ISSN 2278-3091
Volume 9 No.2, March - April 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse184922020.pdf
https://doi.org/10.30534/ijatcse/2020/184922020

On improving fault tolerance of IoT networks through Butterfly Networks implemented at Services Layer

Amirapu Anjana[1], G. Gopi Chand[2], K. Sai Kiran[3], Dr.JKR Sastry[4]. Bhpathi[5]

[1]Koneru Lakshmaiah Education Foundation, Vaddeswaram, amirapuanjana611@gmail.com
[2]Koneru Lakshmaiah Education Foundation, Vaddeswaram, gopichandgutta17@gmail.com@gmail.com
[3]Koneru Lakshmaiah Education Foundation, Vaddeswaram, kkcsai03@gmail.com
[4]Koneru Lakshmaiah Education Foundation (Deemed to be University), Vaddeswaram, drsastry@kluniversity.in
[5]Koneru Lakshmaiah Education Foundation (Deemed to be University), Vaddeswaram, Bhupathi@kluniversity.in

ABSTRACT

IoT networks are being used these days for several purposed and in many domains. IoT Networks contain several layers of networking and each network is built with small to big things. IoT networks are fragile and therefore tend to fail several times defeating the very purpose of building the IoT networks. The networking layers of the IoT network include device layer, controller layer, services layer, gateway layer, networking layer and the storage layer. Most of the computing within the IoT network is undertaken in services layer. With a single server used at this layer, the fault tolerance of the IoT networks is insignificant. Use of clustered servers enhances the fault tolerance of the IoT network.

In this paper, it is shown, how the fault tolerance of the IoT networks improves with use of Butterfly network implemented in the services layer using 4 Servers to process 4 Significant inputs and 4 Significant outputs. The fault tolerance of the IoT network is computed using the FTA model and a probability model that can be used for computing fault tolerance with the services layer when Butterfly network is used for building the IoT network. A Hybrid model of computing the Fault tolerance of the IoT network is presented when services layer is built through a Butter fly model..

Key words: fault Tolerance, IoT Networks, and FTA analysis. Clustered Devices, Butterfly 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. 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 that has the services layer built using the Butterfly model and to improve the fault tolerance of the IoT network using such a model.

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 containthree 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.

Lots of conversion is needed so as to convert an IoT network into a FTA diagram. Figure 2 shows the way clustered devices are converted into FTA diagram Equivalent. FT analysis carried on the prototype model, and the derived FT diagram for the prototype model, is shown in Figure 3. 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 the networking done in specific layer into FTA equivalent tree diagram. Some intermittent dummy devices are included into the FTA diagram**.** The cluster of three sensors are converted into a pair of two and connected through a OR 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 to 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 calculations are 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 fault rate is (1 – 0.84 = 0.16).

3.4 Computing the Fault rate of a Network built through a Butterfly Network

Many networking topologies exist in the literature that includes Butterfly, crossbar, multi-stage, mesh, and such other combination of topologies. It is easy to compute the reliability of these networks due to the existence of probability-based computational methods for computing reliability. The failure rate of the network reduces by implementing different topologies than the tree topology used for the sample IoT network. In a multistage network, there will be multiple stages at which the inputs are processed to transform inputs to outputs. The switching nodes sutured in different processing stages to not all undertake to process, but all move the processed outputs to the nodes situated in a different stage.

The network is built using N X N switches. Each switch takes N inputs and produces N Outputs. The switches

interfaced through different connections that include the cross, straight, lower broadcast, and upper broadcast. A 2 X 2 network is called the butterfly network. The devices are networked using Butterfly model. The Links that are at a distance two are connected to switch box. A 4 X 4 butterfly network achieved through two 2 X 2 Butterfly Networks. Figure 4 shows a 4 X 4 Butterfly network. Consider Φ(0), which is the probability that at least one line out of a switch box at the output stage is functional then the probability is 1-q_l, where $q_l$is the failure rate of the link.

$$\phi(i) = 1 - ql \qquad (1)$$

Where k = Number of stages, ρl = probability that a link fails and Φ (i) is the probability that a switch box in stage K can fail. Φ(k) can be computed using equation (2).

$$\phi(i) = 1 - (1 - pl\phi(i-1))2 \qquad (2)$$

Due to the simplicity of the butterfly topology, it has been contemplated to modify the original prototype IoT network using Butterfly topology at services level to assess the change in the fault tolerance state of the IoT network. Figure 4shows the revised network topology built in the services level using for computing server with 4 Inputs and 4 Outputs.

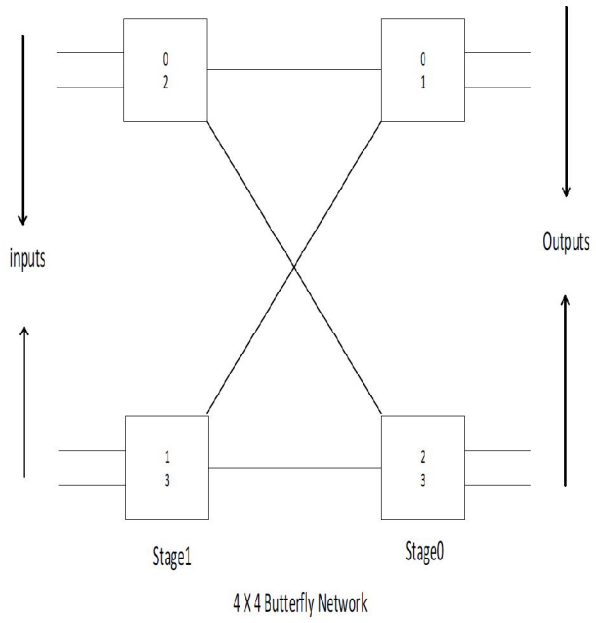

Figure 4: 4 X 4 Butterfly network

3.5 Modifying the Prototype of IoT network using a Butterfly network topology at Services Layer

The prototype network is modified to implement Butterfly network at services Layer. Four servers have been added which are connected in butterfly network so that fault tolerance shall increase due to redundancy introduced. The Modified IoT network is shown in Figure 5.

3.6 Computing Fault Rate of revised IoT network

The fault rate of the revised IoT network is computed in two ways. First a Fault Tree is constructed for the revised IoT network and then fault calculations are carried. The fault tree for the revised IoT diagram is shown in Figure 6. The fault calculations carried based on the revised FTA Diagram is shown in Table 2. From the Table it can be Seen that the success rate is improved from 0.840 t0 0.940 leading to 10% Improvement

Fault calculations are also carried using a Hybrid model. Using the Hybrid Model, Fault calculations are carried in two phases. In pase-1 Fault calculations of the network in the services layer is calculated using a probability model related to butterfly network. Then the network in the services layer is replaced by a node which is assigned with the fault rate computed using the Probability model. The newly arrived FTA diagram is shown in Figure 7. Fault Calculations using the Hybrid model is carried in the following way

In the revised IoT network, a 4X4 butterfly network has been constructed at the service layer considering 4 Inputs (Temperature, Humidity, FAN, and LIGHT. Following are the reliability computation of the butterfly network built at services layer level. Using equation (1) and equation (2)

$$pl = 0.98, ql = 0.02$$

$$\phi(i) = 1 - (1 - pl\phi(i-1))2$$
$$\phi(1) = 1 - (1 - 0.98\phi(1-1))2$$
$$= 1 - (1 - 0.98\phi(0))2$$
$$= 1 - (1 - 0.98(1 - ql))2$$
$$= 1 - (1 - 0.98(1 - 0.02))2$$
$$= 1 - (0.02(0.98))2$$
$$= 1 - (0.0196)2$$
$$= 1 - 0.00038$$
$$= 0.998$$

0.998 Success rate is assigned to a device that replaces the network in the services layer and then the revised FTA diagram is constructed which is used for computing the final Success rate of the network.

The fault calculations made using the revised fault tree diagram is shown in Table 5. From the Table it can be seen that the Success rate is increased to 0.998 leading to an overall improvement of 15%.

4. CONCLUSION

IoT system is built using several layers of networking which include Devices Layer, Controlling Layer, and Services Layer, Gateway layer, Internet layer and storage/Cloud Computing Layer. The fault tolerance of IoT system depends on the Kind of networking used in different layers.

Services Layer is the central part of the IoT network which provides services to the users who interact through the internet. Services layer is also responsible transport data between the gate way and the Gateway. Most of the computing is done at this layer. Any fault in this layer will jeopardize the working of entire IoT system.

Implementing the clustered or networked servers reduces failure rate of the IoT network. Implementing clustered servers connected through crossbar networks greatly enhances the fault tolerance of the IoT network.

Implementing crossbar network at services layers with four Restful servers increased the fault tolerance of the IoT network by more than 11%

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, AniruddhaGokhale, Aaron Paulos, Matthew Gillen, JaiganeshBalasubramanian, PriyaNarasimhan, 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. GholamrezaKakamanshadi, 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 https://doi.org/10.14419/ijet.v7i2.7.10293
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
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
https://doi.org/10.14419/ijet.v7i2.20.11732

19. Poonam Gupta, Kopparti Veera VenkataSatyanarayan, DilipDevchand 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 VenkataSatyanarayan, DilipDevchand 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
https://doi.org/10.30534/ijeter/2020/37822020
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
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
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 AppliedSciences, 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

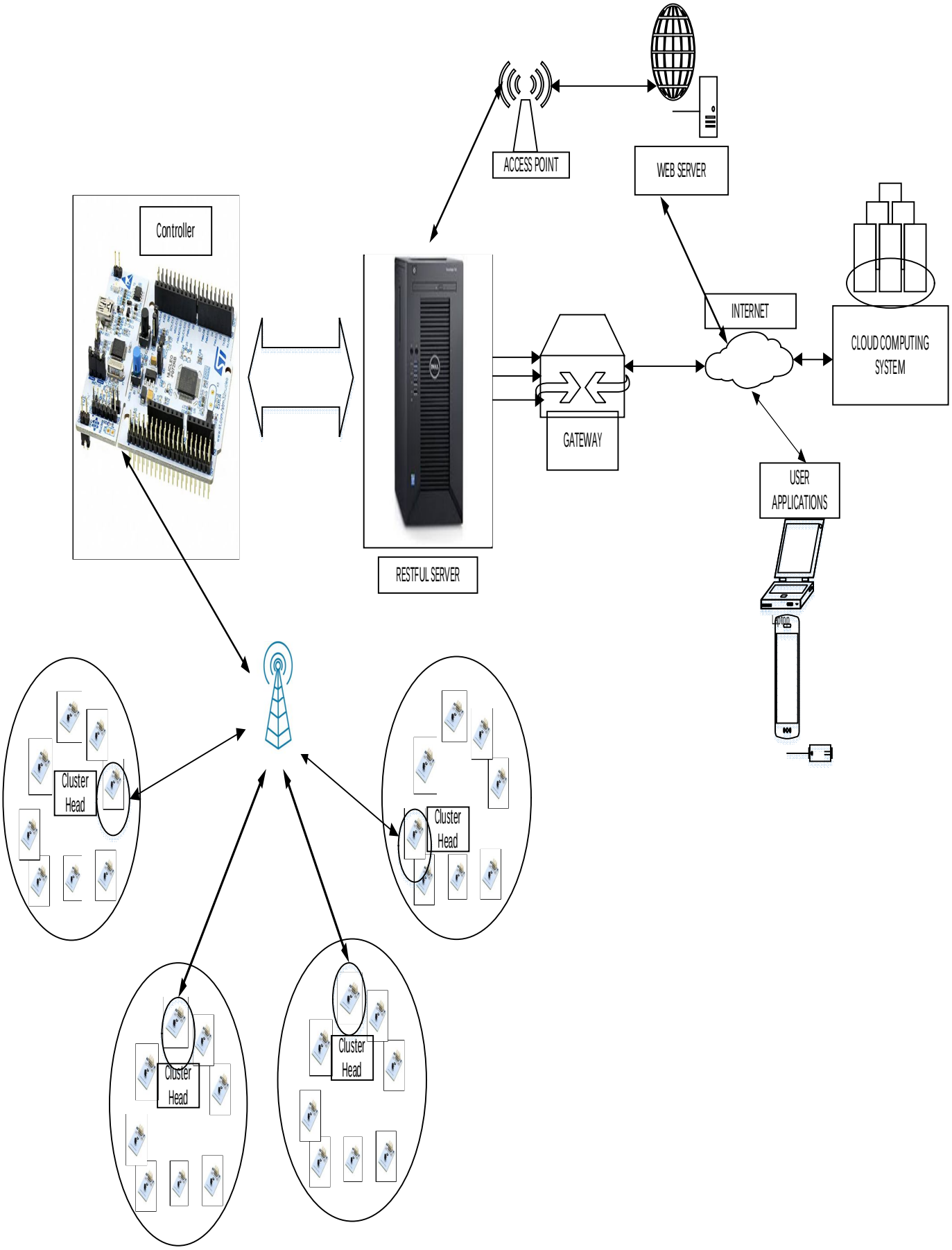

Figure 1:A Prototype IoT Network

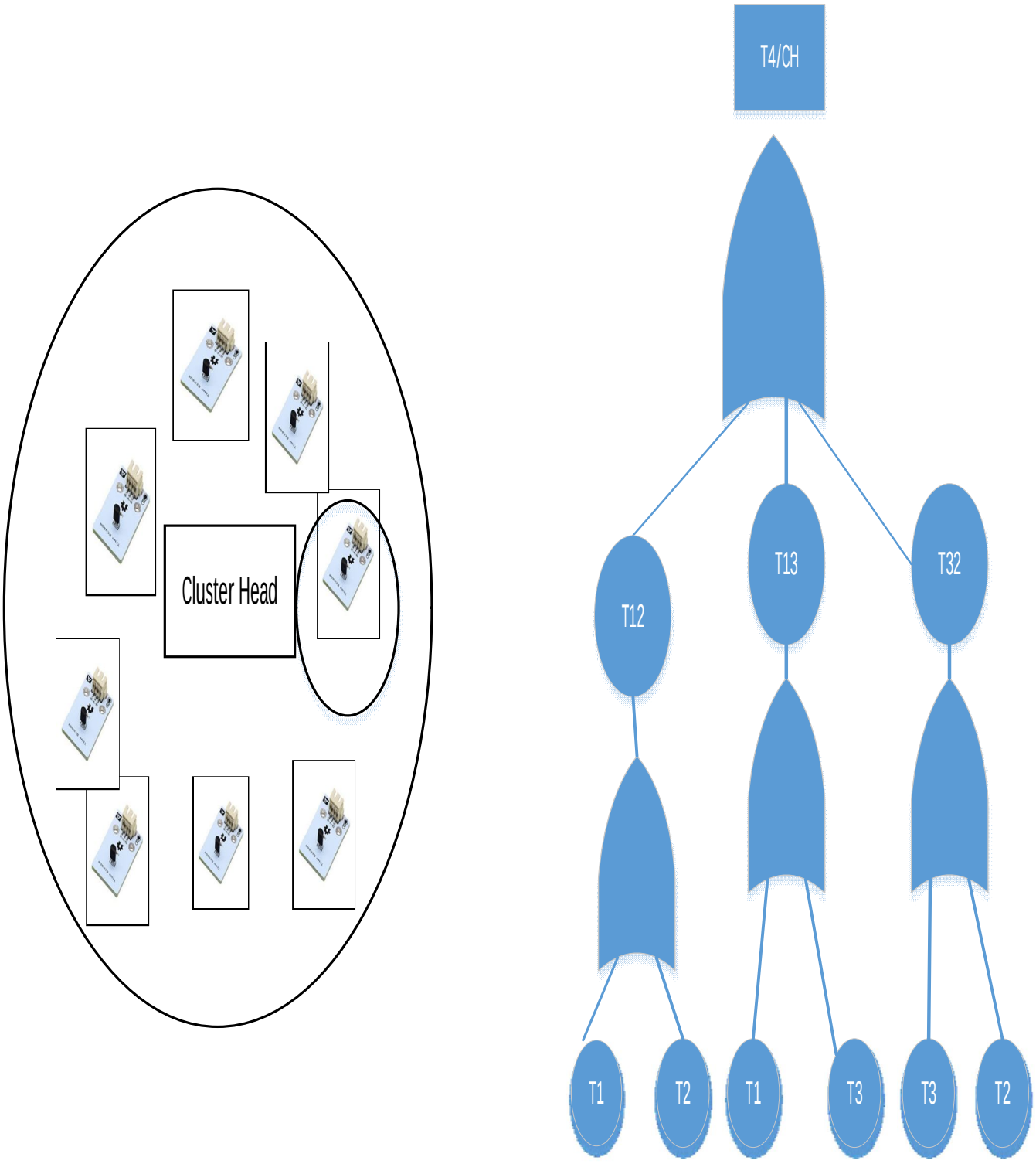

Figure 2: Converting a Cluster to a FTA Diagram

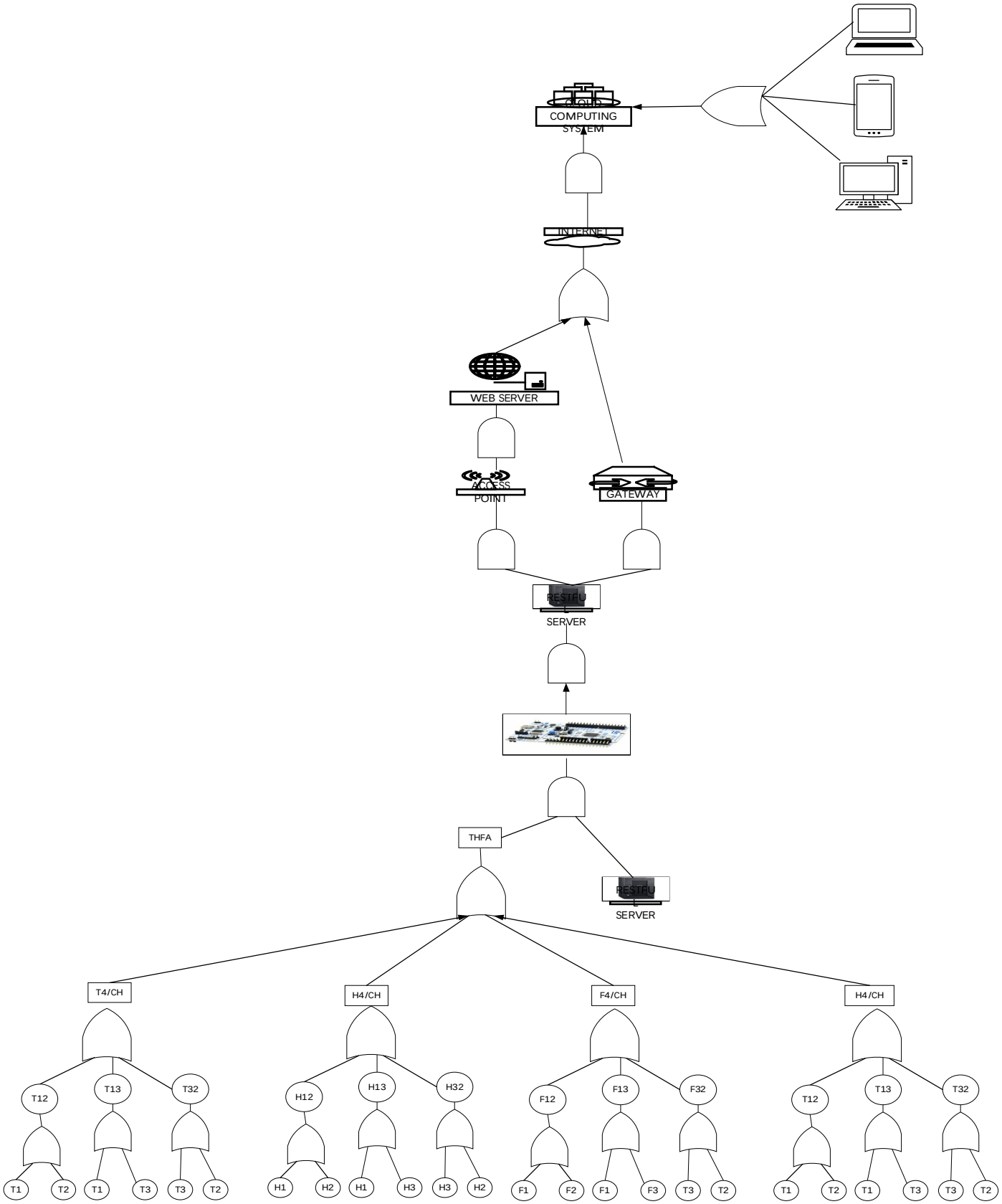

Figure 3: FTA Diagram for Prototype IoT Network

Table 1: Computation of 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	1.00	OR	T1	T2				
				0.950	0.950				0.950
4	Temp-sensor-3	0.95							0.950
5	T23-Dummy	1.00	OR	T2	T3				
				0.950	0.950				0.950
6	T13-Dummy	1.00	OR	T1	T3				
				0.950	0.950				0.950
7	Temp-Sensor-4	0.95	OR	T12	T23	T13			0.950
				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	1.00	OR	H1(0.950)	H2(0.950)				0.950
12	H23-Dummy	1.00	OR	H2(0.950)	H3(0.950)				0.950
13	H31-Dummy	1.00	OR	H3(0.950)	H1(0.950)				0.950
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							0.950
11	FAN-3	0.95							0.950
12	F12-Dummy	1.00	OR	F1(0.950)	F2(0.950)				0.950
13	F23-Dummy	1.00	OR	F2(0.950)	F3(0.950)				0.950

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	1.00	OR	F3(0.950)	F1(0.950)				0.950
15	Humidity-sensor-4	0.95	OR	T12(0.950)	T23(0.950)	T13(0.950)			0.950
16	THFA	0.95	OR	T4(0.950)	H4(0.950)	F4(0.950)	H4(0.950)		0.950
18	CONTROLLER	0.88	AND	THFA(0.950)					0.880
	Server	0.85	AND	CONTROLLER(0.880)					0.850
19	POINT	0.90	AND	Server (0.880)					0.880
20	GATEWAY	0.91	AND	Server (0.880)					0.880
21	WEBSERVER	0.93	AND	POINT 0.880					0.880
22	INTERNET	0.95	AND	WEBSERVER	GATEWAY				0.880
				0.880	0.880				
23	COMPUTING	0.84	AND	Internet (0.880)	User Terminal (0.8400				0.840

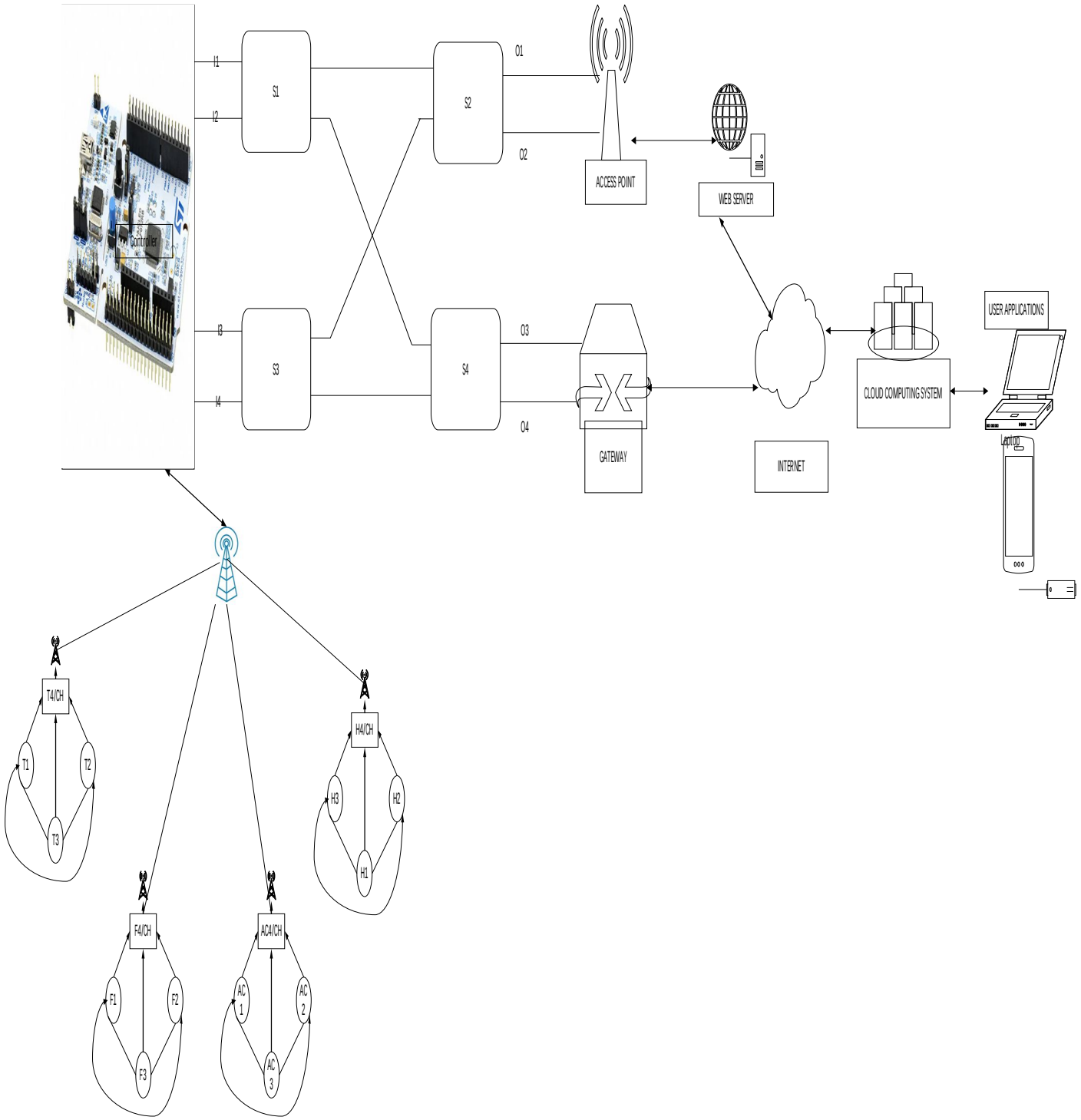

Figure 5: Modified IoT network – Crossbar network introduced at Services Level

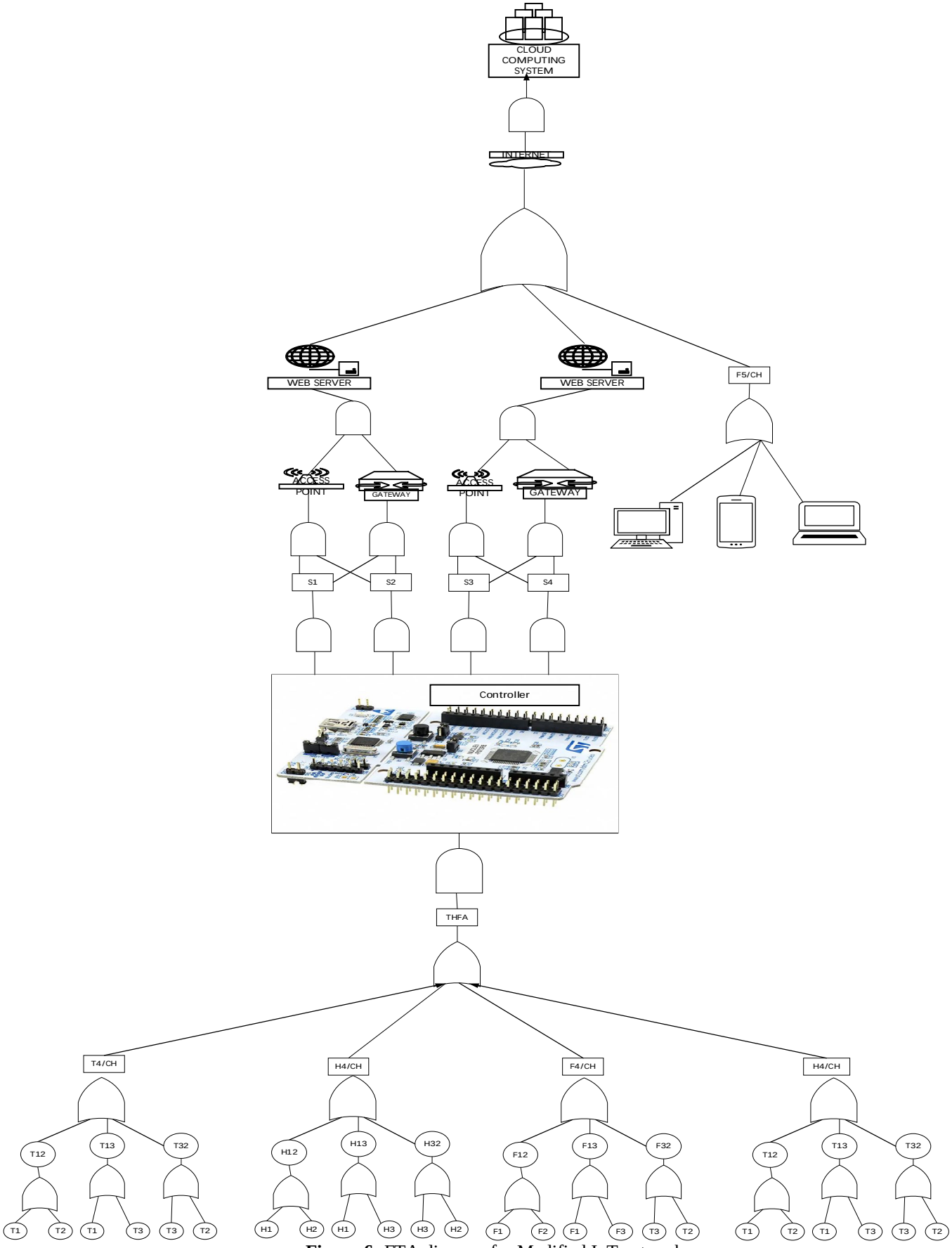

Figure 6 :FTA diagram for Modified IoT network

Table 2: Revised FTA Computations for Revised IoT Network

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	1.00	OR	T1	T2				
				0.950	0.950				0.950
4	Temp-sensor-3	0.95							0.950
5	T23-Dummy	1.00	OR	T2	T3				
				0.950	0.950				0.950
6	T13-Dummy	1.00	OR	T1	T3				
				0.950	0.950				0.950
7	Temp-Sensor-4	0.95	OR	T12	T23	T13			0.950
				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	1.00	OR	H1(0.950)	H2(0.950)				0.950
12	H23-Dummy	1.00	OR	H2(0.950)	H3(0.950)				0.950
13	H31-Dummy	1.00	OR	H3(0.950)	H1(0.950)				0.950
14	Humidity-Sensor-4	0.95	OR	H12	H23	H31			0.950
				0.950	0.950	0.950			
15	FAN-1	0.95							0.950
16	FAN-2	0.95							0.950
17	FAN-3	0.95							0.950
18	F12-Dummy	1.00	OR	F1(0.950)	F2(0.950)				0.950
19	F23-Dummy	1.00	OR	F2(0.950)	F3(0.950)				0.950

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	
20	F31-Dummy	1.00	OR	F3(0.950)	F1(0.950)				0.950
21	Humidity-sensor-4	0.95	OR	T12(0.950)	T23(0.950)	T13(0.950)			0.950
22	THFA	0.95	OR	T4(0.950)	H4(0.950)	F4(0.950)	H4(0.950)		0.950
23	CONTROLLER	0.88	AND	THFA(0.950)					0.880
24	Server-1	0.90	AND	CONTROLLER(0.880)					0.900
25	Server-2	0.90	AND	CONTROLLER(0.880)					0.900
26	Server-3	0.90	AND	CONTROLLER(0.880)					0.900
27	Server-4	0.90	AND	CONTROLLER(0.880)					0.900
28	POINT-1	0.90	AND	Server -1 (0.90)	Server 2 (0.90)				0.900
29	GATEWAY-1	0.91	AND	Server -1 (0.90)	Server 2 (0.90)				0.900
30	POINT-2	0.90	AND	Server -3 (0.90)	Server 4 (0.90)				0.900
31	GATEWAY-2	0.91	AND	Server -3 (0.90)	Server 4 (0.90)				0.900
32	WEBSERVER-1	0.93	AND	POINT-1 0.900	Gateway-1 (0.900)				0.900
33	WEBSERVER-2	0.93	AND	POINT-2 0.900	Gateway-2 (0.900)				0.900
34	INTERNET	0.95	AND	WEBSERVER-1 0.900	WEBSERVER-2 0.900	User Terminal) 0.940			0.940
34	COMPUTING	0.84	OR	Internet (0.940)	User Terminal (0.940				0.940

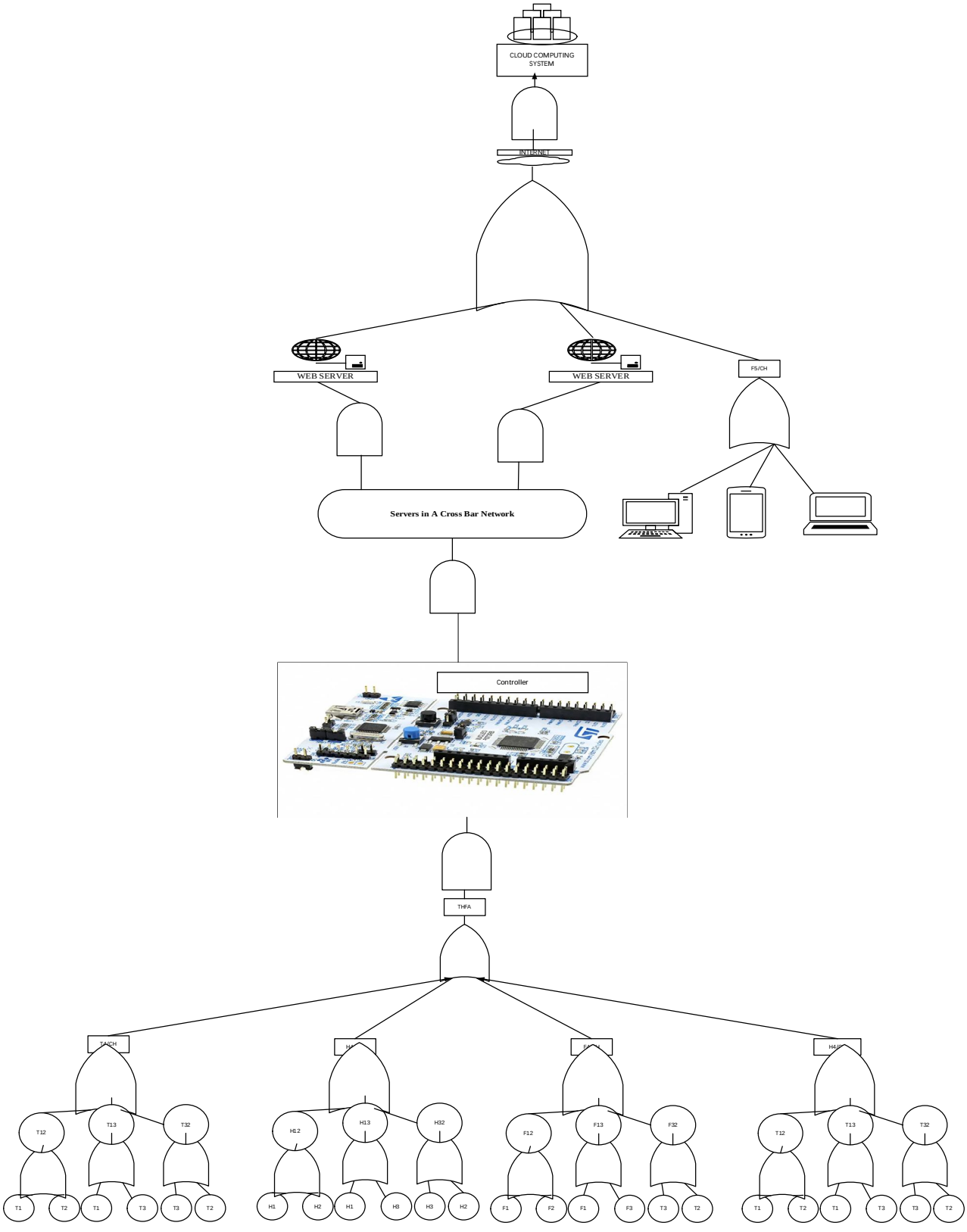

Figure 7: Modified FTA Diagram with Crossbar Network Included at services Layer

Table 3: Revised FTA Computations with a Crossbar network introduced at services Layer

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	1.00	OR	T1	T2				
				0.950	0.950				0.950
4	Temp-sensor-3	0.95							0.950
5	T23-Dummy	1.00	OR	T2	T3				
				0.950	0.950				0.950
6	T13-Dummy	1.00	OR	T1	T3				
				0.950	0.950				0.950
7	Temp-Sensor-4	0.95	OR	T12	T23	T13			0.950
				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	1.00	OR	H1(0.950)	H2(0.950)				0.950
12	H23-Dummy	1.00	OR	H2(0.950)	H3(0.950)				0.950
13	H31-Dummy	1.00	OR	H3(0.950)	H1(0.950)				0.950
14	Humidity-Sensor-4	0.95	OR	H12	H23	H31			0.950
				0.950	0.950	0.950			
15	FAN-1	0.95							0.950
16	FAN-2	0.95							0.950
17	FAN-3	0.95							0.950
18	F12-Dummy	1.00	OR	F1(0.950)	F2(0.950)				0.950
19	F23-Dummy	1.00	OR	F2(0.950)	F3(0.950)				0.950

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	
20	F31-Dummy	1.00	OR	F3(0.950)	F1(0.950)				0.950
21	Humidity-sensor-4	0.95	OR	T12(0.950)	T23(0.950)	T13(0.950)			0.950
22	THFA	0.95	OR	T4(0.950)	H4(0.950)	F4(0.950)	H4(0.950)		0.950
23	CONTROLLER	0.88	AND	THFA(0.950)					0.880
24	Server Crossbar network-1	0.998	OR	CONTROLLER(0.880)					0.998
25	WEBSERVER-1	0.93	AND	Server Crossbar network-1 (0.998)					0.998
26	WEBSERVER-2	0.93	AND	Server Crossbar network-1 (0.998)					0.998
27	INTERNET	0.95	AND	WEBSERVER-1 0.998	WEBSERVER-2 0.998	User Terminal) 0.998			0.998
28	COMPUTING	0.84	OR	Internet (0.9980)					0.998