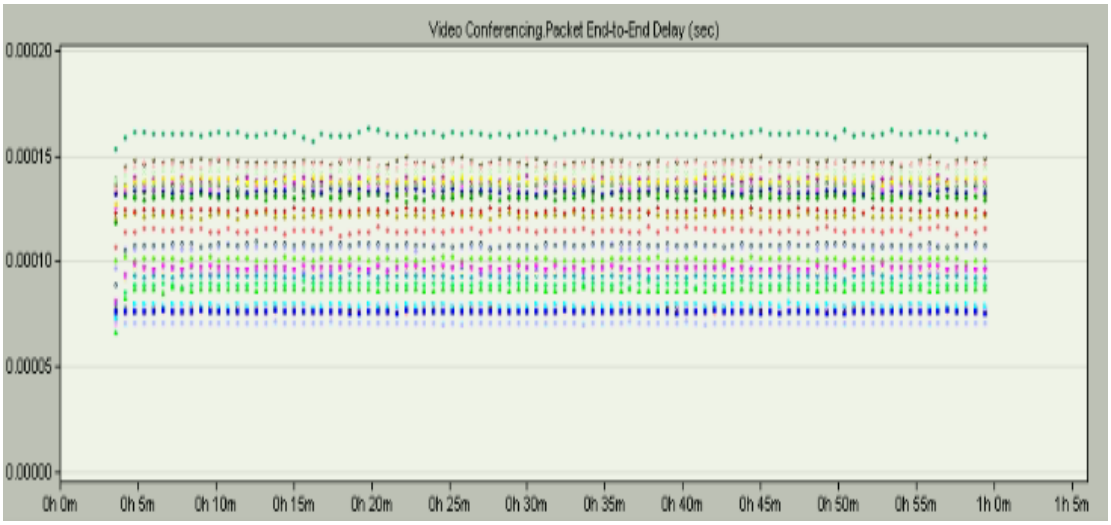

Figure 3.2. 2 Fault-Tolerant, Interference Free Model; Actuator 1 End-to-End delays (33 runs)

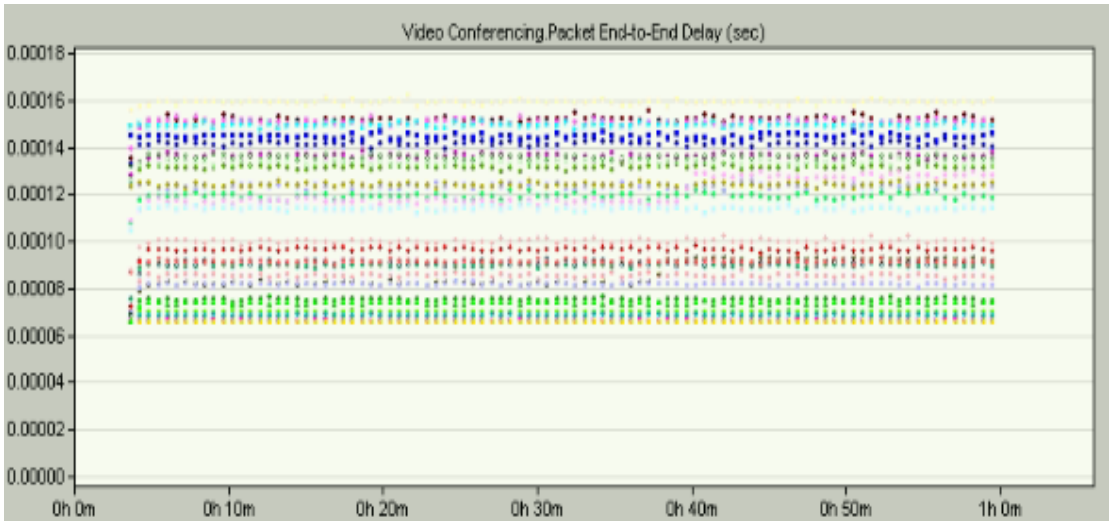

Figure 3.2. 3 Fault-Tolerant, Interference Free Model; Actuator 2 End-to-End delays (33 runs)

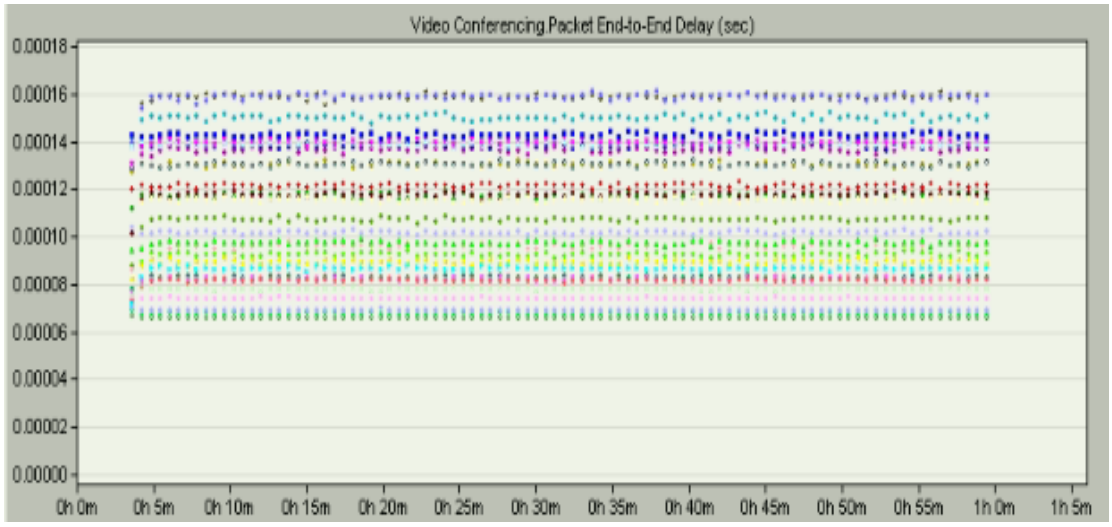

Figure 3.2. 4 Fault-Tolerant, Interference Free Model; Actuator 3 End-to-End delays (33 runs)

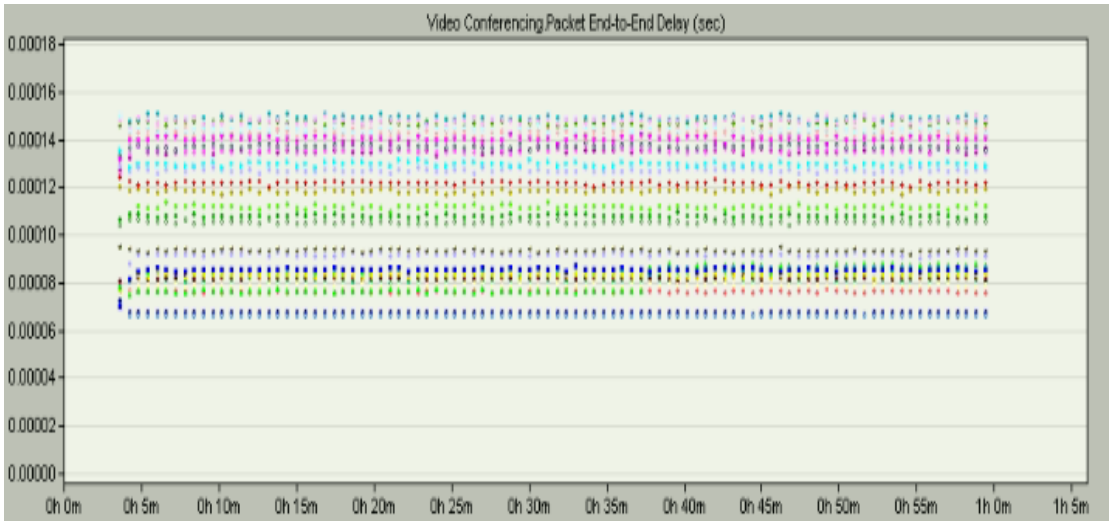

Figure 3.2. 5 Fault-Tolerant, Interference Free Model; Actuator 4 End-to-End delays (33 runs)

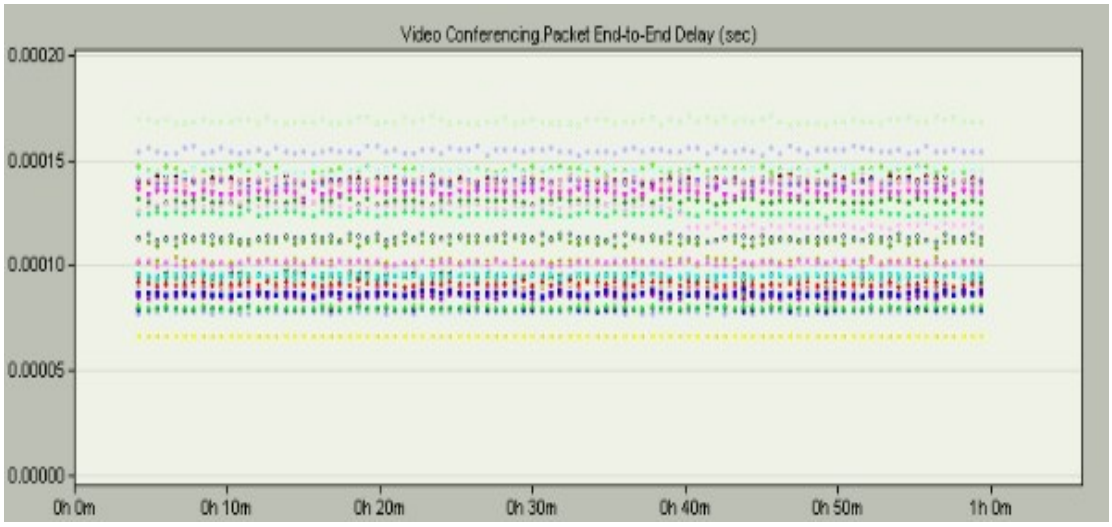

Figure 3.2. 6 Fault-Tolerant, Interference Free Model; Actuator 5 End-to-End delays (33 runs)

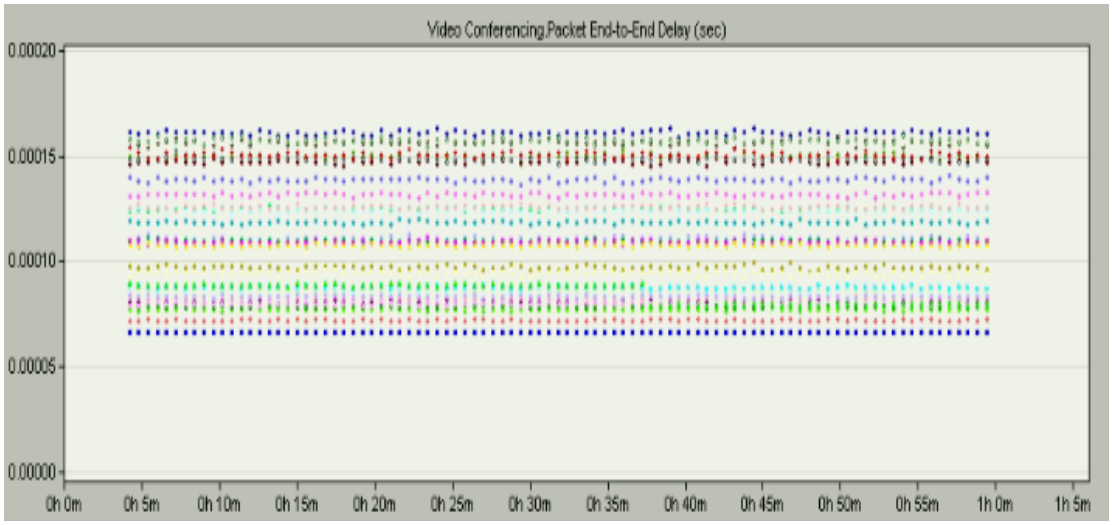

Figure 3.2. 7 Fault-Tolerant, Interference Free Model; Actuator 6 End-to-End delays (33 runs)

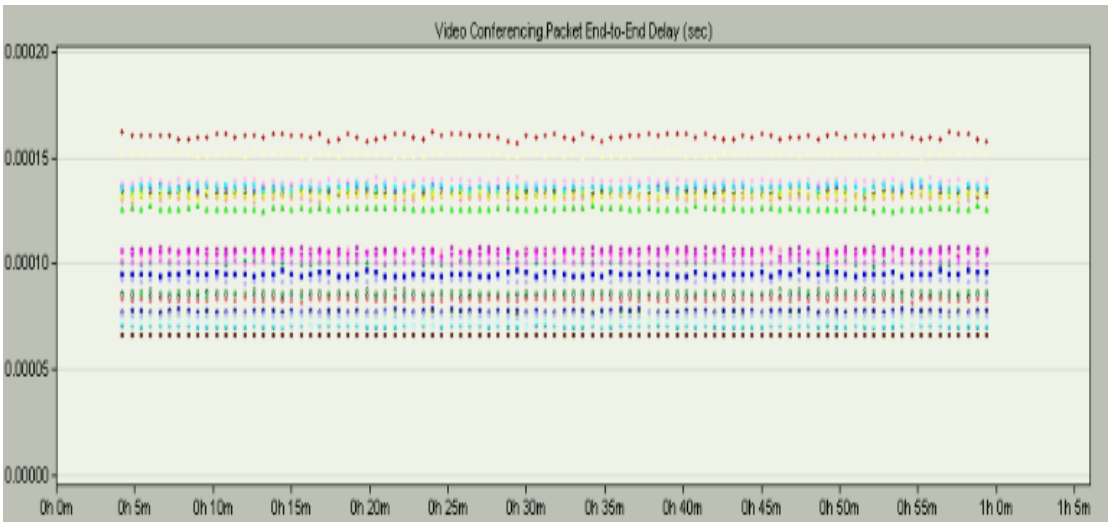

Figure 3.2. 8 Fault-Tolerant, Interference Free Model; Actuator 7 End-to-End delays (33 runs)

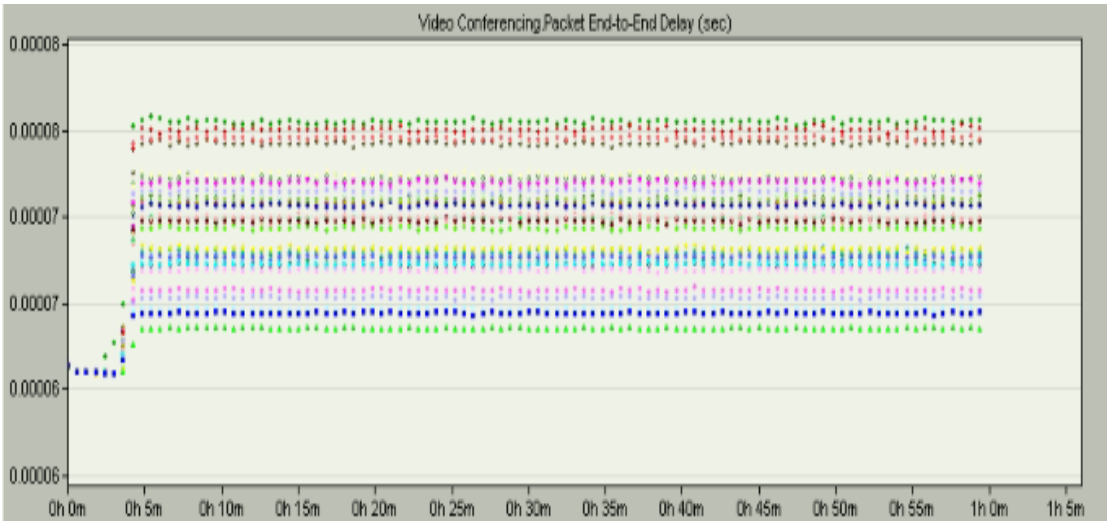

Figure 3.2. 9 Fault-Tolerant, Interference Free Model; Supervisor End-to-End delays (33 runs)

Further, Figure 3.2.10 to Figure 3.2.17 show the graphs of packets received at the 7 actuators and the supervisors. The x-axis represents the simulation time in minutes and the y-axis represents the received traffic in bytes/sec. The graphs show zero packet loss.

For simulating the outgoing traffic, the sensor outgoing stream inter-arrival time is 41msec, while the supervisor outgoing stream inter-arrival time is 21msec.

Recall that sets 1,2,3 and 4 contains 6 sensors while sets 5,6 and 7 contains 5 sensors.

-The supervisor receives the data aggregated from all actuators is calculated as follows:

$\left(\frac{6bytes}{0.041sec}\right)$x 4 + $\left(\frac{5bytes}{0.041sec}\right)$x3

-In addition, the traffic received at actuators 1 → 4 is calculated as follows:

$$\left(\frac{6bytes}{0.021sec}\right)$$

In addition, the traffic received at actuators 5 → 7 is calculated as follows:

$$\left(\frac{5bytes}{0.021sec}\right)$$

Hence, each node will receive the below information:

Actuator 1>> 285bytes/sec

Actuator 2>> 285 bytes/sec

Actuator 3>> 285 bytes/sec

Actuator 4>> 285 bytes/sec

Actuator 5>> 238 bytes/sec

Actuator 6>> 238 bytes/sec

Actuator 7>> 238 bytes/sec

Supervisor>> 950 bytes/sec

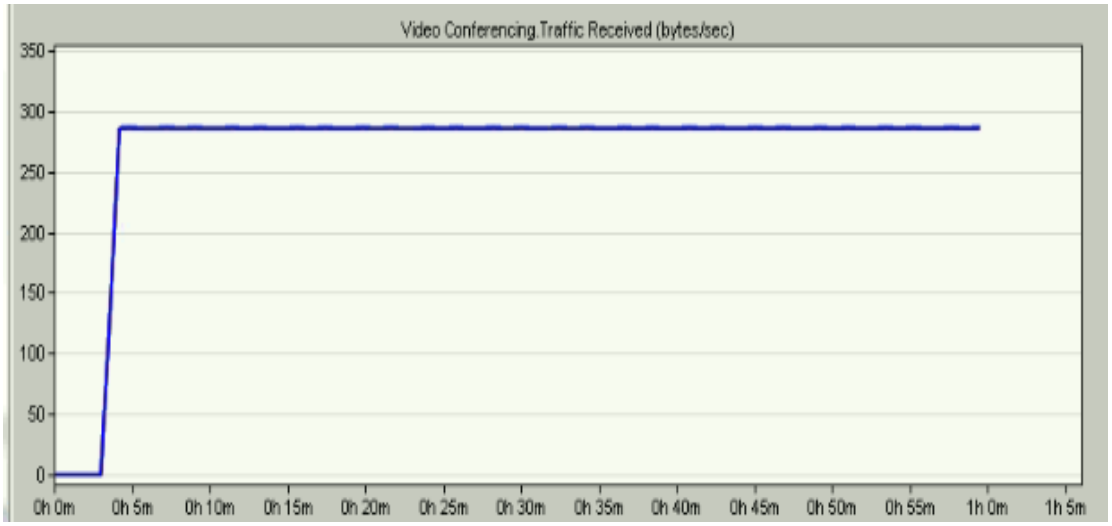

Figure 3.2. 10 Fault-Tolerant, Interference Free Model; traffic received at Actuator 1

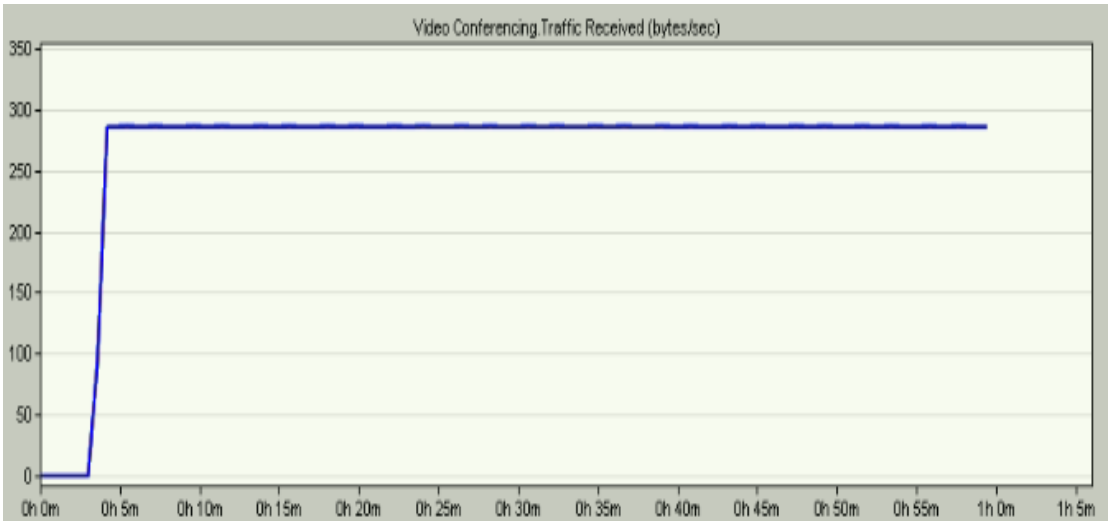

Figure 3.2. 11 Fault-Tolerant, Interference Free Model; traffic received at Actuator 2

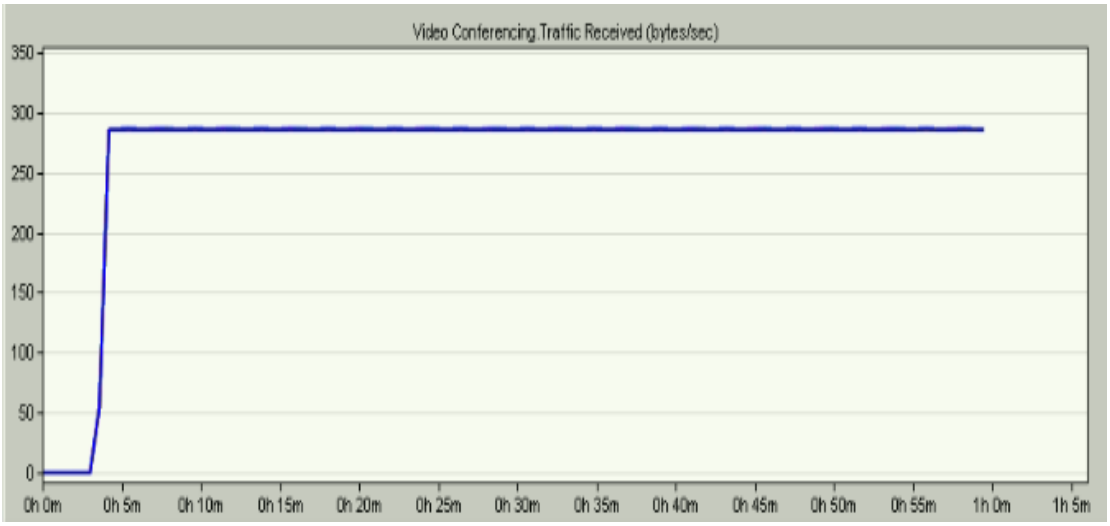

Figure 3.2. 12 Fault-Tolerant, Interference Free Model; traffic received at Actuator 3

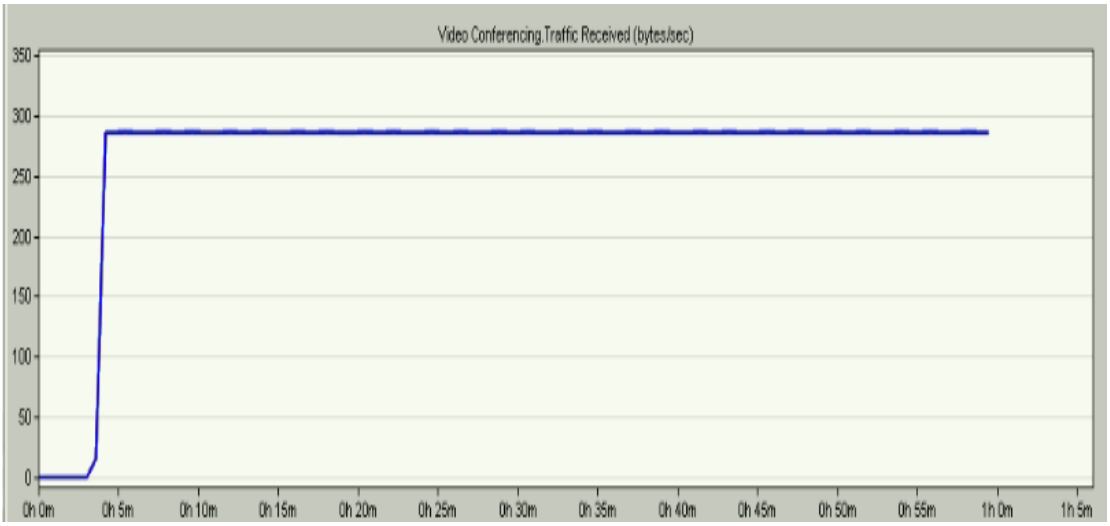

Figure 3.2. 13 Fault-Tolerant, Interference Free Model; traffic received at Actuator 4

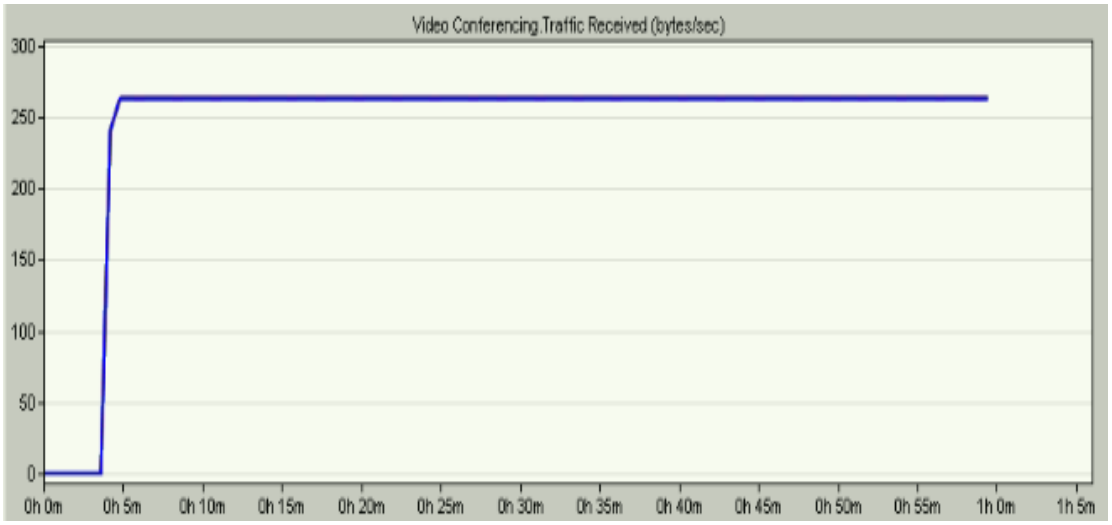

Figure 3.2. 14 Fault-Tolerant, Interference Free Model; traffic received at Actuator 5

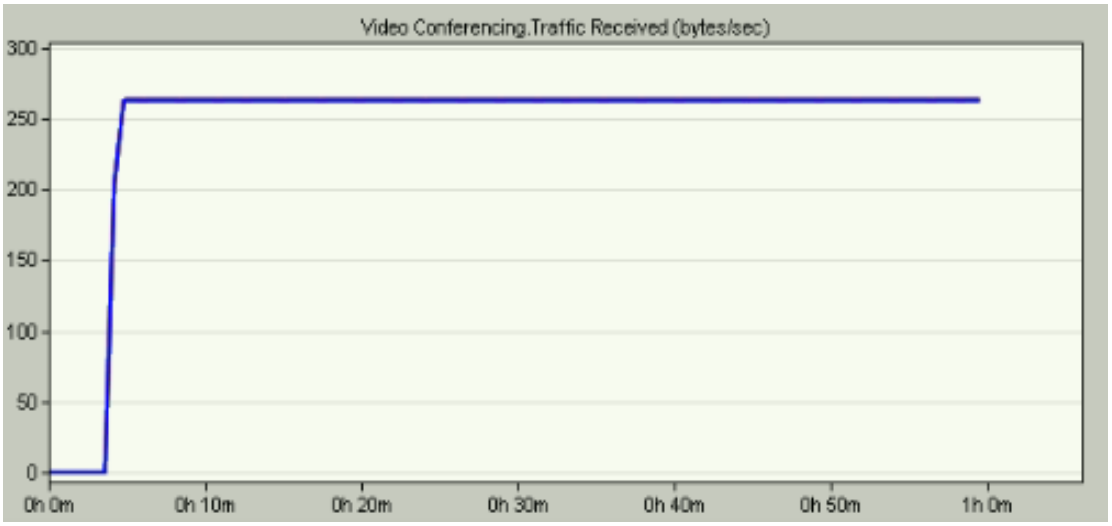

Figure 3.2. 15 Fault-Tolerant, Interference Free Model; traffic received at Actuator 6

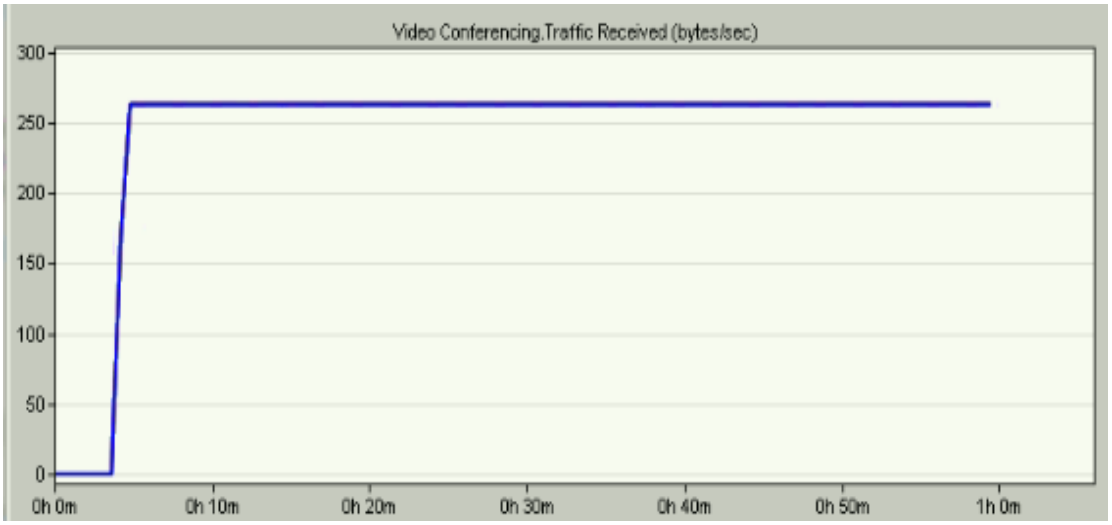

Figure 3.2. 16 Fault-Tolerant, Interference Free Model; traffic received at Actuator 7

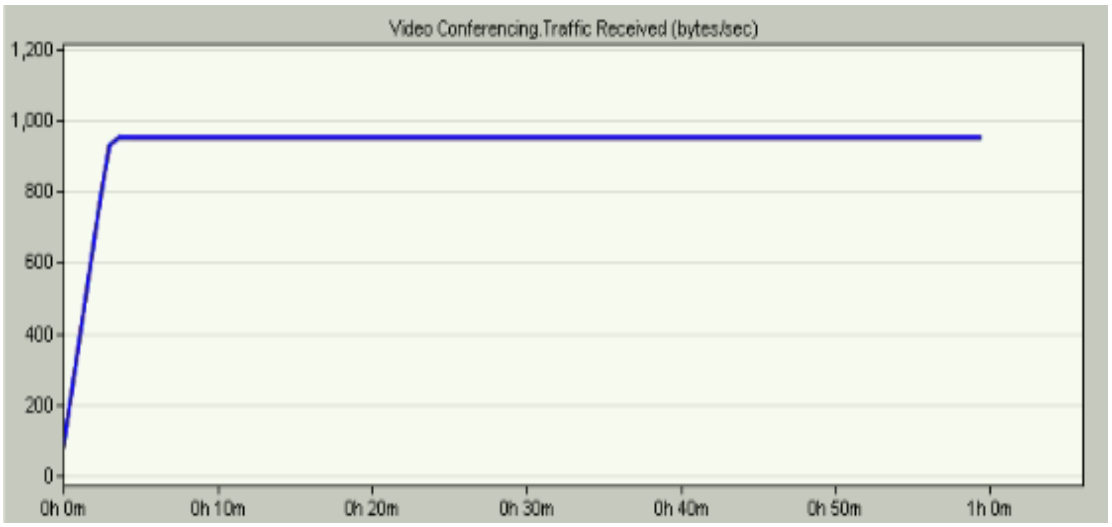

Figure 3.2. 17 Fault-Tolerant, Interference Free Model; traffic received at Supervisor

3.2.2 Fault-tolerant, with interference model results

The simulation results of the interference model are demonstrated in this subsection. The interference simulation is done typically as illustrated in the previous section. Table 3.5 shows the packets' end-to-end delays of the supervisor and all actuators. As shown, all the delays are less than 5 msec; therefore, the delay requirement is met. Figure 3.2.18 to Figure 3.2.25 show the end to end delays of all actuators and the supervisor for 33 runs. The y-axis represents the packet end-to-end delay in seconds while the x-axis represents the simulation time. The different colors of the curves indicate the results of the 33 simulations.

Table 3. 5 95% Confidence Intervals Results For the Fault and Interfered Scenario

Packet end-to-end delay interval	**Delay interval (msec)**
Actuator 1	[3.223814; 3.196413]
Actuator 2	[3.616807; 3.590146]
Actuator 3	[3.883393; 3.851567]
Actuator 4	[4.08042; 4.037759]
Actuator 5	[3.32291; 2.94726]
Actuator 6	[3.64859; 3.61902]
Actuator 7	[3.817418; 3.78496]
Supervisor	[1.72157; 1.70434]

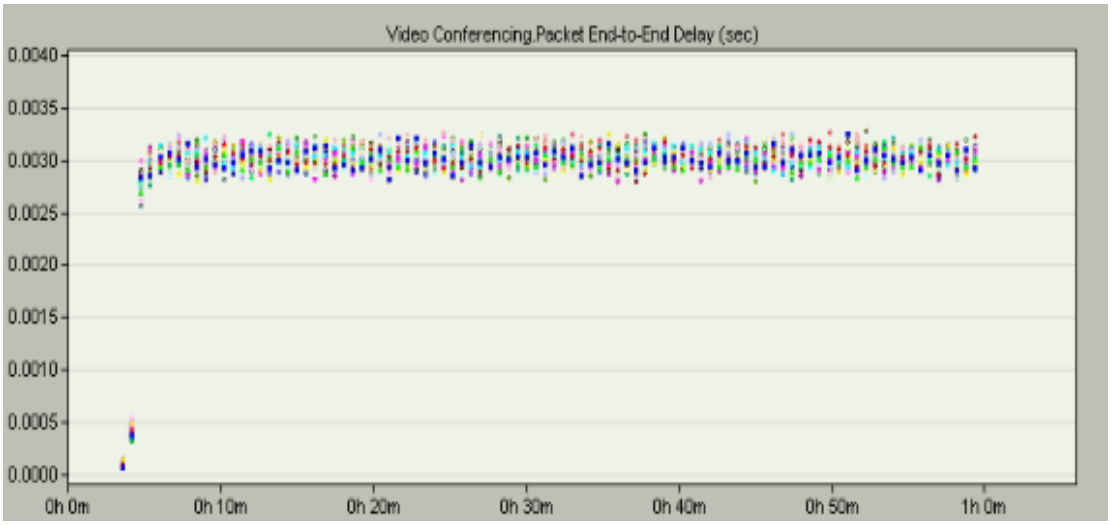

Figure 3.2. 18 Fault-Tolerant, Interference Model; Actuator 1 End-to-End delays (33 runs)

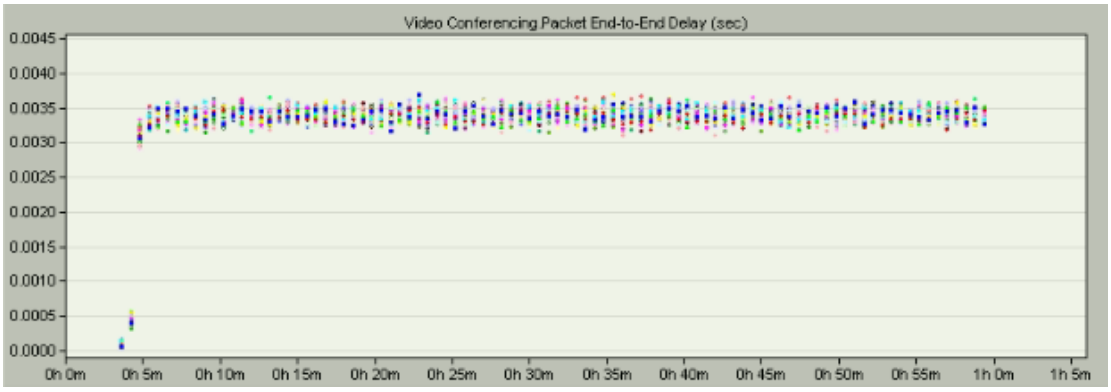

Figure 3.2. 19 Fault-Tolerant, Interference Model; Actuator 2 End-to-End delays (33 runs)

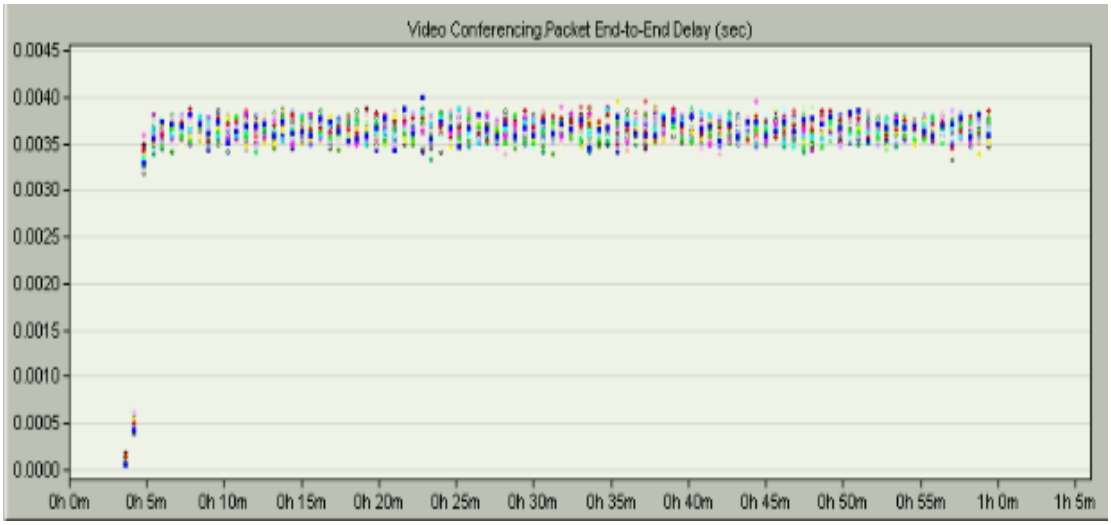

Figure 3.2. 20 Fault-Tolerant, Interference Model; Actuator 3 End-to-End delays (33 runs)

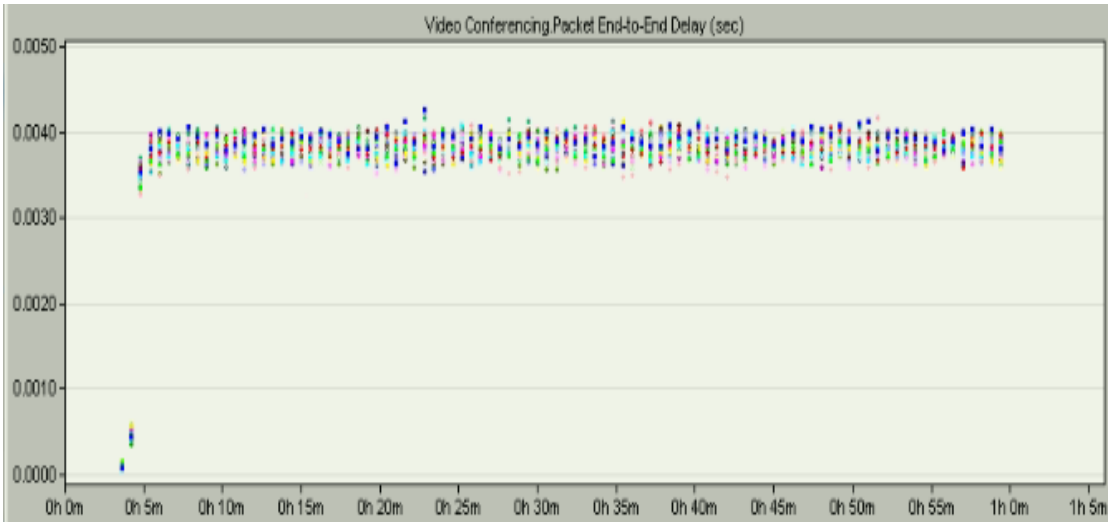

Figure 3.2. 21 Fault-Tolerant, Interference Model; Actuator 4 End-to-End delays (33 runs)

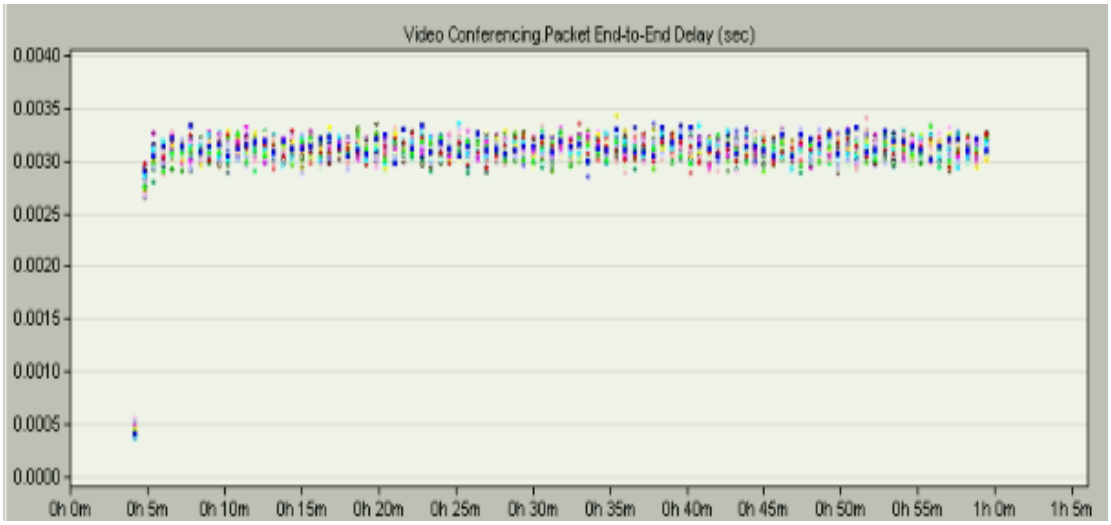

Figure 3.2. 22 Fault-Tolerant, Interference Model; Actuator 5 End-to-End delays (33 runs)

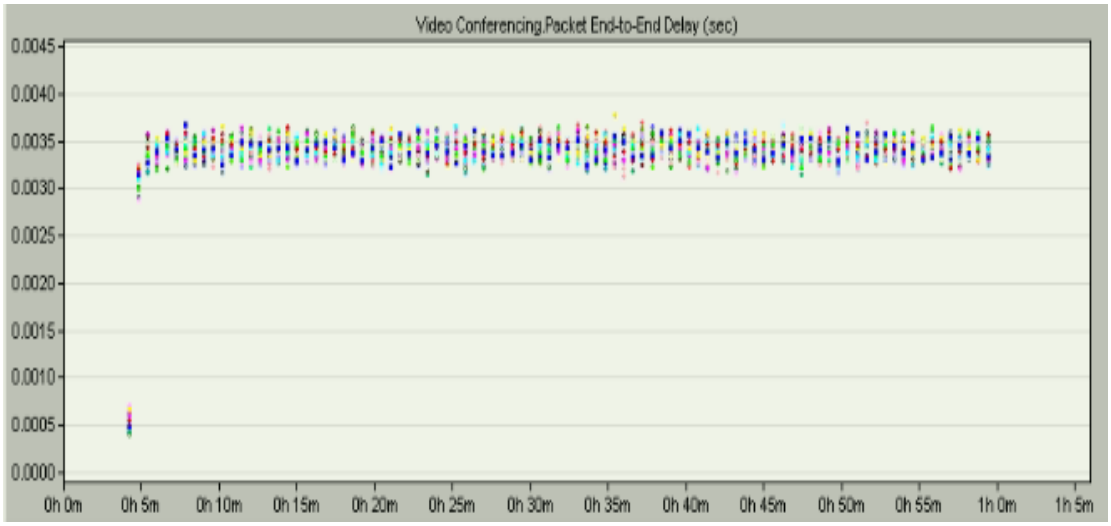

Figure 3.2. 23 Fault-Tolerant, Interference Model; Actuator 6 End-to-End delays (33 runs)

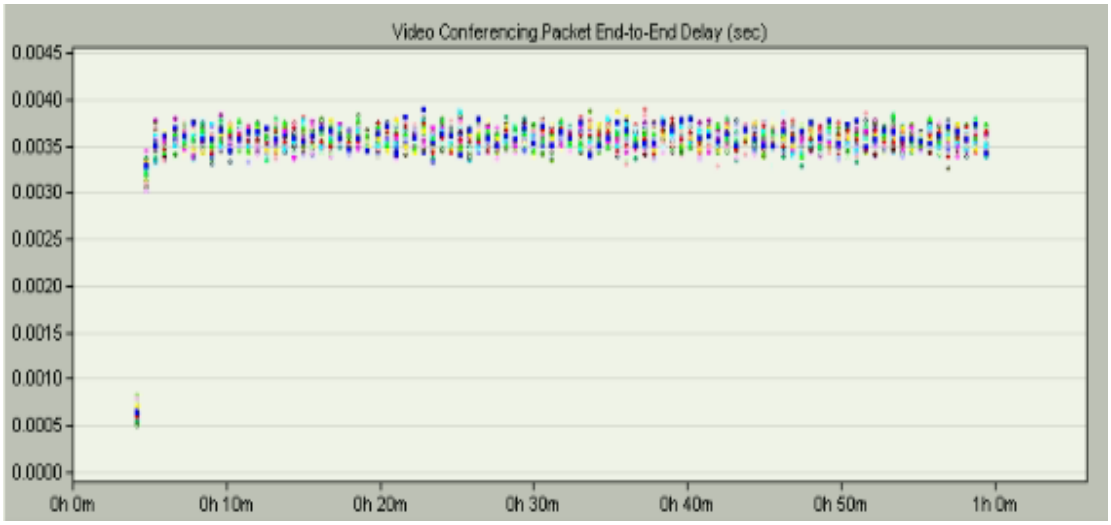

Figure 3.2. 24 Fault-Tolerant, Interference Model; Actuator 7 End-to-End delays (33 runs)

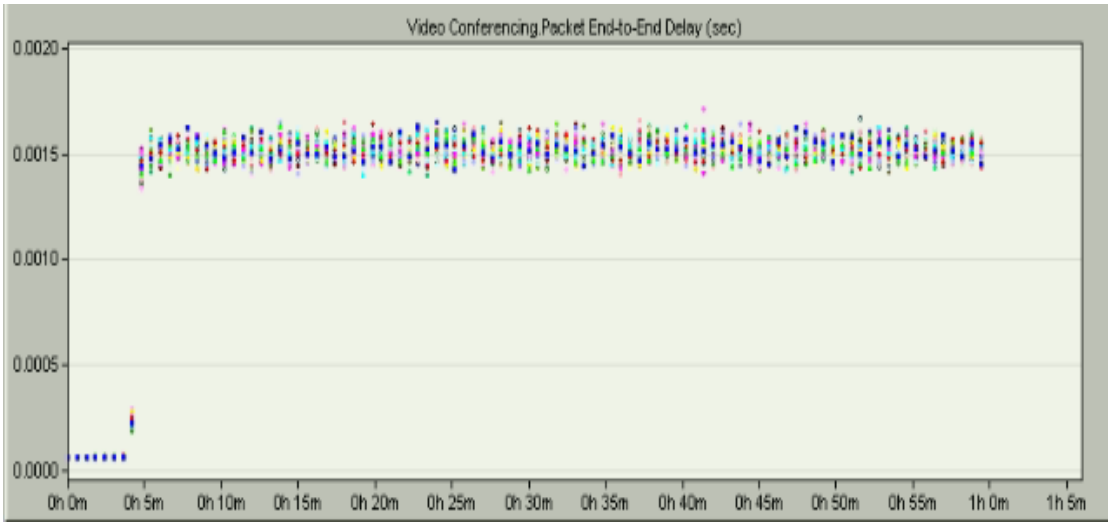

Figure 3.2. 25 Fault-Tolerant, Interference Model; Supervisor End-to-End delays (33 runs)

Further, Figure 3.2.26 to Figure 3.2.33 show the graphs of packets received at the 7 actuators and the supervisors. The x-axis represents the simulation time in minutes and the y-axis represents the received traffic in bytes/sec. The graphs show zero packet loss.

For simulating the outgoing traffic, the sensor outgoing stream inter-arrival time is 41msec, while the supervisor outgoing stream inter-arrival time is 21msec.

Recall that sets 1,2,3 and 4 contains 6 sensors while sets 5,6 and 7 contains 5 sensors.

-The supervisor receives the data aggregated from all actuators is calculated as follows:

$\left(\frac{6bytes}{0.041sec}\right)$x 4 + $\left(\frac{5bytes}{0.041sec}\right)$x3

-In addition, the traffic received at actuators 1 → 4 is calculated as follows:

$$\left(\frac{6bytes}{0.021sec}\right)$$

In addition, the traffic received at actuators 5 → 7 is calculated as follows:

$$\left(\frac{5bytes}{0.021sec}\right)$$

Hence, each node will receive the below information:

Actuator 1>> 285bytes/sec

Actuator 2>> 285 bytes/sec

Actuator 3>> 285 bytes/sec

Actuator 4>> 285 bytes/sec

Actuator 5>> 238 bytes/sec

Actuator 6>> 238 bytes/sec

Actuator 7>> 238 bytes/sec

Supervisor>> 950 bytes/sec

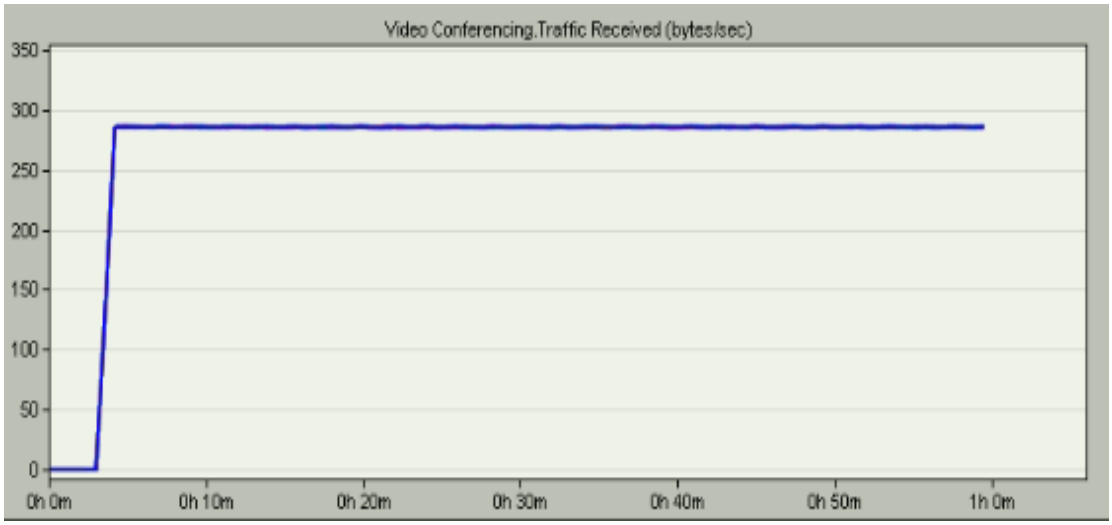

Figure 3.2. 26 Fault-Tolerant, Interference Model; traffic received at Actuator 1

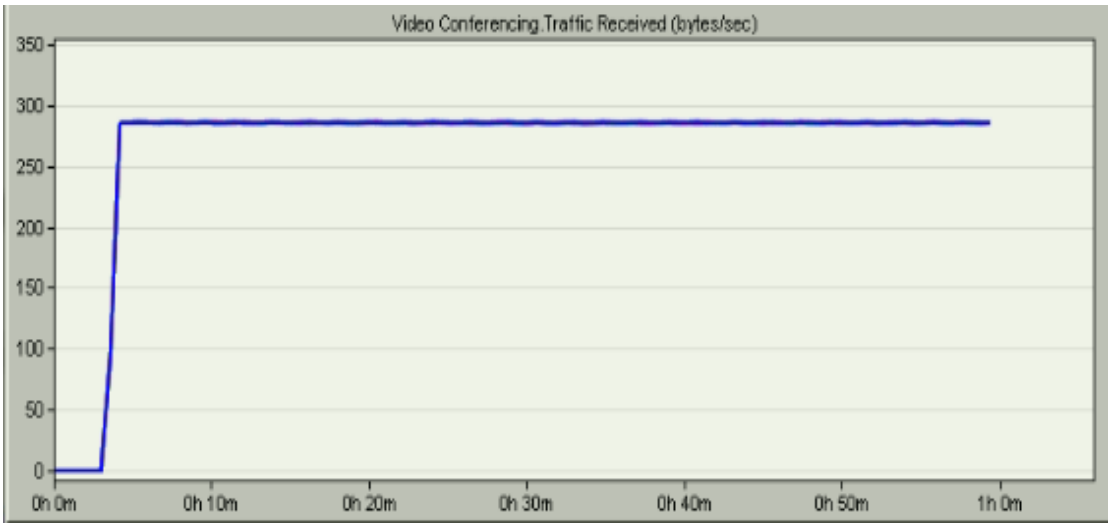

Figure 3.2. 27 Fault-Tolerant, Interference Model; traffic received at Actuator 2

\

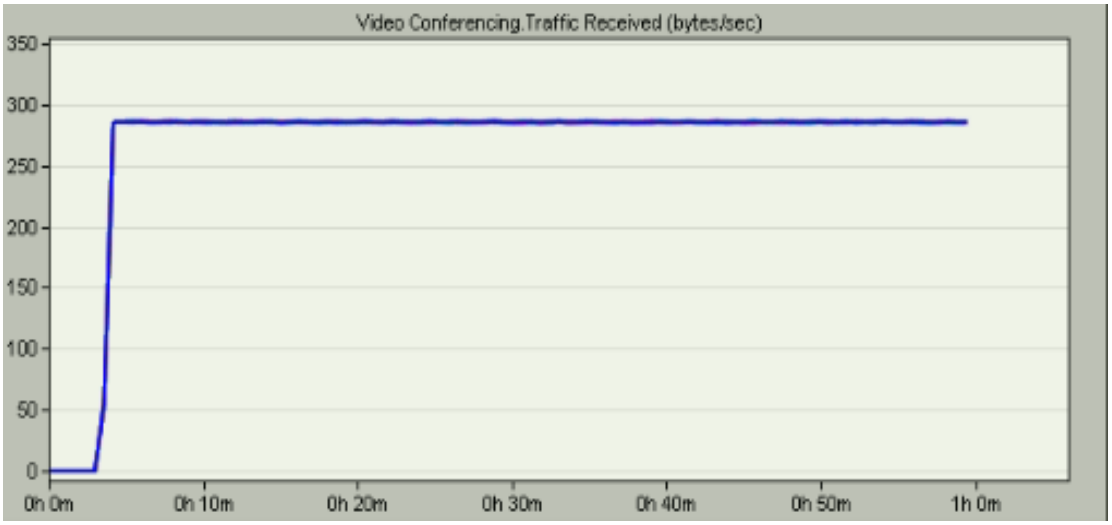

Figure 3.2. 28 Fault-Tolerant, Interference Model; traffic received at Actuator 3

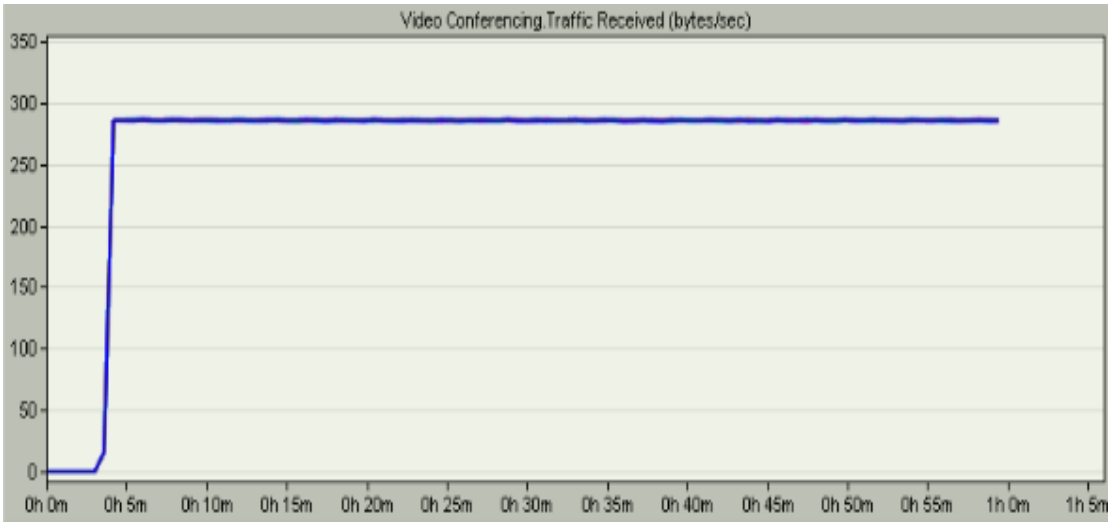

Figure 3.2. 29 Fault-Tolerant, Interference Model; traffic received at Actuator 4

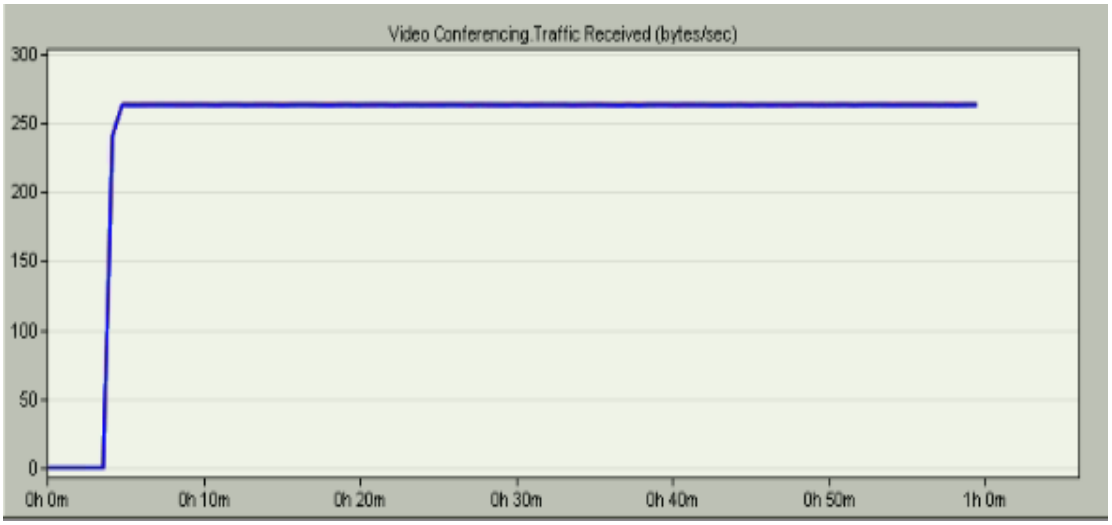

Figure 3.2. 30 Fault-Tolerant, Interference Model; traffic received at Actuator 5

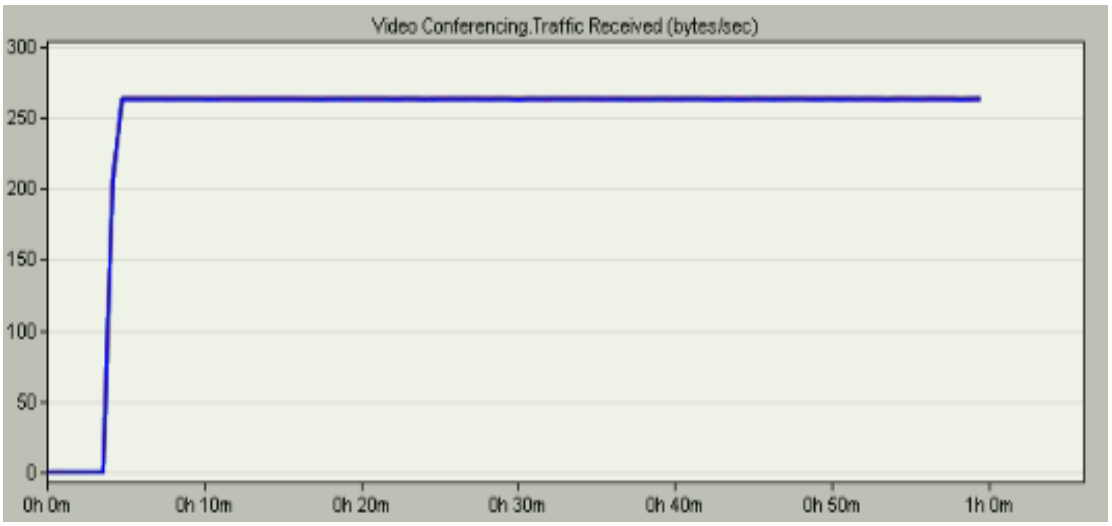

Figure 3.2. 31 Fault-Tolerant, Interference Model; traffic received at Actuator 6

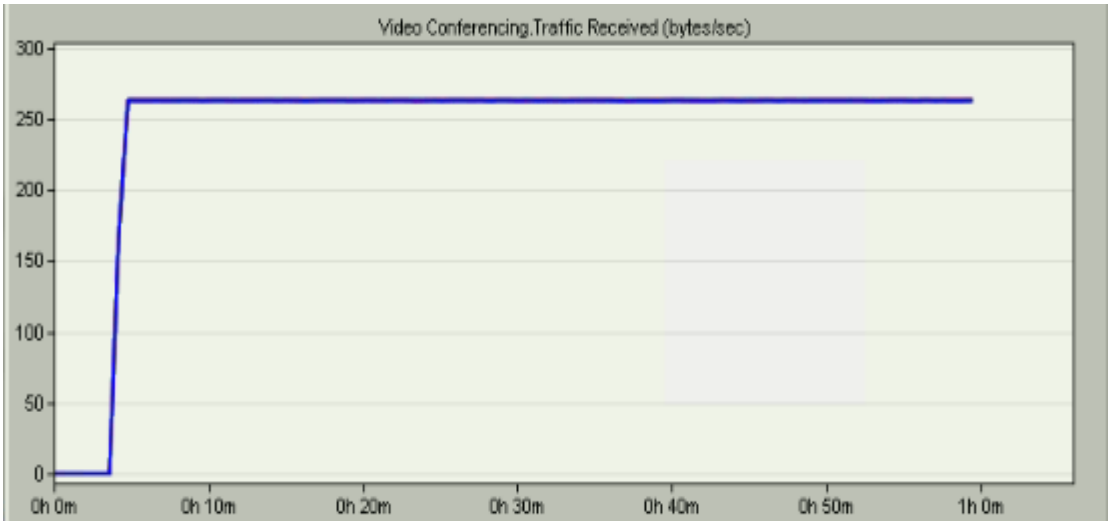

Figure 3.2. 32 Fault-Tolerant, Interference Model; traffic received at Actuator 7

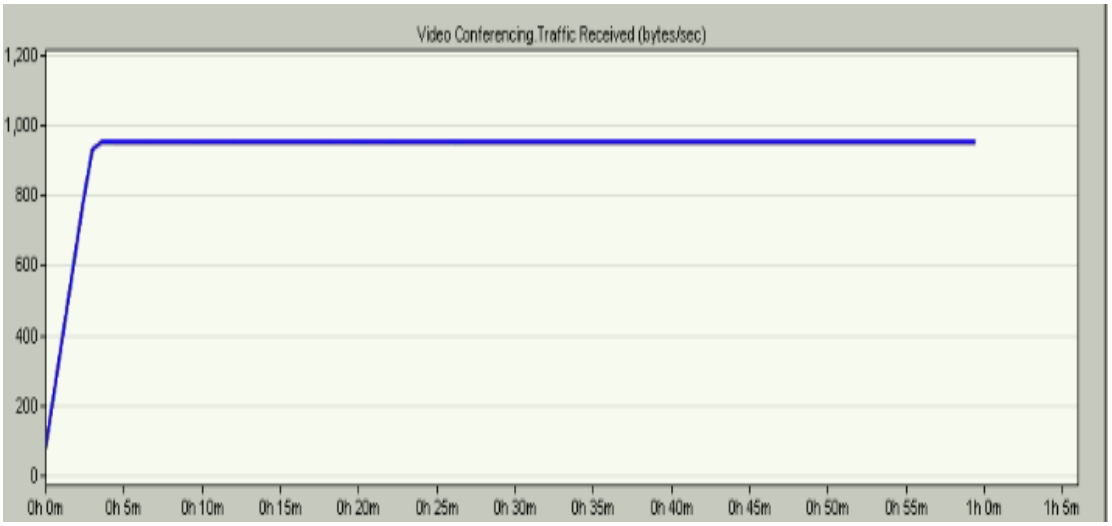

Figure 3.2. 33 Fault-Tolerant, Interference Model; traffic received at Supervisor

Chapter 4

4.1 Reliability Concepts [Smith 2005, Koren 2007]

The general definition of reliability is the absence of failure. That means that the device of interest is still utilized even with its minimum available components. The reliability of complex equipment depend various aspects such as: the manufacturing method, the surrounding environment that the equipment of interest operates, etc. The major role of a reliability engineer is defining and handling the hazards that could harmfully disturb the business operations. In order words, the reliability engineer's major role is reduction of losses. A lot of work and research is already done to understand the stochastic nature of failures as well as to reduce their probability to happen.

Some reliability properties which values the efficiency of the system are probability of survival P(t), mean time to failure (MTTF) which is the average time the system takes to fail, mean time to repair (MTTR) and mean time between failures (MTBF) which is MTTF + MTTR. "T" is a random variable which denotes the time the system is alive. The probability that the system is alive for time "t" is R(t) = P(T>t). Since R (t) is a probability, it always satisfies the below conditions:

$$0 < R(t) < 1 \quad \text{and}$$

$$R(t) = 0; \text{ if } t = \infty$$

$$R(t) = 1; \text{ if } t = 0$$

Bathtub Curve:

As shown in figure 4.1, the component fails with high probability at the beginning of its operation and that is called infant failures. The reason for these early failures can be manufacturing faults. Then, the middle phase represents the service failures which has a certain rate (number of failures/unit time). Finally, by time and after the device has reached the end of its lifetime, the failure rate increases as the systems degrades and such failures are known as "wear-out failures"

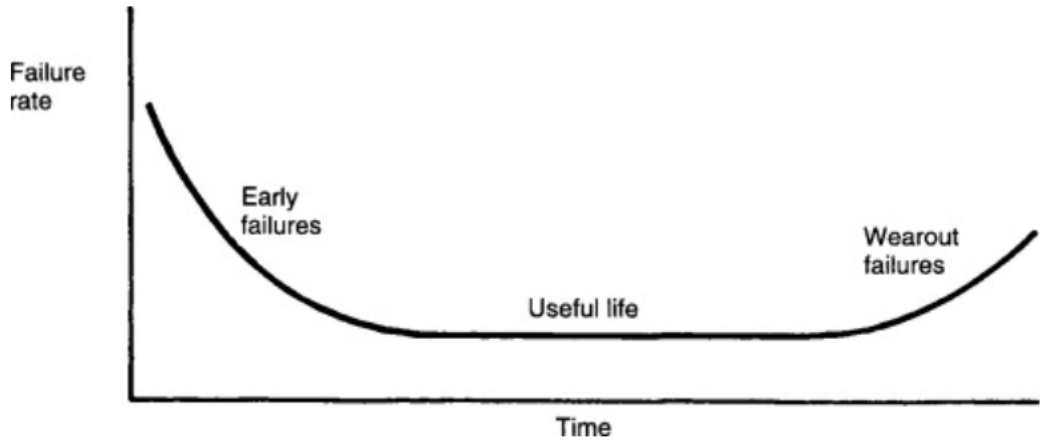

Figure 4. 1 Bathtub curve [Smith 2005]

4.2 MARKOV MODEL [ZIMMERMANN 2010]

A Markov Model is a stochastic model which represents the possible states of a random process. The transitions between the different states depend on certain probabilities. It is assumed that the probability of transitioning from one state to another depends only on the current state. So the difference between markov chain and any other stochastic process is being "memoryless". The Markov property is described using the below equation.

$$P(X_n = i_n \mid X_{n-1} = i_{n-1}) = P(X_n = i_n \mid X_0 = i_0, X_1 = i_1, \ldots, X_{n-1} = i_{n-1}).$$

4.3 MARKOV MODEL RESULTS

The reliability of the system has been calculated using Markov models. All Markov models have been simulated using SHARPE [SHARPE 2020]. SHARPE is the abbreviation for "Symbolic Hierarchical Automated Reliability and Performance Evaluator". This modelling tool was developed by Duke University for analyzing the stochastics models in order to obtain the reliability, availability and performability.

4.3.1 WITHOUT REPAIR

Referring to chapter 3, when the controller (which is part of the actuator's node) fails, the supervisor will take over its function and the sensors data will be sent to the supervisor where the control action will be calculated and sent to the actuator. It was proven in chapter 3 that, even if all 7 controllers stopped working, the supervisor can still handle the system. Accordingly, the Markov model that is shown in figure 4.3.1 has been simulated by SHARPE.

Below is the explanation of the symbols in the figure:

S: Supervisor failure rate

K: Controller failure rate

S7: 7 controllers are working

S6: 6 controllers are working

S5: 5 controllers are working

S4: 4 controllers are working

S3: 3 controllers are working

S2: 2 controllers are working

S1: 1 controller is working

SUP: only the supervisor is working

F: Total system failure

At State S7, where all 7 controllers are working, if one controller fails, the system will transition with failure rate 7k to state S6, where only 6 controllers are working. If the supervisor fails, the system will transition with a failure rate of s to state F. The reason that the system will totally fail upon the supervisor failure is that the supervisor is the interface between the patient's nerves and the MPL. The brain decides the required action and the supervisor sends this required action to the controllers. A supervisor failure will incapacitate the system because the required actions do not reach the controllers anymore.

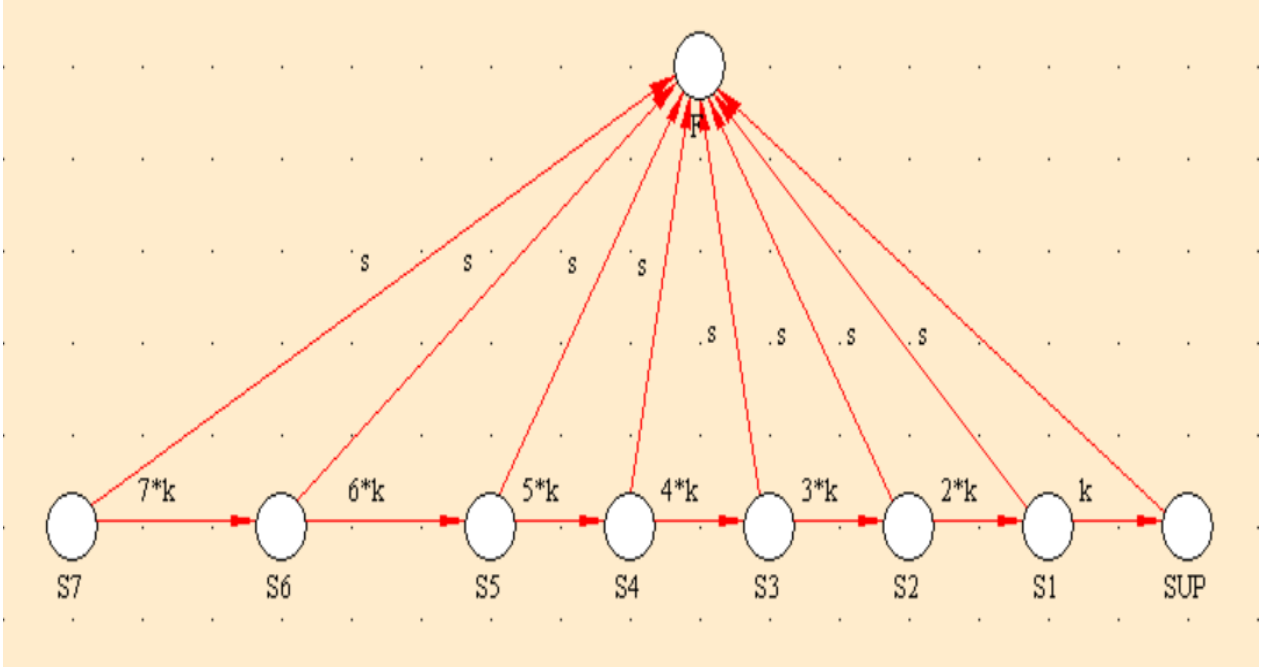

Figure 4.3. 1 Markov model for the system without repair

The Markov model that is shown in figure 4.3.1 has been simulated by SHARPE in order to solve the below mathematical equations:

Let P(t)= [P_{S7} (t) P_{S6} (t) P_{S5}(t) P_{S4}(t) P_{S3}(t) P_{S2}(t) P_{S1}(t) P_{SUP}(t) P_F(t)], where P(0)= [1 0 0 0 0 0 0 0 0] and dP/dt= P x T, where T is the differential state transition rate matrix:

$$\boldsymbol{T} = \begin{bmatrix} -(7k+s) & 7k & 0 & 0 & 0 & 0 & 0 & 0 & s \\ 0 & -(6k+s) & 6k & 0 & 0 & 0 & 0 & 0 & s \\ 0 & 0 & -(5k+s) & 5k & 0 & 0 & 0 & 0 & s \\ 0 & 0 & 0 & -(4k+s) & 4k & 0 & 0 & 0 & s \\ 0 & 0 & 0 & 0 & -(3k+s) & 3k & 0 & 0 & s \\ 0 & 0 & 0 & 0 & 0 & -(2k+s) & 2k & 0 & s \\ 0 & 0 & 0 & 0 & 0 & 0 & -(k+s) & k & s \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -s & s \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

By using Laplace transform, P(t) can be obtained and then R(t) can be calculated as below:

$$R(t)= 1- P_F(t)$$

The supervisor failure rate s is calculated based on the assumption that the supervisor Mean Time To failure (MTTF) is about 10 years so s will be equal to 0.1/year. While k, which is

the controller failure rate, is calculated based on an MTTF of 1 year. Accordingly, k is equal to 1/year. Figure 4.3.2 shows the graph which represents the Reliability Vs time; where s= 0.1/year and k = 1/year. As shown in the figure and in table 4.1, after 4 years, the reliability is 67.03 %. The graph has been drawn with steps = 0.1.

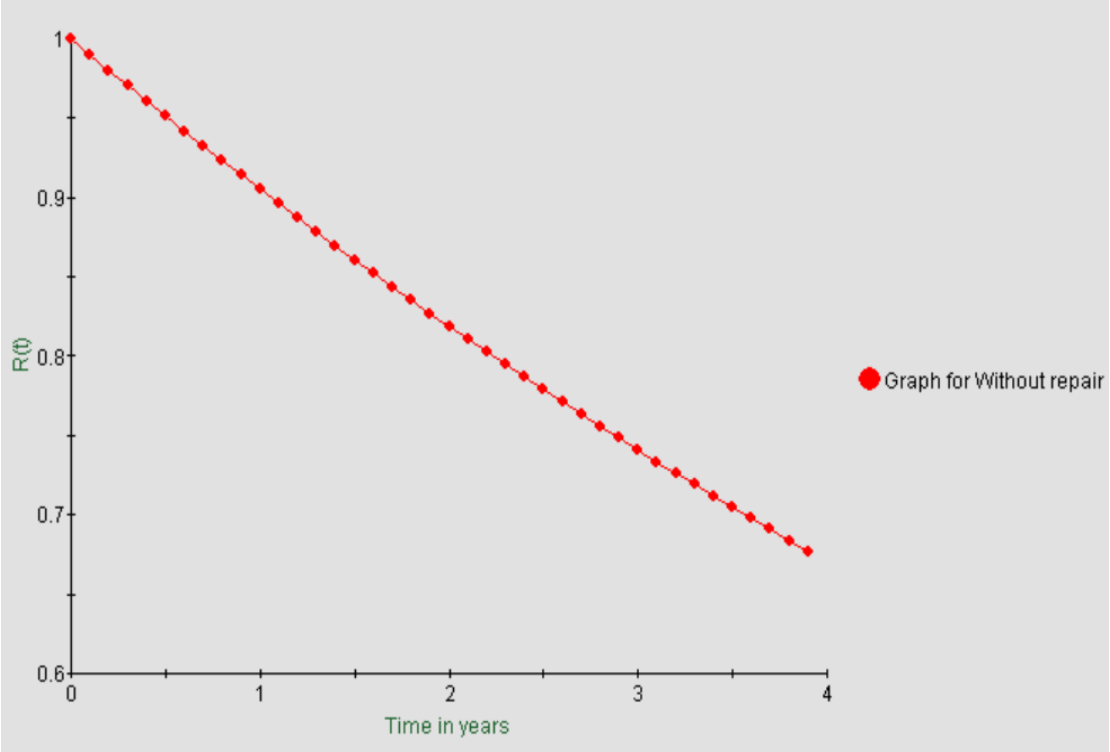

Figure 4.3. 2 Reliability Vs time for the model without repair

Table 4. 1 Reliability values over time for the model without repair

Time in years	R(t)
t=0	1.00000000
t=1	0.904837420
t=2	0.818730762
t=3	0.740818235
t=4	0.670329970

It is important to note here that the reliability of the Supervisor (on its own) is $e^{-s \times t}$. At t=1 year, the reliability of the Supervisor is 90.4837% which is exactly the reliability of the entire

system at the same time. Also, at t=4years, the reliability of the supervisor is 67.032%, which is again the system reliability in Table 4.1. The rationale behind this observation is that the Supervisor is really a single point of failure in the architecture described in chapter 3. Notice that every state in the Markov model of figure 4.3.1 has a transition (with a rate s) to the F state.

Furthermore, consider repair. If repair is introduced into the Markov model of figure 4.3.1, a transition from each of the following states to state S7, have to be added. For a Mean Time To Repair (MTTR) of one day, the repair rate will be 365/year. SHARPE results indicate that the reliability of the system does not any significant change.

In chapter 3, there was an implicit assumption, namely that the supervisor was capable of handling its own load along with that of one or more controllers if they fail. Consequently, the processing capability of the supervisor had to be relatively high. A lower cost supervisor would only be able to handle the tasks of several (but not all) controllers. Without any loss of generality, let the supervisor carry its own tasks along with those of three of the seven controllers. This will reduce the complexity/cost of the supervisor as well as its power consumption. Especially in developing countries, a low-cost supervisor and a long-lasting battery are very attractive features. Next, reliability models will be developed for situations where the supervisor is only capable of handling the task of a few (but not all) controllers upon their failure.

4.3.2 WITH REPAIR

High performance supervisors may not be available or prohibitively expensive in developing countries; hence, an alternative idea is to use a supervisor with lower specifications. The supervisor will NOT be able to carry the tasks of all seven controllers; hence, the repair option must be added to the system. Two models are studied next. In the first model, controller repair is performed as soon as the failure occurs. In the second model, repair is only carried out upon the failure of the third controller.

4.3.2.1 WITH IMMEDIATE REPAIR

Figure 4.3.3 shows the Markov model of the system with the immediate repair where m is the repair rate. With "immediate repair", it is meant that the controller is repaired offline as soon as the failure occurs NOT online repair.

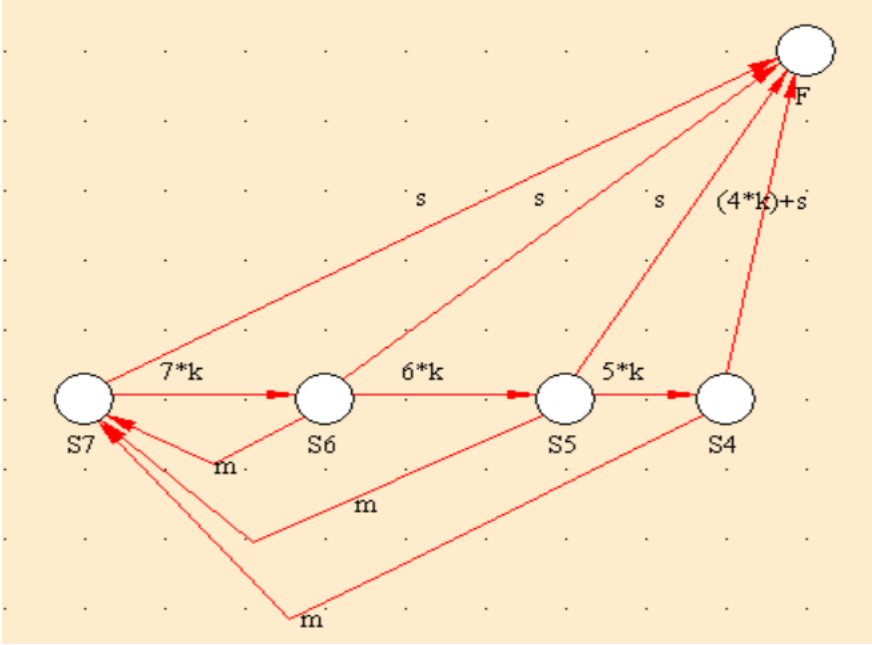

Figure 4.3. 3 Markov model for the system with immediate repair

The Markov model that is shown in figure 4.3.3 has been simulated by SHARPE in order to solve the below mathematical equations:

Let P(t)= [P_{S7} (t) P_{S6} (t) P_{S5}(t) P_{S4}(t) P_F(t)], where P(0)= [1 0 0 0 0] and

dP/dt= P x T, where T is the differential state transition rate matrix:

$$T = \begin{bmatrix} -(7k+s) & 7k & 0 & 0 & s \\ m & -(m+s+6k) & 6k & 0 & s \\ m & 0 & -(m+s+5k) & 5k & s \\ m & 0 & 0 & -(s+m+4k) & 4k+s \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

By using Laplace transform, P(t) can be obtained and then R(t) can be calculated as below:

$$R(t)= 1- P_F(t)$$

Note that the m, which is the repair rate, is calculated based on the Mean Time To Repair (MTTR) which is the time taken by the system to be repaired. Several test cases are done to study system performance. In all test cases, the supervisor failure rate s will be calculated based on MTTF equals 10 years so s equals 0.1/year.

Case 1:

In this test case, the failure rate of the controller k is calculated based on an MTTF equal to 1 year. Accordingly, k equals 1/year, while the repair rate m is evaluated based on an MTTR of 1 day so m equals 365/year. Note that this is a very optimistic repair time/rate in developing countries. Figure 4.3.4 shows the graph which represents the Reliability Vs time; where s= 0.1/year and k = 1/year and m=365/year and the values are shown in table 4.2.

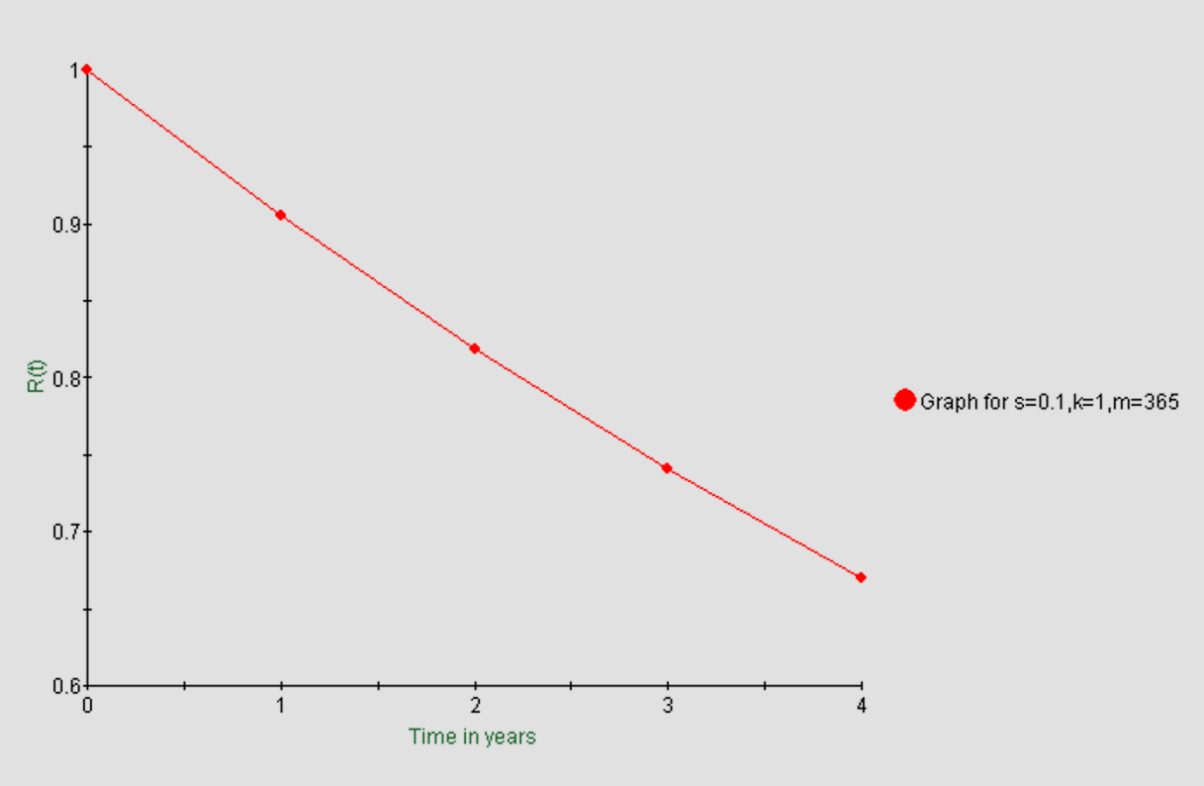

Figure 4.3. 4 Reliability Vs time for the model with immediate repair (s= 0.1/year, k = 1/year and m=365/year)

Table 4. 2 Reliability over time for the model with immediate repair (s= 0.1/year, k = 1/year and m=365/year)

Time in years	R(t)
t=0	1.00000000
t=1	0.904822870
t=2	0.818704382
t=3	0.740782465
t=4	0.670276964

Case 2:

In this test case, the failure rate of the controller k is calculated based on MTTF equals 2 years so k equals 0.5/year. While the repair rate m is evaluated also based on an MTTR of 1 day therefore m equals 365/year. Figure 4.3.5 shows the graph which represents the Reliability Vs time; where s= 0.1/year and k = 0.5/year and m=365/year and the exact values are shown in table 4.3.

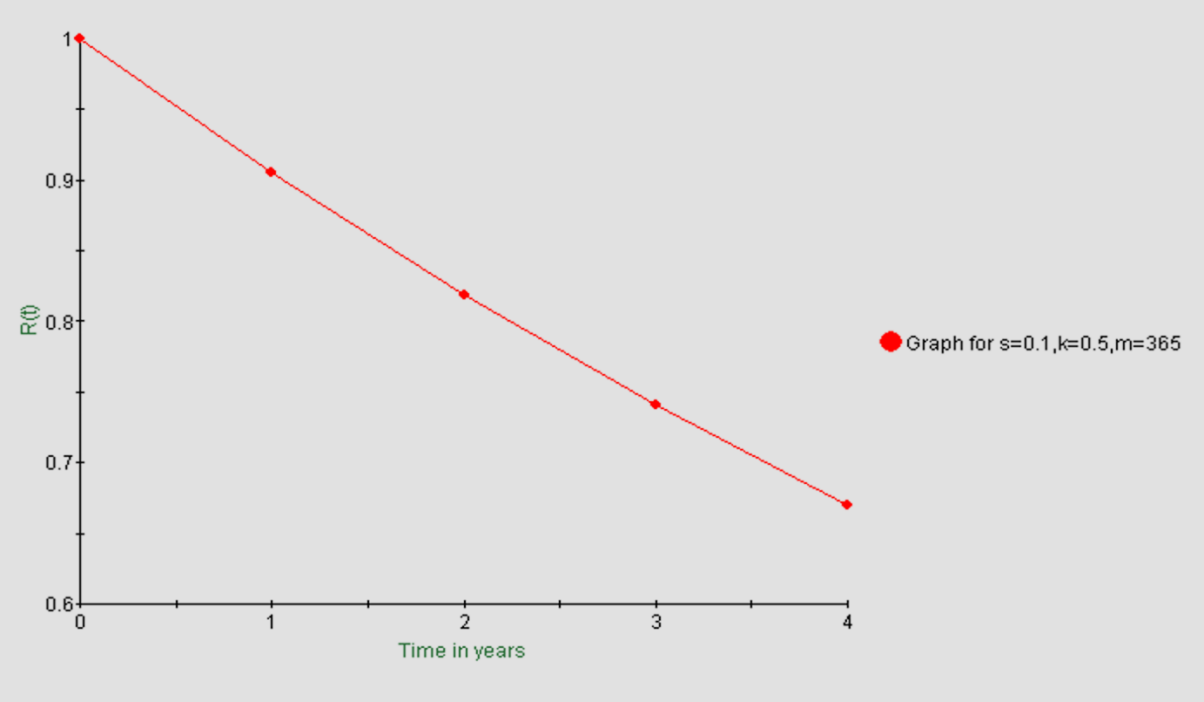

Figure 4.3. 5 Reliability Vs time for the model with immediate repair (s= 0.1/year, k = 0.5/year and m=365/year).

Table 4. 3 Reliability Vs time for the model with immediate repair (s= 0.1/year, k = 0.5/year and m=365/year).

Time in years	R(t)
t=0	1.000
t=1	0.904836528
t=2	0.818729223
t=3	0.740816199
t=4	0.670317684

Note that when the MTTF of the controller increases from 1 to 2 years, the reliability of the system increases slightly. Accordingly, in the following cases, the system reliability will be evaluated using different repair rates m.

Case 3:

In this test case, the failure rate of the controller k equals 0.5/year. While the repair rate m is evaluated based on MTTR equals 1 week; therefore, m equals 52/year. Figure 4.3.6 shows the graph which represents the Reliability Vs time; where s= 0.1/year and k = 0.5/year and m=52/year and the values are shown in table 4.4.

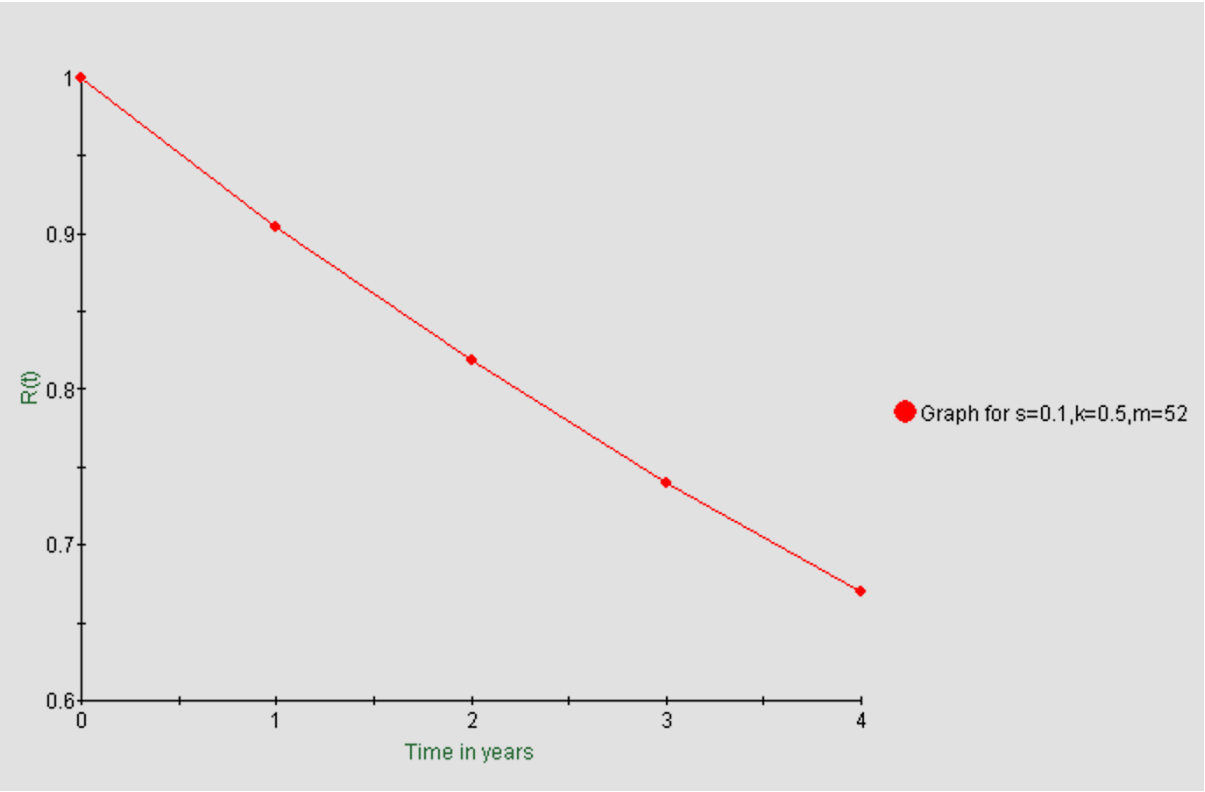

Figure 4.3. 6 Reliability Vs time for the model with immediate repair (s= 0.1/year, k = 0.5/year and m=52/year).

Table 4. 4 Reliability Vs time for the model with immediate repair (s= 0.1/year, k = 0.5/year and m=52/year).

Time in years	R(t)
t=0	1.00000000
t=1	0.904577298
t=2	0.818246723
t=3	0.740155329
t=4	0.669516816

Case 4:

In this next test case, the failure rate of the controller k equals 0.5/year. While the repair rate m is evaluated based on MTTR equals 2 weeks therefore m equals 26/year. Figure 4.3.7 shows the graph which represents the Reliability Vs time; where s= 0.1/year and k = 0.5/year and m=26/year and the values are shown in table 4.5. Comparing the values of table 4.4 and

table 4.5, the reliability has slightly decreased. Comparing all cases, case 2 has shown the best results.

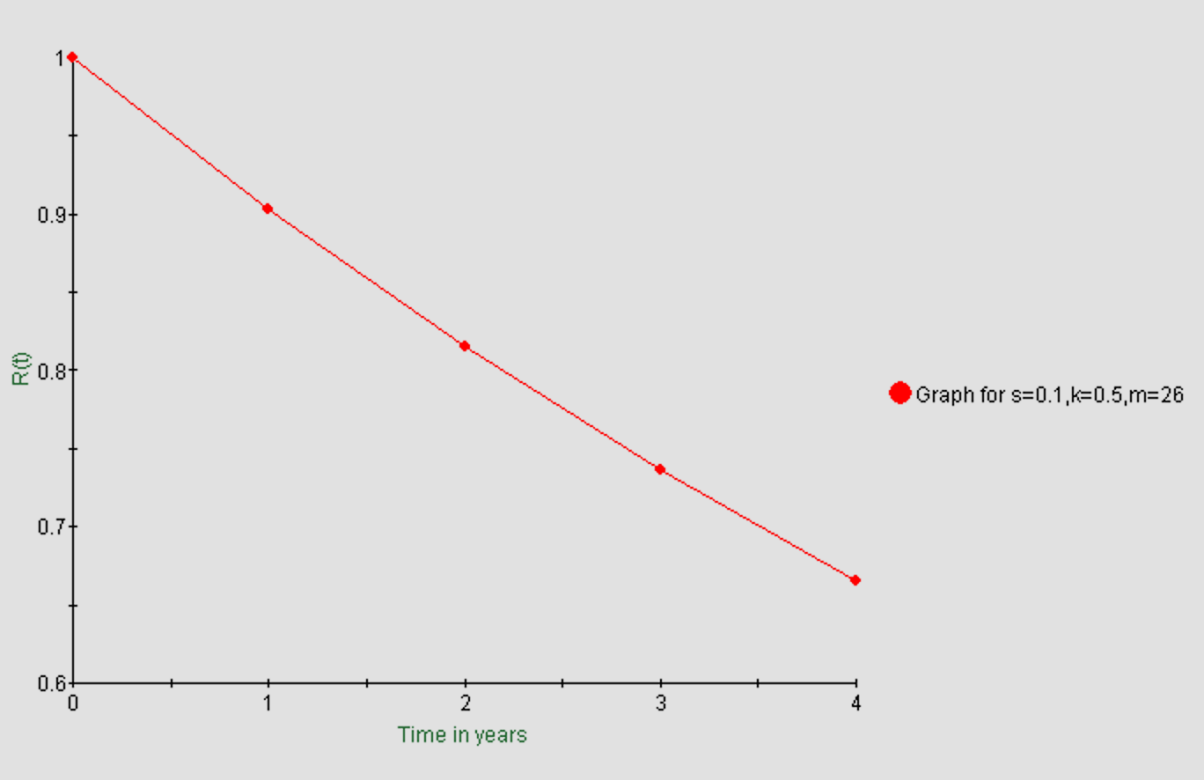

Figure 4.3. 7 Reliability Vs time for the model with immediate repair (s= 0.1/year, k = 0.5/year and m=26/year).

Table 4. 5 Reliability over time for the model with immediate repair (s= 0.1/year, k = 0.5/year and m=26/year).

Time in years	R(t)
t=0	1.00000000
t=1	0.903211536
t=2	0.815626729
t=3	0.736535048
t=4	0.665112972

4.3.2.2 WITHOUT IMMEDIATE REPAIR

This subsection analyzes the performance of the system with repair but only after the third controller failure. Figure 4.3.8 shows the Markov model of the system.

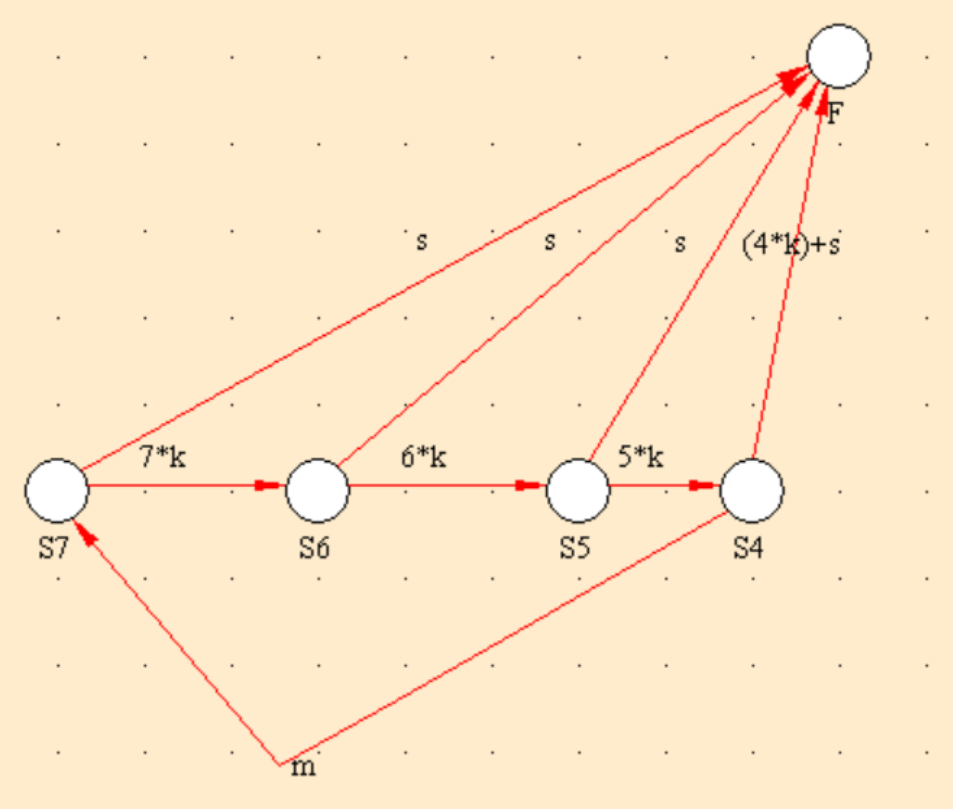

Figure 4.3. 8 Markov model for the system without immediate repair

The Markov model shown in figure 4.3.8 has been simulated by SHARPE in order to solve the below mathematical equations:

Let P(t)= [P_{S7} (t) P_{S6} (t) P_{S5}(t) P_{S4}(t) P_F(t)], where P(0)= [1 0 0 0 0] and

dP/dt= P x T, where T is the differential state transition rate matrix:

$$T = \begin{bmatrix} -(7k+s) & 7k & 0 & 0 & s \\ 0 & -(s+6k) & 6k & 0 & s \\ 0 & 0 & -(s+5k) & 5k & s \\ m & 0 & 0 & -(s+m+4k) & 4k+s \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

By using Laplace transform, P(t) can be obtained and then R(t) can be calculated as below:

$$R(t)= 1- P_F(t)$$

For fair comparison, the same test cases used in calculating system reliability of the previous model are also used for this model. Moreover, in all test cases, the supervisor failure rate s is calculated based on an MTTF of 10 years so s equals 0.1/year.

Case 1:

In this test case, the failure rate of the controller k is calculated based on MTTF equals 1 year. Accordingly, k equals 1/year. While the repair rate m is evaluated based on MTTR equals 1 day so m equals 365/year. Figure 4.3.9 shows the graph which represents the Reliability Vs time; where s= 0.1 and k = 1 and m=365 and the exact values are shown in table 4.6.

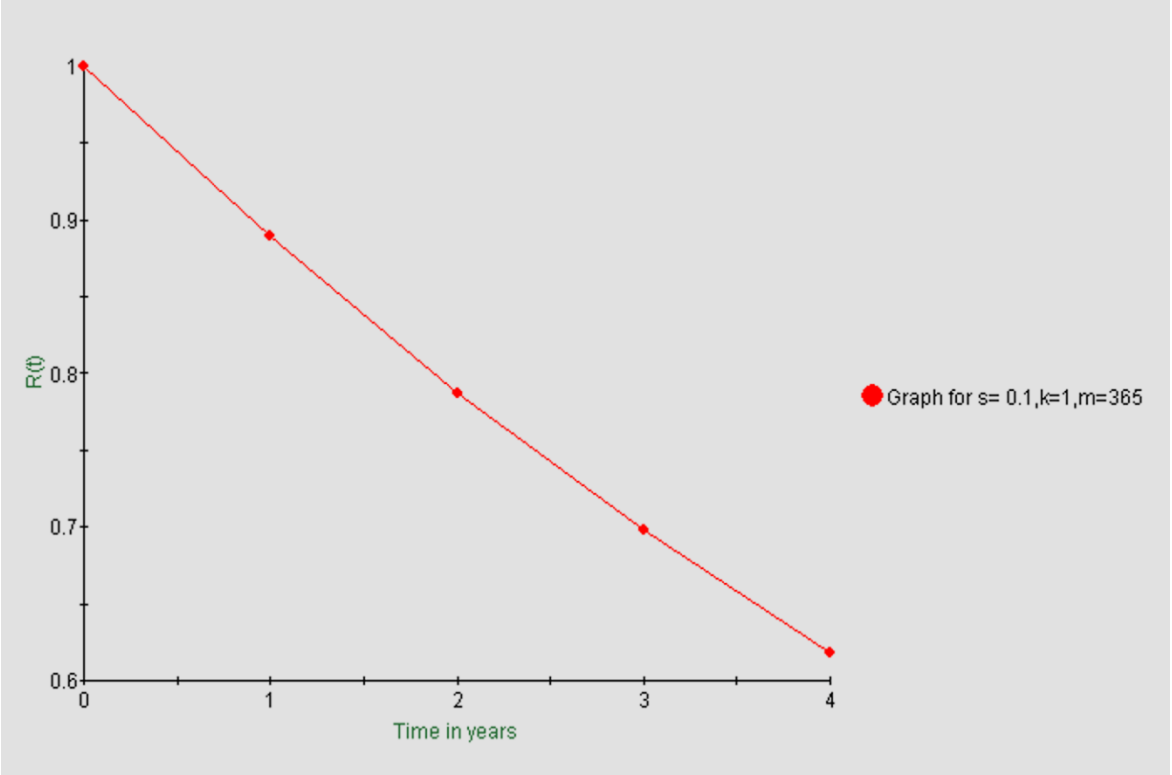

Figure 4.3. 9 Reliability Vs time for the model without immediate repair (s= 0.1, k = 1 and m=365).

Table 4. 6 Reliability over time for the model without immediate repair (s= 0.1, k = 1 and m=365).

Time in years	R(t)
t=0	1.00000000
t=1	0.889036643
t=2	0.787528468
t=3	0.697610402
t=4	0.617958987

Case 2:

In this test case, the failure rate of the controller k is calculated based on MTTF equals 2 years so k equals 0.5/year. While the repair rate m is evaluated also based on MTTR equals 1 day therefore m equals 365/year. Figure 4.3.10 shows the graph which represents the Reliability Vs time; where s= 0.1/year and k = 0.5/year and m=365/year and the exact values are shown in table 4.7.

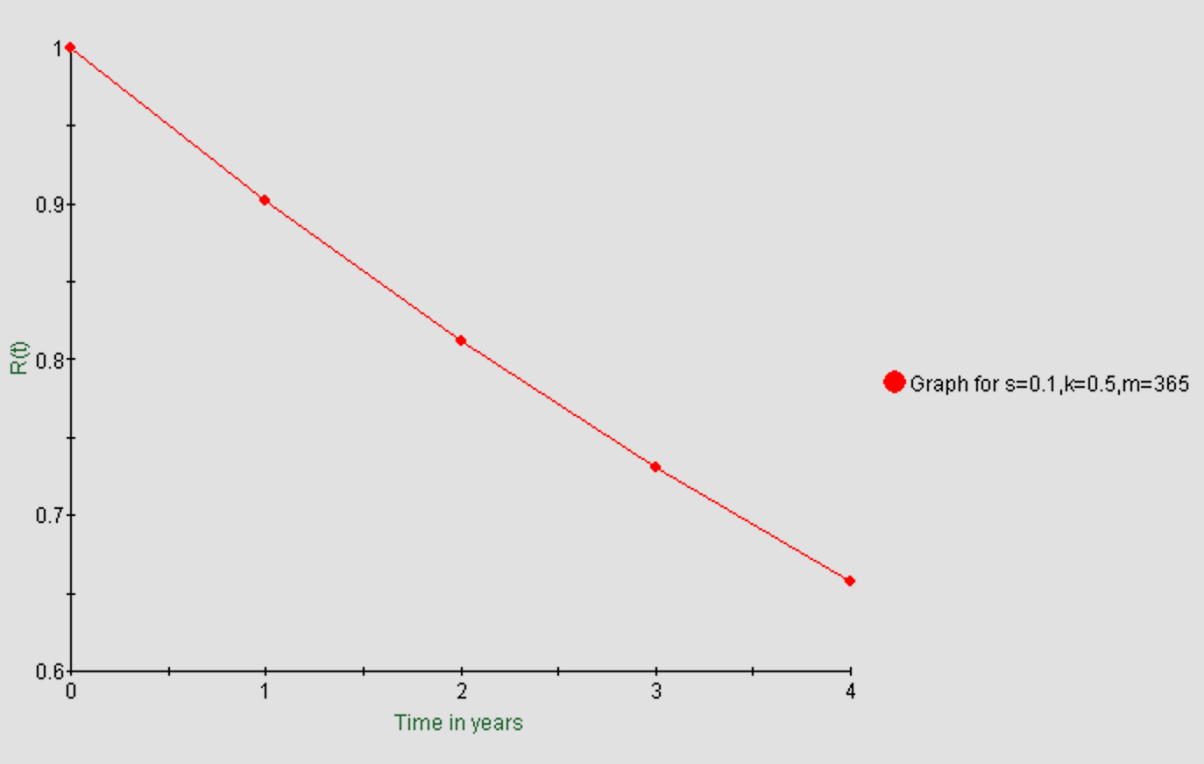

Figure 4.3. 10 Reliability Vs time for the model without immediate repair s= 0.1/year and k = 0.5/year and m=365/year.

Table 4. 7 Reliability Vs time for the model without immediate repair s= 0.1/year and k = 0.5/year and m=365/year.

Time in years	R(t)
t=0	1.00000000
t=1	0.904836528
t=2	0.818729223
t=3	0.740816199
t=4	0.670317684

Case 3:

In this test case, the failure rate of the controller k equals 0.5/year. While the repair rate m is evaluated based on MTTR equals 1 week therefore m equals 52/year. Figure 4.3.11 shows the graph which represents the Reliability Vs time; where s= 0.1/year and k = 0.5/year and m=52/year and the exact values are shown in table 4.8.

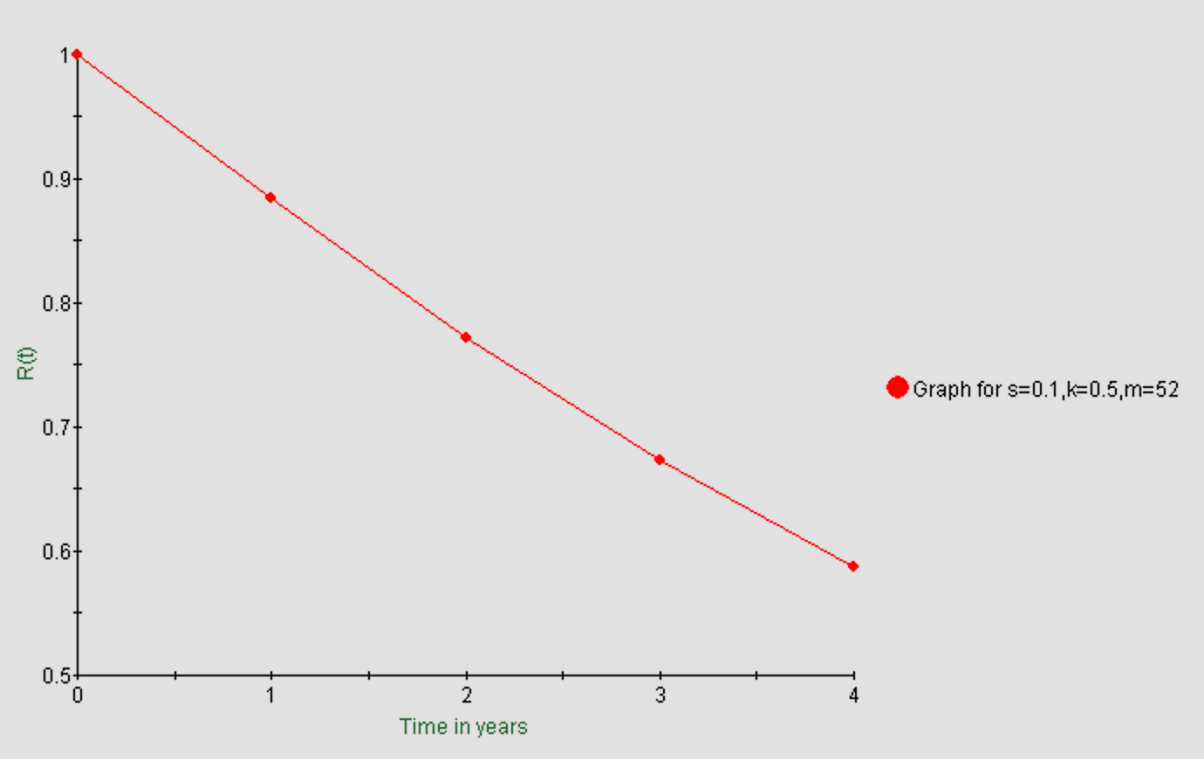

Figure 4.3. 11 Reliability Vs time for the model without immediate repair (s= 0.1/year, k = 0.5/year and m=52/year).

Table 4. 8 Reliability over time for the model without immediate repair (s= 0.1/year, k = 0.5/year and m=52/year).

Time in years	R(t)
t=0	1.00000000
t=1	0.883965516
t=2	0.771384229
t=3	0.673196292
t=4	0.587506567

Case 4:

In this test case, the failure rate of the controller k equals 0.5/year. While the repair rate m is evaluated based on MTTR equals 2 weeks therefore m equals 26/year. Figure 4.3.12 shows the graph which represents the Reliability Vs time; where s= 0.1/year and k = 0.5/year and m=26/year and the exact values are shown in table 4.9. Comparing all cases, case 2 has shown the best results.

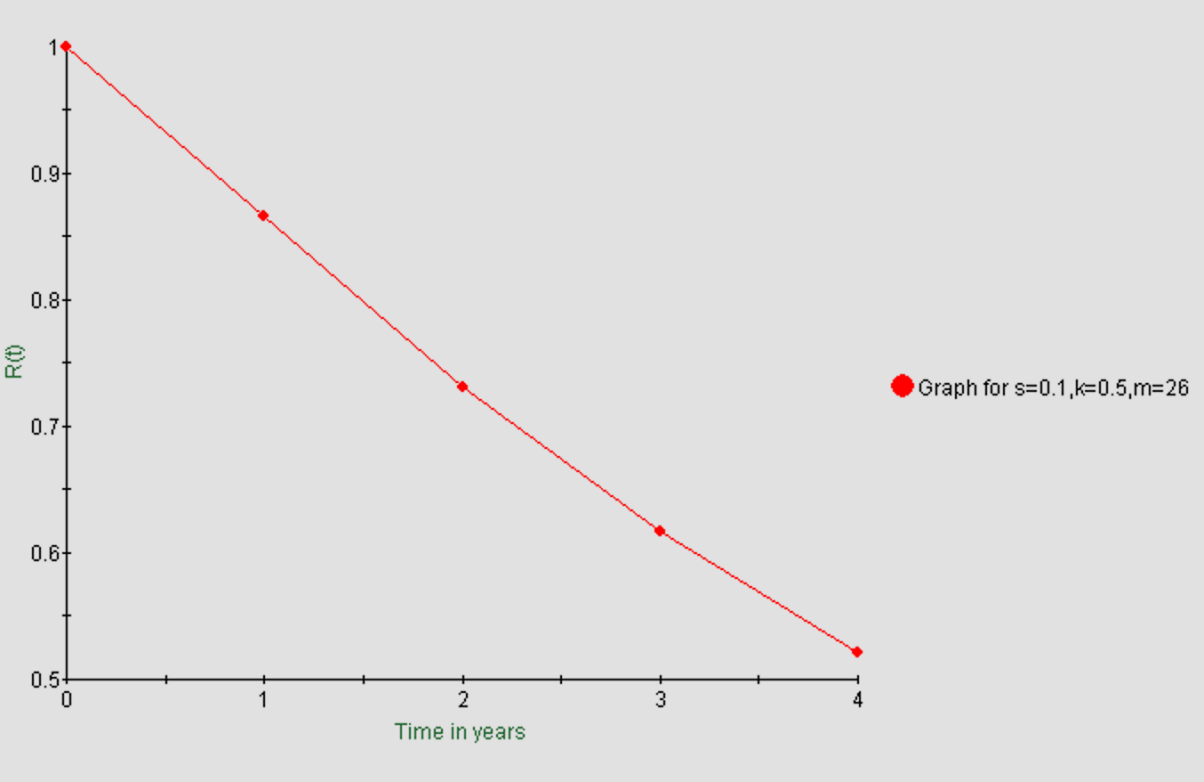

Figure 4.3. 12 Reliability Vs time for the model without immediate repair (s= 0.1/year, k = 0.5/year and m=26/year).

Table 4. 9 Reliability over time for the model without immediate repair (s= 0.1/year, k = 0.5/year and m=26/year).

Time in years	R(t)
t=0	1.00000000
t=1	0.866063970
t=2	0.730939271
t=3	0.617014761
t=4	0.520846447

To conclude, from the test cases of both the Markov models with immediate repair and without immediate repair, it is clear that the reliability values are very similar, therefore, it is better to follow the second scenario which is repairing the controllers only after the failure of controller 3 as the immediate repair of each controller upon its failure is considered an overhead specially that the is no significant difference in the results between both scenarios.

4.3.3 Mechanical reliability

This Section deals with the mechanical reliability of the Johns Hopkins modular prosthetic limb. Remember that, in [Medhat 2020], the architecture had seven motors; three motors in the shoulder, another three motors in the wrist and a motor in the elbow. A failure in any one or more of these motors will severely impede the operation of the arm.

Therefore, it is very important to increase the lifetime of the arm with respect to the mechanical parts, namely the motors. One of the most famous fault-tolerant techniques is the M-of-N architecture. M-of-N systems are example of parallel models that require M out of N modules for the system to work properly. For the triple modular redundancy (TMR), two modules out of the three must be working so as the system keeps functioning. Thus, for module reliability (Rm), the below equation enumerates all the working states [Siewiorek 1998].

$$R_{TMR} = R_m^3 + \binom{3}{2} R_m^2 (1-R_m)$$

The state in which the 3 modules are functioning is represented by R_m^3. The $\binom{3}{2} R_m^2 (1-R_m)$ term denotes the other three states where two modules are alive and the third one is down. It is assumed that all modules are identical [Siewiorek 1998].

A very simple M-of-N technique is the 1-out-of-2 architecture. Instead of using just one motor for every degree of freedom, two identical motors are sued on the same shaft. In the fault-free situation, both motors are operating in parallel. In case one of the motors fails, it will not stop the operation of the shaft and the other motor will handle the motion on its own. An obvious advantage of this technique is the expected increase in lifetime. Had one motor been responsible for the operation, a failure in this motor would totally eliminate one degree of freedom and the arm would have had to undergo repair immediately. With the 1-out-of-2 system suggested, the arm will still operate until the failed motor is repaired. Especially in developing countries, the issue of maintenance is problematic both from a logistical point of view and obviously from a cost point of view. A failed arm may take a long time to repair because of the scarcity of appropriate spare parts [Soliman 2019]. However, if the arm can still function adequately until the spare part becomes available, this would be a great advantage for the user who will only completely lose the use of the arm if the second remaining motor fails before the first one is repaired.

The reliability of this 1-out-of-2 motor architecture is expected to be higher than that of a single motor architecture (single point of failure). To estimate the advantage of this fault-tolerant architecture, the reliability will be obtained. Remember that the reliability is defined as the probability of the architecture being operational at time t given that it was operational at t=0 [siewiorek 1998, Trivedi 2017]. Next Stochastic Petri Nets will be used to calculate the reliability of the 1-out-of-2 motor system.

Petri nets, as shown in figure 4.3.13, contain places (illustrated by circles), transitions (illustrated by boxes or bars), and directed arcs connecting between them. Places may hold tokens, and a certain assignment of tokens to the places of a model corresponds to its model state (called marking in Petri net terms). Transitions represents activities such as the state changes and events. Just like in other discrete event system descriptions, events may be possible in a state, the transition is said to been abled in the marking. If so, the transition fires and change the system state. Transition enabling and firing as well as the consequential marking change are defined by the enabling rule and firing rule of the actual Petri net class [Trivedi 2017].

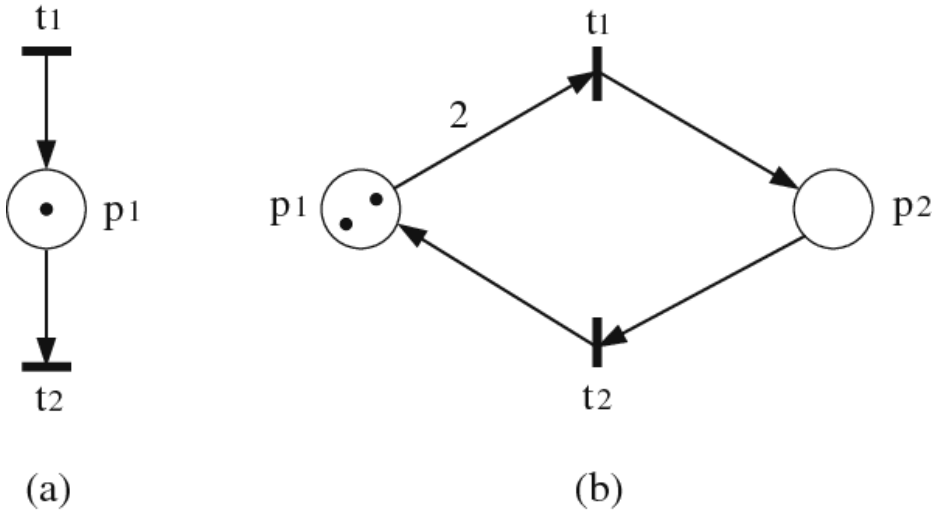

Figure 4.3. 13 Example of petri nets [Trivedi 2017].

Figure 4.3.14 shows the SPN for the 1-of-2 motor architecture under study. Let Lm be the failure rate of the motor and further assume that both motors are identical both from a functional and a reliability point of view. Also, let m be the repair rate of any of the two

motors. In the figure, both states p1 and p2 represents the 2 working motors, if Motor 1 fails, the system will transition to state M2 where only Motor 2 is working. Let Lm1=Lm2= 2/ year and m1=m2= 52/year

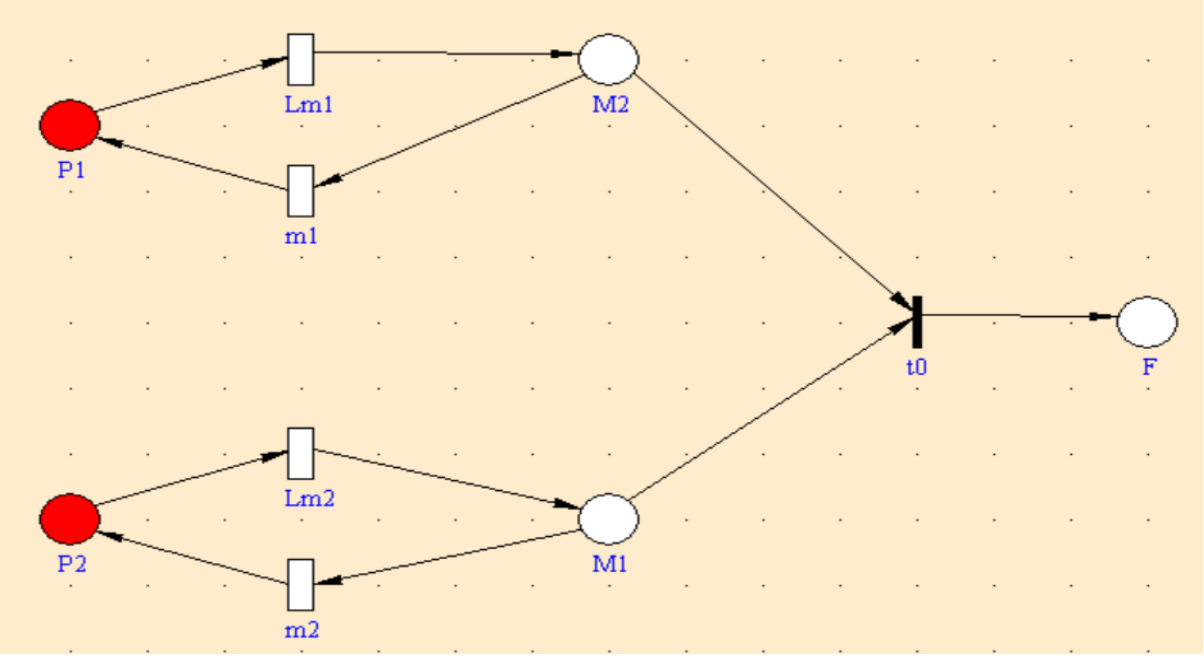

Figure 4.3. 14 Petri Net for the system with mechanical repair

Figure 4.3.15, shows the mechanical reliability curve. After 4 years, the reliability is 57.65%.

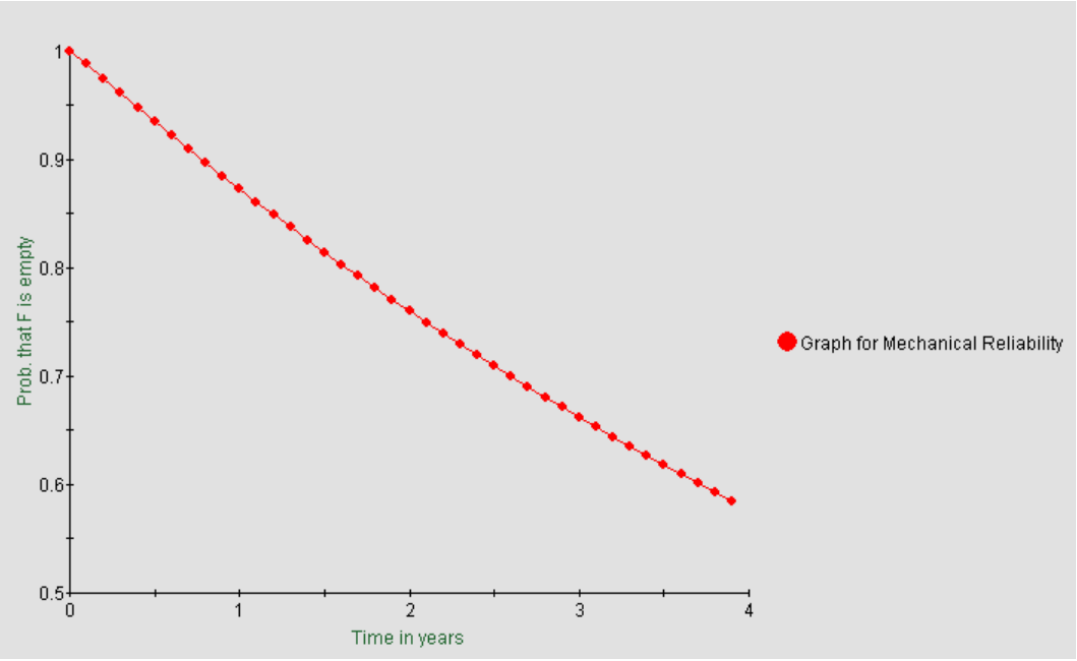

Figure 4.3. 15 Reliability vs time for the model with mechanical repair

Note that the reliability of a conventional one-motor architecture (simplex), and the reliability of 1 set from the mechanical point of view can be calculated using the below equation [Siewiorek 1998].

$$R_{simplex} = e^{-Lm \times t}$$

Table 4.10 shows the difference between the reliabilities of the model with and without the mechanical reliability throughout the 4 years. It is clear that, without the motors redundancy, the reliability dropped to zero after 2 years. Finally, the table shows the mechanical reliability for one set only; in order to calculate the mechanical reliability of all the 7 sets the following equation should be used: $R_{all\ sets} = R_{one\ set}^{7}$

Table 4. 10 Reliability over time for the model with and without the mechanical repair

Year	**Without mechanical reliability**	**With mechanical reliability**
t=0	1.000	1.000
t=1	0.1353	0.8729
t=2	0.0183	0.7602
t=3	0.0024	0.6620
t=4	0.0003	0.5765

Conclusions and Recommendations for Future Work

To conclude, the idea of designing the prosthetic arms with Wireless Body Area Networks can lead to remarkable enhancement in the arms' functionality. Using Wireless communication to exchange packets between the arm's nodes will prevent the wear and tear issue and the mobility limitation issue. This thesis presents a study for a wireless scheme for the Johns Hopkins modular prosthetic limb (MPL) as well as a demonstration for a novel fault tolerant scheme to further improve the arm's reliability.

A former research has proved that it is not a promising idea to use Zigbee for the wireless communication between the sensors, actuators and controller of the arm. The reason is that the 16 non-interfering channels of the Zigbee have been utilized. This will not allow adding more nodes and hence will inhibit the model extension. Therefore, low power WiFi has been used in this research so as to offer higher bandwidth. The results are quite satisfying, even though only one channel was utilized. The deadlines used in this research as the benchmark are provided by the Johns Hopkins design. In addition, it was proved that the system has zero packet loss. Regarding the fault free model, the presented scheme is S2A (sensor to actuator). While the faulty model, the proposed scheme is transformed to S2K (sensor to controller) since it is assumed that the on-board controller (part of the actuator node) is faulty.

Moreover, interference analysis was done to all the scenarios by adding interference packets of size 1Gb on the same channel used for the sensors, actuators and supervisor. Even

when subjected to interference, the model is proved to be quite robust as it has met the system's requirements. Also, the results are based on a 95% confidence analysis.

The reliability of the system was calculated from the electrical and mechanical point of view. It was proved that a system that uses a supervisor with lower specifications will have a very close reliability values to the system that uses very powerful supervisor if it was repaired after the failure of the third controller. Finally, it was proved that the motor redundancy has remarkably increased the system reliability.

For future work, it is planned to study the performance of the full hand architecture including the palm of the hand not just the arm. Also, mechanical reliability will be studied in more depth rather than the redundancy of all motors. Finally, it is suggested to implement a simple prototype to study the concept of the wireless prosthetic arm from a practical point of view.

References

[Boggia 2009] G. Boggia, P. Camarda, V. Divittorio, and L. A. Grieco, “A simulation-based performance evaluation of wireless networked control systems,” IEEE Conference on Emerging Technologies & Factory Automation, Mallorca, Spain, September 2009.

[Burck 2011] J.M. Burck, J.D. Bigelow and S.D. Harshbarger, "Revolutionizing Prosthetics: Systems Engineering Challenges and Opportunities," Johns Hopkins APL Technical Digest, vol. 30, no. 3, pp. 186-197, 2011.

[Crosby 2012] G.V. Crosby, T. Ghosh, R. Murimi and C.A. Chin, “Wireless Body Area Networks for Healthcare: A Survey,” International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC), vol. 3, no. 3, June 2012.

[Dadarlat 2014] M. Dadarlat, J. O'Doherty and P. Sabes, "A learning-based approach to artificial sensory feedback leads to optimal integration", Nature Neuroscience, vol. 18, no. 1, pp. 138-144, 2014. Available: 10.1038/nn.3883.

[Daoud 2007] R.M.Daoud, et al.,"Burst Communication for Traffic Control in Light Urban Areas", International Workshop on Logistics and Transportation, 2007.

[Dekel 2019] E. Dekel, “Low-power Internetconnectivity over Wi-Fi,” Texas Instruments,2019.Available:http://www.ti.com/lit/wp/swry019a/swry019a.pdf

[Faramawy 2012] Y.I. El-Faramawy, M.A. Ibrahim, H.H. Halawa, A. Elhamy, E.E. Abdel Reheem, T.K. Refaat, R.M. Daoud and H.H. Amer, “Multicasting for Cascaded Fault-Tolerant Wireless Networked Control Systems in Noisy Industrial Environments”, Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation ETFA, Krakow, Poland, September 2012.

[Heger 1985] H. Heger, S. Millstein and G.A. Hunter, “Electrically Powered Prostheses for the Adult with an Upper Limb Amputation,” Journal of Bone & Joint Surgery, vol. 67, no. 2, pp. 278-281, 1985.

[Hirata 2011] M. Hirata, et al., “A Fully-Implantable Wireless System for Human Brain-Machine Interfaces Using Brain Surface Electrodes: W-HERBS,” IEICE Transactions on Communications, vol. E94-B, no. 9, pp. 2448-2453, September 2011.

[jhuapl2019]“Prosthetics”JHU/APLBrand.[Online].Available:https://www.jhuapl.edu/Prosthetics/ResearchMPL. [Accessed: 01-Jan-2019].

[Johannes 2011] M.S. Johannes, J.D. Bigelow, J.M. Burck, S.D. Hashbarger, M.V. Kozelowski and T. Van Doren, "An Overview of the Development Process for the Modular Prosthetic Limb," Johns Hopkins APL Technical Digest, vol. 10, no. 3, pp. 207-216, 2011.

[Kaur 2011] R. Kaur, “Wireless Body Area Network & UTS APPLICATION”, International Journal of Engineering Sciences (ISSN), vol. 1, July 2011, pp. 199-216.

[Koren 2007] I. Koren and C. Mani Krishna, “Fault-Tolerant Systems,” Elsevier, 2007.

[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. Available: http://www.eecs.umich.edu/~impact.

[Martin 2011] C. Martin and D. Edeer, “Upper Limb Prostheses: A Review of the Literature with a Focus on Myoelectric Hands,” WorkSafeBC Evidence Based Practice Group, 2011.

[McGee 2014] T.G. McGee, M.P. Para, K.D. Katyal and M.S. Johannes, “Demonstration of force feedback control on the Modular Prosthetic Limb,” Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics SMC, San Diego, CA, USA, October 2014.

[Medhat 2020] M.M.Medhat, R. M.Daoud and H.H.Amer, "A Novel Fault Tolerant Scheme for a Wireless Modular Prosthetic Limb", IEEE International Workshop on Factory Communication Systems (WFCS), 2020.

[Otto 2005] C. Otto, A. Milenkovic, C. Sanders and E. Jovanov, “System Architecture of a Wireless Body Area Sensor Network for Ubiquitous Health Monitoring,” Journal of Mobile Multimedia, vol. 1, no. 4, January 2005, pp. 307-326.

[Perry 2018] N. Perry, C.W. Moran, R.S. Armiger, P.F. Pasquina, J.W. Vandersea, and J.W. Tsao, “Initial clinical evaluation of the Modular Prosthetic Limb,” Frontiers in Neurology, vol. 9, 2018.

[Ravitz 2013] A. Ravitz, et al., “Revolutionizing prosthetics—phase 3,” Johns Hopkins Apl Technical Digest, 31(4), pp.366-376, 2013.

[Saleh 2015] A. Saleh, et al., “A Wireless Body Area Network architecture for a Prosthetic arm”, Proceedings of the 7th International Congress on Ultra Modern Telecommunications and Control Systems ICUMT, Brno, Czech Republic, October 2015.

[SHARPE 2020] Official site for SHARPE: http://sharpe.pratt.duke.edu/[Accessed 28 Jan 2020]

[Siewiorek 1998] D.P. Siewiorek and R.S. Swarz, "Reliable Computer Systems Design and Evaluation," A K Peters, Natick, Massachusetts, 1998.

[Skeie 2002] T. Skeie, S. Johannessen, and C. Brunner, “Ethernet in Substation Automation,” IEEE Control Syst. vol. 22, no. 3, June 2002, pp. 43-51.

[Smith 2005] D.J. Smith, “Reliability, Maintainability and Risk - Practical methods for engineers,” Springer, 2005.

[Soliman 2019] Khalid A. Soliman, Ramez M. Daoud, Hassanein H. Amer and Dina F. Rateb, “Steady-State Availability and Inventory in Fault-Tolerant NCS for Pharmaceutical Process”, Proceedings of the International Conference on Emerging Technologies and Factory Automation ETFA, Zaragoza, Spain, September 2019

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

[Toledo 2009] C. Toledo, L. Leija, R. Munoz, A. Vera and A. Ramirez, “Upper Limb Prostheses for Amputations Above Elbow: A Review,” Pan American Health Care Exchanges (PAHCE), Mexico City-Mexico, March 2009.

[Trivedi 2017] K. Trivedi and A. Bobbio, "Reliability and Availability Engineering - Modeling, Analysis, and Applications," Cambridge University Press, 2017.

[Zhang 2011] X.D. Zhang, Y.X. Wang, Y.N. Li, J.J. Zhang and X.D. Zhang, “An Approach for Pattern Recognition of EEG Applied in Prosthetic Hand Drive,” Journal of Systemics, Cybernetics & Informatics, vol. 9, no. 6, 2011.

[Zimmermann 2010] A.Zimmermann, Stochastic Discrete Event Systems. Berlin: Springer Berlin, 2010.