



# An Approach to Improve the Availability of a Traffic Light System

**Olajide Blessing Olajide<sup>1,\*</sup>, Oke Alice Olufunke<sup>2</sup>, Odeniyi Olufemi Ayodeji<sup>3</sup>,  
Olabiyisi Stephen Olatunde<sup>4</sup>, Adeosun Olusegun Olajide<sup>4</sup>**

<sup>1</sup>Department of Computer Engineering, Federal University, Wukari, Nigeria

<sup>2</sup>Department of Computer Engineering, Ladoké Akintola University of Technology, Ogbomoso, Nigeria

<sup>3</sup>Department of Computer Science, Osun State College of Technology, Esa Oke, Nigeria

<sup>4</sup>Department of Computer Science, Ladoké Akintola University of Technology, Ogbomoso, Nigeria

## Email address:

olajideblessing55@gmail.com (O. B. Olajide)

\*Corresponding author

## To cite this article:

Olajide Blessing Olajide, Oke Alice Olufunke, Odeniyi Olufemi Ayodeji, Olabiyisi Stephen Olatunde, Adeosun Olusegun Olajide. An Approach to Improve the Availability of a Traffic Light System. *International Journal of Intelligent Information Systems*. Vol. 10, No. 4, 2021, pp. 37-43. doi: 10.11648/j.ijis.20211004.11

**Received:** April 13, 2021; **Accepted:** April 30, 2021; **Published:** July 6, 2021

---

**Abstract:** Traffic Light System (TLS) is a standalone safety-critical infrastructure that is used to avert traffic congestion and accidents at a road intersection. It is pertinent that its service must be dependable because any failure could result to loss of lives or resources. The existing fail-safe TLS often experience downtime as a result of inevitable fault developed frequently by its Traffic Light Controller Unit (TLCU) due to harsh weather and other environmental factors exposed to on the roads. Hence, the need for a fault-tolerant TLS that will optimize TLS service delivery even at the event of a faulty TLCU initiated this work. In developing the fault-tolerant TLS, three TLCUs were interfaced using the concept of triple modular redundancy architecture. A disagreement detector was configured to test the viability of the primary TLCU using stationarity process. Markovian process was used to switch a faulty primary TLCU to a good one using majority voter mechanism. The fault-tolerant TLS and existing TLS were simulated using MATLAB R2015a. The performance of the fault-tolerant TLS was evaluated by comparing with that of existing TLS using availability as performance metric. The simulation results revealed that the fault-tolerant TLS yielded 99.9474% availability while simulation results of the existing TLS yielded 97.6199% availability. This work has therefore developed a fault-tolerant TLS that performed better than the existing fail-safe TLS.

**Keywords:** TLS, TLCU, Fail-Safe, Fault-Tolerant, Markovian, Stationarity

---

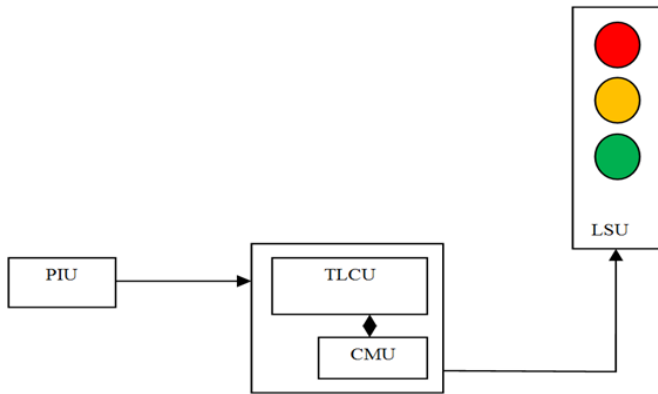
## 1. Introduction

Traffic Light System (TLS) is one of the vital public facilities that play important role to control traffic flows at busy road intersections. It consists of three parts: Light Signal Unit (LSU), Traffic Light Controller Unit (TLCU) and Power and Input Unit (PIU) Udoakah [1]. The TLS is as illustrated in Figure 1.

TLS is a standalone application automated to work independently without the help of any traffic warding officer. Every system is vulnerable to failure and TLS being a system may at times develop fault(s) leading to its failure. Failures of TLS embedded systems are exhibited as TLS downtime

where TLS display all red light flashing or no display at all Sivarao [2]. These failures are caused by the malfunctioning of the embedded TLCU Salami [3]. The role played by the TLS on the roads makes its downtime a situation not healthy for traffic control on major roads. Therefore, an important requirement in TLS is that it should be highly dependable. TLS should have autonomous response and reconfiguration in the presence of components failure so as to provide non-stop services to users. Existing TLS are void of proactive scheme to escape TLS downtime rather only deliver a fail-safe design which has the drawback of unnecessary delay and accidents. The existing fail-safe TLS design comprises the inclusion of a Conflict

Monitor Unit CMU. The CMU monitors the output of the single TLCU and compare it with the expected pre-programmed output. If it discovers any fault in the TLCU, the CMU uses flash transfer relays to put the intersection to flash with all red lights flashing rather than displaying a potential hazardous combination of signals. With this approach it is assumed that the TLS has aspired to fail-safe Latha [4]. Hence to mitigate this problem this work explore fault-tolerant design scheme to enhance the availability of the TLS. Fault-tolerance is the art and science of building systems that continue to operate satisfactorily in the presence of faults Paoli [5].



**Figure 1.** Existing TLS Model (Source: Udoakah and Okure, 2017).

Fault tolerance is a substantial design criterion for critical systems like TLS where the availability of hardware is crucial. Among the numerous fault tolerance design criteria are; Dual Modular Redundancy, Standby replacement and Triple Modular Redundancy (TMR). TMR is the most applied fault masking technique for fault tolerance of software or hardware system Alagoz [6]. These architectural design operations are supported by some algorithm and strategies such as markovian processes and stationarity processes Jaroslaw [7]. A Markov process is a technique used for modelling the states a system can assume in a process and the possible transitions between them. It is widely useful for dependability analysis of complex fault tolerant systems Jaroslaw [7]. The Markov process operates in the following ways: the system is envisioned as being in one of the states at all times throughout the period of interest. The system can be in only one state at a time, and from time to time it makes transition from one state to another state by following one of the set of inter-state transitions Jaroslaw [7]. The concept of stationarity is a mathematical idea constructed to simplify the theoretical and practical development of stochastic processes. To design a proper model, adequate for future forecasting, the underlying time series is expected to be stationary. A stationary process is one whose statistical properties do not change over time Hipel [8].

Though TLS failure is inevitable, this work develops an intelligent fault management technique to improve TLS availability thereby increasing its operation life span. The proposed fault-tolerant TLS model was developed and simulated using MATLAB R2015a. The performance of the

developed fault-tolerant TLS was evaluated and compared with the performance of the existing TLS using availability as a performance metric. Availability is The availability  $A$  of a system is defined as the probability that the system is operating correctly at instant  $t$ , and  $A(t)$  is expressed as equation 1.

$$A(t) = \frac{MTTF}{MTTF + MTTR} \quad (1)$$

The MTTF represents the length of time the system is expected to last in operation until it fails. The MTTF is commonly referred to as the life-time of any product and is expressed as:

$$MTTF = \frac{\text{totaltimeofoperationofsystem}}{\text{totalnumberofembeddedsystems}} \quad (2)$$

The MTTR of a system is used to refer to the time required to repair a system and restored it to full functionality and is expressed as:

$$MTTR = \frac{\text{totalmaintancetimeofsystem}}{\text{totalnumberofrepairsonembeddedsystem}} \quad (3)$$

## 2. Related Works

Rodney [9] designed an Intelligent Machine Controller (IMC) architecture using an IMC nodes, a system coordinator and a real time coordinator. The design did not cater for auto repair of system coordinator. Shalangwa [10] designed an automated traffic light controller using 12 volts automated solar energy power supply, 555 timer connected in astable mode, decade counter, relay circuit and timing sequence selector for red, green, amber and yellow light. The limitation of this system is that its design employed a fail-stop methodology. Sivarao [2] designed a prototype of traffic light electrical and mechanical fault detector system using three modules; the electrical fault detection system, the mechanical fault detection system and the conventional TLS. Dauda [11] developed a TLS that include traffic density detection and signal adjustment system using of five units which are power supply unit, traffic density detection unit, signal adjustment unit, microcontroller unit and display unit. The system did not have any mechanism for handling faults in its constituent component during the TLS operation. Khelassi [12] proposes an over-actuated controller based on reliability analysis and experimented it on linearized aircraft model. The work was further compared to the existing allocation strategy for self actuated controllers to examine their performances. It was concluded that their strategy guarantees the distribution of the desired effort with a high overall reliability. Sparsh [13] provides survey of architectural strategies for improving resilience in computing systems. Their work further advocates some suitable techniques for non volatile memory and 3D-stack processors. Gauri [14] Carried out analysis on the effect of redundancy on queues, delay and speed up of contents download from cloud storage system using fork-join model. The analysis provides practical insights into how many users can access a piece of content simultaneously, and

how fast they can be served. The study recommends that the techniques and insights are applicable to other systems with stochastically varying components.

Existing TLS designs identified in literatures have shown a fail-safe design. A design that permits the system to fail while ensuring that the users are alerted of the failure events to enable them to seek other alternatives if there is any. This further shows that with the fail safe design the downtime experience as a result of system's component failure is unavoidable. The search to minimize the downtime experienced frequently by existing TLS initiated this work.

### 3. Methodology

The developed Fault-tolerant TLS Controller is made up of three units; the Power and Input Unit (PIU), three Traffic Light Controller Unit (TLCU), and Light Signal Unit (LSU) as shown in Figure 2. Three functionally identical TLCUs were deployed in parallel, namely TLCU 1, TLCU 2 and TLCU 3 as shown in Figure 3. The disagreement detector is used to keep record of the possible combination of output from the TLCU and how they are mapped to the output of the majority voter. That is when an active TLCU fails; disagreement detector and majority voter does the work of detecting and switching to available TLCU spare to replace the failed TLCU, thereby restoring the TLS to its operational state. These form the BFS architecture for the fault-tolerant TLS that provide each TLCU the ability to be informed about the status of the immediately adjacent TLCU (direct connection) and if a TLCU fails, the overall system and the structure remain operative, because a connection to the next-but-one module always remains.

#### 3.1. Markov's Assumption for the TLCU Module of the Fault-Tolerant TLS

This work adopts Markov's simplifying assumption which

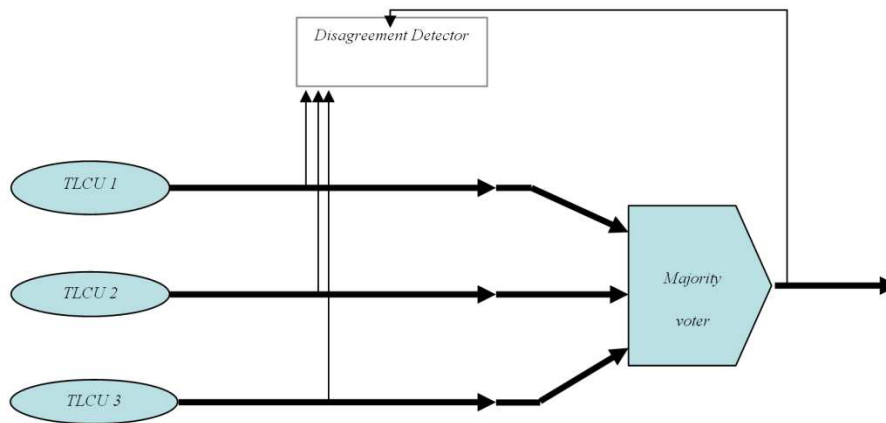


Figure 3. The TMR Architecture of the TLCU.

The conditional probability of transiting from one TLCU to another within the TMR scheme is given as;

$$P\{X_{t+1} = j | X_t = i\} \quad (5)$$

(c) Stationary transition probabilities for each  $i$  and  $j$ ,

assumes that the state to assign next depends only on the present state assignment; it means that the TLCU to assign next depends only on the present status of the primary TLCU. This will thereby exclude all previous check experiences of the TLCU module as a factor in determining the next TLCU to be assigned as primary TLCU. The TMR scheme exhibits the following attributes:

(a) Markovian property;

The TLCU process  $\{X_t\}$  have the Markovian Property;

$$P\{X_{t+1} = j | X_0 = k_0, X_1 = k_1, \dots, X_{t-1} = k_{t-1}, X_t = i\} = P\{X_{t+1} = j | X_t = i\} \quad (4)$$

For  $t = 0, 1, \dots$  and every sequence  $i, j, k_0, k_1, \dots, k_{t-1}$ .

This Markovian property (4) is equivalent to stating that the conditional probability of any "failure and switching event in the TLCU module, given any past event and the present state of the TLCU module  $X_t = i$ , is independent of the past event and depends only upon the present state.

(b) Transition probabilities

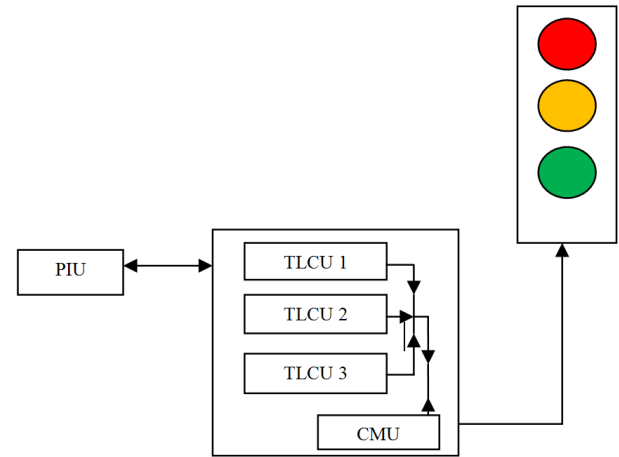


Figure 2. The Block Diagram of the Fault-tolerant TLS.

$$P\{X_{t+1} = j | X_t = i\} = P\{X_1 = j | X_0 = i\} \text{ for all } t = 0, 1, \dots \quad (6)$$

then the (one-step) transition probabilities are said to be stationary and are denoted by  $P_{ij}$ . Thus, having stationary transition probabilities; (6) implies that the transition

probabilities do not change in time. The existence of stationary (one-step) transition probabilities also implies that, for each  $i, j$ , and  $n$  ( $n = 0, 1, 2, \dots$ ),

$$P\{X_{t+n} = j | X_t = i\} = P\{X_n = j | X_0 = i\} \text{ for all } t = 0, 1, \dots \quad (7)$$

These conditional probabilities are denoted by  $P_{ij}^{(n)}$ . Thus,  $P_{ij}^{(n)}$  is just the conditional probability that the random variable  $X$ , starting in state  $i$ , will be in state  $j$  after exactly  $n$  steps. Because  $P_{ij}^{(n)}$  is conditional probability, it will be nonnegative and since the process must make a transition into some state, it must satisfy the properties

$$P_{ij}^{(n)} \geq 0 \text{ for all } i \text{ and } j; n = 0, 1, 2 \quad (8)$$

and

$$\sum_j P_{ij}^{(n)} = 1 \text{ for all } i; n = 0, 1, 2 \quad (9)$$

### 3.2. Stationarity Process for the TLCU Module of the Fault-Tolerant TLS

This research assumed that transition probabilities do not change with the passage of time. The term used to describe this assumption is stationarity Kalla [15]. The matrix  $P$  denote the one-step transition probabilities for any time. The stationary transition processes are explained as follows:

$$P\{X_0 = 1 \rightarrow X_1 = 3 \rightarrow X_2 = 1\} = P_{13}P_{31} \quad (10)$$

In equation 10, the two-step path probability is given by the product of the two one-step transition probabilities comparing the path. Finding the corresponding probabilities for the other two paths, and collecting results gives

$$P\{X_0 = 1 \rightarrow X_2 = 1\} = P_{11}P_{11} + P_{12}P_{21} + P_{13}P_{31} \quad (11)$$

Equation 11 describes the probability that TLCU 1 is still the one declared active (primary controller) after a failure is noticed, given that TLCU 1 was the immediate past controller (TLCU) used. By similar logic, the probabilities for using TLCU 2 and TLCU 3 may be obtained at a notice of failure as indicated by equations 12 and 13, given that initial active controller was TLCU 1.

$$P\{X_0 = 1 \rightarrow X_2 = 2\} = P_{11}P_{12} + P_{12}P_{22} + P_{13}P_{32} \quad (12)$$

$$P\{X_0 = 1 \rightarrow X_2 = 3\} = P_{11}P_{13} + P_{12}P_{23} + P_{13}P_{33} \quad (13)$$

The three equations (11, 12 and 13) together give the probability distribution of the random variable  $X_2$  given that  $X_0 = 1$ . If initial primary controller is TLCU 2, three different expressions are got for the distribution of  $X_2$ , and three more for when TLCU 3 is used for the initial TLS service. There are total of nine expressions required to specify the distribution of  $X_2$  under the various possible initial conditions. As it was for  $X_2$ , it is convenient to use a matrix arrangement. Let  $P^{(2)}$  denote the matrix of these probabilities, and let  $P_{ij}^{(2)}$  represent the  $ij$ th entry.

$$P^{(2)} = \begin{bmatrix} P_{11}^{(2)} & P_{12}^{(2)} & P_{13}^{(2)} \\ P_{21}^{(2)} & P_{22}^{(2)} & P_{23}^{(2)} \\ P_{31}^{(2)} & P_{32}^{(2)} & P_{33}^{(2)} \end{bmatrix} \quad (14)$$

$P^{(2)}$  is called the two-step transition matrix; its elements, two-step transition probabilities. They give conditional probabilities for the states at time 2 under varying possible conditions for the state at time zero. In equation 14, each row is a probability distribution. Loosely speaking,  $P_{ij}^{(2)}$  is the probability of “going” from state  $i$  to state  $j$  in two steps. It is also the probability which previously designated as  $P\{X_0 = I \rightarrow X_2 = j\}$ . Equation 15 express  $P_{ij}^{(2)}$  in terms of the one-step transition probabilities thus:

$$P_{11}^{(2)} = P_{11}P_{11} + P_{12}P_{21} + P_{13}P_{31} \quad (15)$$

Thus, all the ways that the event could occur were considered.

### 3.3. Simulation

Simulink models of MATLAB R2015a were developed and simulated to investigate the performance of the Fault-tolerant TLS as well as the existing fail-safe TLS design. The model in Figure 4 depicts the existing fail-safe TLS design and Figure 5 depicts the fault-tolerant TLS model presented in this research. The models consist of some blocks that perform specific functions. At the start of the Simulink model run, the Smat block takes inputs that are specified by the matlab logic in its workspace to initialize the model, the constant block then displays health status of the TLCUs, the output of the TLCU is then converged to the Triple Modular Redundancy block. This block will send signal to the Data Log block to save the output of specified array from the simulink model. It also sends another signal through the compare to zero block to display the current active TLCU out of the three TLCUs. This same signal is sent to the Light Signal Unitblock that controls the Vlight blocks to display the colour that reflects a distinct input value. Vlight blocks displays green or red light which indicates movement permission or movement denial respectively on the vertical or horizontal lane where it's displayed.

Each of the two TLS models is simulated for ten different runs. At each run instance the models were run till total failure states is achieved in order to elicit comparative analysis of both. The LSU displayed all red lights flashing at total failure for both models through the CMU. In other words when the single TLCU of the Fail-safe TLS model fails the model was made to display all red lights as a fail-safe method. In the same manner, Fault-tolerant TLS model displays all red lights flashing when the last TLCU of the three TLCU fails after two switching in the TMR architecture. At every state a TLCU (module) is either set to ON or OFF automatically through the help of the simulation software logic. If a module is in the ON (1) state, it implies that it is connected to the entire system and its signal is seen by the Voter. On the other hand, if the module is in the DOWN (0)

state, it implies that it is disconnected from the entire system and its signal is not seen by the Voter.

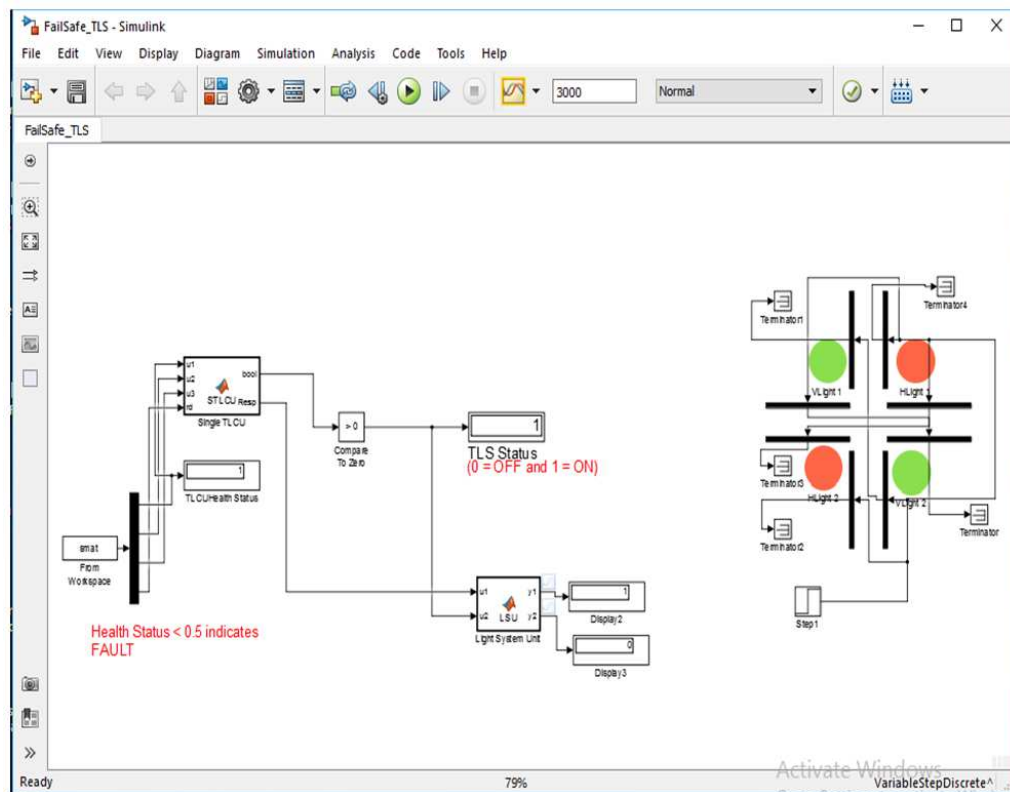


Figure 4. Existing TLS Model.

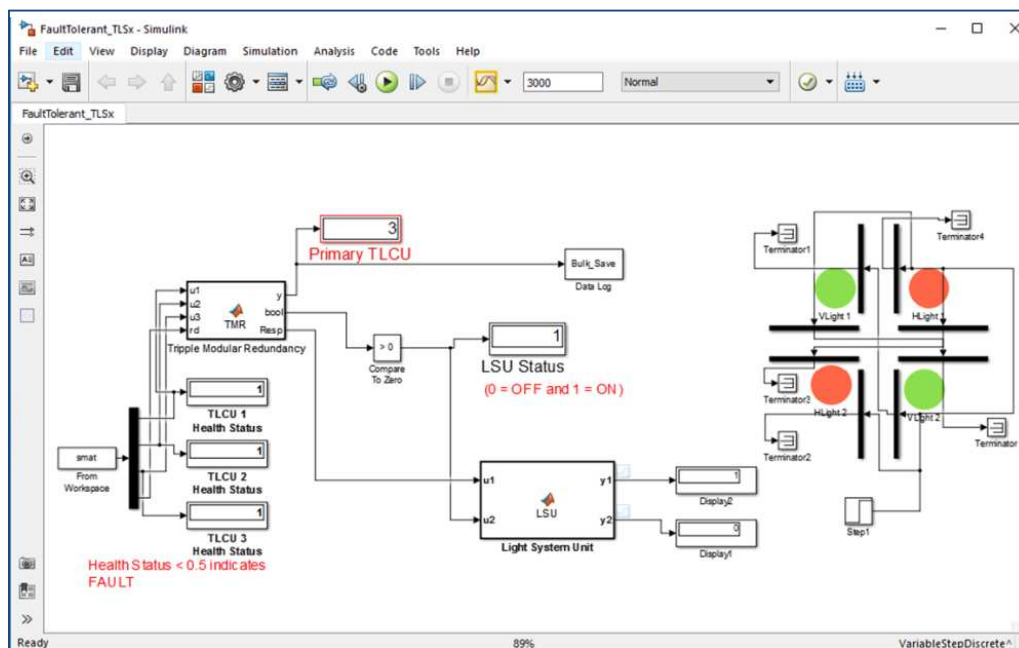


Figure 5. Fault-tolerant TLS Model.

The Fault-tolerant TLS model consists of three TLCUs (module) and a voter. The Voter will be ON if and only if at least two modules are ON. This assumption is a TMR logic that dictates that in a network of three modules, at least two of the modules must be operating satisfactorily. After every simulation, the signal outputs of the modules and the Voter

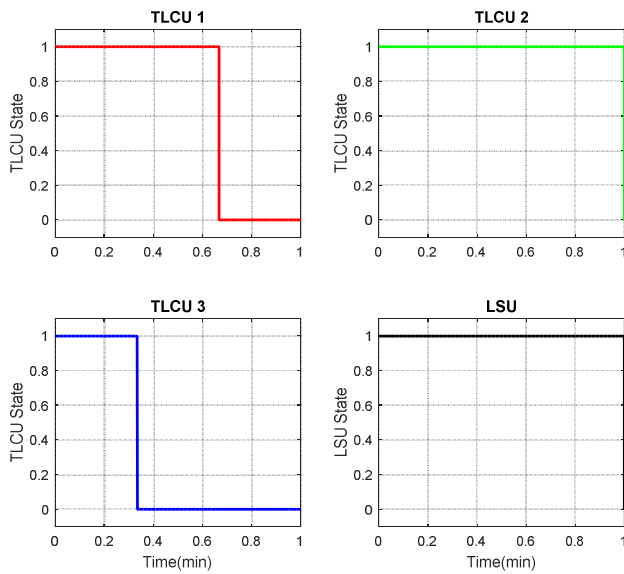
are saved in a data file for plotting and performance evaluation. At the start of the simulation, the three modules will start transmitting their signal states (1 or 0) to the voter and the data logger. Whenever a module sends a signal, the If block checks whether it is in ON or DOWN state. If it is in the DOWN state, the If Action block display 0 and sends it to



the data logger; but if it is in ON state, integer value 1 is sent to the data logger. A written MATLAB code was used to achieve this. The switching state transition of the three TLCUs of the Fault-tolerant TLS model is illustrated in Table 1 and Figure 6. The logic is derived from the TMR protocol that dictates that all the three TLCUs or two consistent outputs of two TLCUs will outvote the deviated output or wrong output of the remaining failed TCU. In other words the TLS behaves as a conventional fail-safe TLS when two TLCUs have failed hence the TLS can no longer be referred to as a fault-tolerant TLS. A healthy TCU output is indicated by 1 and failed TCU by 0 on the table.

**Table 1.** Fault-tolerant TLS Model States and Switching Operation.

TLCU 1	TLCU 2	TLCU 3	TMR Voter	Fault-tolerant TLS Operation state
1	1	1	1	Fault-tolerant Active
1	1	0	1	Fault-tolerant Active
1	0	1	1	Fault-tolerant Active
0	1	1	1	Fault-tolerant Active
1	0	0	0	Fail-safe Active
0	0	1	0	Fail-safe Active
0	1	0	0	Fail-safe Active
0	0	0	0	Total Failure



**Figure 6.** Switching Transition of the three TLCUs of the Fault-tolerant TLS Model.

The outputs of the TLS are also indicated by the voter

**Table 2.** Availability of the Existing Fail-safe TLS Model.

Runs	1	2	3	4	5	6	7	8	9	10
Availability (%)	99.3188	97.8031	96.6200	96.8580	96.5323	96.8811	99.0788	97.4621	96.6846	98.9597
MTTF (Secs)	0.0538	0.0206	0.0177	0.0164	0.0173	0.0202	0.0435	0.0199	0.0177	0.0357
MTTR (Secs)	0.0004	0.0005	0.0006	0.0005	0.0006	0.0006	0.0004	0.0005	0.0006	0.0004

**Table 3.** Availability of the Fault-tolerant TLS Model.

Runs	1	2	3	4	5	6	7	8	9	10
Availability (%)	99.9952	99.9954	99.8484	99.9903	99.9689	99.8602	99.9944	99.9908	99.8617	99.9969
MTTF (secs)	0.0012	0.7915	0.6026	0.6892	0.5994	0.5809	0.9933	0.0199	0.6125	1.0115
MTTR (secs)	0.0001	0.0000	0.0009	0.0001	0.0002	0.0008	0.0001	0.0008	0.0008	0.0000

column on the tables. The result on the graph shows that the Fault-tolerant TLS automatically selects TLCU 3 as the initial primary controller at the TLS start up. It was ON for 0.35 seconds and DOWN for 0.65 seconds of the entire 1 minute simulation time. At 0.35 seconds TLCU 3 failed and the TLS switches to TLCU 1. This is a typical scenario of a Fail-safe single TCU operation. Also, the TLCU 1 was ON for 0.65 seconds and DOWN for 0.35 seconds of the entire 1 minute simulation time. At 0.65 seconds the TLCU 1 failed and the Fault-tolerant TLS Switched to TLCU 2. TLCU 2 was ON for the entire 1 minute of simulation time. The simulation ends when there are no spare to replace the failed TLCUs. The LSU was active all through the 1 minute simulation time after which it displays all red lights flashing (fail-safe mode).

## 4. Results and Discussions

The two models were simulated for ten runs each simultaneously to investigate their availability performance. The Fault-tolerant TLS model makes use of TMR architecture consisting of three TLCUs. While the conventional Fail-safe TLS makes use of a single TCU. Table 2 shows the availability, MTTF and MTTR results of the existing Fail-safe TLS model with a single TCU while Table 3 shows the availability, MTTF and MTTR results of the Fault-tolerant TLS model with three TLCUs (module) as computed by the MATLAB simulation program. The average availability of the Fail-safe model is 97.6199% and the average availability of the Fault-tolerant TLS model is 99.9474%. A perfect system that offers 100% availability is one that does not experience any downtime throughout its operation time in its life cycle; a situation not realistic in practice. The availability of a system can only be enhanced to give a value that is closer to 100% in practice. Hence the fault-tolerant TLS model delivers a difference of 0.0526% to the perfect system availability. And the fail-safe TLS model had an availability difference of 2.3801%. Therefore the fault-tolerant TLS model was able to deliver an availability that outperforms the existing Fail-safe TLS model by 2.3275%. This information is also represented graphically in Figure 7. The line graph of the Fault-tolerant TLS model shows high degree of closeness to 100% which depicts high availability of service as oppose to the line graph of the Fail-safe TLS model

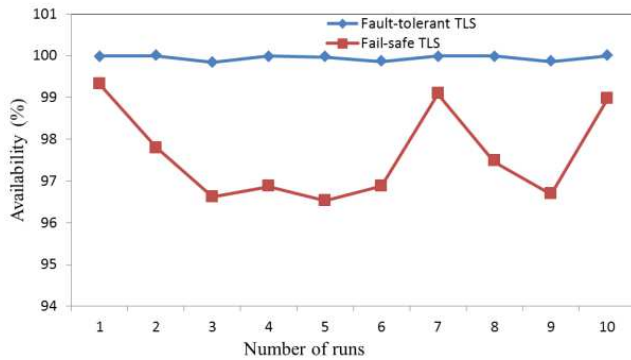


Figure 7. Fault-tolerant TLS Model versus Fail-safe TLS Model Availability.

## 5. Conclusion and Recommendations

This work presented the development of a proactive Fault-tolerant TLS controller for abating the downtime problem that is prevalent in Existing TLS. This research proffers solution to this quandary by designing a TMR scheme of three TLCUs for the TLS. The TMR architecture was modelled using Markovian and Stationarity processes. The availability performance evaluation showed that the Fault-tolerant TLS controller outperforms the existing TLS controller.

Further research recommendation in this field includes the development of algorithms that support adaptive fault tolerance in a distributed environment and real time implementation of this simulated research in order to further justify the suitability of this approach.

## References

- [1] Udoakah, N. and Okure, G. (2017). Design and Implementation of a Density Based Traffic Light Control with surveillance System, *Nigerian Journal of Technology (NIJOTECH)*, 36 (4): 1239-1248, ISSN: 2467-8821, pp 23-31.
- [2] Sivarao, K. S., Mazran, E. and Anand, T. J. (2010). Electrical and Mechanical Fault Alert Traffic Light System Using Wireless Technology. *International Journal of Mechanical and Mechatronics Engineering*, 10 (4): 15-18.
- [3] Salami, S. O., Akinyele, A. O., Sarumi, A. J. and Kesinro, K. K. (2013). Design and Construction of a Close Loop Traffic Light Control System. *International Journal of Engineering Research and Technology (IJERT)*, ISSN: 2278- 0181. 2 (12): 2107-2115.
- [4] Latha, J. R. and Suman, U. (2015). Intelligent traffic Light Controller. *International Journal of Emerging Engineering Research and Technology*, 3 (3): 38-50.
- [5] Paoli, A., Sartini, M. and Lafortuneb, S. (2011). Active fault tolerant control of discrete event systems using online diagnostics. *Automaticagooglescholar* 2011; 47. pp 639–649.
- [6] Alagoz, J. M., Tsao, J. Y. and Simmons, J. A. (2008). "Prospects for LED lighting," *Proc. SPIE* 5187, pp 227–233.
- [7] Jaroslaw, S. (2012). "Reliability and Statistics in Transportation and Communication," *Proceedings of the 12<sup>th</sup> International Conference (RelStat'12)*, 17–20 October 2012, Riga, Latvia, ISBN 978-9984-818-49-8 Transport and Telecommunication Institute, Lomonosova 1, LV-1019, Riga, Latvia, pp 362–369.
- [8] Hipel, K. W. and McLeod, A. I. (1994). "Time Series Modelling of Water Resources and Environmental Systems", Amsterdam, Elsevier 1994, pp 30-42.
- [9] Rodney, A. (2006). Design and Implementation of a Modular Controller for Robotic Machines, a Published Ph.D thesis, Presented to the department of Mechanical Engineering, University of Saskatchewan, pp 44-70.
- [10] Shalangwa, D. A. (2010). Design and Implementation of a Model (ADS-3g) of a Traffic Light using Automated Solar Power Supply, *Journal of Electrical Engineering Research*, <http://www.academicjournals.org/jeeer>, ISSN-2141-2367@2010 Academic Journals, 2 (3): 57-67.
- [11] Dauda, E., Abdulkadir H., Emmanuel, G. and Kafilat, I. (2019). Design and Development of a Traffic Density Detection and Signal Adjustment System, *Asian Journal of Applied Science and technology (AJAST)*, 3 (1): 86-98.
- [12] Khelassi A. D. And Theilliol P. W. (2014): "Control Design for over-actuated Systems Based on Reliability Indicator", Centre de Recherche en Automatique de Nancy (CRAN) CNRS UMR 7039, Nancy-Universit'e BP 70239, 54506 Vandœuvre Cedex, France, at: <https://www.researchgate.net/publication/225070811>, pp 1-5.
- [13] Sparsh M. And Jeffery S. V. (2015): "A Survey of Technique for Modelling and Improving Reliability of Computing System", *IEEE Transaction on Parallel and Distributing Systems*, at: <https://www.researchgate.net/publication/275211206>, pp 1-8.
- [14] Gauri J. (2016): "Efficient Redundancy Techniques to Reduce Delay in Cloud Systems", a Ph.D Thesis Submitted to the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology., pp 1- 209.
- [15] Kalla H., Girault A. And Sorel Y. (2010). An Hybrid Redundancy Approach to Tolerate Transient Faults in Multiple Bus Architectures. In *Proceedings of Pacific Tim International Symposium on Fault-tolerant Systems (PRFTS)*, Taipei, Taiwan, pp 425-431.