ISSN 2347 - 3983
Volume 8. No. 4, April 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter08842020.pdf
https://doi.org/10.30534/ijeter/2020/08842020

An Approach to Compute Fault Tolerance of an IoT Network having Clustered Devices Using Cross bar Networks

Geethika Reddy A[1], Y. Upendra[2], Dr.JKR Sastry[3]. Mr. Bhpathi[4]
[1]Koneru Lakshmaiah Education Foundation (Deemed to be University), Vaddeswaram,geethikareddyavula@gmail.com
[2]Koneru Lakshmaiah Education Foundation (Deemed to be University), Vaddeswaram, upendranaidu80088@gmail.com
[3]Koneru Lakshmaiah Education Foundation (Deemed to be University), Vaddeswaram, drsastry@kluniversity.in
[4]Koneru Lakshmaiah Education Foundation (Deemed to be University), Vaddeswaram, Bhupathi@kluniversity.in

ABSTRACT

Enhancing the fault tolerance of IoT networks is a challenging task as the network is achieved using small and fragile things, local and remote thing, less computing to heavy computing. IoT Networks contain several layers and each layer is built with different clustered and Un-clustered things. In each layer different architecture is followed making it difficult to computing fault tolerance of the network especially when clustered architectures are used.

FTA (Fault Tolerance Analysis) is one Technique normally used for computing fault tolerance of an IoT network. This technique becomes complicated when a different type of Topologies are used for connecting things in different Layers of a network.

In this paper a Hybrid Methodology is presented to compute fault tolerance of an IOT network when clustered devices are connected in a Cross bar network within the device Layer.

Keywords: fault Tolerance, IoT Networks, FTA analysis. Clustered Devices, Cross bar Networks

1. INTRODUCTION

IoT networks are being used extensively these days for implementing different applications that require variety of things that include Gate ways, controllers, Restful services servers, Controllers, clusters, base stations, couplers and such other devices. Some of the devices like sensors and actuators are small and the failure rate of such small devices is high.
Some of the applications being built include Home Automation, Aerospace, Automobile, Defense etc. Continuous operation of these systems is critical and therefore must be made fault Tolerant.

Fault tolerance as such could be as an integral part of the design of IoT based system. Fault tolerance must be in-built as part and parcel of the very IoT system itself. A typical IoT system must cater for implementation of many of the fault-tolerant strategies that have, in the literature.

The fault tolerance of IoT networks greatly improves when networking is done using networking systems that introduce redundancy. IoT is a network of physical objects or 'things' that can interact with each other to share information and take action. The Internet of Things (IoT) is the interconnection of uniquely identifiable embedded computing devices within the existing Internet infrastructure.

Every device in an IoT network fail and therefore needs to be made fail free. Failure as such can happen due to breakdown, malfunctioning, or security leakage. Failures in IoT can happen at any level of an IoT network. Wherever the failure, the IoT shall become in-operational and serves no purpose.

Faults within any network are bound to happen due to various reasons. A network called fault-tolerant when it functions even normally when faults occur while the network is in use. IoT networks are fragile and therefore, must be made fault-tolerant. IoT networks used in the medical domain must be fault-tolerant as any misinformation flow will cause a devastating effect even to the extent of loss of human life. A small fault may lead to serious negative results.

When a Fault happens, generally, the data acquired is lost. Data must be preserved and retained at any cost. Use of Non-volatile memory within IoT based systems will help in recovering from the loss of the normal operation when a fault occurs. Fault tolerance is essential even at the cost of incurring overhead die to use of non-volatile memories.

The common approach to enhance the fault tolerance is Making a process to be running through several instances and adding many devices in parallel such that when one fails, there is another instance/device to take over. Computing fault tolerance is as such complex due to the existence of many intricate issues.

Fault tolerance of network generally expressed quantitatively in terms of success or failure rate is the rate of failure of topmost nodes existing in a Fault Tree — the success rate computed as 1 – Failure Rate. In a typical network success rate is the probability that at least one transmission path exists from a transmitting device to the destination device, the failure rate obtained by subtracting the success rate from 1.

An IoT network typically contains many layers such as device, controller, restful service, gateway, internet and storage & and computing layers. Many devices are interconnected through the realization of a subnet in each of the layers generally using the same topology. The topology to be used as such is dependent on the kind of devices used and the fault-tolerant characteristics of those devices. A single topology as such may not be suitable for all layers of the IoT network. The network can be designed using different network topologies and architecture and different implementation methods and a certain level of redundancy built into the system.

Many types of faults happen within IoT networks, and each of the faults must be considered and find the methods to mitigate the same. IoT networks typically are recognized into several layers. A fault-tolerance computing model used could differ from layer to layer. Fault tolerance of a network generally computed using a single computational model. A single computational model is generally not stuffiest as the networks in each layer may have deterministic or probabilistic behavior. Choice of a topology suiting to the fault tolerance level of the devices contained in each layer and choice of the proper method to compute fault tolerance of a sub-net will lead to high fault-tolerant IoT network

Fault tolerance of the IoT networks is the most critical issue as small things tend to fail quite often.The failure rate of IoT networks is also high due to complexity of networking done in different layers of the IoT networks. Failure of IoT networks can happen due to several reasons that include breakdowns, malfunctioning.
Fault Tree Analysis is the technique used quite often for computing the fault rate of any system including the IoT based System.

Computing the Fault rate of an IoT network is some time complex and becomes infeasible due to the existence of different kinds of topologies used in different layers. In such a case use of FTA based computation of fault rate is quite complicated. In the device layer, cluster of devices are used for sensing and transmitting the data to the controller layer with cluster heads selected dynamically for transmission of the data dynamically. Converting such as Cluster of devices is to a FTA is complicated. Some of the clusters can be converted into Crossbar based Multi Stage networks to reduce the fault rate. However is becomes infeasible to convert a crossbar into a FTA requiring Hybrid strategy to compute Fault rate of heterogeneous IOT networks in which a different networking topology is used.

In this paper a method is presented which uses a Hybrid approach for computing the Fault rate of an IoT networks considering that the cluster of devices are represented as a Crossbar Network

2. PROBLEM DEFINITION

The main problem is computing the Fault rate of an IoT network in which clustered devices are connected into a Crossbar network within the device layer of an IoT Network.

3. RELATED WORK

Maheswari et al.,[1], have presented different kinds of failures that can happen in a mobile network that include power failures, energy failures, and network failures such as node and link failures. They have presented different techniques considering a subset of a set of failures and have shown the reliability of the network and the way the reliability enhanced through consideration of other aspects of fault tolerance that include alternative power, energy, and the network management.

Choreography is a mechanism generally used to define object interaction dynamically not withholding any of the statically defined object linkages. This technique generally affects the coherence that exists between the objects. Due to this reason, there could be loss of messages flowing across various objects contained in an application. There could be several faults occurring due to this reason leading to the failure of a system. Sylvain Cherrier et al.,[2], have proposed the method that synchronizes, de-synchronizes and a re-synchronizes the objects such that coherence between the objects intact leading to failure-free systems while dynamically configurable systems implemented through the process of choreography.

WSN networks are fault-prone due to loss of communication link, loss of data during transmission and, missing sensor nodes, etc., due to the occurrence of various factors such as asymmetric communication links, dislocation of sensor node and collision, radio interference, environmental impact, and power depletion. There are several mechanisms presented in the literature that includes cluttering, inducing redundancy, deployment of objects dynamically to mitigate the failures that can happen within WSN networks. GholamrezaKakamanshadi et al.,[3], have presented an analysis of the techniques considering the weakness and strengths of the mechanisms and arrived at suitable mechanisms deployed given a composer of failure situations.

Customers are using Cloud computing for meeting their IT requirements. However, the users are concerned with the security and availability of the data as cloud computing infrastructure can be affected due to attacks by malicious users and due to the generation of different types of faults that happen due to failure of either Hardware or software. Susmitha et al.,[4], have discussed the challenges that one should address while using cloud computing for meeting their IT requirements. One such challenge is to create fault tolerance within a network that connects various physical and logical resources. An architectural framework has been recommended implementation of which will provide fault tolerance within the network

Huge data is collected using the IoT network, which is made available to several local and remote users. Routing the information across the IoT network must cater for faults that may occur while the IoT system is in running state. Zaki Hasan et al.,[5], presented a routing algorithm which is capable of constructing and recovering and also selecting k-disjoint paths that are fault free and then communicate the data across those few selected fault-free paths, The authors considered optimization of energy required to communicate the data across the network while ensuring that minimum delay in communicating the data. They have compared PMSO with other similar algorithms and shown the efficiency and effectiveness of the algorithm.

Implementing a fault-tolerant IoT based system is complex as one has to deal with many of the dynamically evolvable and coupled systems. Alexander Power et al.,[6], built a framework using Micro-services. In the framework, they have included the support required for the IoT system to tolerate the faults when they happen through the inclusion of machine learning processes. The machine learns when the faults happen and then take tolerant actions immediately so that the network will fail free.

A cloud-based IoT network architecture proposed by Jatinder Grover et al.,[7]. The architecture built with the components required for making the network survive even in the presence of failure of the edge servers. The network recognized as different hierarchies, and the communication is re-directed to different hierarchy when a fault noticed in a different hierarchy. They have included mobile agents on the servers that share the system states, data, and other agents if the system fails at fog, edge, mist, or cloud. Inclusion of these components will help re-direction in the case of any server failure.

Generally, mission-critical real-time systems implemented through distributed embedded systems. The real-time characteristics of an embedded system mapped to the requirements of a distributed system which are dynamic. Most of the techniques available for computing the fault tolerance of a system don not considered the distributed considerations of a system. FTA based systems consider every working component and the connectivity between them, whereas the distributed systems built through logical models that describe connectivity between the components. Paul Rubel et al.,[8] have presented approaches /techniques using which FTA applied for computing the fault tolerance of distributed embedded systems. They have considered three FT based techniques/ approaches that include auto-configuration of dynamic systems, mixed-mode communication, and maintenance of redundancy into peer-peer communication. They have described an integrated system that combines an off the shelf middleware with different FT based techniques that have been the advanced models implemented by them.

All the devices in an IoT network interconnected as a subnet in the bottom-most layer of the network. The protocols used for effecting communication between the devices are also pre-identified and taken in to count while designing the IoT based systems. In this process, there could be a possibility that unlike devices may be connected leading to the generation of unwanted faults during the working of these devices. On the other hand, Chen Wang1 et al.,[9], have recommended the analysis of data generated by the respective devices and established/predict the logical relationships between those devices which can be used as a basis to predict faults and maintenance requirements of an application/objects. Generally, this needs fault diagnosis and in a way, enhancing the fault tolerance/reliability through periodic maintenance of the devices which are predicted to be error-prone.

Cloud computing technologies deal with a large amount of data, so it is cost-effective for implementing IT-based solutions. Many issues are to be addressed considering the usage of the cloud. Among all, fault tolerance and securing the data are the most important issues. DBK Kamesh et al.,[10, have presented that a fault occurring in one device might lead to faults occurring in one or more connected devices. They have implemented a design method to achieve high reliability, which leads to improvising the fault tolerance of the networks that connect clouds.

For developing an IoT network, three things focused; the network should be efficient, economical, and robust. Kai Fan et al.,[11], have presented random topologies, which promises high performance by reducing the cost of network establishment. It automatically explores to build temporary routing when unpredicted failure occurs, which will not affect the overall network. By implementing these methods, they have improved the fault tolerance and availability of the Networks.

The architecture of an IoT network designed considering the possibility of occurrence of the faults within the network. A fault-tolerant architecture proposed by AsadJaveda et al.,[12], used for implementing a variety of IoT based applications. In the architecture, they have considered the placement of software stacks at different locations for making deployment decisions at run time. They have also considered many other issues such as long-distance network connectivity, faults happening within edge devices, harsh operating environment, etc. In the architecture that included the issue of processing that should take place at both the edges of devices and the cloud.

A cluster or a leader node used for communicating within the IoT, WSN, and Adhoc networks. The node must be selected such that it has maximum energy or located to the extreme left of the network such that it would be the last node. If the head node or the leader node fails, the entire IoT network will fail. Routing algorithms are the key to any communication. Routing algorithms must be intelligent to elect a cluster head when a fault happens such that fail free communication happens. Ahc`Ene Bounceur1 et al.,[13],

have expressed that the leader must be elected dynamically considering the paths that must have failed. They have presented an algorithm for electing a leader through the use of a local minimum as a root and the concept of flooding is used to determine a spanning tree for routing the communication over the spanning tree. The two spanning trees coincide, the better one is selected, and the other ignored. The root of the spanning tree will be the leader through which the communication is affected. IoT is a layered network which is having different layers; it deals with many heterogeneous subnetworks.

Failure rates of an IoT system are dependent on network topology as the faults can happen within the network hardware device and even can happen in the software that runs in different layers. Every IoT based must be scalable, maintainable, and highly reliable. Failure of an IoT system will lose its identity and leads to customer dissatisfaction. One has to implement quite number strategies to make an IoT system more reliable. Many authors considered the reducing levels of the performance as a kind occurrence of faults with IT and therefore performance of an IoT system must also be considered for assessing the fault tolerance of the IoT based system [14][15[16][17][18][19][20][21][22][23]. Various Approaches have been presented in the Literature which are either related to networking or computing fault tolerance of different types of IoT Networks [24][25][26][[27][28][29[30][31][32].

GAP

None of the methods presented in the literature considered the issue of computing the fault rate when different topologies are considered in different layers of a IoT network.

3. INVESTIGATIONS AND FINDINGS

3.1 Overview of prototype IoT network

An IoT network typically contains several layers of networking that include Device Layer, Controlling Layer, Services layer, gateway Layer, and cloud computing Layer. The IoT network must be fault-tolerant at every layer. In this paper, an approach has been built considering all the layers in the network while exploring the fault tolerance in device layer while assuming that the fault tolerance of the layers in the network fixed and no variances noticed in those layers.

A typical IoT network developed for carrying the experimentation shown in Figure 1. The IoT network has been built considering all the layers situated in a typical and comprehensive IoT network that include device layer, controller layer, services layer, gateway layer, and computing layer and the Devices in the device layer is connected as a cluster.

Four clusters are included in the device layers. The first clusters contain three temperatures sensors which are connected completely with an elected Cluster Head which communicate with a base station. There are three more clusters similar to the Temperature sensors which include Humidity Sensing cluster, Air-condition Cluster and a FAN Clusters each communication through its cluster head with the Base Station.

In the next layer the base station is connected to a Controller in a peer to Peer to connection and the Controller is connected to a restful services server using again a peer to peer connection. The services server keeps the status of device and provides the API required for providing the status of a device or transmitting the data routed from a device through the controller to a cloud through either a Gateway or through web service server. Both the WEB server and the gate way connected to the Internet on to which the cloud is interfaced. The remote users are connected to the cloud or to the restful server through the Internet. The prototype network is simple mostly connected using a peer to peer or a parallel connection except that the devices are connected through a Cluster.

3.2 Construction the FTA (Fault Tree) for the prototype network

Given an IoT network, analysis has to be carried to find the fault tolerance strength of the network. Fault tolerance of a network is generally achieved through Fault tree Analysis. Fault tree analysis is an analytical technique. In this approach, an undersized state of the system is defined and then the same is analysed in terms of environment, operation, safety, criticality, etc. and then find different ways in which the undesired event can occur. A fault tree is a graphical model that has all combinations of the faults, both sequential and parallel that can occur, leading to an undesirable event. The faults as such can be hardware faults, network faults, software faults, or faults occurring due to human error. The basic interrelations between the faults and the events are depicted using a fault tree. The undesired event will be the top node of the fault tree. A fault tree is not a model that can capture all system failures or that causes that lead to system failures.

The top node of a fault tree relates to the occurrence of a specific event, which is a kind of system failure. The faults tree deals with those faults that lead to the top event. There can be many and many faults that could be related to the top of the event, making the construction of the tree complex. To avoid this few venerable and most important faults are selected and modelled into the tree. AND gates and OR gates are used to show the relationships among the faults that can occur on different devices. A fault tree model is not a quantitative model, and In fact, it is a qualitative model that can be measured quantitatively

In the fault tree, gates used for passing through the effect of the faults up the ladder to reach the root node. The relationship between the events modelled through the gates. It shows how the lower order events trigger higher-order

events. The output from a gate is the higher-order event. The lower order events are the inputs to the gates. The gates are not like logic gates. The gates are just symbolic to show what output event raised due to the occurrence of the lower order events. The occurrence of an output event due to the occurrence of one or more input events modelled through the OR gate, the occurrence of the output events when all input events occur modelled through AND gate.

Assessing fault tolerance of IoT networks is required as the devices in the network are fragile and lightweight. The fault tolerance of an IoT network is majorly dependent on the way various hardware elements are interconnected and the kind of devices selected for achieving the network. Network topology is the most important aspects considered from the perspective of fault tolerance of the IoT network. The topology as such takes care of many failure conditions that can generally happen within an IoT network. Many methods/models are in existence for computing the reliability of any given network, the most important being reliability analysis through fault tree analysis and probability models.

FT analysis carried on the prototype model, and the derived FT diagram for the prototype model, is shown in Figure 2. The relationships among different elements that form the network are connected through OR and gates to simulate the failure model of the prototype IoT network. Fault computations carried through compilation of failure rates of the devices and the failure of one device due to failure of other devices based on the relationships that exist among the devices through AND or OR relationships among the devices. The fault computations are undertaken using a bottom-up approach until the root node arrives. The failure rate of the root node is considered to be the failure rate of the IoT network. The failure rates of each of the device obtained through Manufacturer data. Lots of dependency is created for converting a Device Cluster into a FTA equivalent. Some intermittent dummy devices are included into the FTA diagram. Figure 3 shows the Cluster and the equivalent FTA. The cluster of three sensors are converted into a pair of two and connected through aOR gate, the output of which is connected to a dummy device. The dummy devices are connected to a cluster head through another OR gate. It has become possible yo connect like this as only 3 temperatures are considered. Think of the complication when more number such sensors exists in a cluster.

3.3 Computing the Fault rate of the prototype network through computing across the FTA diagram

After having constructed a fault tree, fault rate is computed through tabulating the Fault tree considering the relationships exhibited in the FTA and the fault data supplied by the manufacturer. The Tabulated Fault tree is shown Table-1. The computation of Fault Rate through Generation of the Table can be done using the following algorithm.

Algorithm For generation Fault Rate computation Table

Step-1:

Capture a Repository of the Hardware elements contained in the IoT network containing the Fault Rate of each of the device

Step-2:

Capture the relation (OR, AND) of each of the device with its preceding devices

Step-3:

Adjust the Fault rate of each device by applying the Relationships on the Fault rates of its preceding Devices

If the relationship is OR, find the least fault rate considering the fault rates of the Preceding devices and assign the computed fault rate to the outgoing device

If the relationship is AND, find the Product of fault rates considering the fault rates of the Preceding devices and assign the computed fault rate to the outgoing device

Step-4

Find the Fault rate of device that has no more parent devices, which is the fault rate of entire IoT network

This algorithm is applied and Table-1 is generated. From the Table it can be seen that the computed rate of the prototype IoT network is 0.84 and the auly arte is (1 – 0.84 = 0.16).

3.4 Computing the Fault rate of a Network built through Crossbar technology

A crossbar network topology deals with N Inputs (Fault rates of the Incoming Devices) and M Outputs (Fault rates of the Outgoing devices, There is one switch Box associated with each input and output which is an additional hardware element added into the network. The switch box in row I and column j is responsible for connecting the network input on row I to the network output on row i. The box as such is called as ij Box. A typical a Non fault tolerant and its equivalent Tolerant bigrams are shown in Figure 4 and Figure 5.

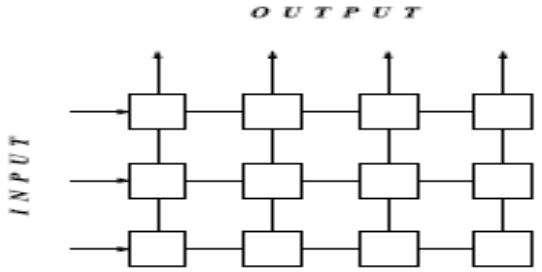

Figure 4 : Non fault Tolerant Network

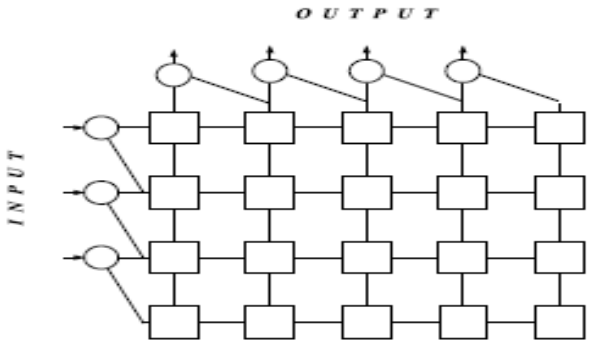

Figure 5 : Fault Tolerant Network with addition of switch boxes

Each switchbox Forwards the data received from its left link to its right link, which means propagate the data horizontally and also foreword the data flow through its bottom link to its top link. The switch box also is capable of moving data from its left lint to the Top Link. Every Link at most can carry only one data element and each switch box will be able to process two data elements at the same time. A witch box can forward data from its left link to its right link while at the same time forward the data from its bottom link to Top Link.

Example

If input 3 is to be sent to output 5, the data will be received by switchbox (3,1) which will forward it to (3,2) and so on until it reaches (3,5) which will then foreword it to (2,5) and then to (1,5) which will then be sent to the connected device.

The routing strategy is rather obvious. For example, if we want to send a message from input 3 to output 5, we will proceed as follows. The input will first arrive to switchbox (3, 1), which will forward it to (3, 2) and so on, until it reaches switchbox (3, 5). This switchbox will turn the message into column 5 and forward it to box (2, 5), which will send it to box (1, 5), which will send it to its destination.

From this network one can see that any input-output combination can be realised as long as there is no collision at the output (No two inputs are competing for the same output line). This network thus is quire suitable when process is quite faster and just involves transmission such as transmission of data from a sensor.The connectability of the crossbar can be analysed to assess the failure rates of the Individual components.

q_l = probability that a Link is Faulty
$1-q_i$ = probability that a Link and the switch box is not Faulty

Counting from 1, for input *i*to be connectable to output *j*, we have to go through a total of *i*+ *j* links. The probability that all of them are fault free is P_l^{i+j}

Probability that a a network will be fault-free =
The fault calculations of the revised FTA diagram are shown in the **Table 2**.

$$Q = \sum_{i=1}^{N}\sum_{j=1}^{M} p_\ell^{i+j} = p_\ell^2 \frac{1-p_\ell^N}{1-p_\ell}\frac{1-p_\ell^M}{1-p_\ell} \quad \text{Equ. 1}$$

3.4 Modifying the Prototype of IoT network using a Crossbar topology at Device level

The prototype network is modified to implement Crossbar network at the device layer which originally contain the clusters of devices. The Modified IoT network at the Device level is shown in Figure 6 and the Modified total IoT network is shown in Figure 7. Converting the Crossbar network to a Fault Tree Diagram is not feasible or not Complex. Thus there is a need for a new strategy to compute the Fault Tolerance of an IoT network.

Table 2: Cross Bar Fault rate computation

Expression	Value
P_l	0.1
$1 - P_l$	0.9
Number of Inputs (m)	3
Number of outputs (n)	3
P_l**2	0.01
P_l**3	0.001
p^{l**m}	0.001
$1-p^{l**m}$	0.999
$1-p^{l**n}$	0.999
$(1-p^{l**n}/ 1-p_l)$	1.11
$((p_l**2) * (1-p_l**n)/1-P_l)$	0.0111
$(1-p^{l**n}/ 1-p_l) * ((p_l**2) * (1-p_l**n)/1-P_l)$	0.012321
Success Rate	1 - 0.012321 = 0.987679

The IoT network is now segment into two parts, the device layer part and the rest of the network. The fault rate of the device layer is computed using the equation 1 considering each of the clusters individually and then taking an OR of the outputs obtained. Calculation of Fault Rate of Crossbar network is shown in Table 3.

The Success rate is attached to the dummy device included in revised FTA diagram shown in Figure8. The FTA diagram of the revised IoT network considering a set of clustered devices as a single component is shown in Figure 8. From the table it can be seen that the success rate at the computing

end rose to 0.965. as against the success rate that is computed for Original IoT network being 0.840. In increase success rate of 0.12 is achieved by introducing the Crossbar network at Device level

4. CONCLUSIONS

IoT based Systems must be designed keeping in view of the issue of Fault Tolerance without which the IoT based systems will become nonoperational and become out of use.

Building Fault tolerant IoT based system is complex due to involves meant too many small things like sensors and actuators

There is a need to build fault tolerance in every layer of the IoT based system. It is highly difficult to build fault tolerant systems especially when clustered devices are used in any of the layers.

Addition of redundancy is absolutely required for building fault tolerant IoT based systems. Use of Cross Bar based networking system at Device level considering the device clusters will greatly improve the performance of the IoT based Systems. The Improvement will certainly to the extent of minimum of 12%.

REFERENCES

1 D.Maheshwari, A. Dhanalakshmi, Fault Tolerance in Mobile ad hoc Network: A Survey. *International Journal of Advanced Research in Computer Science and Software Engineering*,Volume 3, Issue 3, pp: 191-195

2 Sylvain Cherrier, Yacine M. Ghamri-Doudane, StéphaneLohier, and Gilles Roussel, (2014). Fault-recovery and Coherence in the Internet of Things Choreographies. *IEEE WF-IoT,HAL Id: hal-00957056*, pp:1-7 https://doi.org/10.1109/WF-IoT.2014.6803224

3 Chen Wang1, Hoang Tam, (2015). An IoT Application for Fault Diagnosis and Prediction, *IEEE, 978-1-5090-0214-6/15,DOI 10.1109/DSDIS.2015.97*, pp: 726-731

4 S. L. Sushmitha, Dr. D. B. K. Kamesh, Dr. J. K. R. Sastry, V. V. N. Sri Ravali, Y. Sai Krishna Reddy, (2016). Building Fault Tolerance within clouds for Providing Uninterrupted Software as Service. *Journal of Theoretical and Applied Information Technology*,Vol.88. No.1,ISSN: 1992-8645, pp: 65-76

5 Mohammed Zaki Hasan and Fadi Al-Turjman, (2017). Optimizing Multipath Routing With Guaranteed Fault Tolerance in the Internet of Things. *IEEE Sensors Journal,Digital Object Identifier 10.1109/JSEN.2017.2739188*. VOL. 17, NO. 19, pp: 6463-6473

6 Alexander Power and Gerald Kotonya (2018). A Microservices Architecture for Reactive and Proactive Fault Tolerance in IoT Syste*ms.IEEE, 978-1-5386-4725-7/18/$31.00,DOI:10.1109/wowmom.2018.8449789*, pp: 1-6

7 Jitender Grover and Rama Murthy Garimella, (2018). Reliable and Fault-Tolerant IoT-Edge Architecture. *DOI: 10.1109/ICSENS.2018.8589624*, pp:1-4

8 Paul Rubel, Aniruddha Gokhale, Aaron Paulos, Matthew Gillen, Jaiganesh Balasubramanian, Priya Narasimhan, Joseph Loyall, Richard Schantz, (2007). Fault-tolerant approaches for distributed real-time and embedded systems. *(DARPA) under contract NBCHC030119https://ieeexplore.ieee.org/document/4455043*, pp: 1-8

9 Gholamreza Kakamanshadi, Savita Gupta, Sukhwinder Singh, (2015). A Survey on Fault Tolerance Techniques in Wireless Sensor Networks. IEEE, 978-1-4673-7910-6/15/$31.00, pp: 168-173

10 DBK Kamesh, JKRSastry, Ch. Devi Anusha, P. Padmini, G. Siva Anjaneyulu, (2016). Building Fault Tolerance within Clouds at Network Level. International Journal of Electrical and Computer Engineering, Vol. 6, No. 4, pp: 1560-1569. https://doi.org/10.11591/ijece.v6i4.10676

11 Kai Fan, Jiapeng Lu, Dazhen Sun, Yong Jin, Ruimin Shen, Bin Sheng, (2017). Failure Resilient Routing Via IoT Networks. *978-1-5386-3066-2/17, DOI 10.1109/iThings-GreenCom-CPSCom-SmartData*.

12 AsadJaved, KeijoHeljanko, Andrea Buda, and KaryFrämling, (2018). A Fault-Tolerant IoT Architecture for Edge and Cloud. IEEE, 978-1-4673-9944-9/18, DOI:10.1109/wf-IoT.2018.8355149, pp: 813-818.

13 AhceneBounceur, MadaniBezoui, MassinissaLounis, Reinhardt Euler, CiprianTeodorov, (2018). A New Dominating Tree Routing Algorithm for Efficient Leader Election in IoT Networks. *IEEE 978-1-5386-4790-5/18*,DOI:10.1109/ccnc.2018.8319292, pp:1-2

14 Murty, A. S. R., Teja, K., Naveen, S. (2018). Lathe performance monitoring using IoT. *International Journal of Mechanical Engineering and Technology (IJMET)*, Volume 9, Issue 4, pp. 494–501

15 Rambabu, K., Venkatram, N. (2018). Traffic flow features as metrics (TFFM): Detection of application layer level DDOS attack scope of IoT traffic flow. *International Journal of Engineering and Technology (UAE)*, 7(2), pp. 203-208

16 K., Prabu, A.V., Sai Prathyusha, M., Varakumari, S., (2018). Performance monitoring of UPS battery using IoT. *International Journal of Engineering and Technology(UAE)*, 7 (2.7), pp: 352-355

17 Poonam Jain, S., Pooja, S., Sripath Roy, K., Abhilash, K, Arvind, B. V., (2018). Implementation of asymmetric processing on multi-core processors to implement IOT applications on GNU/Linux framework. *International Journal of Engineering and Technology (UAE)*, 7 (2.7), pp:710-713 https://doi.org/10.14419/ijet.v7i2.7.10928

18 Raja Sekhara Naidu, G., Venkatram, N, (2018). Urban climate monitoring system with IoT data analytics.

International Journal of Engineering and Technology (UAE), 7 (2.20), pp: 5-9

19 Poonam Gupta, Kopparti Veera Venkata Satyanarayan, Dilip Devchand Shah, (2018). Development and testing of message scheduling middleware algorithm with SOA for message traffic control in the IoT environment.*International Journal of Intelligent Engineering and Systems.* Vol.11, No. 5,DOI: 10.22266/ijies2018.1031.28, pp: 301-313

20 Poonam Gupta, Kopparti Veera Venkata Satyanarayan, Dilip evchand Shah, (2018). IoT multitasking: Development of hybrid execution service-oriented architecture (HESOA) to reduce response time for IoT application. *Journal of Theoretical and Applied Information Technology*, 96(5), pp. 1398-1407,

21 Yasaswini, A., DayaSagar, K. V, Shri Vishnu, K., Hari Nandan V, Prasadara Rao, P. V. R. D., (2018), Automation of an IoT hub using artificial intelligence techniques. *International Journal of Engineering and Technology(UAE), 27DOI: 10.14419/ijet.v7i2.7.10250*, Vol 7, No 2.7, pp. 25-30

22 Ramaiah, C. H., Parimala, V. S., Kumar, S. P., Reddy, G. B., Rahul, Y. (2018). Remote monitoring through the tab. *International Journal of Mechanical Engineering and Technology*, Volume 9, Issue 1, January 2018, pp. 490–498

23 Y. Shanmukha Sai, K. Kiran Kumar, (2018). Internet of things and their applications. *International Journal of Engineering and Technology(UAE)*, Vol 7, No 2.7, pp. 422-427

24 Dr. JKR Sastry, Mr. Bhupathi, Enhancing Fault Tolerance of IoT Networks within Device Layer, International Journal of Engineering technology and Engineering Research, Volume 8, Issue 2, 2020, 491-509

25 Dr. J, Sasi Bhanu, Dr. JKR Sastry, P. Venkata Sunil Kumar, B. Venkata Sai, K.V. Sowmya, Enhancing Performance of IoT Networks through High Performance Computing, International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, Issue 3, 2019, 432-442 https://doi.org/10.30534/ijatcse/2019/17832019

26 J. Rajasekhar, Dr. JKR Sastry, An Approach to hybridisation of embedded system networks, International Journal of Engineering & Technology 7 (2.7) (2018) 384-389 https://doi.org/10.14419/ijet.v7i2.7.10748

27 T. Pavithra, J. K. R. Sastry, Strategies to handle heterogeneity prevalent within an IOT based network, International Journal of Engineering & Technology, 7 (2.7) (2018) 77-83 https://doi.org/10.14419/ijet.v7i2.7.10264

28 Bhupathi, Dr. JKR Sastry, A framework for effecting fault tolerance within IoT network, Jour of Adv. Research in Dynamical & Control Systems, Vol. 10, 02-Special Issue, 2018

29 K.V. Sowmya, Dr. JKR Sastry, Performance evaluation of IOT systems – basic issues, International Journal of Engineering & Technology, 7 (2.7) (2018) 131-137 https://doi.org/10.14419/ijet.v7i2.7.10279

30 M. Sai Rama Krishna, J K R Sastry , J Sasi Bhanu, Building Fault Tolerance Within Wireless Sensor Networks: A Butterfly Model, Research Journal of Applied Sciences 12 (2): 139-147, 2017

31 J K. R. Sastry, K. Sai Abhigna, R. Samuel, D. B. K. Kamesh, Architectural models for fault tolerance within clouds at infrastructure level, ARPN Journal of Engineering and Applied Sciences, VOL. 12, NO. 11,JUNE 2017

32 JammalamadakaRajasekhar, JKR. Sastry. Building composite embedded systems based networks through hybridisation and bridging I^2C and CAN, Journal of Engineering Science and Technology, Vol. 15, No. 2 (2020) 858 - 881

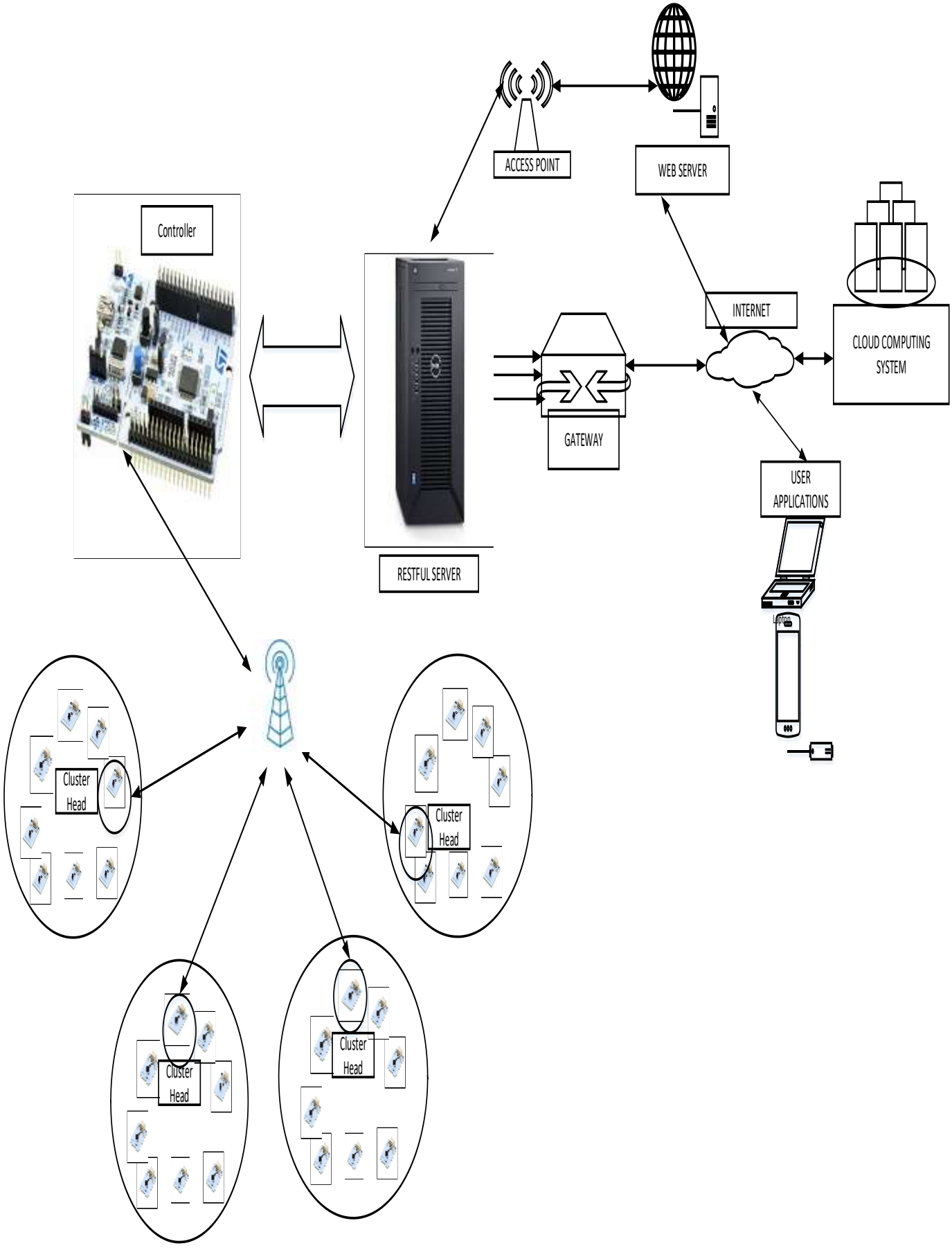

Figure 1 :A Prototype IoT Network

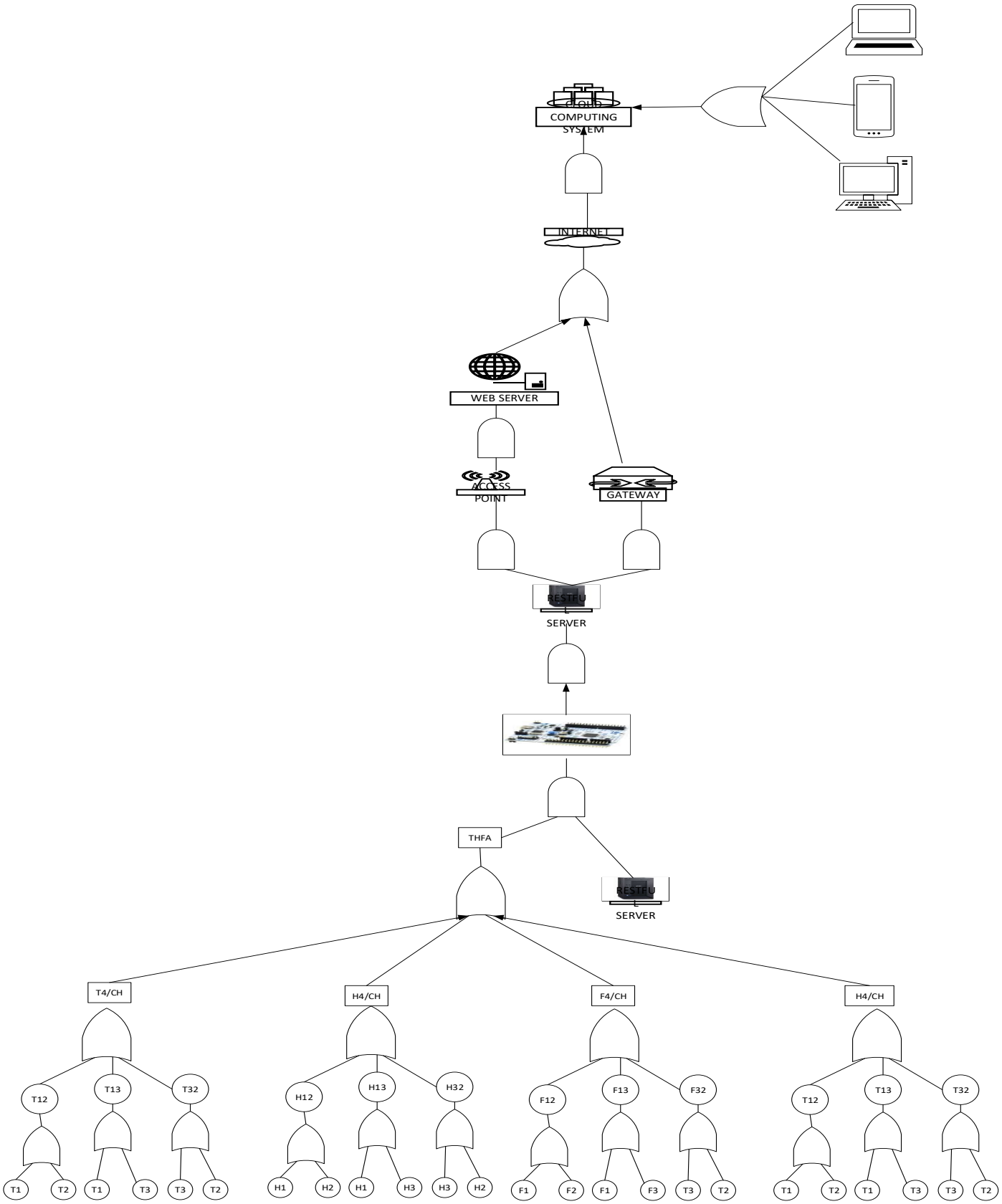

Figure 2 : FTA Diagram for Prototype IoT Network

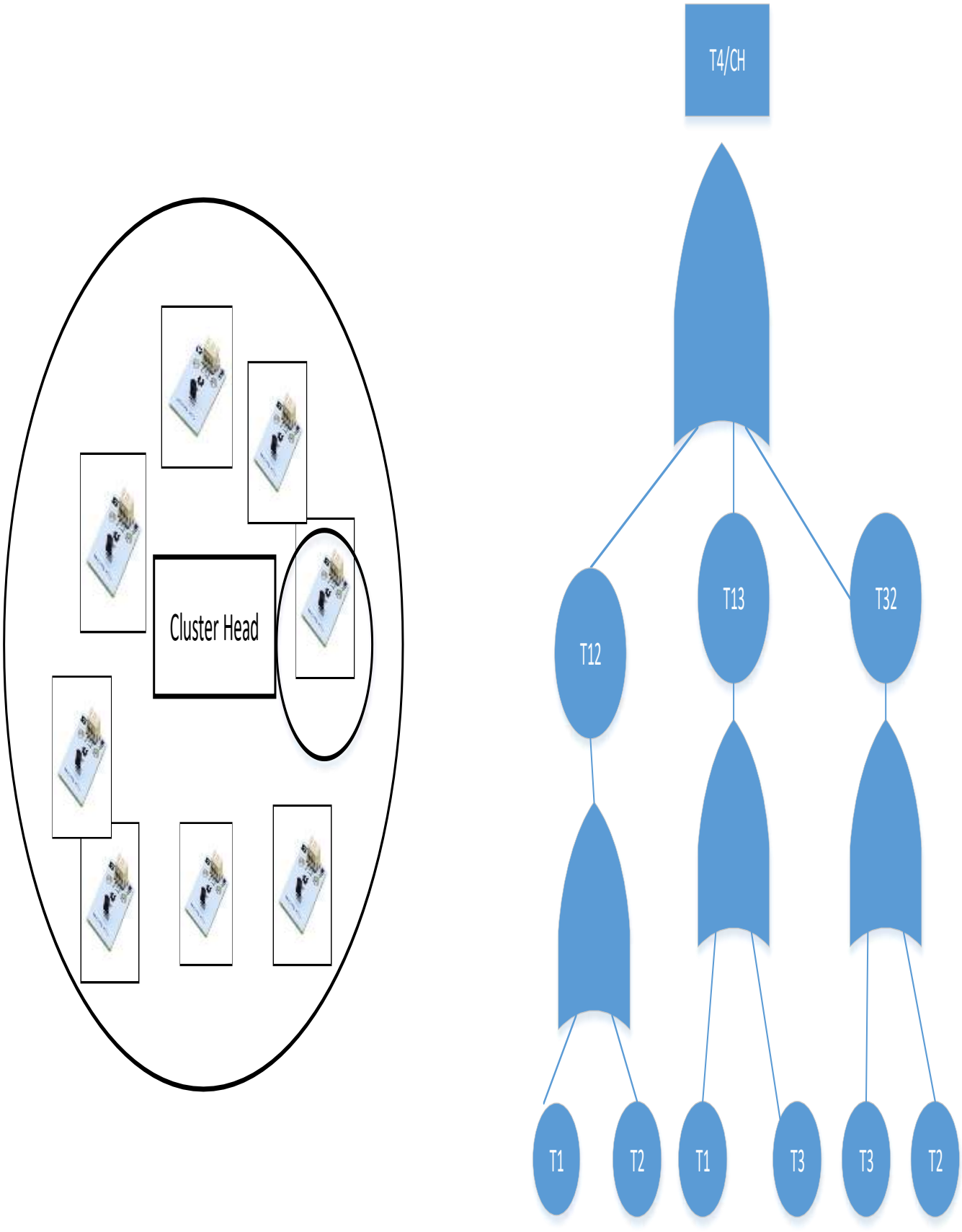

Figure 5 : Converting a Cluster to a FTA Diagram

Table 1: Computation Fault Tolerance of a Prototype IoT Network Using FTA

\

Sl.no	Device	Success Rate	Gates used For Connection	Preceding Devices					
				Device name D1	Device name D2	Device name D3	Device name D4	Device name D5	Combined Success Rate
				Success Rate S1	Success Rate S2	Success Rate S3	Success Rate S4	Success Rate S5	
1	Temp-Sensor-1	0.95							0.950
2	Temp-Sensor-2	0.95							0.950
3	T12-Dummy	1	OR	T1	T2				
				0.950	0.950				0.950
4	Temp-sensor-3	0.95							0.950
5	T23-Dummy	1	OR	T2	T3				
				0.950	0.950				0.950
6	T13-Dummy	1	OR	T1	T3				
				0.950	0.950				
7	Temp-Sensor-4	0.95	OR	T12	T23	T13			0.95
				0.95	0.95	0.95			
8	Humidity-Sensor-1	0.95							0.950
9	Humidity-Sensor-2	0.95							0.950
10	Humidity-Sensor-3	0.95							0.950
11	H12-Dummy	1	OR	H1(0.950)	H2(0.950)				0.95
12	H23-Dummy	1	OR	H2(0.950)	H3(0.950)				0.95
13	H31-Dummy	1	OR	H3(0.950)	H1(0.950)				0.95
14	Humidity-Sensor-4	0.95	OR	H12	H23	H31			0.950
				0.950	0.950	0.950			
9	FAN-1	0.95							0.950
10	FAN-2	0.95							0.95
11	FAN-3	0.95							0.950
12	F12-Dummy	1	OR	F1(0.950)	F2(0.950)				0.95
13	F23-Dummy	1	OR	F2(0.950)	F3(0.950)				0.95

Sl.no	Device	Success Rate	Gates used For Connection	Preceding Devices					
				Device name D1	Device name D2	Device name D3	Device name D4	Device name D5	Combined Success Rate
				Success Rate S1	Success Rate S2	Success Rate S3	Success Rate S4	Success Rate S5	
14	F31-Dummy	1	OR	F3(0.950)	F1(0.950)				0.95
15	Humidity-sensor-4	0.95	OR	T12(0.950)	T23(0.950)	T13(0.950)			0.95
16	THFA	0.95	OR	T4(0.950)	H4(0.950)	F4(0.950)	H4(0.950)		0.95
17	SERVER	0.93							0.93
18	CONTROLLER	0.95	AND	THFA(0.950)	SERVER(0.93)				0.88
19	POINT	0.9							0.9
20	GATEWAY	0.91							0.91
21	WEBSERVER	0.93							0.93
22	INTERNET	0.95	AND	WEBSERVER 0.93	GATEWAY 0.91				0.84
23	COMPUTING	0.84							0.84

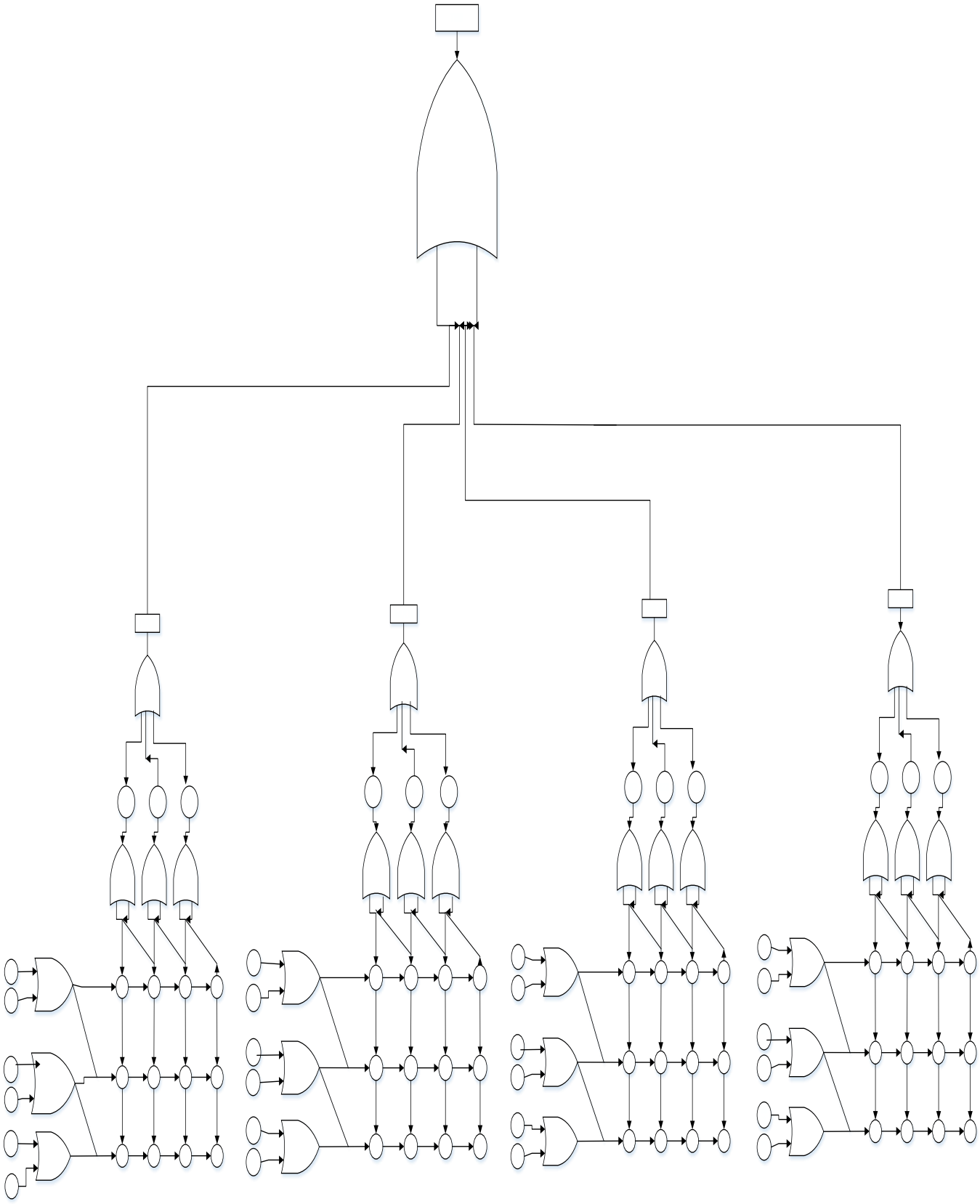

Figure 6: Converting the Clustered Devices to a Crossbar network

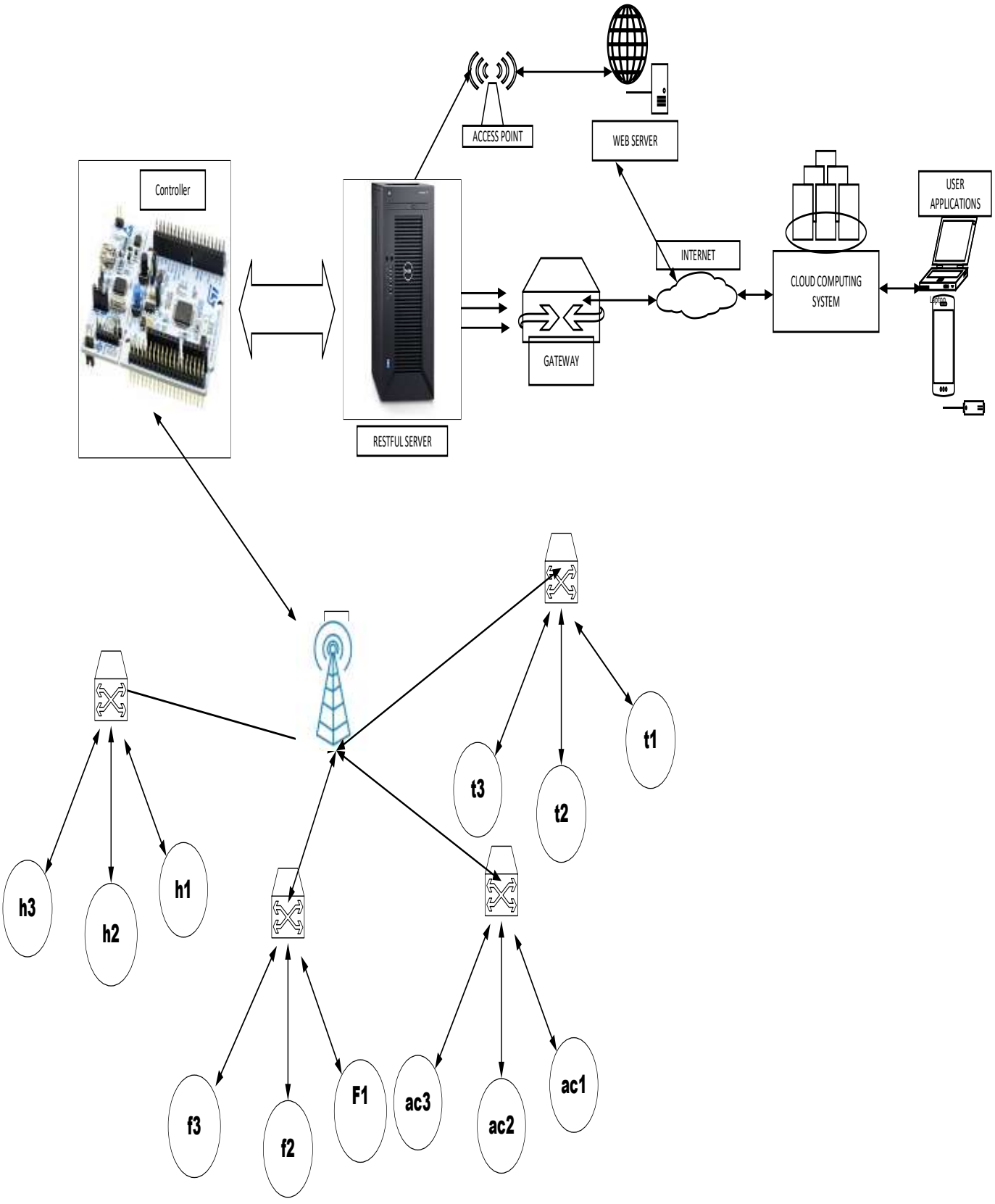

Figure 7: Modified Prototype IoT network

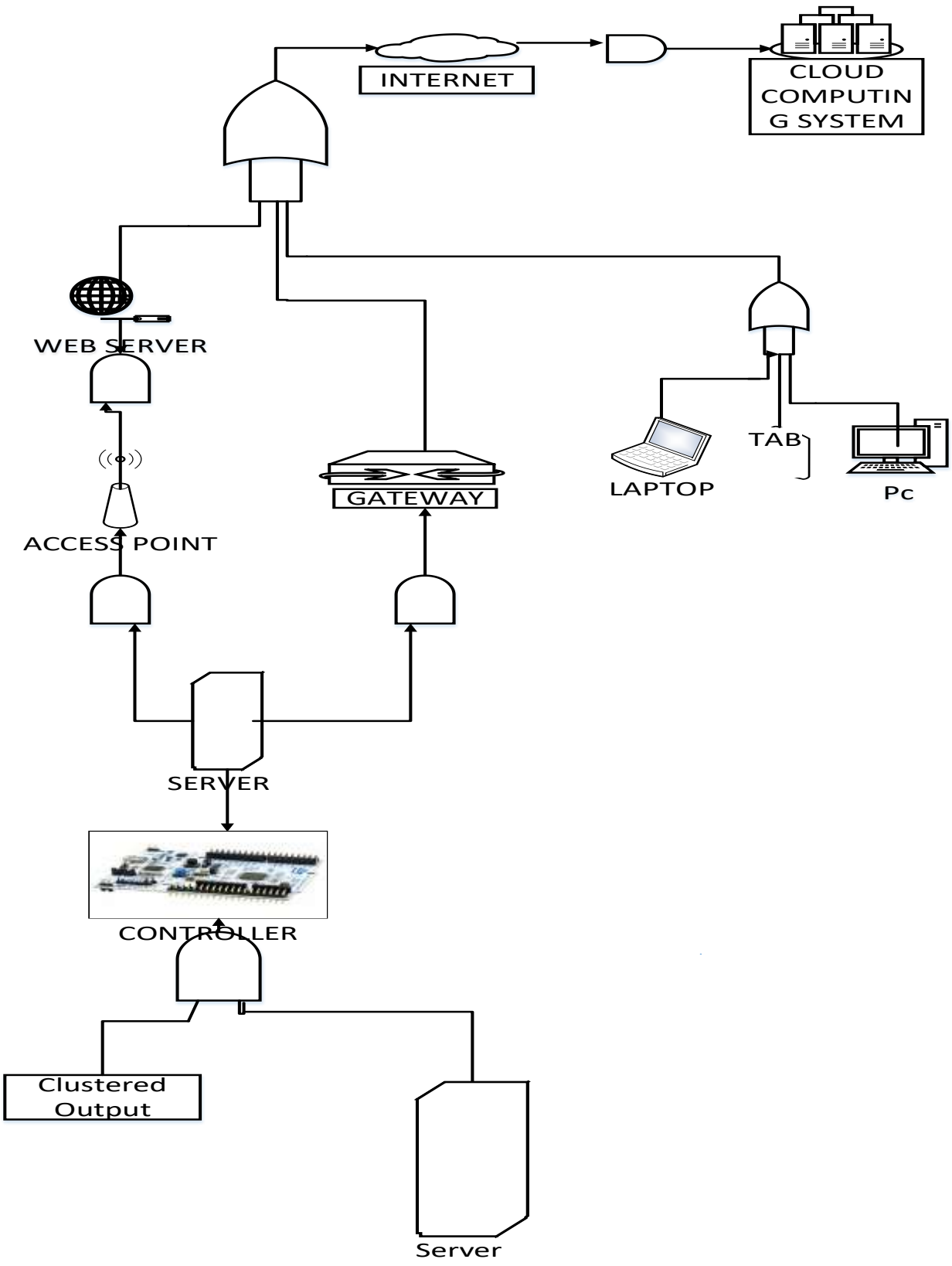

Figure 8: FTA for a Modified IoT Network

Table 3: Revised FTA Computations

Sl.no	Device	Success Rate	Gates used For Connection	Preceding Devices					
				Device name D1	Device name D2	Device name D3	Device name D4	Device name D5	Combined Success Rate
				Success Rate S1	Success Rate S2	Success Rate S3	Success Rate S4	Success Rate S5	
1	Temp-Sensor-1	0.98							0.98
2	Temp-Sensor-2	0.98							0.980
3	T12-Dummy	1	OR	T1	T2				
				0.980	0.980				0.980
4	Temp-sensor-3	0.98							0.980
5	T23-Dummy	1	OR	T2	T3				
				0.980	0.980				0.980
6	T13-Dummy	1	OR	T1	T3				
				0.980	0.980				
7	Temp-Sensor-4	0.98	OR	T12	T23	T13			0.98
				0.98	0.98	0.98			
8	Humidity-Sensor-1	0.98							0.98
9	Humidity-Sensor-2	0.98							0.980
10	Humidity-Sensor-3	0.98							0.980
11	H12-Dummy	1	OR	H1(0.980)	H2(0.980)				0.98
12	H23-Dummy	1	OR	H2(0.980)	H3(0.980)				0.98
13	H31-Dummy	1	OR	H3(0.980)	H1(0.980)				0.98
14	Humidity-Sensor-4	0.98	OR	H12	H23	H31			0.980
				0.980	0.980	0.980			
9	FAN-1	0.98							0.98
10	FAN-2	0.98							0.98
11	FAN-3	0.98							0.980
12	F12-Dummy	1	OR	F1(0.980)	F2(0.980)				0.98
13	F23-Dummy	1	OR	F2(0.980)	F3(0.980)				0.98
14	F31-Dummy	1	OR	F3(0.980)	F1(0.980)				0.98

Sl.no	Device	Success Rate	Gates used For Connection	Preceding Devices					
				Device name D1	Device name D2	Device name D3	Device name D4	Device name D5	Combined Success Rate
				Success Rate S1	Success Rate S2	Success Rate S3	Success Rate S4	Success Rate S5	
15	Humidity-sensor-4	0.98	OR	T12(0.980)	T23(0.980)	T13(0.980)			0.98
16	THFA	0.98	OR	T4(0.980)	H4(0.980)	F4(0.980)	H4(0.980)		0.98
17	SERVER	0.98							0.93
18	CONTROLLER	0.95	AND	THFA(0.980)	SERVER(0.98)				0.96
19	POINT	0.95							0.95
20	GATEWAY	0.97							0.95
21	WEBSERVER	0.99							0.95
22	INTERNET	0.95	AND	WEBSERVER 0.99	GATEWAY 0.97				0.9605
23	COMPUTING	0.9605							0.9605