The American University in Cairo
School of Sciences and Engineering

FAULT TOLERANT ETHERNET TRAIN NETWORK
A Thesis Submitted to
The Electronics Engineering Department

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

by Mai Mahmoud Hassan Salem Ibrahim

under the supervision of Prof. Hassanein H. Amer and Dr. Ramez M. Daoud
May/2013
Approval Sheet Goes Here
DEDICATION

I would dedicate this thesis to my dear Mother. She is the one person who always believed in me even when I did not. I would also like to dedicate it to my sister, grandfather and to all my professors.

Mom: without you I would have achieved nothing in my life. Any minor success I achieve is because of your faith in me, hard work and your sleepless nights. I know I have not been the best daughter but you have been more than the greatest mother created on earth. Just being your daughter has made me fight to try and be worthy of that position. No words are enough to express how grateful I am to you or to illustrate your scarifies that did for me. If I keep writing books could be made and still they are not enough. I pray to Allah that I can always make you happy and proud.

Sister: you are the source of joy in my life. I cannot imagine my life without your lovely smile. I know you look up to me. I just hope that I would never disappoint you. I promise you to always have your back.

Grandpa: You have always been my source of safety. Your success story has been my inspiration and it will always be. When I was young I wanted to be like you. I know I’m not like you but all I hope for is to make you proud enough to call me a granddaughter. Even when you were away, I felt you were there pushing me to be better and pleasing you has been and will always be my target in life. You are my father, grandfather and role model.

I would like to especially dedicate this thesis to Prof. Hassanein Amer and Dr. Ramez M. Daoud. Your have been my beacon for almost 10 years. Everything I learned from you has made me a better engineer and a better person. You are my role models in every aspect of my career life. I would not have been an engineer in the first place without your support and believe in me. Dr. Amer, the nickname you gave me is and will always be the main source of strength and dedication. It made me determined during hard times to be worthy of it. I cannot be grateful enough to have both of you as my mentors and professors.

Moreover, I would like to dedicate this thesis to Dr. Sherif Abdel Azeem. You are my god father. If there is any good in me it’s all because of you. I’m just proud to be one of your students. One day I wish I would be merely as good of a person as you.

Last but not least, I would like to thank all the professors who taught me at AUC, namely, Dr. Ayman El Ezabei, Dr. Ahmed Abu Auf, Dr. Ali Darwish and Dr. Karim Seddik. Without your effort and dedication I would not have been and engineer. Thank you for every second of effort spent to teach me. I’ll always be in debt to you all.

May Allah grant you all happiness in this life.
ACKNOWLEDGMENTS

I would like to acknowledge both my supervisors: Prof. H.H. Amer and Prof. R.M. Daoud for their endless support throughout this thesis.

I would also like to acknowledge Prof. H. ElSayed, Dr. M. Dakroury, Eng. Tarek Refaat, and Eng. M. Mostafa for their generous assistance throughout my work.

I would finally like to acknowledge my examiners: Prof. Magdy El Soudani and Prof. Sherif Abdel Azeem and the graduate program directors: Dr. Ayman El Ezabei and Dr Karim Seddik.
TABLE OF CONTENTS

ABSTRACT .......................................................................................................................... 1

I. INTRODUCTION ............................................................................................................ 2

II. LITERATURE REVIEW ................................................................................................ 5

II.1. CONTROL PROTOCOLS IN TRAINS ................................................................. 5

II.1.1. Control Area Network (CAN) .......................................................................... 5

II.1.2. Train Communication Network (TCN) ............................................................ 5

II.1.3. LonWorks Protocol ......................................................................................... 6

II.2. ETHERNET IN NCS .............................................................................................. 8

II.2.1. Ethernet in Industrial Control Systems .......................................................... 8

II.2.2. Ethernet in Automotive On-board Networks ................................................ 9

II.2.3. Ethernet Train Control Network ...................................................................... 9

II.2.4. Fault tolerant Train Network at the Controller Level ................................... 11

III. TRAIN NETWORK RELIABILITY ENHANCEMENT ........................................ 12

III.1. ONE WAGON MODEL WITH TMR AT THE SENSOR LEVEL ..................... 12

III.1.1. Simulated Scenarios and Outcomes .............................................................. 15

III.2. TWO WAGONS NETWORK MODEL ............................................................... 21

III.2.1. Two Wagons Simulated Scenarios ............................................................... 22

III.3. RELIABILITY ANALYSIS OF THE TRAIN NETWORK MODEL .................. 25

III.3.1. Reliability Model of the Fault-Tolerant Controller ....................................... 25

III.3.2. Efficiency of Error Detection/Reconfiguration Mechanisms ...................... 31

IV. HYBRID NETWORK MODEL AND CONTROLLER HIERARCHICAL
ARCHITECTURE ............................................................................................................. 36

IV.1. HYBRID CONTROL NETWORK MODEL .......................................................... 36

IV.1.1. Simulation Outcomes ..................................................................................... 38

IV.2. HIERARCHICAL ARCHITECTURE AT THE CONTROLLER LEVEL .............. 40

IV.2.1. Unique Simulated Scenarios in Passive Supervisor ...................................... 43

V. ACTIVE SUPERVISOR AND RELIABILITY MODELING ............................... 50

V.1. HYBRID NETWORK MODEL WITH ACTIVE SUPERVISOR .......................... 50

V.1.1. Unique Simulated Scenarios in Active Supervisor ........................................... 52
V.1.2. Passive and Active Supervisor Outcomes Comparison.............................60
V.2. MARKOV MODELING AND RELIABILITY ANALYSIS COMPARISON..............62
   V.2.1. Network Reliability Comparison ..........................................................69
VI. CONCLUSIONS ...........................................................................................76
APPENDIX – CONFIDENCE ANALYSIS..............................................................78
REFERENCES ..................................................................................................80
LIST OF FIGURES

FIGURE 1: OVERVIEW OF THE TCN STRUCTURE [16] .................................................. 6
FIGURE 2: A SINGLE NETWORK MANAGER [19] .......................................................... 7
FIGURE 3: MULTIPLE NETWORK MANAGERS [19] .......................................................... 7
FIGURE 4: ONE WAGON TRAIN NETWORK [40] .............................................................. 10
FIGURE 5: ENHANCED ONE WAGON TRAIN NETWORK [43] .......................................... 11
FIGURE 6: TMR ONE WAGON TRAIN NETWORK MODEL ........................................... 13
FIGURE 7: SCENARIO ONE (B), CONTROL PACKET END-TO-END DELAY ..................... 16
FIGURE 8: SCENARIO TWO (A), CONTROL PACKET END-TO-END DELAY ..................... 17
FIGURE 9: SCENARIO THREE (A), CONTROL PACKET END-TO-END DELAY ................... 18
FIGURE 10: SCENARIO FOUR (B), CONTROL PACKET END-TO-END DELAY .................. 19
FIGURE 11: TWO WAGONS NETWORK MODEL ......................................................... 22
FIGURE 12: SCENARIO ONE (B), TWO WAGONS END-TO-END DELAY ........................ 23
FIGURE 13: SCENARIO TWO (B), TWO WAGONS END-TO-END DELAY ........................ 24
FIGURE 14: PARALLEL SYSTEM RELIABILITY BLOCK DIAGRAM ................................. 25
FIGURE 15: SERIES SYSTEM RELIABILITY BLOCK DIAGRAM ..................................... 26
FIGURE 16: RELIABILITY COMPARISON ....................................................................... 26
FIGURE 17: CONSERVATIVE MARKOV MODEL ............................................................. 27
FIGURE 18: EFFECT OF COVERAGE IN CONSERVATIVE MODEL ................................. 29
FIGURE 19: REALISTIC MARKOV MODEL ...................................................................... 33
FIGURE 20: EFFECT OF COVERAGE IN REALISTIC MODEL .......................................... 35
FIGURE 21: HYBRID ONE WAGON TRAIN NETWORK MODEL ....................................... 37
FIGURE 22: HYBRID MODEL - ONE WAGON - FF - K TO A DELAY (G1) ......................... 39
FIGURE 23: HYBRID MODEL - ONE WAGON – FF - K TO A DELAY (G3) ....................... 39
FIGURE 24: TWO-WAGON UNIT NETWORK MODEL WITH A SUPERVISOR NODE ........ 40
FIGURE 25: NETWORK FAILURE PATHS IN PASSIVE .................................................... 43
FIGURE 26: NETWORK FAILURE PATHS IN PASSIVE SUPERVISOR (CONT’D) ............... 44
FIGURE 27: PASSIVE - KIKeIjEjS CONTROL END-TO-END DELAY – G3 ...................... 47
FIGURE 28: PASSIVE - KIKeIjS CONTROL END-TO-END DELAY - G2 ....................... 48
FIGURE 29: PASSIVE - END-TO-END DELAY WITH S OPERATIONAL – G1 ................... 48
FIGURE 30: TWO WAGON TRAIN MODEL WITH ACTIVE SUPERVISOR ...................... 51
FIGURE 31: NETWORK FAILURE PATHS IN ACTIVE SUPERVISOR ............................... 52
FIGURE 32: NETWORK FAILURE PATHS IN ACTIVE SUPERVISOR (CONT’D) ...............53
FIGURE 33: ACTIVE - KiKfEiEjS CONTROL END-TO-END DELAY – G2 ..............56
FIGURE 34: ACTIVE - KiKjS CONTROL END-TO-END – G3 ................................57
FIGURE 35: ACTIVE - END-TO-END DELAY WITH ONLY S OPERATIONAL (S) – G2 ....57
FIGURE 36: PASSIVE SUPERVISOR - MARKOV MODEL ...................................62
FIGURE 37: ACTIVE SUPERVISOR - MARKOV MODEL .......................................66
FIGURE 38: RELIABILITY - ACTIVE VS. PASSIVE - C=0.9 - \( \lambda_s=0.5 \) ........71
FIGURE 39: RELIABILITY - ACTIVE VS. PASSIVE - C=0.95 - \( \lambda_s=0.5 \) ........71
FIGURE 40: RELIABILITY - ACTIVE VS. PASSIVE - C=0.9 - \( \lambda_s=0.8 \) ........72
FIGURE 41: RELIABILITY - ACTIVE VS. PASSIVE - C=0.95 - \( \lambda_s=0.8 \) ........72
FIGURE 42: RELIABILITY - ACTIVE VS. PASSIVE - C=0.9 - \( \lambda_s=1 \) ...............73
FIGURE 43: RELIABILITY - ACTIVE VS. PASSIVE - C=0.95 - \( \lambda_s=1 \) ...............73
LIST OF TABLES

TABLE 1: SUMMARY OF TMR SIMULATION OUTCOMES .......................................................... 20
TABLE 2: HYBRID MODEL - ONE WAGON - TOTAL END-TO-END DELAYS (MS) .............. 38
TABLE 3: PASSIVE SUPERVISOR TOTAL END-TO-END DELAYS (MS) ............................. 49
TABLE 4: ACTIVE SUPERVISOR TOTAL END-TO-END DELAYS (MS) .............................. 58
TABLE 5: NUMBER OF STATES WITH ENABLED ENTERTAINMENT .................................. 61
TABLE 6: RELIABILITY (T) FOR PASSIVE SUPERVISOR .................................................. 69
TABLE 7: RELIABILITY (T) FOR ACTIVE SUPERVISOR .................................................. 70
TABLE 8: TRAFFIC HANDLED BY SUPERVISOR IN EACH STATE ................................. 75
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>Anti-lock braking system</td>
</tr>
<tr>
<td>ACT</td>
<td>Actuator</td>
</tr>
<tr>
<td>AP</td>
<td>Access Point</td>
</tr>
<tr>
<td>CAN</td>
<td>Controller Area Network</td>
</tr>
<tr>
<td>CIP</td>
<td>Common Industrial Protocol</td>
</tr>
<tr>
<td>CSMA/CD</td>
<td>Carrier Sense Multiple Access with Collision Detection</td>
</tr>
<tr>
<td>DB</td>
<td>Database</td>
</tr>
<tr>
<td>EPS</td>
<td>Electronic Stability Program</td>
</tr>
<tr>
<td>FF</td>
<td>Fault-Free</td>
</tr>
<tr>
<td>FT</td>
<td>Fault-Tolerance/Fault-Tolerant</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer protocol</td>
</tr>
<tr>
<td>G1</td>
<td>Group 1</td>
</tr>
<tr>
<td>G2</td>
<td>Group 2</td>
</tr>
<tr>
<td>G3</td>
<td>Group 3</td>
</tr>
<tr>
<td>Gb</td>
<td>Gigabit</td>
</tr>
<tr>
<td>GbE</td>
<td>Gigabit Ethernet</td>
</tr>
<tr>
<td>HTTP</td>
<td>Web browsing</td>
</tr>
<tr>
<td>IEC</td>
<td>International Electrotechnical Commission</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IS</td>
<td>Intermediate Switch</td>
</tr>
<tr>
<td>LONWORKS</td>
<td>Local Operation Network</td>
</tr>
<tr>
<td>MAVQ</td>
<td>Minimum acceptable video quality</td>
</tr>
<tr>
<td>MCD</td>
<td>Maximum acceptable delay</td>
</tr>
<tr>
<td>MS</td>
<td>Main Switch</td>
</tr>
<tr>
<td>MVB</td>
<td>Multi-function Vehicle Bus</td>
</tr>
<tr>
<td>NCS</td>
<td>Networked Control System</td>
</tr>
<tr>
<td>NTSC</td>
<td>National Television System Committee</td>
</tr>
<tr>
<td>SAs</td>
<td>Sensors &amp;/or Actuators</td>
</tr>
<tr>
<td>SEN</td>
<td>Sensor</td>
</tr>
<tr>
<td>TCN</td>
<td>Train Control Network/s</td>
</tr>
<tr>
<td>TMR</td>
<td>Triple Modular Redundancy</td>
</tr>
<tr>
<td>UIC</td>
<td>Union Internationale des Chemins de Fer</td>
</tr>
<tr>
<td>WTB</td>
<td>Wire Train Bus</td>
</tr>
</tbody>
</table>
LIST OF SYMBOLS

c   System Coverage
Ei   Entertainment Server in wagon i
Ej   Entertainment Server in wagon j
Ki   Controller (or Control Server) in wagon i
Kj   Controller (or Control Server) in wagon j
R_K  Reliability of the Controller
R_E  Reliability of the Entertainment Server
R_cont_FT  Reliability of 2 wagon system with fault tolerance
R_cont  Reliability of 2 wagon system without fault tolerance
S   Supervisor
Wi   Wagon i in the network
\lambda_K  Failure Rate of the Controller
\lambda_E  Failure Rate of the Entertainment Server
\lambda_S  Failure Rate of the Supervisor
ABSTRACT

The American University in Cairo, Egypt
Fault Tolerant Ethernet Train Network
Name: Mai Mahmoud Hassan Salem Ibrahim
Supervisors: Prof. Hassanein H. Amer and Dr. Ramez M. Daoud

Ethernet use for control networks is an emerging topic of study. Research discusses the implementation of train control networks using Ethernet. In this research, an Ethernet Train Control Network for one wagon with a mixed communication load environment (real-time and non-real-time traffic) was simulated. Both the control and entertainment loads were implemented on top of Gigabit (GbE) Ethernet, each with a dedicated controller/server.

Triple Modular Redundancy (TMR) implementation of the control network at the sensor level was tested. Results for the case of introducing on-board passengers’ entertainment (Wi-Fi and video streaming) were presented based on OPNET simulations. Additionally, the network model was further modified to have a control load with mixed sampling periods. It was shown that this system could tolerate the failure of one controller in one wagon.

In a two wagon scenario, fault tolerance (FT) at the controller level was studied, and simulation results showed that the system could tolerate the failure of 3 controllers. The hybrid model scenario was successful in meeting the packet end-to-end delay with zero packet loss in all OPNET simulated scenarios.

A supervisor was further interconnected to both wagons. Two fault tolerance analyses were presented by studying the reallocation of the control load according to the number of Servers failed. In the first analysis the supervisor was acting in a passive manner where it only acts when all the other Controllers/Entertainment Servers fail. The other analysis presented an active supervisor that acts once a Controller fail.

The main measuring metric used is the maximum real-time control packet end-to-end delay and ensure it meets its constraints. All simulations are conducted on OPNET Network Modeler and results are subjected to a 95% confidence analysis.
I. INTRODUCTION

Networks are categorized as either control networks or data communication networks. Control networks are used to transmit control packets which are small in size and must meet time-critical requirements. Since control packets are real time, the transmission of such packets has to be successfully assured within the required time period [1]. There are several control protocols used in train control networks such as Local Operating Networks (LonWorks), Train Communication Networks (TCN) and Controller Area Network (CAN) [2, 3]. The LonWorks protocol is common in Europe, while the TCN is used in the USA. CAN is also used but not widely as TCN or LonWorks [2, 3].

Since Ethernet appeared in the world of wired communication systems, the implementation of Ethernet as a communication medium for Networked Control System became a must. Ethernet is based on the CSMA/CD mechanism. Even though Ethernet is non-deterministic by nature, this did not stop researchers in academia and industry from using the Ether-Channel as a communication medium for Control Systems. With the introduction of switches, the non-deterministic nature of Ethernet is partially resolved. Now, different problems exist such as queuing delays and queue lengths. The non-deterministic nature of Ethernet was first thought to be problematic because of the real-time constraints inherent in control systems; however, research shows that Ethernet (or IEEE Std 802.3) performed well in Networked Control System either by changing packet format for real-time control messages, or by giving higher priority for these messages [4, 5, 6]. The standardization process for the use of Ethernet in control is also under way [7, 8]. Rockwell Automation and the ODVA organization proposed the Ethernet/IP as an industrial version of Ethernet and they have developed the Common Industrial Protocol (CIP) [9, 10]. Furthermore, TT Ethernet and FTT Ethernet are in the course of standardization [11, 12].
In railway systems, in order to assure safety and efficiency along the journey the train network has to ensure that four main aspects are controlled within a train; these aspects are [13]:

1. **Train protection**: it is the process of preventing any possible accidents, route interlocking or exceeding of the speed limits, e.g., route interlocking and over speed protection.

2. **Train operation**: it is the process of controlling the train motion during the journey and regulating the journey termination at the stations. Also, it includes controlling of the doors and air conditioning in the train, e.g., speed regulation and door control.

3. **Train supervision**: it is the process of guaranteeing that the train follows the schedule and the proper path while recording any malfunction, e.g., performance modification and malfunction recording

4. **Communication**: it is the process of enabling the train to communicate with other system elements and allowing exchange of information, e.g., passenger service and status of the system.

In addition to these main aspects, passenger demand to have more information and entertainment services on-board of trains has been increasing. As Ethernet networks have a relatively wide bandwidth in comparison to other control protocols used in trains, it allows providing some entertainment services beside the control services.

Chapter 2 summaries the literature review. First, CAN, TCN and LonWorks as control protocols in trains are presented. Additionally, the use of Ethernet (IEEE 802.3) without any modifications in the context of NCS is illustrated. Ethernet is used as a control protocol in industrial system, automotive on-board networks and in one-wagon train networks. This proved to be successful not only for pure control loads but also when mixing real-time and non-real-time entertainment messages. The introduction of fault-tolerance into these schemes is also shown.

Since safety and efficiency are critical in railways, further research to enhance the train network reliability while meeting the control packet end-to-end delay requirement was essential. In Chapter 3, using the same model presented in the literature, enhancing one-wagon train network reliability at the sensor level using triple modular redundancy (TMR) is demonstrated. In addition, a two-wagon model is
also simulated to ensure its functionality. The reliability for this model is also calculated.

Chapter 4 focuses on investigating a more realistic train wagon model with multiple sampling periods and different. Pyramid architecture at the controller level is studied. A passive supervisor is added to the network.

In chapter 5, an active supervisor rather than a passive one is integrated into the network. The network performance is presented and a comparison with the network with passive supervisor is performed. Furthermore, a reliability study is conducted to establish the best performing supervision scheme.

This thesis is concluded in Chapter 6.
II. LITERATURE REVIEW

II.1. CONTROL PROTOCOLS IN TRAINS

There are various types and products used in trains nowadays. Consequently, there are different types of networks in trains. There are networks that function in a train wagon. Other networks control the overall train functionality. There are many different protocols that are used in operating the trains. Among the various networks used in trains are: Control Area Network (CAN), Train Communication Network (TCN) platform, and LonWorks platform.

II.1.1. CONTROL AREA NETWORK (CAN)

The CAN protocol was designed by Robert Bosch for use in automotive controls networks. The CAN protocol is a deterministic one. Moreover, its performance is prioritized with short messages and is capable of detecting faults in the network through extensive error detection mechanism [14]. This protocol has other non-automotive applications, such as in trains and is applied widely. One reason for the spread of the CAN is the simplicity of its components [14]. The limitations of the CAN occur in the access of the medium because the nodes must listen to the bus while transmitting packets/messages, and the control word length is at least double the propagation, which prevents this protocol from having high speeds except for short distances [14]. In trains, TCN and LonWorks are dominating train control networks in the market more than CAN protocol.

II.1.2. TRAIN COMMUNICATION NETWORK (TCN)

The manufacturing of a train is a very complicated task concerning many electronic subsystems, from different manufacturers, as well as various standards and protocols applied. These subsystems need to be integrated by cabling, control and providing interfaces between these subsystems. Moreover, there is a need to eliminate some problems such as trains having different vehicles/wagons from different vendors/manufactures and the frequent separation or addition of new wagons. These operations can be done daily with high frequency. The interoperability issue can be
achieved only through setting international standards. The International Electrotechnical Commission (IEC), in collaboration with the Union Internationale des Chemins de Fer (UIC), designed the TCN standard [15].

TCN is composed of two different networks with somewhat different protocols. The first protocol is Wire Train Bus (WTB). It is used along the length of the entire train [16]. It interconnects vehicles over hand-plug jumper cables or automatic couplers. The Multi-function Vehicle Bus (MVB) is the second protocol. It is used for networks within a single wagon. It connects equipment within a wagon, or within different wagons in closed train sets. MVB controller provides redundancy at the physical layer, i.e., a device transmits on the redundant lines, but listens to one while monitoring the other. Figure 1 shows the structure of a TCN network.

Figure 1: Overview of the TCN Structure [16]

II.1.3. LONWORKS PROTOCOL

The idea behind LonWorks platform is developing a technology which addresses “the networking of everyday device” [17]. It is a product of Echelon Corporation in which it is built on a low bandwidth protocol created by Echelon Corporation for networking devices over different media.

LonWorks is widely used in many fields, mainly in the building automation, transportation, medical, industrial, home, utility, etc. [17]. It is also popular for the automation of various functions within buildings such as lighting. The reason that LonWorks is involved in all these markets and others is that the platform was designed to control various and general devices.

LonWorks is used as a control network in trains network Management. Network management refers to maintenance and administration of network nodes [18]. Train network is always changing in terms of the nodes connected to the network; hence the network management of a train is never fixed. This type of network management is known to have a “continuous installation” scenario [13]. There are 2 scenarios for
network management in LonWorks trains: single network manager or multiple network managers. In single network manager there is only one network manager node in the whole train and all the devices in the train are connected to this manager. The advantage of such system is that there are some message types that will travel with very high speed [19]. While in multiple network managers system each device in a train car is connected to a local network manager called Proxy Nodes. All Proxy nodes are connected to a single train manager that is found in one of the locomotive cars of the train [19]. Figures 2 and 3 illustrate the LonWorks different management.

**Figure 2: A single network manager [19]**

![Single Network Manager](image)

**Figure 3: Multiple network managers [19]**

![Multiple Network Managers](image)
II.2. ETHERENET IN NCS

Since Ethernet is a communication non-deterministic protocol, the use of Ethernet without modification as a control protocol is a challenge and has been studied in multiple researches [1, 20, 21, 22, 23, 24, 25, 26, 27]. It has even been studied for adoption in aerospace systems [28]. Below is a summary of the work done regarding Ethernet integration in Industrial control systems as well as terrestrial transportation systems (vehicular and railway). These researches show the possibility of using Ethernet as a control protocol in the automation field despite of its non-deterministic nature. Therefore, further research is required to simulate the use of Ethernet in a realistic train model.

II.2.1. ETHERENET IN INDUSTRIAL CONTROL SYSTEMS

Automated work cells consist of sensors (sen.), controllers and actuators (act.) connected over the network [29]. Sensor nodes, known as source nodes because the data flow originates at these nodes, are smart nodes. Actuator nodes, also called sink nodes because the data flow ends at these nodes, are also smart. Smart sensors/actuators (SAs) have network communication capabilities and sometimes they have self diagnostic and calibration features as well. The smart sensor collects physical data from the process under control and sends it in a packet format after encapsulation over the communication network to reach the controller. At the controller the data is de-capsulated, analyzed and processed; a control word is generated, encapsulated and sent over the network once again to reach the smart actuator node. The smart actuator collects the control word from the network and applies it to the physical process after de-capsulation.

In [29, 30], the authors tested Fast and Gigabit (Gb) Switched Ethernet Networked Control System for automated work cell implementation. In this model, each machine is an isolated control LAN. An overhead of non-real-time applications was also added to the real-time control data flow to create a mixed traffic environment. The IEEE 802.3 or Ethernet protocol was used without modifications as the control network protocol [31]. OPNET [32] simulations revealed the success of the tested model [30].

A fault-tolerant model at the controller level was also tested in [33, 34] to form a linear model simulating the in-line process. In these simulations, different controllers
connected on the same network in a linear topology could back up each other in case one controller failed. A fault-tolerant hierarchical model was also introduced and tested in [35, 36] introducing a supervisor node on top of many controllers connected in the form of a continuous line.

II.2.2. ETHERNET IN AUTOMOTIVE ON-BOARD NETWORKS

The success of the Ethernet use in industrial Networked Control System was the motive behind trying its implementation in automotive on-board networks. Electronic Stability Program (ESP), Anti-lock braking system (ABS), drive by-wire, brake by-wire, electronic assisted steering and many other applications are good examples of today’s electronics applications on-board of a moving vehicle. Other functions include passive as well as active safety and entertainment on board of the vehicles.

Based on the Ford model in [37], a 90-node model was successfully tested using OPNET simulations for pure control automotive on-board network [38]. This model was further modified by the authors to include Triple Model Redundancy (TMR) at the sensor level [39] and fault-tolerance at the controller level [37].

II.2.3. ETHERNET TRAIN CONTROL NETWORK

Due to the current technological advancement, entertainment and multimedia are becoming a necessity on board of moving vehicles. Moreover, as previously presented, researchers proved that Ethernet could be used as a control protocol in industrial machinery and in cars [37]. Consequently, Ethernet, with its wide bandwidth, has evolved as a promising technology in train control networks over the currently used protocols such as Local Operating Networks (LonWorks), Train Communication Networks (TCN) and Controller Area Network (CAN) [2, 3]. In [40], it was shown that the use of Ethernet, as a control protocol in trains, could allow carrying an entertainment load on top of the control load. This was achieved without jeopardizing the packet end-to-end delay requirement of the control data. A GbE network model had been proposed as a control and entertainment network within a one 60-seat train wagon (W) [41]. The network consisted of 250 nodes, the maximum number of sensors and actuators currently allowable in train standards [42], divided into 125 sensors and 125 actuators; i.e., a 1:1 sen:act [40]. Examples of such smart
sensor nodes are the ones used in ventilation systems (HVAC), door lock monitor, brake systems, friction monitor systems, etc… Such type of sensor nodes is used for vehicle health monitoring applications. Additionally, there were two categories of entertainment traffic added to the control traffic. The first load was in the form of video streams. The second load was a Wi-Fi traffic produced from mobile wireless nodes (laptops). These laptops were running web browsing (HTTP), email, File transfer protocol (FTP) and database (DB) applications [40]. The sensors, actuators and video screens were all connected via a main switch to a controller that handles control and entertainment loads. However, the Wi-Fi nodes were connected to the controller and the main switch via a wireless access point (AP).

With a packet payload of 32 bytes [42], the sensor to actuator control packet end-to-end delay was measured using OPNET simulations. This measured delay included all the processing, propagation, encapsulation and de-capsulation delays. All simulations were run for 16ms and 1ms sampling periods separately. Figure 4 illustrates the network model used in [40].

Figure 4: One Wagon Train Network [40]
II.2.4. FAULT TOLERANT TRAIN NETWORK AT THE CONTROLLER LEVEL

In order to increase system reliability, two controllers were used instead of one controller [43]. A Control Server (Controller) handled the control load and an Entertainment Server handled the entertainment traffic (video streams and Wi-Fi load). Furthermore, to enhance system reliability, the Entertainment Server acted as a backup for the Controller. The figure below shows the enhanced network model that was simulated with OPNET.

In case the Controller failed, the Entertainment Server handled both the control traffic and entertainment load. For the Entertainment Server to handle the control traffic, sensors sent their packets to both the Controller and Entertainment servers. In case of the Entertainment Server failure, the system dropped both the video streams and the Wi-Fi services.

Since the Entertainment Server acted as backup for the Controller, both the Entertainment Server and the Controller received all the control packets from the sensors. This was achieved while meeting the sensor-actuator packet delay requirements. Figure 5 shows the enhanced network model.

**Figure 5: Enhanced One Wagon Train Network [43]**
III. TRAIN NETWORK RELIABILITY ENHANCEMENT

The network model simulated is a GbE network in a star topology with optical fiber connections. The model consisted of 2 main networks: the control network and the entertainment network sharing the same resources. The network model was enhanced in this research at the sensor level through using triple modular redundancy. The network model is simulated using OPNET network simulator. In this chapter the reliability enhancement at the sensor level is illustrated.

III.1. ONE WAGON MODEL WITH TMR AT THE SENSOR LEVEL

In this enhanced model each train wagon contains a total of 250 nodes [42]. In [40] the sensor used was 1:1. However, the control network in the model studied next has a sensor:actuator of 3:1 for triple modular redundancy (TMR) to have a total of 500 nodes per wagon. All the sensors and actuators are arranged in subnets. There are 4 subnets; each had 125 nodes in which there are 1 subnet for actuators (125 actuators) and 3 subnets for sensors (375 sensors). All the control services are handled by the Controller which was connected to the subnets through a main switch.

The entertainment network consists of 2 main subnets; the video streaming subnet and the Wi-Fi subnet. The Video Stream subnet is composed of video screens which are connected to the Entertainment Server through the main switch to receive video streams. There are 2 video stream qualities in use:

1. The minimum acceptable video quality (MAVQ) with 128×240 pixels resolution and 16 bits/pixel. The frame rate for this quality is 24 frames/second [45].

2. The uncompressed DVD quality is based on National Television System Committee (NTSC) standard with a resolution of 240×352 pixels and 24 bits/pixel. The used frame rate is 29.97 frames/second [46].

Furthermore, the Wi-Fi subnet consists of Wi-Fi laptop nodes connected to the Entertainment Server through a wireless AP and then the main switch. The Wi-Fi nodes have HTTP, FTP, DB, and email as running applications. Figure 6 shows the network schematic used during the simulations.
In this research, there are 2 approaches used to increase the reliability of the network.

First, the Entertainment Server acts as a backup for the Controller as in [43]. In case of failure of the Controller, the Entertainment Server handles both the control and the entertainment loads with a degraded performance in the number of video streams supplied from the Entertainment Server [43]. For the Entertainment Server to deal with the control data, it has to receive signals from the sensors. Additionally, there is a watchdog signal between both the Controller and the Entertainment Server to ensure the proper functionality of the Controller [43]. This watchdog signal are sent at a rate of 1ms with a packet size of 32 bytes. In case of failure of the Entertainment Server, all the Entertainment services are dropped.

Second, a Triple Modular Redundant (TMR) model for the sensors is used. In this model, each sensor is triplicated. Identical data packets are sent from each of the three sensors to the Controller. The Controller compares all three packets and if they are within the allowable tolerance range, it forwards one decision packet to the appropriate actuator. On the other hand, if one of the three packets does not match the others, the controller concludes that the sensor responsible for sending such a faulty packet has failed and therefore ignores its packets. The final decision is taken by the controller based upon the other two packets and the system will not fail even though one of the sensors did fail [47, 48, 49].
The first consequence for having this more reliable model is having extra hardware in which the number of sensors is tripled to be 375 rather than 125. Furthermore, there is added hardware due to the presence of the Controller and the Entertainment Server rather than just one controller. Another main consequence is the increase in the number of packets in comparison to the number of packets in [40] and [43]. Therefore, the number of packets passing through the main switch increases dramatically.

**Switch Forwarding Rate**

The main switch used in the model presented in [40, 43] was the generic switch in OPNET. This generic switch has a packet forward rate of 0.5 Mpps. However, with the increase in the number of packets due to the triple modular redundancy of the sensors, this forward rate is not enough to pass both the control packets and video stream packets. Consequently, this forward rate has to be increased to a value that is high enough to pass all the packets and meet the maximum allowable delay for the control packets. In this research it is found that the minimum forward rate that would ensure meeting all the time constraints in all simulated scenarios is 2.2 Mpps. This number is rather small compared to the forwarding rate of the 128 port switches available in the market. For example, the Cisco Catalyst 3560 series GbE switches have a minimum forward rate of 38.2 Mpps [50]. Therefore the presented results are pessimistic. If a switch with a forward rate of 38.2 Mpps had been used, the end-to-end delay would have been smaller.

In this triple modular redundant model the simulation scenarios are a combination of having:

- Two controllers or one controller (simulating the failure of the Controller).
- Two different sampling rates (1 ms and 16 ms).
- Two video qualities (DVD and MAVQ).

The exact scenarios used are further explained in the simulation outcomes and results section later.
III.1.1. SIMULATED SCENARIOS AND OUTCOMES

The main metric for all the models is the maximum sensor-to-actuator packet end-to-end delay. The sensor-to-actuator packet end-to-end delay is the sum of packet delay at the Controller and at the actuator. This delay takes into account processing, propagation, encapsulation and de-capsulation delays.

In this triple modular redundant model the simulation scenarios are a combination of having different number of controllers, different sampling rates and different video qualities. The scenarios are divided into 4 main scenarios. Scenarios One and Two simulate the fault free (FF) network model while scenarios three and four simulate the network when the Controller fails and the Entertainment Server handles both the control and the entertainment loads.

In simulation outcomes figures below the X-axis represent the simulation time in minutes and the Y-axis represent the delay in seconds. Also, in the figures below the red dots indicate the end-to-end delay from the acting Controller/Server to the actuator node and the blue dots indicate the end-to-end delay from the sensor to the Controller. The total packet end-to-end delay is the sum of the red and blue dots.

III.1.1.1. SCENARIO ONE

In Scenario One (a), the network consists of:

- 16 ms Sampling period
- Two Controllers
- Minimum video quality (MAVQ)

The worst sensor to actuator packet end-to-end delay resulting from this scenario is 9.47 µs while loading the network with entertainment data from 60 Wi-Fi laptops and 60 video streams. Due to the higher forwarding rate of the switch than in [40] and [43], the queuing delay at the switch decreased; this resulted in a significant decrease in the end-to-end delay in comparison to [43]. Despite the fact that the worst delay is 0.059% of the maximum allowable delay, which is the same as the sampling period, no more traffic is added as it is assumed that the wagon contains only 60 seats [41].
In Scenario One (b), the network consists of:

- 1 ms Sampling period
- Two Controllers
- MAVQ

The network would be able to carry up to 60 video streams and 60 Wi-Fi nodes with an end-to-end delay of 11.4 μs which is 1.14% of the maximum allowable delay. Figure 7 shows the OPNET delay outcomes for Scenario One (b).

Figure 7: Scenario One (b), Control Packet End-to-End Delay

III.1.1.2. SCENARIO TWO

In Scenario Two (a), the network consists of:

- 16 ms Sampling period
- Two Controllers
- DVD Quality

Even though the produced delay for this scenario is 9.9 μs (0.062% of maximum acceptable delay(MCD)), 60 Wi-Fi nodes and only 15 video streams were added onto the network which is not the maximum number of seats in a train wagon. This limitation in the number of video streams is caused by the saturation of the Gb links used in the network. The figure below shows the OPNET delay outcomes for Scenario Two (a).
In Scenario Two (b), the network consists of:

- 1 ms Sampling period
- Two Controllers
- DVD Quality

The worst end-to-end delay is 12.885 $\mu$s, i.e., 1.29% of the MCD requirement. The network is loaded with 60 Wi-Fi nodes and 15 video streams.

### III.1.1.3. Scenario Three

In Scenario Three (a), the network consists of:

- 16 ms Sampling period
- One Controller
- MAVQ

This scenario simulates the case when the Controller fails and the Entertainment Server handles both the control and Entertainment Data. In comparison with the outcomes of [40], the increase in the switch forwarding rate still plays a major role in decreasing the packet end-to-end delay. The produced delay is 12.25 ms with 60
video streams and 60 Wi-Fi nodes. The figure below shows the OPNET delay outcomes for Scenario Three (a).

**Figure 9: Scenario Three (a), Control Packet End-to-End Delay**

In Scenario Three (b), the network consists of:
- 1 ms Sampling period
- One Controller
- MAVQ

The worst sensor to actuator delay is 0.56 ms. The entertainment traffic is produced from 12 video streams and Wi-Fi nodes. The limitation for the number of video streams now is meeting the time constraints for the control data since both types of data (control and entertainment) are handled by the Entertainment Server.

**III.1.1.4. SCENARIO FOUR**

In Scenario Four (a), the network consists of:
- 16 ms Sampling period
- One Controller
- DVD quality

In this scenario, the number of video streams carried by the network, besides 60 Wi-Fi nodes, is 14 with a control packet end-to-end delay of 14.05 ms.
In Scenario Four (b), the network consists of:

- 1 ms Sampling period
- One Controller
- DVD quality

Here the number of video streams supported by the network shrinks down to 3 streams as the amount of produced control packets is more as compared to the 16 ms scenario. Therefore, to avoid the congestion of the links, the number of streams had to be decreased in order not to degrade the performance of the control network. The delay output of OPNET for this scenario is shown in the figure below.

Figure 10: Scenario Four (b), Control Packet End-to-End Delay
All the sensor-to-actuator packet end-to-end delays are less than the maximum allowable delays. Moreover, zero packet loss was achieved. The following table summarizes resulting end-to-end delays for each scenario simulated with the entertainment load carried by the network.

Table 1: Summary of TMR Simulation Outcomes

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>Num of Controllers</th>
<th>Sampling Period (ms)</th>
<th>Num. of Video Stream</th>
<th>End-to-End Delay (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>2</td>
<td>16</td>
<td>60*</td>
<td>9.47</td>
</tr>
<tr>
<td>b</td>
<td>2</td>
<td>1</td>
<td>60*</td>
<td>11.4</td>
</tr>
<tr>
<td>Scenario 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>2</td>
<td>16</td>
<td>15**</td>
<td>9.9</td>
</tr>
<tr>
<td>b</td>
<td>2</td>
<td>1</td>
<td>15**</td>
<td>12.885</td>
</tr>
<tr>
<td>Scenario 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>1</td>
<td>16</td>
<td>12*</td>
<td>12252.8</td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>1</td>
<td>60*</td>
<td>561.2</td>
</tr>
<tr>
<td>Scenario 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>1</td>
<td>16</td>
<td>14**</td>
<td>14051.2</td>
</tr>
<tr>
<td>b</td>
<td>1</td>
<td>1</td>
<td>3**</td>
<td>715.7</td>
</tr>
</tbody>
</table>

* MAQV ** DVD video quality
III.2. TWO WAGONS NETWORK MODEL

The reliable one wagon train model presented in the above section is further expanded into a two wagon model [44]. Each wagon still consists of 500 SAs with a 3:1 sen:act ratio. This is to implement TMR at the sensors level. Like in the one wagon model, all the sensors and actuators are connected via the main switch in each wagon to the Controller. Moreover, 60 video streams and 60 Wi-Fi laptops constitute the entertainment load. The video streams quality used is MAVQ. Additionally, the Wi-Fi nodes are running the same applications mentioned in the above section. The entertainment load is handled by the Entertainment server in each wagon [51].

The two wagons are interconnected at the main switch level via a 10GbE optical fiber cable. For each Controller and each Entertainment Server in each wagon to act as backup for other servers, they need to receive the data from the sensors in both wagons. Consequently, each sensor multicasts its packet to all Controllers and Entertainment Servers in both wagons. In other words, each sensor forwards four replicas of its packet to two Controllers and two Servers in the network model.

In this two wagon model, all 4 servers act as backup to each other. This means that if a Controller within one wagon fail, the Controller of the second wagon will carry the control load of both wagons. For this fault tolerant model to be achievable, the four servers share a watchdog signal among them with a packet size of 32 bytes and sent every 1ms. Consequently, if one controller fails the remaining ones will be able to identify that. The other controller in the other wagon will act and take over the control load of the first wagon. Figure 11 shows the two wagon network model.

The main bottleneck to be monitored is the main switches forwarding rate. In the one wagon train model presented above a forwarding rate of 2.2Mpps is used. However, since the control load is three times more than in the one wagon train wagon, therefore, the forwarding rate has to be increased up to 6.6Mpps. This forwarding rate allows the control packet end-to-end delay to just meet the delay requirement. This increase in the forward rate will not cause any implementation obstacles since the forwarding rate of the 128 port switches available in the market is much larger than 6.6Mpbs. As mentioned before the Cisco Catalyst 3560 series Gigabit Ethernet switches have a minimum forward rate of 38.2Mpbs [50].
III.2.1. TWO WAGONS SIMULATED SCENARIOS

The main metric under study is still the maximum sensor-to-actuator packet end-to-end delay. In this study, two main scenarios are simulated. The first scenario shows the network outcomes in normal full functioning manner with all four controllers (one Controller and one Entertainment Server per wagon) for both 16ms and 1ms sampling periods. In this scenario, the Controller handles the control data while the Entertainment Server handles the video streaming and the Wi-Fi load within its wagon while receiving the control packets from all sensors. The packet end-to-end delay is calculated from a sensor to the actuator within the same wagon.

In the second scenario, the worst case scenario is simulated in which all servers fail except one server. If this server is a Controller, it handles the control load of both wagons. If this server is an entertainment server, then it drops the entertainment load and handles all the control load of both wagons.

Processing, propagation, encapsulation and de-capsulation delays are all considered in the delay measurement.

As in the previously mentioned the X-axis represent the simulation time in minutes and the Y-axis represent the delay in seconds in the OPNET figures below. Also, the red dots indicate the end-to-end delay from the acting Controller/Server to the actuator node and the blue dots indicate the end-to-end delay from the sensor to the Controller. The total packet end-to-end delay is the sum of both delays.
III.2.1.1. SCENARIO ONE

In Scenario One (a) the network consists of:

- 16ms Sampling period
- Two Controllers and two Entertainment Servers

The maximum sensor to actuator packet end-to-end delay measured in each wagon. The delay produced was less than 1% of the maximum allowable delay. This delay is produced while loading the network with data from 60 Wi-Fi laptops and 60 video streams. Since it is assumed that the wagon contains only 60 seats, adding more traffic would be irrelevant.

In scenario One (b) the network has:

- 1ms Sampling period
- Two Controllers and two Entertainment Servers

60 video streams and 60 Wi-Fi nodes were attached to the network. The produced end-to-end delay is still below 10% of the maximum allowable delay. The figure below shows the delay outcomes for this scenario.

Figure 12: Scenario One (b), Two wagons End-to-End Delay
III.2.1.2. SCENARIO TWO

In scenario Two (a) the network consists of:

- 16ms Sampling period
- One controller carrying all control functions in both wagons

All entertainment loads have been dropped in this scenario as it is considered a critical situation. The sensor to actuator packet end-to-end delay still has a worst value that is less than 1% of the maximum allowable delay.

In scenario Two (b) the network includes:

- 1 ms Sampling period
- One controller carrying all control functions in both wagons

The worst end-to-end delay, with no entertainment load, is less than 2% of the maximum allowable delay. Figure 13 shows the simulation outcome.

**Figure 13: Scenario Two (b), Two wagons End-to-End Delay**
III.3. RELIABILITY ANALYSIS OF THE TRAIN NETWORK MODEL

III.3.1. RELIABILITY MODEL OF THE FAULT-TOLERANT CONTROLLER

For the control function to fail within one wagon, BOTH the Entertainment Server and the Controller should fail. However in a two wagons model, the control function of the whole network will fail after the failure of both Controllers and both Entertainment Servers [52]. Consequently, the control function is achieved through a parallel system consisting of two Controllers and two Entertainment Servers [48, 49]. Figure 14 illustrates the system reliability block diagram.

Figure 14: Parallel system reliability block diagram

If \( R_K(t) \), also referred to as \( R_K \), is the reliability of a Controller and \( R_E(t) \), also referred to as \( R_E \), is the reliability of an Entertainment Server. Furthermore, let \( R_{cont, FT}(t) \), also referred to as \( R_{cont, FT} \), be the reliability of the control function. Then,

\[
R_{cont, FT}(t) = R_{cont, FT} = 1 - (1 - R_K)^2(1 - R_E)^2
\]  

(1)

The term \( (1 - R_K) \) is the probability of failure of a Controller while \( (1 - R_E) \) is the probability of failure of an Entertainment Server.
The fault-tolerant two wagon model is expected to enhance the reliability of the control function, which is worth studying. If no fault-tolerance is applied, the control functionality within any wagon would fail by the failure of its Controller. The figure below shows the system reliability model without fault tolerance.

**Figure 15: Series system reliability block diagram**

Let $R_{\text{cont}}(t)$, also referred to as $R_{\text{cont}}$, indicates the control function reliability with no fault tolerance. Then,

$$R_{\text{cont}}(t) = R_{\text{cont}} = R_K^2$$

(2)

For an exponentially-distributed failure time, the failure rate is constant. If $\lambda_K$ is the Controller failure rate, then $R_K = R_K(t) = e^{-\lambda_K t}$ [48, 49]. For simplicity, assume the Entertainment Server is assumed to have the failure rate $\lambda_E = \lambda_K = \lambda$. Therefore, it has a failure rate $R_E = R_E(t) = e^{-\lambda_E t} = e^{-\lambda_K t} = e^{-\lambda t}$. Applying a numerical example using $\lambda_E = \lambda_K = \lambda = 1$/month, Figure 16 shows $R_{\text{cont,FT}}$ and $R_{\text{cont}}$ for a period of 2 month. The improvement in reliability is clear.

**Figure 16: Reliability Comparison**
III.3.1.1. Effect of Coverage on System Reliability

In the above analysis, the switching of the control tasks from a failed Server to another operational server is assumed to be always successful. This is somehow an optimistic view. In real life, there is always the possibility that the switching mechanism would fail. For example, if the inter-communication between the operational Servers fails, a Server in one wagon may assume that the other Server in the other wagon has failed and hence, will take over its tasks. Such a conflict will cause a system failure [53, 54]. As the reconfiguration process successfulness is not guaranteed, it has to be considered in the reliability model. The probability of successful detection/reconfiguration is called coverage [48, 49, 55, 56]. The coverage is defined as the proportion of faults from which a system can automatically recover [53]. The coverage is included in reliability/availability models and it’s determined by the user. Any small mistake in the calculation of the coverage leads to false reliability/availability estimations [49]. Moreover, if the coverage of a system decreases, the system reliability is expected to decrease as well.

Figure 17 shows the Markov model that will be used to calculate the reliability of the control function ($R_{cont, FT}$) discussed above. In this model, it is assumed, for simplicity, that $\lambda_K = \lambda_E = \lambda$. Consequently, the failure of any the four controllers/servers may cause a system failure with a probability $(1-c)$, where $c$ is the coverage. In other words, if $c=1$, the system is really a parallel system with four components; it fails after the failure of the fourth component.

Figure 17: Conservative Markov Model
Any state in Figure 17 indicates the number of operational server in that state. This is why the initial state is called “4” and the final (absorbing) state is called “0”. Moving from state “i” to state “i-1” (for i=1 to 4) occurs when one of the operational components fails and the recovery is successful (with a probability c). Remember that c is the coverage parameter. For example, moving from state 4 to state 3 occurs at a rate of \(4\times \lambda \times c\) because there are four components that can fail and the system will only move to state 3 if the recovery is successful. If the recovery is not successful, the system moves from state 4 directly to state 0 (the failure state) at a rate \(4\lambda(1-c)\).

Finally,

\[
R_{\text{cont,FT}}(t) = R_{\text{cont,FT}}(0) - \exp(-4\lambda t)
\]

(3)

The Markov model can be described by the following Chapman-Kolmogorov equations (in matrix form) [48]:

\[
\frac{dP(t)}{dt} = P(t) \times T
\]

(4)

Where

\[
\frac{dP(t)}{dt} = \begin{bmatrix}
\frac{dP_4(t)}{dt} \\
\frac{dP_3(t)}{dt} \\
\frac{dP_2(t)}{dt} \\
\frac{dP_1(t)}{dt} \\
\frac{dP_0(t)}{dt}
\end{bmatrix}
= \begin{bmatrix}
\frac{dP_4}{dt} \\
\frac{dP_3}{dt} \\
\frac{dP_2}{dt} \\
\frac{dP_1}{dt} \\
\frac{dP_0}{dt}
\end{bmatrix}
\]

\[
P(t) = [P_4(t) \quad P_3(t) \quad P_2(t) \quad P_1(t) \quad P_0(t)] = [\begin{bmatrix}
P_4 \\
P_3 \\
P_2 \\
P_1 \\
P_0
\end{bmatrix}]
\]

and

\[
T = \begin{bmatrix}
-4\lambda & 4c\lambda & 0 & 0 & 4\lambda(1-c) \\
0 & -3\lambda & 3c\lambda & 0 & 3\lambda(1-c) \\
0 & 0 & -2\lambda & 2c\lambda & 2\lambda(1-c) \\
0 & 0 & 0 & -\lambda & \lambda \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
\]
Given that $P_4(0) = 1$ and $P_3(0) = P_2(0) = P_1(0) = P_0(0) = 0$, the Chapman-Kolmogorov equations can be solved in closed form as follows to find the probability of being in a certain state:

\[
P_4(t) = P_4 = e^{-4\lambda t}
\]  

(5)

\[
P_3(t) = P_3 = 4c \left( e^{-3\lambda t} - e^{-4\lambda t} \right)
\]  

(6)

\[
P_2(t) = P_2 = 6c^2 \left( e^{-2\lambda t} - 2e^{-3\lambda t} + e^{-4\lambda t} \right)
\]  

(7)

\[
P_1(t) = P_1 = 4c^3 \left( e^{-\lambda t} - 3e^{-2\lambda t} + 3e^{-3\lambda t} - e^{-4\lambda t} \right)
\]  

(8)

\[
P_0(t) = P_0 = 1 - \left( P_4(t) + P_3(t) + P_2(t) + P_1(t) \right)
\]  

(9)

If $\lambda = 1$/month (as above), the effect of coverage can be seen in Figure 18. $R_{cont,FT}$ is shown for several values of the coverage parameter $c$. The effect of an incorrect estimation of the coverage is clear. For example, using $c=0.95$ instead of 0.9 will yield a reliability of 38.27% instead of 32.97% after two months.

**Figure 18: Effect of Coverage in Conservative Model**
It is interesting to note here, that for $c=1$, the system is really a parallel system (as mentioned above). Solving the Markov model of Figure 17 with $c=1$ will produce the same $R_{cont\_FT}$ as in equation (1) with $\lambda_K = \lambda_E$. Alternatively, for $c=0$, the system becomes a series system where the failure of any Controller/Entertainment Server causes the failure of the control function. In equations (5) through (9), if $c=0$ then $R_{cont\_FT} = e^{-4\lambda t}$. This is the reliability of a series system with 4 identical components and each component has a failure rate of $\lambda$. 
III.3.2. Efficiency of Error Detection/Reconfiguration Mechanisms

The model described in the previous section is very conservative. In the previous model, it is assumed that the failure of any component can lead to a system failure if the recovery is not successful. A more realistic scenario is studied. Let $K_1$ and $K_2$ be the controllers in the two wagons. Also, let $E_1$ and $E_2$ be the two entertainment servers. Assume that the recovery strategy is as follows: When one of the controllers ($K_1$ or $K_2$) fails, the control function is switched to the other controller. This other controller handles the control function of both wagons. As a precautionary measure, the entertainment in both wagons is shut down. The system will now consist of one controller handling the control function of both wagons while $E_1$ and $E_2$ are in hot stand-by mode [48]. If the remaining controller fails, the control function is switched to one of the entertainment servers. A system failure occurs when the second entertainment server fails.

If the first entertainment server to fail is $E_1$ (or $E_2$), it is realistic to assume that the entertainment is shut down in wagon 1 (or wagon 2) without affecting the control function. Based on these assumptions, the Markov model in Figure 17 has to be modified to reflect the new scenario. Figure 19 shows the improved model. Again, any state indicates the operational components in that state. Remember that $\lambda_E$ is the failure rate of the Entertainment Server and $\lambda_K$ is the failure rate of the Controller.

The initial state is $2K2E$. This is the fault free (FF) state; both controllers and both entertainment servers are operational. A failure of one of the controllers takes the system to state $1K2E$. This transition will only occur if the recovery is successful and the operational controller is able to take over the tasks of the failed controller and handle the entire control function in both wagons. This is why the transition rate from state $2K2E$ to state $1K2E$ is $2c\lambda_K$. If the recovery is not successful, the control function fails and the system moves to state $F$ (i.e., the control function failure state) at a rate of $2\lambda_K(1-c)$. Also, a failure of one of the Entertainment servers moves the model to state $2K1E$. Since two servers can fail, the transition rate from $2K2E$ to $2K1E$ is $2\lambda_E$. The coverage does not affect this transition as mentioned in the assumptions leading to this more realistic scenario.
In state 1K2E, a failure of one of the two entertainment servers moves the system to state 1K1E at a rate of $2\lambda_E$. The failure of the remaining operational controller takes the system to state 2E at a rate of $c\lambda_K$ and to state F at a rate of $(1-c)\lambda_K$.

In state 2K1E, the failure of $K_1$ or $K_2$ moves the system to state 1K1E; the transition rate is $2c\lambda_K$. If the recovery is not successful, the system moves to state F at a rate of $2\lambda_K(1-c)$. The failure of the remaining Entertainment server takes the system to state 2K at a rate of $\lambda_E$. Here again, the recovery process is not involved because the entertainment is shut off and the control function is not affected.

In state 2K, $E_1$ and $E_2$ have failed but the control function has not been affected since $K_1$ and $K_2$ are both operational. If either $K_1$ or $K_2$ fails, the model moves to state 1K; the control of both wagons is handled by the remaining operational controller. The coverage affects this transition and therefore, the transition rate from 2K to 1K is $2c\lambda_K$; also, the transition from 2K to F is $2(1-c)\lambda_K$.

In state 2E, both controllers have already failed and one the Entertainment Servers is controlling both wagons. If either $E_1$ or $E_2$ fails, the control function is switched to the remaining operational server. Here again, the coverage is involved in the transition as shown in Figure 19.

In state 1K1E, the situation is more complex. One Entertainment Server has already failed as well as one of the controllers. The remaining controller is in charge of the control function of both wagons and the entertainment is turned off in both wagons. The remaining entertainment server acts as a hot stand-by for the remaining controller. If the server fails first, the system moves to state 1K without affecting the control function; consequently, the transition rate from 1K1E to 1K is $\lambda_E$. However, if the controller fails before the Entertainment Server, the control function is switched to the Entertainment Server and the coverage affects the transition; the transition rate from state 1K1E to state 1E is $c\lambda_K$ and the one from 1K1E to F is $(1-c)\lambda_K$.

Finally, in state 1E, the failure of the remaining entertainment server causes a system failure at a rate of $\lambda_E$. The same argument applies for state 1K where the failure of the remaining Controller causes a system failure at a rate of $\lambda_K$. 

32
The system can again be described by the Chapman-Kolmogorov equations. The row vectors $\dot{P}(t)$ and $P(t)$ are:

$$
\dot{P}(t) = \begin{bmatrix}
\frac{dP_{2K2E}(t)}{dt} & \frac{dP_{2K1E}(t)}{dt} & \frac{dP_{1K2E}(t)}{dt} & \frac{dP_{2K}(t)}{dt} & \frac{dP_{1K1E}(t)}{dt} & \frac{dP_{1E}(t)}{dt} & \frac{dP_{2E}(t)}{dt} & \frac{dP_{1K}(t)}{dt} & \frac{dF(t)}{dt}
\end{bmatrix}
$$

$$
P(t) = \begin{bmatrix}
P_{2K2E}(t) & P_{2K1E}(t) & P_{1K2E}(t) & P_{2K}(t) & P_{1K1E}(t) & P_{2E}(t) & P_{1K}(t) & P_{1E}(t) & F(t)
\end{bmatrix}
$$

$$
= \begin{bmatrix}
P_{2K2E} & P_{2K1E} & P_{1K2E} & P_{2K} & P_{1K1E} & P_{2E} & P_{1K} & P_{1E} & F
\end{bmatrix}
$$
and the transition matrix $T$ is:

$$
T = \begin{bmatrix}
-2(\lambda_E + \lambda_K) & 2\lambda_E & 2c\lambda_K & 0 & 0 & 0 & 0 & 2(1-c)\lambda_K \\
0 & -\lambda_E - 2\lambda_K & 0 & \lambda_E & 2c\lambda_K & 0 & 0 & 0 \\
0 & 0 & -2\lambda_E - \lambda_K & 0 & 2\lambda_E & c\lambda_K & 0 & 0 \\
0 & 0 & 0 & -\lambda_E - \lambda_K & 0 & \lambda_E & c\lambda_K & (1-c)\lambda_K \\
0 & 0 & 0 & 0 & -2\lambda_E & 0 & 2c\lambda_E & 2(1-c)\lambda_K \\
0 & 0 & 0 & 0 & 0 & \lambda_K & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \lambda_K & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
$$

Given that the system starts in state $2K2E$, these differential equations can be solved and the probabilities of being in each of the model states will be as follows:

$$
P_{2K2E} = e^{-2(\lambda_E + \lambda_K)t} 
$$

(10)

$$
P_{2K1E} = 2e^{-(\lambda_E + 2\lambda_K)t}(1 - e^{-\lambda_E t}) 
$$

(11)

$$
P_{1K2E} = 2ce^{-(2\lambda_E + \lambda_K)t}(1 - e^{-\lambda_K t}) 
$$

(12)

$$
P_{2K} = e^{-2(\lambda_E + \lambda_K)t} - 2e^{-(\lambda_E + 2\lambda_K)t} + e^{-2\lambda_K t} 
$$

(13)

$$
P_{1K1E} = 4c(e^{-2(\lambda_E + \lambda_K)t} + e^{-(\lambda_E + \lambda_K)t}) 
- e^{-(\lambda_E + 2\lambda_K)t} 
$$

(14)

$$
P_{2E} = [c^2e^{-2\lambda_E t}[e^{-2\lambda_K t} - 2e^{-\lambda_K t} + 1] 
$$

(15)

$$
P_{1K} = 4ce^{-(\lambda_E + 2\lambda_K)t} + 2ce^{-(2\lambda_E + \lambda_K)t} - 2ce^{-2(\lambda_E + \lambda_K)t} - 4ce^{-(\lambda_E + \lambda_K)t} 
- 2ce^{-2\lambda_K t} + 2ce^{-\lambda_K t} 
$$

(16)

$$
P_{1E} = (X)e^{-(2\lambda_E + \lambda_K)t} - (Y)e^{-(2\lambda_E + 2\lambda_K)t} 
	imes 2c^2(e^{-(\lambda_E + 2\lambda_K)t}) 
- 4c^2e^{-(\lambda_E + \lambda_K)t} 
- 2c^3e^{-2\lambda_E t} + (W)e^{-\lambda_E t} 
$$

(17)
Where

\[ X = \frac{4c^2 \lambda_K + 4c^3 \lambda_E}{\lambda_K + \lambda_E} \]

\[ Y = \frac{4c^2 \lambda_K + 2c^3 \lambda_E}{2\lambda_K + \lambda_E} \]

\[ W = 2c^2 + 2c^3 + \frac{4c^2 \lambda_K + 2c\lambda_E}{2\lambda_K + \lambda_E} - \frac{4c^2 \lambda_K + 4c^3 \lambda_E}{\lambda_K + \lambda_E} \]

Figure 20 compares \( R_{\text{cont, FT}} \) for the two models: the conservative model in Figure 17 and the improved model in Figure 19. Several values of the coverage parameter are used. For all these values, the reliability is higher in the improved model. Note that the difference between \( R_{\text{cont, FT}|c=0.9} \) for the realistic model and \( R_{\text{cont, FT}|c=0.95} \) for the conservative model, is very small. All calculations were verified using the SHARPE program [57].

**Figure 20: Effect of Coverage in Realistic Model**
IV. HYBRID NETWORK MODEL AND CONTROLLER

HIERARCHICAL ARCHITECTURE

IV.1. HYBRID CONTROL NETWORK MODEL

The previously simulated models used only one sampling period per scenario, either 1ms or 16ms, which is unrealistic. In an actual train wagon a combination of different sampling periods are present. Additionally, the simulated sen:act is constant to be 1:1 while in an actual train wagon the sen:act is not constant.

The new proposed model in this section utilizes the same Gigabit Ethernet infrastructure used previously without modifications based on the IEEE 802.3 standard [31] for the whole network. It also follows the regulations described in IEC 61375 train standard [42, 59]. Several sampling periods are mentioned in the standard, concerning the SAs, however the most common values are 1ms and 16ms [42]. According to the train network standard, 16ms sampling period is the most used within a train wagon, while 1ms is the smallest sampling period within a wagon [42]. The new model incorporates both sampling periods in a single control network, representing the different possible applications of SAs requiring different sampling periods [58].

A typical number of SAs on a single train wagon is around 250 with sampling periods 1ms (minority) and 16ms (majority) [42]. These will be broken down into 3 groups. Group 1 (G1) consists of 30 sensors and 30 actuators (1:1 ratio) operating with the most demanding sampling period of 1ms. Group 2 (G2) consists of 100 sensors and 50 actuators (2:1 ratio) operating at 16ms sampling period. Finally, Group 3 (G3) also operating at 16ms sampling period, consists of 30 sensors and 10 actuators (3:1 ratio).

This design minimizes the number of switches, using standard 128 port switches, readily available in the market. Also note that the locations of the SAs had been chosen to increase the distance between the switches and the controller, maximizing the trip distance to simulate a worst case scenario. The main switch (MS1) utilizes a forwarding rate of 6.6Mbps.

The entertainment load in the train model can be described in terms of the number of streams, the quality of the video screens as well as the number of Wi-Fi nodes and
the number of applications per node. In the worst case scenario, in a large wagon of 60 seats [41], the network supported 60 different and simultaneously played video streams (one per seat), as well as one Wi-Fi user per seat. Each Wi-Fi user run several simultaneous applications: Database access, email, web browsing and file transfer. This gives a total of 60 Video streams and 60 Wi-Fi users, running 4 simultaneous applications requiring access to the network. The Wi-Fi access is provided via the Access Point (AP) in the middle of the wagon, maximizing coverage area. The network distribution is illustrated in Figure 21.

**Figure 21: Hybrid one wagon train network model**

The applications used for the Wi-Fi nodes are the generic heavy applications built-in OPNET. The video streaming applications had to be custom set to allow DVD quality streaming. DVD quality videos are available in a variety of formats, utilizing different levels of compression and encoding to decrease file size on disk. In previous scenarios uncompressed DVD quality was used. However, to further mimic a realistic model a compressed DVD video stream is simulated in this hybrid model. Streaming a DVD quality file requires a broadband connection ranging from 450Kb/s to 3.5Mb/s depending on the resolution [60]. In the worst case scenario, it took the full length of the film to stream the actual file from the hard disk. This means that the video can
experience significant lag if there are any lost packets. In order to avoid such a case, a streaming rate of 5Mb/s is used instead of 3.5Mb/s.

IV.1.1. SIMULATION OUTCOMES

The hybrid model proposed simulates a single wagon model with 60 video streams and 60 Wi-Fi users. The first scenario simulates the fault free case, where both servers are fully functional; $K_1$ handling all control data and $E_1$ handling the entertainment load. In the second scenario (fault tolerant), one of the two controllers fails. The traffic, now both control and entertainment, is handled by the remaining server while maximizing the number of video streams and Wi-Fi users, without jeopardizing the control load.

In order to gauge the performance of the system, end-to-end delay and packet loss must be monitored. In all simulations, zero packet loss (no packets dropped or delayed) is observed and total end-to-end delays across all SAs are within their respective constraints (37). A 95% confidence analysis is applied to all results. Figures 22 and 23 show examples of the results obtained from different scenarios. In all figures, the x-axis represents the simulation time in minutes and seconds, while the y-axis shows the delay in seconds. With all video streams at DVD quality and all Wi-Fi users running the full load of applications described above, the maximum total end-to-end delay (with a 95% confidence) for each group of SAs is shown in Table 2.

These results guarantee that all SAs operating with a sampling period of 1ms have an end-to-end delay of less than 1ms, and those operating at 16ms have an end-to-end delay of less than 16ms (with 95% confidence). Note that the number of Wi-Fi users is found to be unaffected due to the fact that the load (restricted to 6Mbps) is not comparable to the control or the video streaming loads.

<table>
<thead>
<tr>
<th>No. of Operational Servers</th>
<th>Control End-to-End Delay (µs)</th>
<th>No of Video Streams</th>
<th>No of Wi-Fi Users</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$G1$</td>
<td>$G2$</td>
<td>$G3$</td>
</tr>
<tr>
<td>2</td>
<td>0.010</td>
<td>0.011</td>
<td>0.011</td>
</tr>
<tr>
<td>1</td>
<td>0.751</td>
<td>0.870</td>
<td>0.876</td>
</tr>
</tbody>
</table>

Table 2: Hybrid Model - One Wagon - total end-to-end delays (µs)
Figure 22: Hybrid model - One Wagon - FF - K to A delay (G1)

Figure 23: Hybrid model - One Wagon – FF - K to A delay (G3)
IV.2. HIERARCHICAL ARCHITECTURE AT THE CONTROLLER LEVEL

For the purpose of this study, two of the one wagon model are combined to form a two wagon unit interconnected using the intermediate switch with a supervisory server (Supervisor) connected to this switch as well [61]. This unit can be considered as the main building unit of the train such as the Siemens Desiro diesel or electric multiple unit (DMU or EMU) [62]. A full train can be formulated by conencting several of this two-wagons building unit. All the main switches and the intermediate switch have a forwarding rate of 6.6Mbps which is much lower than the rate of the switches available in the market such as the Cisco Catalyst 3560 Gigabit Ethernet switch [50].

Video streams and Wi-Fi applications represent the entertainment load within the train network. The quality of the video streams used is the standardized compressed DVD quality with a speed of 5MBps [46]. Since, a wagon has a maximum of 60 seats, 60 video streams are run simultaneously on video screens connected to the main switch of each wagon [41]. The Wi-Fi load is simulated using 60 Wi-Fi laptops (1 laptop/seat) connected to each wagon main switch via a wireless router. These laptops are running the same four applications: HTTP, FTP, DB, and Email. Figure 24, shows the two wagons train unit network design.

Figure 24: Two-Wagon Unit Network Model with a Supervisor Node
The model has two dedicated controllers per wagon. $W_1$ has $K_1$ and $E_1$, where $K_1$ was a controller, handling the control load only, and $E_1$ is the entertainment server. The same is true for $W_2$.

To further increase safety on board of the train, a fault tolerance hierarchal model at the controller level is introduced. In each train wagon there are two Servers, the Controller and the Entertainment server representing the first step in the hierarchical model. There is a watchdog signal between all four servers (two Controllers and two Entertainments Servers) every 1ms with a 32 byte packet size. The next level of the controller hierarchy is a **Passive Supervisor** ($S$) in each two-wagon unit receiving the same watchdog signal from all four Servers.

To enhance network reliability, the Entertainment Server acts as a backup for the same wagon Controller. In case a Controller ($K$) fails, the Entertainment Server ($E$) of the same wagon drops the entertainment load and handles the control load only. The Entertainment Server will detect the failure of the Controller when it does not receive the watchdog signal from the Controller. In case the Entertainment Server failed the entertainment load is dropped. This conservative approach is followed to ensure the safety of the train operation. In order for the Entertainment Server to take over the control load, the sensors in both wagons send their data to all four servers while only the one handling the control load made decision and send the control words to the corresponding actuators.

In case both the Controller and the Entertainment Server of a wagon fail, the Entertainment server of the other wagon ($W_j$) will drop the entertainment load and would carry the Control load of the first wagon ($W_i$). If further problems occur in which the Controller or the Entertainment Server of $W_j$ failed, the remaining Server (Controller or Entertainment Server) will carry the control load of $W_j$ and all the entertainment would be dropped while the Passive Supervisor would come into action and carry the Control load of $W_i$. The Supervisor will carry the control load of both wagons if all 4 servers failed. For the Supervisor to handle the control load, the sensors of both wagons start sending their data to the Supervisor after the failure of two servers.

In order for this algorithm to succeed, there is an assumption that the reliability of the Supervisor is higher than that of the four servers and hence it would fail last.
In order to improve the security within the train, four door-cameras per wagon have been added to the network, one camera at each door [63]. Each camera transmits a video stream to the Supervisor to be monitored by the train driver. The video streams are simulated using the low quality video conferencing OPNET default application. This quality runs at a rate of 15 frames/second, each of has 9 bits/pixel and 128×240 pixels resolution. This door-camera streaming is never dropped in case of any server failure.
IV.2.1. **Unique Simulated Scenarios in Passive Supervisor**

Since there are 5 servers in the network (2 Controllers, 2 Entertainment Servers and one Supervisor), then there are $5! = 120$ possible failure paths the network can go through. However, as it’s assumed that the Supervisor is the most reliable node of all 5 servers and that it would fail last, there are only $4! = 24$ possible failure paths as shown below. Each state represent the operational servers in the two wagon network. Also, State $E_2S\_W_2$ refers to the situation where $E_2$ and $S$ are the only operational servers while $E_2$ is carrying the Load of $W_2$. State $E_2S\_W_1$ refers to the situation where $E_2$ and $S$ are the only operational servers while $E_2$ is carrying the Load of $W_1$. The same applies to states $E_1S\_W_1$ and $E_1S\_W_2$. The different paths are shown below:

*Figure 25: Network Failure Paths in Passive*
Figure 26: Network Failure Paths in Passive Supervisor (Cont’d)
In the above paths, there are 18 different states (other than the failure state) that the network goes through, which are:

1. $K_1K_2E_1E_2S$
2. $K_2E_1E_2S$
3. $K_1E_1E_2S$
4. $K_1K_2E_2S$
5. $K_1K_2E_1S$
6. $E_1E_2S$
7. $K_2E_2S$
8. $K_2E_1S$
9. $K_1E_2S$
10. $K_1E_1S$
11. $K_1K_2S$
12. $K_1S$
13. $K_2S$
14. $E_iS_\ W_1$
15. $E_iS_\ W_2$
16. $E_2S_\ W_1$
17. $E_2S_\ W_2$
18. $S$

Since the two-wagons network model is symmetric, then several of these states are similar. For example, the state $K_1K_2E_2S$ and state $K_1K_2E_1S$ are similar where is one wagon with controller failed and the other wagon is fault free. Consequently, there are 11 unique states/scenarios which are analyzed using OPNET; any further scenario can be related to one of the analyzed scenarios. In all these scenarios, the Supervisor is running otherwise the whole network will fail since it was considered the last node to fail.

The scenarios are as follow (i and j represent the wagon number to be either 1 or 2 interchangeably):

- **Scenario 1**: The fault free state in which are 4 servers are functioning ($K_1K_2E_1E_2S$).
- **Scenario 2**: One Controller fails ($W_j$) and the other 3 servers are up and running ($K_iE_1E_2S$). In this case the Entertainment Server of the same wagon ($W_j$) would carry its control load.
- **Scenario 3**: One Entertainment Server fails and the other 3 servers are up and running ($K_1K_2E_1S$).
- **Scenario 4**: The Controller and the Entertainment Server of the same wagon ($W_j$) fail ($K_1E_1S$). The Entertainment Server of the other wagon ($W_i$) will drop its entertainment load and carry the control load of $W_j$.
- **Scenario 5**: The Controller of $W_j$ and the Entertainment Server of $W_i$ fail or vice versa ($K_1E_2S$). The control load of $W_j$ will be handled by the Entertainment Server of $W_j$.
- **Scenario 6**: The Entertainment Servers of both wagons fail ($K_1K_2S$). Then the entertainment load is dropped and no change happens to the control load.
- **Scenario 7**: The Controllers of both wagons fail (E_i,E_j,S). The Entertainment Server of each wagon shall carry the control load of its own wagon and drop its entertainment load.

- **Scenario 8**: The Entertainment Servers of both wagons fail and the Controller of one wagon (W_j) fails (K_i,S). The Supervisor will carry the control load of W_j while the Controller W_i will carry its own control load.

- **Scenario 9**: The Entertainment Server of one wagon fails (W_j) and both Controllers fail in which Controller of wagon W_i fails before the Controller of wagon W_j. The Entertainment Server will carry its own wagon control load (W_i) while the supervisor carries the other wagon (W_j) control load (E_i,S_W_i).

- **Scenario 10**: The Entertainment Server of one wagon fail (W_j) and both Controllers fail in which Controller of wagon W_j fails before the Controller of wagon W_i. The Entertainment Server will carry the control load of the other wagon (W_j) while the supervisor carries the first wagon (W_i) control load (E_i,S_W_j).

- **Scenario 11**: All 4 servers fail, and leave the Supervisor carrying the control load of both wagons (S).
IV.2.1.1. SIMULATION RESULTS WITH PASSIVE SUPERVISOR

Adding a passive fault tolerant hierarchical model at the Controller level is a method to further improve the network reliability. The main performance metric is still the control packet end-to-end delay for all the sensors and actuators. This delay should not exceed the corresponding sensor/actuator sampling period [37]. Additionally, the rate of packet loss has been monitored. The OPNET simulations have proved zero packet loss, i.e., there are no dropped or delayed packets in the system. A 95% confidence analysis is applied to all results. The delay for the door-cameras as well as the entertainment load (if present) have been observed and the results showed that they are always within the acceptable limits to avoid video flickering (in case of door-camera or video streaming) and to avoid user frustration due to long response time regarding the Wi-Fi services. Figures 27 to 29 show a sample of OPNET results from different scenarios.

As previously mentioned the X-axis represent the simulation time in minutes and the Y-axis represent the delay in seconds in the OPNET figures below. Also, the red dots indicate the end-to-end delay from the acting Controller/Server to the actuator node and the blue dots indicate the end-to-end delay from the sensor to the Controller. The total packet end-to-end delay is the sum of both delays.

Figure 27: Passive - KiKjEiEjS control end-to-end delay – G3

![Graph showing OPNET results]

Figure 27: Passive - KiKjEiEjS control end-to-end delay – G3
Figure 28: Passive - KiKjS control end-to-end delay - G2

Figure 29: Passive - End-to-end delay with S operational – G1
Table 3 illustrates the output of the OPNET simulations. The end-to-end delay for the 1ms sensors/actuators is found to be below 1ms and the end-to-end delay of the 16ms sensors and actuators is proved to be below 16ms. The presence of the entertainment load is indicated in the below table with the wagon name that it is running in. As previously mentioned, the entertainment load per wagon consists of 60 video streaming and 60 Wi-Fi laptops running 4 different applications; HTTP, FTP, DB, and email. The presented delays include all processing, propagation, encapsulation and de-capsulation delays. The delay for the door-cameras as well as the entertainment load (if present) have been observed and the results showed that they are always with the acceptable limits to avoid video flickering (in case of door-camera or video streaming) and to avoid user frustration due to long response time regarding the Wi-Fi services.

Table 3: Passive Supervisor total end-to-end delays (µs)

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>Server handling</th>
<th>Control End-to-End Delay (µs)</th>
<th>Entertainment Enabled in Wagon</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Control load</td>
<td>G1</td>
<td>G2</td>
</tr>
<tr>
<td></td>
<td>Wᵢ</td>
<td>Wᵢ</td>
<td>G1</td>
</tr>
<tr>
<td>KᵢKⱼEᵢEⱼS</td>
<td>Kᵢ</td>
<td>Kⱼ</td>
<td>33.2</td>
</tr>
<tr>
<td>KᵢEᵢS</td>
<td>Kᵢ</td>
<td>Eᵢ</td>
<td>17.3</td>
</tr>
<tr>
<td>KᵢKⱼEᵢS</td>
<td>Kᵢ</td>
<td>Kⱼ</td>
<td>18.6</td>
</tr>
<tr>
<td>KᵢEᵢS</td>
<td>Kᵢ</td>
<td>Eᵢ</td>
<td>15.3</td>
</tr>
<tr>
<td>KᵢEⱼS</td>
<td>Kᵢ</td>
<td>Eⱼ</td>
<td>14.1</td>
</tr>
<tr>
<td>KᵢKⱼS</td>
<td>Kᵢ</td>
<td>Kⱼ</td>
<td>15.2</td>
</tr>
<tr>
<td>EᵢEⱼS</td>
<td>Eᵢ</td>
<td>Eⱼ</td>
<td>15.2</td>
</tr>
<tr>
<td>KᵢS</td>
<td>Kᵢ</td>
<td>S</td>
<td>10.5</td>
</tr>
<tr>
<td>EᵢS_Wᵢ</td>
<td>Eᵢ</td>
<td>S</td>
<td>10.5</td>
</tr>
<tr>
<td>EᵢS_Wⱼ</td>
<td>S</td>
<td>Eᵢ</td>
<td>10.2</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>12.4</td>
</tr>
</tbody>
</table>

All the results were obtained after a 95% confidence analysis. The results shown represent the mean value of the maximum packet end-to-end delay obtained from all runs. Furthermore, the delays for the door-cameras and the video streaming were below the acceptable delay requirements. As per [64], the OPNET results presented in this research are comparable to hardware implementation outcomes as the maximum transmission rate for the control load is 0.256 Mbps.
V. Active Supervisor and Reliability Modeling

V.1. Hybrid Network Model with Active Supervisor

In this study, the same network architecture modeled in the previous chapter is used for comparison purposes. Two of the one-wagon networks are connected together via the intermediate switch using a 10GbE link. The supervisor is connected to the intermediate switch. In each wagon there are 250 SAs. These SAs are divided into three groups G1, G2 and G3. In group G1, there is a 1:1 sen:act running at a sampling period of 1ms. The nodes in G2 and G3 are running at a sampling period of 16 ms with a 1:2 sen:act and 1:3 sen:act respectively. The nodes are located at positions that would ensure maximum propagation and therefore maximum end-to-end delay [65].

All nodes within a wagon are connected to the wagon’s MS using a GbE links. The IS, and both MSs have a forwarding rate of 6.6Mpps. The entertainment load is the same as in the previous chapter. It consists of 60 Wi-Fi nodes as well as 60 compressed DVD video streams running at a rate of 5Mbps [46]. This means that for each seat of the wagon 60 seats has a Wi-Fi laptop and a video stream running [41]. The two-wagon train unit network model is in Figure 30.

In each wagon, there are 2 operational Servers in case of FF state; one Controller (K) and one Entertainment Server (E). They handle the control and the entertainment loads of the wagon respectively. All five servers (two Controllers, two entertainment server and one supervisor) know the state of the other server through sharing a watchdog signal of 32 bytes that is sent every 1ms. Moreover, there are 4 cameras per wagon located at each door to enhance safety [63]. They send video signals directly to the Supervisor for safety monitoring purposes by the train driver. The reliability of the Supervisor is still assumed to be the highest as in the previous chapter to ensure it has the lowest probability of failure and, therefore the longest lifetime for comparison purposes.
Figure 30: Two wagon train model with Active Supervisor

![Diagram of Two Wagon Train Model with Active Supervisor](image)

**Active Supervisor**

Unlike in the previous chapter, the Supervisor acts as an **Active Supervisor**. This means that the supervisor acts as the primary backup to the Controllers (Ks) in each wagon. If the Controller in either wagon fails, the Supervisor handles the control load of the wagon immediately. Additionally, in case the Controller of the second wagon fails, the Supervisor in such case will handle the control load of both wagons.

The fault-tolerance relation between the Controllers in both wagons is no longer present, i.e., they no longer carry each other’s control load. Furthermore, the Entertainment Servers do not act as backups to the Controllers and do not handle any control load unlike the presented case in the previous chapter. The same conservative approach in regard to entertainment services is followed; if the Entertainment Server of any wagon fails, the entertainment services within that wagon are dropped due to the high safety requirements in train operations. However, the Entertainment Server does not drop the entertainment load to handle any control load.

As the Supervisor is the only backup node to the Controllers; the sensors send their data only to their corresponding Controller and to the Supervisor. So, for example, the sensors in $w_1$ only send their data to $K_1$ and the Supervisor only as these are the only nodes to handle the control load.
V.1.1. UNIQUE SIMULATED SCENARIOS IN ACTIVE SUPERVISOR

The same approach used with the passive supervisor network is used in this study to find the unique states that the network goes through. As the assumption that the Supervisor is the last to fail, therefore, there are only 24 possible failure paths. Each state represent the operational servers in the two wagon network. It is important to note here that since the entertainment server no longer handles any control load so the two states $E_1S_{-W_1}$ and $E_1S_{-W_2}$ now represent the same state which is $E_1S$. The same case is applicable for cases $E_2S_{-W_1}$ and $E_2S_{-W_2}$. The different paths that the network go through are shown in the below figures:

**Figure 31: Network Failure Paths in Active Supervisor**

![Network Failure Paths Diagram](image-url)
Figure 32: Network Failure Paths in Active Supervisor (Cont’d)
In the above paths, there are 16 different states (other than the failure state) that the network gets through, which are:

1. $K_1K_2E_1E_2S$
2. $K_2E_1E_2S$
3. $K_1E_1E_2S$
4. $K_1K_2E_2S$
5. $K_1K_2E_1S$
6. $E_1E_2S$
7. $K_2E_2S$
8. $K_2E_1S$
9. $K_1E_2S$
10. $K_1E_1S$
11. $K_1K_2S$
12. $K_2S$
13. $K_2S$
14. $E_1S$
15. $E_2S$
16. $S$

Since the two-wagons network model is symmetric, then several of these states are similar. For example, the state $K_1K_2E_2S$ and state $K_1K_2E_1S$ are similar where there is one wagon with failed controller and the other wagon is fault free. Consequently, there are 10 unique states/scenarios which had been analyzed using OPNET; any further scenario could be related to one of the analyzed scenarios. In all these scenarios, the Supervisor was running otherwise the whole network will fail since it was considered the last node to fail.

The scenarios are as follow (i and j represent the wagon number to be either 1 or 2 interchangeably):

- **Scenario 1**: The fault free state in which 4 servers are functioning ($K_iK_jE_iE_jS$)
- **Scenario 2**: One Controller fails ($W_j$) and the other 3 servers are up and running ($K_iE_iE_jS$). In this case the supervisor will carry the control load of $W_j$.
- **Scenario 3**: One Entertainment Server fails and the other 3 servers are up and running ($K_iK_jE_iS$). The entertainment load in $W_j$ is dropped.
- **Scenario 4**: The Controller and the Entertainment Server of the same wagon ($W_j$) fail ($K_iE_iS$). The Entertainment Server of the other wagon ($W_i$) will drop its entertainment load and the Supervisor will carry the control load of $W_i$.
- **Scenario 5**: The Controller of $W_j$ and the Entertainment Server of $W_i$ fail or vice versa ($K_iE_iS$). The control load of $W_j$ will be handled by the Supervisor.
- **Scenario 6**: The Entertainment Servers of both wagons fail ($K_iK_jS$). Then the entertainment load in both wagons is dropped and no change happen to the control load.
- **Scenario 7**: The Controllers of both wagons fail ($E_iE_jS$). The Supervisor carry the control load of both wagons.
• **Scenario 8:** The Entertainment Servers of both wagons fail and the Controller of one wagon (Wj) fails (K,S). The Supervisor will carry the control load of Wj while the Controller Ki will carry its own control load.

• **Scenario 9:** The Entertainment Server of one wagon fails (Wj) and its entertainment load is dropped. Also, both Controllers fail and the Supervisor will carry the control load of both wagons. The entertainment server in Wi is still handling its entertainment load (Ei,S).

• **Scenario 10:** All 4 servers fail, and left the Supervisor carrying the control load of both wagons (S).
V.1.1.1. **Simulation Results with Active Supervisor**

Adding an active fault tolerant hierarchal model at the Controller level is a method to further improve the network reliability. The main performance metric is still the control packet end-to-end delay for all the sensors and actuators. This delay should not exceed the corresponding sensor/actuator sampling period [37]. Additionally, the rate of packet loss has been monitored. The OPNET simulations have proved zero packet loss, i.e., there are no dropped or delayed packets in the system. A 95% confidence analysis is applied to all results. The delay for the door-cameras as well as the entertainment load (if present) have been observed and the results showed that they are always with the acceptable limits to avoid video flickering (in case of door-camera or video streaming) and to avoid user frustration due to long response time regarding the Wi-Fi services. Figures 33 to 35 show a sample of OPNET results from different scenarios.

As previously mentioned the X-axis represent the simulation time in minutes and the Y-axis represent the delay in seconds in the OPNET figures below. Also, the red dots indicate the end-to-end delay from the acting Controller/Server to the actuator node and the blue dots indicate the end-to-end delay from the sensor to the Controller. The total packet end-to-end delay is the sum of both delays.

**Figure 33: Active - KiKjEiEjS Control end-to-end delay – G2**
Figure 34: Active - KiKjS control end-to-end – G3

Figure 35: Active - End-to-end delay with only S operational (S) – G2
In the passive supervisor, there were 11 scenarios simulated while here, only 10 states are simulated. This is due to the fact that, in the passive supervisor case scenario EiS appeared twice as EiS_Wi and EiS_Wj. In the first case the Entertainment Server (Ei) was carrying the control load of Wagon Wi, while in the second case, it was carrying the control load of the other Wagon Wj. As the Entertainment Server in the active supervisor case does not handle any control load, consequently, both cases end up being identical. In the active case, the supervisor node carries the control load of both wagons in case of controllers’ failure while each entertainment server handles its own wagon entertainment load.

Table 4 illustrates the output of the OPNET simulations. The end-to-end delay for the 1ms sensors/actuators is found to be below 1ms and the end-to-end delay of the 16ms sensors and actuators is proved to be below 16ms. The presence of entertainment load is indicated in the below table with the wagon name that it is running in. As previously mentioned, the entertainment load per wagon consists of 60 video streaming and 60 Wi-Fi laptops running 4 different applications; HTTP, FTP, DB, and email. The presented delays include all processing, propagation, encapsulation and de-capsulation delays.

Table 4: Active Supervisor total end-to-end delays (µs)

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>Server handling Control load</th>
<th>Control End-to-End Delay (µs)</th>
<th>Entertainment Enabled in Wagon</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>W1i</td>
<td>W1j</td>
<td>G1</td>
</tr>
<tr>
<td>K,Ej,S</td>
<td>K</td>
<td>S</td>
<td>17.12</td>
</tr>
<tr>
<td>K,Ej,S</td>
<td>K</td>
<td>K</td>
<td>17.98</td>
</tr>
<tr>
<td>K,Ej,S</td>
<td>K</td>
<td>S</td>
<td>14.41</td>
</tr>
<tr>
<td>K,Ej,S</td>
<td>Ki</td>
<td>S</td>
<td>9.77</td>
</tr>
<tr>
<td>K,K,S</td>
<td>Ki</td>
<td>K</td>
<td>15.07</td>
</tr>
<tr>
<td>Ei,Ej,S</td>
<td>S</td>
<td>S</td>
<td>17.02</td>
</tr>
<tr>
<td>Ei,S</td>
<td>K</td>
<td>S</td>
<td>11.85</td>
</tr>
<tr>
<td>E,S</td>
<td>S</td>
<td>S</td>
<td>9.36</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>12.4</td>
</tr>
</tbody>
</table>
All the results were obtained after a 95% confidence analysis. The results shown represent the mean value of the maximum packet end-to-end delay obtained from all runs. The maximum deviation (Δ) from these means is 0.411µs. Furthermore, the delays for the door-cameras and the video streaming were below the acceptable delay requirements. As per [64], the OPNET results presented in this research are comparable to hardware implementation outcomes as the maximum transmission rate for the control load is 0.256 Mbps.
V.1.2. PASSIVE AND ACTIVE SUPERVISOR OUTCOMES

COMPARISON

In the network model with the active supervisor, the sensors send their data to their corresponding controller and the supervisor node only rather than sending 4 different streams to all Servers. Therefore, the delay is lower in the Active supervisor architecture in the fault-free scenario (\(K_iK_jE_iE_jS\)).

In other scenarios such as \(K_iE_iE_jS\), the Controller node (\(K_i\)) carries the control load of wagon \(W_i\) while the Entertainment Server (\(E_j\)) has dropped its entertainment load and is handling the control load of \(W_j\) in the passive supervisor mode. With the active supervisor, since \(E_j\) does not drop its entertainment load, \(S\) handles the control load of wagon \(W_j\). Hence, the passenger can still enjoy the on-board services and will not be affected by the failure that occurred.

Also, in case of \(E_iE_jS\) in the active model, the supervisor \(S\) handles the control load of both wagons but the entertainment services are still running in both wagons. In the passive supervisor case, each of the Entertainment Servers carries its own wagon control load after dropping its entertainment load. Therefore, the delay in the Active Supervisor case is somewhat higher when compared to the Passive supervisor case, due to the introduced latency caused by the video stream and Wi-Fi packets within the network fabric.

In scenario (\(S\)), the delay is the same in the active or passive models since all the entertainment is dropped in both cases and the sensors only send their data to the supervisor. This is verified by observing the forwarded traffic by the intermediate switch, in which the same amount of traffic (133.9Mbps) is forwarded in both cases.

The main benefit when comparing the active supervisor case to the passive supervisor case presented is that the passenger will not be aware of a failure except when the entertainment server of the wagon fails. This case is valid in scenarios \(K_iK_jE_iE_jS\), \(K_iE_iE_jS\) and \(E_iE_jS\); in these three states, the entertainment is functional in both wagons, other than when compared with the passive supervisor case. On the other hand, in the passive scenario, the entertainment is functional in both wagons in the fault-free scenario only. This means that when only one server fails, the passenger will be aware of the failure in case of the passive supervisor. Also, only one wagon will experience the failure of the entertainment in the active case in scenarios \(K_iK_jE_iS\), \(K_iE_iE_jS\) and \(E_iE_jS\).
K,E,S, K,E,S, and E,S. When comparing with the passive scenario, the passengers will enjoy the entertainment services in one wagon in scenarios K,E,E,S and K,E,E,S only.

Table 5 shows a comparison between the Active Supervisor architecture and the passive Supervisor architecture with respect to the number of states that have the entertainment enabled in either one or two wagons. Due to the symmetric nature of the network, the states: K,E,E,S, K,E,E,S, K,E,E,S, K,E,S, K,E,S and E,S are duplicated. Consequently, in the Active Supervisor architecture, the 10 states are expanded to 16 and, in the Passive Supervisor architecture; the 11 states are expanded to 18.

Note finally that, in the Active Supervisor architecture, when the controller of a wagon fails, its only backup is the Supervisor node. In the passive supervisor network model, for each failing controller, there are 4 other machines that act as backups.

Table 5: Number of states with enabled Entertainment

<table>
<thead>
<tr>
<th>Entertainment enabled in</th>
<th>Passive Supervisor</th>
<th>Active Supervisor</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Wagons</td>
<td>1/18 states</td>
<td>4/16 states</td>
</tr>
<tr>
<td>1 Wagon</td>
<td>4/18 states</td>
<td>8/16 states</td>
</tr>
</tbody>
</table>
V.2. MARKOV MODELING AND RELIABILITY ANALYSIS

COMPARISON

After proving that both network models, with a passive and active supervisor, meet the control end-to-end delay requirement, the important finding is which scenario is better performing. The way to discover the better performing scenario is through analyzing the reliability of both systems and comparing them.

In analyzing the reliability of both scenarios, it was important to note that the switching of the control tasks from a failed Server to another operational server is not always a successful process. Therefore, the probability of successful detection/reconfiguration, i.e. the coverage, is included in the Markov models [48, 49, 55, 56]. As previously mentioned, the coverage is defined as the proportion of faults from which a system can automatically recover [53]. Figure 36 shows the Markov model for the network model with passive supervisor.

Figure 36: Passive Supervisor - Markov Model
In the above model, each state indicates the operational servers within the network. Any state where the Supervisor fails before any other node is considered a failure state. This is because the train driver or observer cannot supervise the network behavior and therefore is in a failure state of supervising the network. Moreover, $\lambda_K$, $\lambda_E$ and $\lambda_S$ represent the failure rate of the Controller, Entertainment server and the supervisor respectively. The coverage $c$ appears in the transitions between states where the control function will be shifted from one failed server to another operational one. This transition will only occur if the recovery is successful and the operational controller is able to take over the tasks of the failed controller and handle the control function. If the recovery is not successful, the control function fails and the system moves to state F (i.e., the control function failure state).

The initial state is the fault-free (FF) state where all servers are operational. If one of the Controllers fails ($K_j$), the system moves to state $K_iE_iE_jS$ with a transition rate of $2c\lambda_K$. The factor of 2 is due to the fact that the failure of either $K_1$ or $K_2$ will lead to this state. In case one of the entertainment servers fails instead, the system moves to the $K_iK_jE_iS$ state at a rate of $2\lambda_E$. Here, the coverage is not included in the transition rate as there is no shifting of any control function. In case the supervisor fails or during the shift from the FF state to the $K_iE_iE_jS$ the recovery of the control function was unsuccessful, the system directly goes to the failure state at a transition rate of $2(1-c)\lambda_K+\lambda_S$.

In the state $K_iE_iE_jS$ when the remaining controller fails, the system transits to the $E_iE_jS$ at a rate of $c\lambda_K$. If $E_i$ fails, it moves to the $K_iE_jS$ state at a rate of $\lambda_E$, while if $E_j$ fails, the system transits to the $K_iE_iS$ state with a transition rate of $c\lambda_E$. Note here that incase of failure of $K_i$ or $E_j$ only the coverage is included as those transitions include shifting of the control function to $E_i$. If the control function is not reconfigured successfully or the supervisor fails, the system fails to the failure state at a rate of $(1-c)\lambda_K+(1-c)\lambda_E+\lambda_S$.

In the state $K_iK_jE_iS$ when the controller $K_i$ fails, the system transits to the $K_iE_iS$ at a rate of $c\lambda_K$. If $K_j$ fails, it moves to the $K_iE_iS$ state at a rate of $c\lambda_K$, while if $E_i$ fails, the system transits to the $K_iK_jS$ state with a transition rate of $\lambda_E$. Note here that incase of failure of $K_i$ or $K_j$ only the coverage is included as those transitions include shifting of the control function to $E_i$ or $E_j$ respectively. If the control function is not
reconfigured successfully or the supervisor fails, the system fails to the failure state at a rate of $2(1-c)\lambda_K + \lambda_S$.

In state $E_iE_jS$, $K_i$ and $K_j$ have failed and the control function has been shifted to $E_i$ and $E_j$ respectively. If either $E_i$ or $E_j$ fails, the model moves to state $E_iS_W_i$; the control of $W_i$ is handled by $E_i$ and the control load of $W_j$ is handled by $S$. The coverage affects this transition and therefore, the transition rate from $E_iE_jS$ to $E_iS_W_i$ to is $2c\lambda_E$; also, the transition from $E_iE_jS$ to failure is $2(1-c)\lambda_E + \lambda_S$.

In state $K_iK_jS$, $E_i$ and $E_j$ have failed but the control function has not been affected since $K_i$ and $K_j$ are both operational. If either $K_i$ or $K_j$ fails, the model moves to state $K_iS$; the control of failed controller is handled by $S$. The coverage affects this transition and therefore, the transition rate from $K_iK_jS$ to $K_iS$ is $2c\lambda_K$; also, the transition from $K_iK_jS$ to $F$ is $2(1-c)\lambda_K + \lambda_S$.

In the state $K_iE_jS$ when the controller $K_i$ fails, the system transits to the $E_jS_W_j$ at a rate of $c\lambda_K$. If $E_i$ fails, it moves to the $K_iS$ state at a rate of $c\lambda_E$. Note here that in case of failure of $K_i$ or $E_i$ only the coverage is included as those transitions include shifting of the control function to $S$. If the control function is not reconfigured successfully or the supervisor fails, the system fails to the failure state at a rate of $(1-c)\lambda_K + (1-c)\lambda_E + \lambda_S$.

Similarly, in the state $K_iE_jS$ when the controller $K_i$ fails, the system transits to the $E_iS_W_i$ at a rate of $c\lambda_K$. If $E_j$ fails, it moves to the $K_iS$ state at a rate of $c\lambda_E$. In case of failure of $K_i$ or $E_j$, the coverage is included as those transitions include shifting of the control function to $S$. If the control function is not reconfigured successfully or the supervisor fails, the system fails at a rate of $(1-c)\lambda_K + (1-c)\lambda_E + \lambda_S$.

In states $K_iS$, $E_iS_W_i$, and $E_iS_W_j$, when the operational server other than the supervisor fail, namely $K_i$ or $E_i$, the system moves to the $S$ state at a rate of $c\lambda_K$, $c\lambda_E$ or $c\lambda_E$ in that order. In case of non-recovery of the control function, or failure of $S$, the system fails with a rate of $(1-c)\lambda_K + 2\lambda_S$, $(1-c)\lambda_E + 2\lambda_S$ or $(1-c)\lambda_E + 2\lambda_S$ in the same order. When the system is in state $S$ and the supervisor fail it moves to the failure state at a rate of $\lambda_S$. 

64
The system can be described by the Chapman-Kolmogorov equations. The row vectors \( \hat{P}(t) \) and \( P(t) \) are:

\[
\dot{P}(t) = \begin{bmatrix}
\frac{dP_{K_1K_1}}{dt} & \frac{dP_{K_1K_2}}{dt} & \frac{dP_{K_1K_3}}{dt} & \frac{dP_{K_1K_4}}{dt} & \frac{dP_{K_1K_5}}{dt} & \frac{dP_{K_1K_6}}{dt} & \frac{dP_{K_1K_7}}{dt} & \frac{dP_{K_1K_8}}{dt} & \frac{dP_{K_1K_9}}{dt} \\
\frac{dP_{K_2K_1}}{dt} & \frac{dP_{K_2K_2}}{dt} & \frac{dP_{K_2K_3}}{dt} & \frac{dP_{K_2K_4}}{dt} & \frac{dP_{K_2K_5}}{dt} & \frac{dP_{K_2K_6}}{dt} & \frac{dP_{K_2K_7}}{dt} & \frac{dP_{K_2K_8}}{dt} & \frac{dP_{K_2K_9}}{dt} \\
\frac{dP_{K_3K_1}}{dt} & \frac{dP_{K_3K_2}}{dt} & \frac{dP_{K_3K_3}}{dt} & \frac{dP_{K_3K_4}}{dt} & \frac{dP_{K_3K_5}}{dt} & \frac{dP_{K_3K_6}}{dt} & \frac{dP_{K_3K_7}}{dt} & \frac{dP_{K_3K_8}}{dt} & \frac{dP_{K_3K_9}}{dt} \\
\frac{dP_{K_4K_1}}{dt} & \frac{dP_{K_4K_2}}{dt} & \frac{dP_{K_4K_3}}{dt} & \frac{dP_{K_4K_4}}{dt} & \frac{dP_{K_4K_5}}{dt} & \frac{dP_{K_4K_6}}{dt} & \frac{dP_{K_4K_7}}{dt} & \frac{dP_{K_4K_8}}{dt} & \frac{dP_{K_4K_9}}{dt} \\
\frac{dP_{K_5K_1}}{dt} & \frac{dP_{K_5K_2}}{dt} & \frac{dP_{K_5K_3}}{dt} & \frac{dP_{K_5K_4}}{dt} & \frac{dP_{K_5K_5}}{dt} & \frac{dP_{K_5K_6}}{dt} & \frac{dP_{K_5K_7}}{dt} & \frac{dP_{K_5K_8}}{dt} & \frac{dP_{K_5K_9}}{dt} \\
\frac{dP_{K_6K_1}}{dt} & \frac{dP_{K_6K_2}}{dt} & \frac{dP_{K_6K_3}}{dt} & \frac{dP_{K_6K_4}}{dt} & \frac{dP_{K_6K_5}}{dt} & \frac{dP_{K_6K_6}}{dt} & \frac{dP_{K_6K_7}}{dt} & \frac{dP_{K_6K_8}}{dt} & \frac{dP_{K_6K_9}}{dt} \\
\frac{dP_{K_7K_1}}{dt} & \frac{dP_{K_7K_2}}{dt} & \frac{dP_{K_7K_3}}{dt} & \frac{dP_{K_7K_4}}{dt} & \frac{dP_{K_7K_5}}{dt} & \frac{dP_{K_7K_6}}{dt} & \frac{dP_{K_7K_7}}{dt} & \frac{dP_{K_7K_8}}{dt} & \frac{dP_{K_7K_9}}{dt} \\
\frac{dP_{K_8K_1}}{dt} & \frac{dP_{K_8K_2}}{dt} & \frac{dP_{K_8K_3}}{dt} & \frac{dP_{K_8K_4}}{dt} & \frac{dP_{K_8K_5}}{dt} & \frac{dP_{K_8K_6}}{dt} & \frac{dP_{K_8K_7}}{dt} & \frac{dP_{K_8K_8}}{dt} & \frac{dP_{K_8K_9}}{dt} \\
\frac{dP_{K_9K_1}}{dt} & \frac{dP_{K_9K_2}}{dt} & \frac{dP_{K_9K_3}}{dt} & \frac{dP_{K_9K_4}}{dt} & \frac{dP_{K_9K_5}}{dt} & \frac{dP_{K_9K_6}}{dt} & \frac{dP_{K_9K_7}}{dt} & \frac{dP_{K_9K_8}}{dt} & \frac{dP_{K_9K_9}}{dt} \\
\end{bmatrix}
\]

\[
P(t) = \begin{bmatrix}
P_{K_1K_1} & P_{K_1K_2} & P_{K_1K_3} & P_{K_1K_4} & P_{K_1K_5} & P_{K_1K_6} & P_{K_1K_7} & P_{K_1K_8} & P_{K_1K_9} \\
P_{K_2K_1} & P_{K_2K_2} & P_{K_2K_3} & P_{K_2K_4} & P_{K_2K_5} & P_{K_2K_6} & P_{K_2K_7} & P_{K_2K_8} & P_{K_2K_9} \\
P_{K_3K_1} & P_{K_3K_2} & P_{K_3K_3} & P_{K_3K_4} & P_{K_3K_5} & P_{K_3K_6} & P_{K_3K_7} & P_{K_3K_8} & P_{K_3K_9} \\
P_{K_4K_1} & P_{K_4K_2} & P_{K_4K_3} & P_{K_4K_4} & P_{K_4K_5} & P_{K_4K_6} & P_{K_4K_7} & P_{K_4K_8} & P_{K_4K_9} \\
P_{K_5K_1} & P_{K_5K_2} & P_{K_5K_3} & P_{K_5K_4} & P_{K_5K_5} & P_{K_5K_6} & P_{K_5K_7} & P_{K_5K_8} & P_{K_5K_9} \\
P_{K_6K_1} & P_{K_6K_2} & P_{K_6K_3} & P_{K_6K_4} & P_{K_6K_5} & P_{K_6K_6} & P_{K_6K_7} & P_{K_6K_8} & P_{K_6K_9} \\
P_{K_7K_1} & P_{K_7K_2} & P_{K_7K_3} & P_{K_7K_4} & P_{K_7K_5} & P_{K_7K_6} & P_{K_7K_7} & P_{K_7K_8} & P_{K_7K_9} \\
P_{K_8K_1} & P_{K_8K_2} & P_{K_8K_3} & P_{K_8K_4} & P_{K_8K_5} & P_{K_8K_6} & P_{K_8K_7} & P_{K_8K_8} & P_{K_8K_9} \\
P_{K_9K_1} & P_{K_9K_2} & P_{K_9K_3} & P_{K_9K_4} & P_{K_9K_5} & P_{K_9K_6} & P_{K_9K_7} & P_{K_9K_8} & P_{K_9K_9} \\
\end{bmatrix}
\]

and the transition matrix \( T \) is:

\[
T = \begin{bmatrix}
-2(\lambda_E + \lambda_K) - \lambda_S & 2c\lambda_K & 2\lambda_E & 0 & 0 \\
0 & -(\lambda_K + 2\lambda_E + \lambda_S) & 0 & c\lambda_K & \lambda_E \\
0 & 0 & -(2\lambda_K + \lambda_E + \lambda_S) & 0 & 2c\lambda_K \\
0 & 0 & 0 & -2\lambda_E - \lambda_S & 0 \\
0 & 0 & 0 & 0 & -2\lambda_E - \lambda_S \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]


The active supervisor network Markov model is shown in figure 37.

Figure 37: Active Supervisor - Markov Model

Similar to the passive Markov model, each state indicates the operational servers within the network. Also, the same assumption is still valid; in case that the Supervisor fails before any other node; it is considered a failure state. Additionally, $\lambda_K$, $\lambda_E$ and $\lambda_S$ represent the failure rate of the Controller, Entertainment server and the supervisor respectively. Consistently, the coverage $c$ appears in the transitions between states where the control function will be shifted from one failed server to another operational one.

The initial state is the FF state where all servers are operational. If one of the Controllers fails ($K_i$), the system moves to state $K_iE_iE_jS$ with a transition rate of $2c\lambda_K$. The factor of 2 is due to the fact that the failure of either $K_1$ or $K_2$ will lead to this state. In case one of the entertainment servers fails instead, the system moves to the $K_iK_jE_iS$ state at a rate of $2\lambda_E$. Here, the coverage is not included in the transition rate.
as there is no shifting of any control function. In case the supervisor fails or during the shift from the FF state to the $K_iE_iE_jS$ the recovery of the control function was unsuccessful, the system directly goes to the failure state at a transition rate of $2(1-c)\lambda_K + \lambda_S$.

In the state $K_iE_iE_jS$ when the remaining controller fails, the system transits to the $E_iE_jS$ at a rate of $\lambda_K$. If $E_i$ fails, it moves to the $K_iE_jS$ state at a rate of $\lambda_E$, while if $E_j$ fails, the system transits to the $K_iE_iS$ state with a transition rate of $\lambda_E$. Note here that incase of failure of $K_i$ only the coverage is included as those transitions include shifting of the control function to $S$. If the control function is not reconfigured successfully or the supervisor fails, the system fails to the failure state at a rate of $(1-c)\lambda_K + \lambda_S$.

In state $E_iE_jS$, $K_i$ and $K_j$ have failed and the control function has been shifted to $S$. If either $E_i$ or $E_j$ fails, the model moves to state $E_iS$; the entertainment load is dropped in both cases with no effect on the control load handled by $S$. The coverage does not affect this transition and therefore, the transition rate from $E_iE_jS$ to $E_iS$ is $2\lambda_E$; also, the transition from $E_iE_jS$ to the failure state is $\lambda_S$.

In state $K_iK_jS$, $E_i$ and $E_j$ have failed but the control function has not been affected since $K_i$ and $K_j$ are both operational. If either $K_i$ or $K_j$ fails, the model moves to state $K_iS$; the control of failed controller is handled by $S$. The coverage affects this transition and therefore, the transition rate from $K_iK_jS$ to $K_iS$ is $2c\lambda_K$; also, the transition from $K_iK_jS$ to $F$ is $2(1-c)\lambda_K + \lambda_S$.

In the state $K_iE_iS$ when the controller $K_i$ fails, the system transits to the $E_iS$ at a rate of $2c\lambda_K$. If $E_i$ fails, it moves to the $K_iS$ state at a rate of $\lambda_E$. Note here that incase of failure of $K_i$ only the coverage is included as those transitions include shifting of the control function to $S$. If the control function is not reconfigured successfully or the supervisor fails, the system fails to the failure state at a rate of $(1-c)\lambda_K + \lambda_S$. 
Similarly, in the state $K_iE_jS$ when the controller $K_i$ fails, the system transits to the $E_iS$ state at a rate of $c\lambda_K$. If $E_j$ fails, it moves to the $K_jS$ state at a rate of $\lambda_E$. Only in case of failure of $K_i$, the coverage is included as those transitions include shifting of the control function to S. If the control function is not reconfigured successfully or the supervisor fails, the system fails at a rate of $(1-c)\lambda_K+\lambda_S$.

In states $K_iS$ and $E_iS$, when $K_i$ or $E_j$ fail, the system moves to the $S$ state at a rate of $c\lambda_K$ or $\lambda_E$ in that order. In case of non-recovery of the control function, or failure of $S$, the system fails with a rate of $(1-c)\lambda_K+\lambda_S$ or $\lambda_S$ respectively. When the system is in state S and the supervisor fail it moves to the failure state at a rate of $\lambda_S$.

The system can be described by the Chapman-Kolmogorov equations. The row vectors $P(t)$ and $\dot{P}(t)$ are:

$$
\dot{P}(t) = \begin{bmatrix}
\frac{dP_{K_iK_jE_iE_jS}}{dt} & \frac{dP_{K_iE_jE_jS}}{dt} & \frac{dP_{E_jE_jS}}{dt} & \frac{dP_{K_jE_jS}}{dt} & \frac{dP_{K_jE_jS}}{dt} & \frac{dP_{E_jE_jS}}{dt} & \frac{dP_{E_jS}}{dt} & \frac{dP_{E_jS}}{dt} & \frac{dF}{dt}
\end{bmatrix}
$$

$$
P(t) = \begin{bmatrix}
P_{K_iK_jE_iE_jS} & P_{K_iE_jE_jS} & P_{K_jE_jS} & dP_{E_jE_jS} & P_{K_jE_jS} & P_{K_jE_jS} & P_{K_jE_jS} & P_{E_jS} & P_{E_jS} & F
\end{bmatrix}
$$

and the transition matrix $T$ is:

$$
T = \begin{bmatrix}
-2(\lambda_E + \lambda_K) - \lambda_S & 2c\lambda_K & 2\lambda_E & 0 & 0 & 0 & 0 & 2(1-c)\lambda_K + \lambda_S \\
0 & -2(\lambda_K + \lambda_E + \lambda_S) & 0 & c\lambda_K & \lambda_E & 0 & 0 & (1-c)\lambda_K + \lambda_S \\
0 & 0 & -2(2\lambda_K + \lambda_E + \lambda_S) & 0 & c\lambda_K & \lambda_E & 0 & (1-c)\lambda_K + \lambda_S \\
0 & 0 & 0 & -2\lambda_E - \lambda_S & 0 & c\lambda_K & \lambda_E & 0 & (1-c)\lambda_K + \lambda_S \\
0 & 0 & 0 & 0 & 0 & -2\lambda_E - \lambda_S & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
\end{bmatrix}
$$
V.2.1. NETWORK RELIABILITY COMPARISON

The reliability with respect to time was obtained for both models using SHARPE software package [57]. SHARPE, (Symbolic Hierarchical Automated Reliability and Performance Evaluator) is a tool used for analyzing systems performance. It is used for modeling of system reliability and performability using Markov and semi-Markov reward models as well stochastic Petri nets [57]. A case study is presented next to illustrate the use of the Markov models in the determination of system reliability. Let the failure rates of the controllers and the entertainment servers be the same, i.e., \( \lambda_K = \lambda_E = \lambda = 1/\text{month} \). Moreover, as the Supervisor is assumed to fail last in both cases (passive and active modes), three different values for its failure rate were chosen for \( \lambda_s = 0.5, 0.8 \) and 1/month.

The coverage is determined by the user and any small mistake in the calculation of the coverage leads to false reliability estimations [49]. Moreover, if the coverage of a system decreases, system reliability is expected to decrease as well. Consequently, different values for the coverage were used specifically 95% and 90%.

Tables 6 and 7 show the output from SHARPE for the system reliability for all combinations of the coverage and \( \lambda_s \) for both passive and active supervisor.

Table 6: Reliability (t) for Passive Supervisor

<table>
<thead>
<tr>
<th>Time (Month)</th>
<th>( C=0.9 - \lambda_s=0.5 )</th>
<th>( C=0.95 - \lambda_s=0.5 )</th>
<th>( C=0.9 - \lambda_s=0.8 )</th>
<th>( C=0.95 - \lambda_s=0.8 )</th>
<th>( C=0.9 - \lambda_s=1 )</th>
<th>( C=0.95 - \lambda_s=1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.1</td>
<td>0.93</td>
<td>0.94</td>
<td>0.90</td>
<td>0.91</td>
<td>0.89</td>
<td>0.90</td>
</tr>
<tr>
<td>0.2</td>
<td>0.87</td>
<td>0.89</td>
<td>0.82</td>
<td>0.84</td>
<td>0.79</td>
<td>0.80</td>
</tr>
<tr>
<td>0.3</td>
<td>0.81</td>
<td>0.84</td>
<td>0.74</td>
<td>0.76</td>
<td>0.70</td>
<td>0.72</td>
</tr>
<tr>
<td>0.4</td>
<td>0.76</td>
<td>0.79</td>
<td>0.67</td>
<td>0.70</td>
<td>0.62</td>
<td>0.64</td>
</tr>
<tr>
<td>0.5</td>
<td>0.71</td>
<td>0.74</td>
<td>0.61</td>
<td>0.64</td>
<td>0.55</td>
<td>0.58</td>
</tr>
<tr>
<td>0.6</td>
<td>0.66</td>
<td>0.70</td>
<td>0.55</td>
<td>0.58</td>
<td>0.49</td>
<td>0.52</td>
</tr>
<tr>
<td>0.7</td>
<td>0.62</td>
<td>0.66</td>
<td>0.50</td>
<td>0.53</td>
<td>0.43</td>
<td>0.46</td>
</tr>
<tr>
<td>0.8</td>
<td>0.58</td>
<td>0.62</td>
<td>0.45</td>
<td>0.49</td>
<td>0.39</td>
<td>0.42</td>
</tr>
<tr>
<td>0.9</td>
<td>0.54</td>
<td>0.59</td>
<td>0.41</td>
<td>0.45</td>
<td>0.35</td>
<td>0.37</td>
</tr>
<tr>
<td>1</td>
<td>0.51</td>
<td>0.56</td>
<td>0.38</td>
<td>0.41</td>
<td>0.31</td>
<td>0.34</td>
</tr>
<tr>
<td>1.1</td>
<td>0.48</td>
<td>0.52</td>
<td>0.34</td>
<td>0.38</td>
<td>0.27</td>
<td>0.30</td>
</tr>
<tr>
<td>1.2</td>
<td>0.45</td>
<td>0.50</td>
<td>0.31</td>
<td>0.35</td>
<td>0.25</td>
<td>0.27</td>
</tr>
<tr>
<td>1.3</td>
<td>0.42</td>
<td>0.47</td>
<td>0.29</td>
<td>0.32</td>
<td>0.22</td>
<td>0.25</td>
</tr>
<tr>
<td>1.4</td>
<td>0.40</td>
<td>0.44</td>
<td>0.26</td>
<td>0.29</td>
<td>0.20</td>
<td>0.22</td>
</tr>
<tr>
<td>1.5</td>
<td>0.37</td>
<td>0.42</td>
<td>0.24</td>
<td>0.27</td>
<td>0.18</td>
<td>0.20</td>
</tr>
<tr>
<td>1.6</td>
<td>0.35</td>
<td>0.40</td>
<td>0.22</td>
<td>0.25</td>
<td>0.16</td>
<td>0.18</td>
</tr>
<tr>
<td>1.7</td>
<td>0.33</td>
<td>0.38</td>
<td>0.20</td>
<td>0.23</td>
<td>0.14</td>
<td>0.16</td>
</tr>
<tr>
<td>1.8</td>
<td>0.31</td>
<td>0.36</td>
<td>0.18</td>
<td>0.21</td>
<td>0.13</td>
<td>0.15</td>
</tr>
<tr>
<td>1.9</td>
<td>0.30</td>
<td>0.34</td>
<td>0.17</td>
<td>0.19</td>
<td>0.11</td>
<td>0.13</td>
</tr>
<tr>
<td>2</td>
<td>0.28</td>
<td>0.32</td>
<td>0.15</td>
<td>0.18</td>
<td>0.10</td>
<td>0.12</td>
</tr>
</tbody>
</table>
Table 7: Reliability (t) for Active Supervisor

<table>
<thead>
<tr>
<th>Time (Month)</th>
<th>Active Supervisor Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C=0.9 - λ_s=0.5</td>
</tr>
<tr>
<td>0</td>
<td>1.00</td>
</tr>
<tr>
<td>0.1</td>
<td>0.93</td>
</tr>
<tr>
<td>0.2</td>
<td>0.87</td>
</tr>
<tr>
<td>0.3</td>
<td>0.82</td>
</tr>
<tr>
<td>0.4</td>
<td>0.77</td>
</tr>
<tr>
<td>0.5</td>
<td>0.72</td>
</tr>
<tr>
<td>0.6</td>
<td>0.68</td>
</tr>
<tr>
<td>0.7</td>
<td>0.64</td>
</tr>
<tr>
<td>0.8</td>
<td>0.60</td>
</tr>
<tr>
<td>0.9</td>
<td>0.56</td>
</tr>
<tr>
<td>1</td>
<td>0.53</td>
</tr>
<tr>
<td>1.1</td>
<td>0.50</td>
</tr>
<tr>
<td>1.2</td>
<td>0.47</td>
</tr>
<tr>
<td>1.3</td>
<td>0.45</td>
</tr>
<tr>
<td>1.4</td>
<td>0.42</td>
</tr>
<tr>
<td>1.5</td>
<td>0.40</td>
</tr>
<tr>
<td>1.6</td>
<td>0.38</td>
</tr>
<tr>
<td>1.7</td>
<td>0.36</td>
</tr>
<tr>
<td>1.8</td>
<td>0.34</td>
</tr>
<tr>
<td>1.9</td>
<td>0.32</td>
</tr>
<tr>
<td>2</td>
<td>0.31</td>
</tr>
</tbody>
</table>

As shown in Figures 38 through 43, the active supervisor model consistently shows better system reliability when compared with the passive supervisor model. System reliability is then compared for both supervisory models at time t=1 month. Using the active model improves system reliability by 4.9% for c=90% and by 2.4% for c=95% for all values of λ_s at time t=1 month. When taking λ_s = λ_K = λ_E =1/month for comparison purposes, the active supervisor model is still performing better than the passive supervisor by a difference of 1.5% for c=90% and a difference of 0.8% only for c=95%. This is due to the fact that, as the failure rate of the supervisor increases, both systems move to the failure state faster. Taking the coverage of 90% with λ_s=0.5/month, the active model has a better reliability by a difference of 2.5%; while for c=95%, the active model reliability is performing better by a difference of 1.4%. With λ_s=0.8/month, the active model has a better reliability by a difference of 1.8%; while for c=95% the active model reliability is better performing by a difference of 1.0%. It is noted that for the same λ_s, the difference in the reliability between the
active and passive models becomes smaller as the coverage increases. This is due to
the fact that the coverage affect more transitions in the passive model than in the
active model.

Figure 38: Reliability - Active vs. Passive - $C=0.9 \cdot \lambda_s=0.5$

![Reliability - Active vs. Passive - C=90% - $\lambda_s=0.5$](image)

Figure 39: Reliability - Active vs. Passive - $C=0.95 \cdot \lambda_s=0.5$

![Reliability - Active vs. Passive - C=95% - $\lambda_s=0.5$](image)
Figure 40: Reliability - Active vs. Passive - C=0.9 - $\lambda_s=0.8$

Figure 41: Reliability - Active vs. Passive - C=0.95 - $\lambda_s=0.8$
Figure 42: Reliability - Active vs. Passive - $C=0.9$ - $\lambda_s=1$

![Reliability - Active vs. Passive - $C=0.9$ - $\lambda_s=1$](image)

Figure 43: Reliability - Active vs. Passive - $C=0.95$ - $\lambda_s=1$

![Reliability - Active vs. Passive - $C=0.95$ - $\lambda_s=1$](image)
It is noticed that decreasing $\lambda_s$ has a bigger impact on system reliability than the effect of decreasing the coverage. For example, system reliability improves when using $\lambda_s = 0.5$/month compared to $\lambda_s = 0.8$/month for $c=90\%$ by $26\%$ for the active and passive models. However, the reliability only improves by $6.9\%$ for the active model and by $9.6\%$ for the passive model when using $c=95\%$ rather than using $c=90\%$ for time $t = 1$month and any $\lambda_s$.

Therefore, to conclude the finding of this Markov modeling and reliability analysis for this specific case study, the active supervisor model is more reliable than the passive supervisor model.

The main advantage of the passive supervisor appears when observing the load on the supervisor. For example, in the FF state $K_iK_jE_iE_jS$, the passive supervisor receives 240 packets per second from the door-cameras and sends zero packets per second but when using the active mode, it receives 76490 packets per second for both control packets and door-camera packets so it handles more load. The below table illustrates the traffic handled by the supervisor in each state for both the passive and active models. It can be noticed that, in the passive model, the supervisor handles less load in most states which is expected to decrease its probability of failure. Moreover, an additional advantage for the passive model would be when the supervisor fails before any of the other four servers. For example, if the supervisor fails after the failure of $K_i$, in the active model, this means that the control operation in $W_i$ totally fails. However, in the passive model, the control load of $W_i$ will be handled by $E_i$ and the control functionality can still be operational but the door-cameras as well as the supervision functionality will be lost.
Table 8: Traffic handled by Supervisor in each state

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>Passive Model Traffic (Packets/Sec)</th>
<th>Active Model Traffic (Packets/Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Received</td>
<td>Sent</td>
</tr>
<tr>
<td>K_i, K_j, E_i, E_j, S</td>
<td>240</td>
<td>0</td>
</tr>
<tr>
<td>K_i, E_i, E_j, S</td>
<td>240</td>
<td>0</td>
</tr>
<tr>
<td>K_i, K_j, E_i, S</td>
<td>240</td>
<td>0</td>
</tr>
<tr>
<td>K_i, E_i, S</td>
<td>76490</td>
<td>0</td>
</tr>
<tr>
<td>K_i, E_j, S</td>
<td>76490</td>
<td>0</td>
</tr>
<tr>
<td>K_i, K_j, S</td>
<td>76490</td>
<td>0</td>
</tr>
<tr>
<td>E_i, E_j, S</td>
<td>76490</td>
<td>0</td>
</tr>
<tr>
<td>K_i, S</td>
<td>76490</td>
<td>67500</td>
</tr>
<tr>
<td>E_i, S, W_i / E_i, S</td>
<td>76490</td>
<td>67500</td>
</tr>
<tr>
<td>E_i, S, W_j / E_i, S</td>
<td>76490</td>
<td>67500</td>
</tr>
<tr>
<td>S</td>
<td>76490</td>
<td>67500</td>
</tr>
</tbody>
</table>
VI. Conclusions

In the field of Networked Control Systems (NCS), Ethernet is a strongly emerging technology. Even though Ethernet is a non-deterministic network, recent research has shown that it could be used in the context of control networks in industrial and automotive environments. Furthermore, Ethernet was shown to be successful in the implementation of a Networked Control System connecting sensors, controllers and actuators on a train wagon together with entertainment loads (wired and wireless). In this research, two of the one-wagon train models were combined. Furthermore, entertainment server was added in each wagon. OPNET simulations showed that control packet end-to-end delay requirements are successfully met while zero packets were dropped. This end-to-end delay (from sensor to actuator through the controller) included all processing, propagation and queuing delay. One advantage of such architecture is its fault-tolerant aspect. OPNET simulations further showed that one controller could handle the control payload of two wagons. Consequently, the system could tolerate the failure of up to three servers.

An analysis was then performed to compare the reliability of a one-wagon train network with the two-wagon train model discussed in this research; it is shown that there was a significant improvement in reliability. Reliability was expected to increase because controller failures did not necessarily cause system failure. However, the error detection and system reconfiguration, needed to be successful in order to improve the reliability. The coverage parameter describes the probability of successful error detection and reconfiguration. The effect of the coverage on the reliability analysis was shown. Furthermore, a novel fault-tolerant reliability scheme for the two-wagon train model was developed. This scheme aimed at increasing the reliability of the control function in the presence of the coverage parameter. A Markov model was then used to calculate the modified system reliability. This reliability is then compared to the reliability previously obtained for the two-wagon fault-tolerant scheme with coverage. It is proven that the proposed scheme had a higher reliability. All results were compared to estimates produced by the SHARPE software package and were found to be identical.

Further in the research, a new Hybrid model for the network while adding door-cameras to enhance passenger safety was studied. This model had a hierarchical
architecture and included a passive supervisor. OPNET simulations proved that the functionality of this model in the fault free condition. A fault-tolerance scheme is then investigated to prevent system failure in case of one or more Controller/Entertainment Server fail. A greater importance was given to the Control load over the Entertainment load to ensure train functionality. Different scenarios have been simulated with OPNET to mimic the different unique network states. In both the fault free and faulty scenarios, the simulations showed that the control packet end-to-end delay is within acceptable limits with zero control packet losses. The network will still be functioning after the failure of both Controllers and both Entertainment Servers.

A new role was then defined for the supervisor. As soon as either Controller fails, it acts as a backup for the failed Controller and handles its control load; therefore, it became an active node. For safety purposes, no other node acted as backup for any failed Entertainment Server; the entertainment was dropped when the Entertainment server failed. All possible combinations of operational Servers/Controllers were simulated using OPNET. It was shown that the control packet end-to-end delays met the control requirements and that no packet was dropped. The network was proven to function properly even after the failure of all Controllers and Entertainment Servers; the Supervisor was able to successfully carry the control load of both wagons. It was also shown that this architecture has the advantage of keeping entertainment services operational for a longer period when compared to other hierarchical architectures in the literature.

Markov models were formulated for the network model with passive and active supervisors. The system reliability was then found using SHARPE for different recovery and reconfiguration coverage and for different failure rate of the supervisor node. These models could be used as a design or observation tool for the system to determine the supervisor to be used based on its failure rate to maintain a certain system reliability. It has been found, for a specific case study, that the network reliability is always better when using active supervisor rather than passive supervisor even when the supervisor rate is the same as the controller and entertainment server failure rates. The main edge for the passive model appears when noticing the load on the Supervisor node as its is much less in the passive supervisory mode than in the active mode.
APPENDIX – CONFIDENCE ANALYSIS

All results subjected to a confidence analysis follow the following calculations.

Let:

- $X$: random variable (maximum end-to-end delay).
- $\mu$: Average of $X$
- $\sigma^2$: Variance of $X$
- $X_i$: sample of $X$ obtained during $i^{th}$ OPNET simulation (using different seed)
- $n$: No. of OPNET simulations
- $x$: Sample mean
- $s^2$: Sample variance

\[ x = \frac{1}{n} \sum_{i=1}^{n} X_i \]  
(1)

\[ s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (X_i - x)^2 \]  
(2)

In OPNET Network Modeler, a ‘seed’ value is required. This seed is used to initialize different random number generator equations. These equations are used to simulate the different behavior of non-deterministic aspects. Based on the Central Limit Theorem (CLT), if the distribution of a random variable is unknown, the distribution of its sample mean will approach a normal distribution, as the number of samples increases. The sample mean also approaches the ensemble mean and the variance of the sample mean is a scaled version of the ensemble mean (mean of $x = \mu$ = mean of $X$ and variance of $x = \sigma_x^2 = \frac{\sigma^2}{n}$ where $\sigma^2$ = variance of $X$ [32, 49].

Therefore, the confidence level is defined as the probability that $x$ is below a certain distance from $\mu$:

\[ z = \frac{x - \mu}{\sigma_x} \]  
(3)
$z$: is a normal random variable (mean= 0 & variance = 1).

$$P(-z_\alpha < z < z_\alpha) = \alpha$$  \hspace{1cm} (4)

$$P \left[ \left| \frac{x - \mu}{\sigma_x} \right| < z_\alpha \right] = \alpha$$  \hspace{1cm} (5)

By using 33 simulations, $n > 30$ and hence the sample standard deviation $s$ can be used instead of $\sigma$ as it is difficult to find $\sigma_x = \frac{\sigma}{\sqrt{n}}$. The Normal distribution will be used and $z_\alpha$ is calculated for a confidence level $\alpha = 95\%$. 
REFERENCES


[31] IEEE 802.3 Standard.


