

Article

Not peer-reviewed version

Enhancing the Fault Tolerance of a Multi-Layered IoT Network through Rectangular and Interstitial Mesh in the Gateway Layer

[Sastry Jammalamadaka](#)*, [Bhupati CH](#), Sasi Bhanu Jammalamadaka, Kamesh Balakrishna Duvvuri, [Raja Rao Budaraju](#)

Posted Date: 9 August 2023

doi: 10.20944/preprints202308.0642.v1

Keywords: IoT networks Services Layer; Gateway Layer; Rectangular and Interstitial Mesh networks; Computing Fault tolerance



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Enhancing the Fault Tolerance of a Multi-Layered IoT Network through Rectangular and Interstitial Mesh in the Gateway Layer

JKR Sastry ^{1,*}, Bhupati Ch ², J S Bhanu ³, DBK Kamesh ⁴ and Rajarao Budaraju ⁵

¹ K L Deemed to be University

² K L Deemed to be University

³ CMR College of Engineering and Technology

⁴ Mallareddy Engineering College for women

⁵ Oracle America Inc

* Correspondence: drsastry@kluniversity.in

Abstract: Most IoT systems meant for implementing mission-critical systems are multi-layered based. The majority of the computing is done in the services and gateway layer. The Gateway layer connects the Internal section of the IoT to the cloud through the Internet. The failure of any node between the services servers and the gateways will isolate the entire network leading to zero tolerance. A networking topology is required to connect the service servers to the gateways, yielding 100% fault tolerance. Empirical formulation of the model chosen to connect the service's servers to gateways through routers is required to compute the fault tolerance of the network. Rectangular and Interstitial Mesh have been proposed in this paper to connect the service servers to the gateways through the servers that yield 0.999 fault tolerance of the IoT network. An empirical formulation is also presented using which the Fault tolerance of the IoT network can be computed. An enhancement of 11% in the fault tolerance capability of an IoT network has been achieved by implementing Rectangular and Interstitial Mesh in the gateway layer of the network.

Keywords: IoT networks; services layer; gateway layer; Rectangular and Interstitial Mesh networks; computing fault tolerance

1. Introduction

In an IoT network, many devices such as sensors, actuators, servers, gateways, controllers, base stations etc. [1] communicate as per the established communication system. IoT networks range from small local to remotely connected Global systems [2]. Many small devices (Nano Scale) and newer communication Technologies [3] are being developed and used daily.

Many smaller devices are evolving rapidly, and millions of devices are connected to the cloud through the Internet. The direct connection of sensors/actuators to the cloud requires minimum infrastructure. However, implementing such a system leads to too many bottlenecks, including latency, speed of data transmission, data handling capability and proprietary protocols.

The computation, storage and network resources are brought closer to the IoT devices through the implementation of the edge/fog computing method, which helped in eliminating the bottle networks involved when sensors are directly communicating with the cloud through the Internet. (Pan and McElhannon, 2018[4]; Adhinugraha et al., 2020) [5].

A gateway connects the local devices to the Internet through routers wherever required. A gateway helps small devices connect to the Internet, thus avoiding the necessity of connecting the small devices directly to the Internet. A gateway allows small devices in an IoT network to be connected to traditional networks [6,7]. A gateway is like a relay point to transmit data from a local IoT network to a Global network.

The way the small devices are connected to the Gateway through the routers and the way the devices communicate through the Gateway is a major issue. The networking topology will be used to

connect the devices to the Gateway through a major challenge. Most of the topologies presented in the literature concentrated on cost reduction rather than enhancing or maintaining the fault tolerance of the IoT network. The number of paths that get reduced when a fault happens depends on the Routing topology [8]. It takes time to reconstruct networking topology when a fault happens, causing a delay in response time. This is unacceptable when the IoT system is implemented for a real-time system.

Several gateways may have to be used to handle the traffic. The use of multiple gateways ensures that there is no single point of failure [9,10]. An IoT network's reliability is also enhanced when multiple gateways are used.

The network traffic is expected to divert to another gateway when a gateway fails. Network broadcasting is generally used to find an alternative gateway [11].

When many gateways fail at once, broadcasting the message to all the nodes will block the entire network. Locating a replacement gateway located many hops away increases communication latency [12]. A Voronoi Model can be used to determine distance coverage to a gateway [13,14].

Several networking topologies can be used to establish a network between the services servers, routers, and gateways. Mesh network topology reduces network congestion to the cloud [15]. Mesh networking topology helps locate the processing, storage, or analytic services near the location from which data is routed to the cloud, thereby reducing network traffic [16]. In the case of mesh networks, sensors/servers are connected to the routers and the routers to the gateways. The data is processed at the devices (servers, routers and gateways). These devices are used not only for computing but also for routing. Mesh network is reliable, and there is no single point of failure [17].

Numerous routes exist in a mesh network which can be used for transmitting the data. A mesh network can be easily expandable by adding many processing units, including servers, Routers and gateways. The multi-hop approach can cover larger areas of processing devices [18]. The devices formed into a mesh network need not be equidistant. The devices are located such that they are within the broad costing range. There should be routes from the sensing/servicing devices so that data is moved to one of the gateways in the network. When any Gateway is out of order, other operational gateways must share the load.

A Voronoi diagram can be used to partition the processing nodes based on the nearest neighbour concept [19,20]. Voronoi diagram (VD) represents a road network with weights attached to the roots instead of considering Euclidean distances between the devices. A network graph can be developed considering the relationships between the computing resources with the help of a network Voronoi diagram (NVD)[21–23]. A higher-order Voronoi diagram can be constructed (HOVD) by segregating the cells in the network into different regions and then connecting the regions [24,25]. A reverse k-nearest neighbour method (RkNN) can also be used to regroup the cell [26]. Most of the focus presented in the literature is reaching out to the nearest neighbour node as fast as possible so that communication progresses in the event of failure of one or more processing nodes. None have shown the extent to which the fault tolerance of the network is improved in quantitative terms.

Modern IoT networks are built using multiple layers, including devices, controllers, services, Gateway, and the cloud. Each layer implements a different networking topology. In addition to computing the fault tolerance of a specific layer, there is a need to compute the overall fault tolerance of the IoT network. Sensors or not connected to the routers directly. Most of the processing is done in the services layer. The use of mesh networks for connecting the service's servers to the gateways has yet to be discussed. A pragmatic method to computing the fault tolerance of the mesh network considering the services servers, routers, and gateways still needs to be attempted.

2. Related work

T. Saha et al. [27] proposed a gateway which acts as an interface between the Internet and the rest of the IoT network. The Gateway is designed to deal with hardware software and connection failures and the overall load balance of the network. Their proposed framework involves a set of observers connected with the Gateway and sensors connected with the observers. The framework built into the

observers comprises prevention and detection algorithms to prevent communication failures between the sensor node and Gateway, provide alternative reliable transmission paths, and detect node faults in early stages. However, they have yet to present the extent of enhancement of the fault tolerance capacity of the IoT network due to the implementation of proposed models.

Ma et al. [28] proposed a method to determine the communication paths between nodes in a wireless sensor network (WSN). However, their method assumes an environment aggregating data from the sensor nodes to the sink nodes. The IoT sensor network assumes that the deployed resources cooperate rather than aggregate the data into one node. Using a multi-routing tree to construct the topology guarantees fault tolerance while minimizing power consumption.

Ismail et al. [29] proposed a resource allocation method assuming a mobile network environment. Their method is to find the nearest access point for the device to communicate. A mobile device can connect to multiple access points simultaneously. This differs from the IoT environment in which communication is performed by selecting one Gateway from the candidate's gateways.

Li et al. [30] researched a method to construct a routing tree for a P2P network in distributed interactive applications by using a genetic algorithm. Their method aims to determine the route between two nodes in the environment, optimize the communication speed, and do not consider fault tolerance.

Karthikeya et al. [31] proposed a method to optimize the placement of the Gateway. Their method can minimize the introduction cost of the Gateway into the environment by optimizing the gateway placement.

Kim et al. [32] researched a method to solve the resource allocation problem to the Gateway in the IoT environment. Their method uses a genetic algorithm to find an optimum path from a source node to a destination node. This method does not aim at the quantification of improvement in the fault tolerance of the network in a specific layer or the entire network. [33] Takahashi, R have proposed a method to generate a fault-tolerant networking and routing method using a genetic algorithm such that all the communication paths do not concentrate on one Gateway. Leonardi et al., 2018 [34], and Rondón et al., 2019 [35] have proposed a mesh network for massive device deployment to ensure no single point of failure. They have not discussed fault rate computations considering the mesh networks.

He et al., 2007 [36] and Seyedzadegan et al., 2013 [37] have presented gateway management techniques, such as optimal gateway placement, to handle massive devices and high traffic demand within an IoT network. Shih and Wu, 2016 [38] have presented multi-protocol gateways to support heterogeneous networks. They have yet to focus on the issue of fault tolerance.

Jean-Philippe et al., 2004 [39] have expressed that the Mesh networks also are prone to both unplanned blackouts (Natural disasters, overload, wear out, bugs attacks on devices) and planned blackouts (Hardware replacements and maintenance). They have expressed that a gateway recovery strategy is required to maintain network reliability.

Médard et al. (2002). [40] and Choi et al. (2004) [41] proposed loopback methods through the provision of pre-computed backup paths, which should be used during a network failure.

Lakshmanan et al. (2009). [42] and Kawai et al. (2014) [43] have proposed to use of multi-gateways to avoid traffic load on a specific gateway.

Adhinugraha et al., 2021 [44] have provided pre-computed k backup routers using an order HVD. All these works focussed on a failure of a single gateway. They have not considered the issue of the failure of multiple chained gateways.

In the first method, the reserved resources will only be utilized if something happens in the network. Unlike pre-computed resources, the second method does not require reserved or pre-computed resources since the alternate plan is determined by embedding a specific recovery algorithm in the network after the incident occurs. The main advantage of calculating the recovery plan on-the-fly is to reduce the allocation of unnecessary resources. The cost of reserving resources depends on the resources needed as an alternate plan.

The authors of Kim and Shin (2011) [45] proposed an automatic recovery system (ARS) to enable a wireless mesh network (WMN) to recover from local link failures to maintain network availability automatically.

Route calculations such as SDNMesh and RADAR have been proposed by (Gilani et al., 2020) [46] (and Sarkar et al., 2007) [47], respectively, from which effective routes can be determined after node failures.

Mahiddin and Sarkar, (2019).) [48] have proposed a method to handle the traffic so that every Gateway will have a small hit. Minh et al. (2014) [49] have proposed a method called OEMAN to assign one mobile device as a substitute access point when the main Gateway fails. [50] Kiki Adhinugraha has presented a generalized model based on the k-hops Voronoi diagram to compute k backup gateways and evaluate different gateway diversion strategies to divert a router output to a gateway with the minimum hops. Kiki Adhinugraha et al. [51] have proposed a hops Voronoi diagram (HVD) method to distribute. Dependent routers are based on the nearest number of hops from the router. Moghaddam et al. [52] have classified the mechanisms proposed in the literature, which claim to be improving or retaining the fault tolerance level of the IoT network.

Research Gap

The methods proposed in the literature focussed on improving the fault tolerance of an IoT network, considering typical implementation ignoring the presence of several layers in the network and the need to communicate between the layers. Empirical formulations that can be used to compute the fault tolerance considering a single layer or entire network have yet to be presented. The fault tolerance capability of the IoT network has been explained concerning the availability of an alternate path for communication in the event of the failure of a device. The realistic computation of the fault tolerance level of the IoT network has yet to be explained.

Research contributions

1. This paper proposes a rectangular mesh network with interstitial meshes with built-in redundancy routers to serve as backups to the other surrounding routers.
2. An empirical formulation which can be used for computing the fault tolerance of such a network can be computed using a probability model.
3. A computation method/approach which can be used to compute the fault tolerance of the entire IoT network.

3. The method

The flow diagram that depicts the execution of the proposed methods is shown in Figure 1. The blocks relating to Metrics development, Initial prototype development and its FTA development, implementing a crossbar network in the device layer, implementing fault detection and isolation method in the sensing and actuating devices to counter the possible fault injection in the device layer and its related FTA model and the improvement in Fault tolerance capability of the IoT network have been explained by Bhupathi et al. [53,54].

Further improvements have been made to the IoT network by adding a second base station to the controller layer. The second base station has been connected to the cluster heads through an intelligent relay-based network with built-in redundancy to tackle the issue of the remoteness of the base station from the cluster heads. An intelligent pathfinding algorithm is implemented in each relay to find the shortest deal with the least traffic to reduce the latency tremendously [55].

The fault tolerance capability of the IoT network has been improved further by implementing the Load balancing feature within the controllers, which are networked through an I2C network [56]. The service servers are added with an intelligent service to estimate missing data [57].

The IoT network is further improved by adding Rectangular and Interstitial Mesh in the Gateway Layer to improve the fault tolerance in this network layer. A new reliability model has been proposed to compute fault tolerance considering a rectangular Interstitial Mesh network.

An FTA diagram for the newly introduced network is developed and combined with the FTA

diagram of the prototype model. Fault tolerance values have been computed through the generation of a fault table.

A comparison of the Stage improvements in the fault tolerance of the IoT network due to the implementation of different methods in the Gateway has been presented.

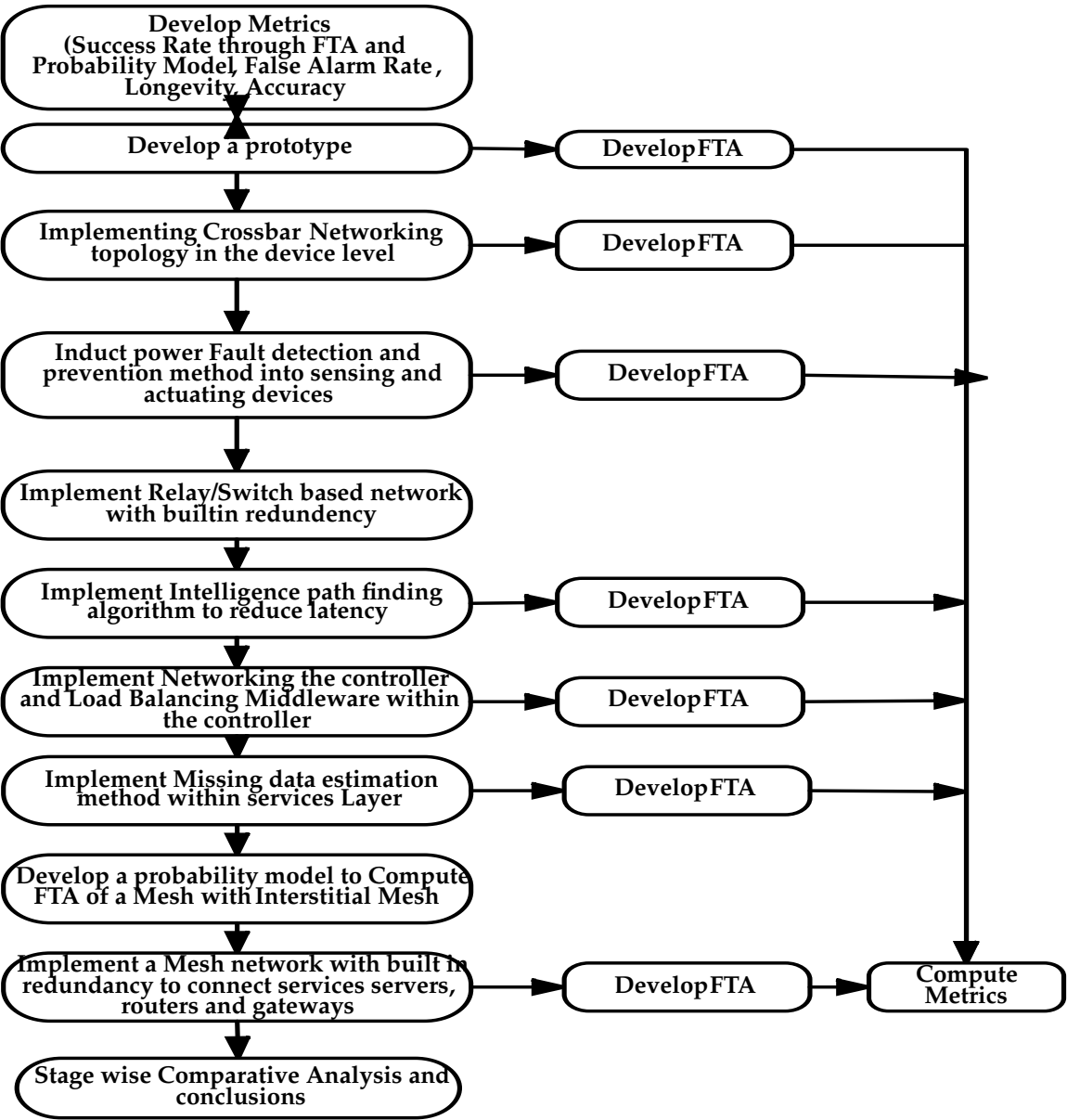


Figure 1. The overall computational method for enhancing the fault tolerance up to the Gateway layer.

4. Example IoT network

Different methods implemented at various levels for enhancing the IoT network’s fault tolerance are shown in Table 1. A sample IoT network is presented in Figure 2. The sample IoT system implements the following in the device, controller, and services layer of the IoT network to enhance the fault tolerance of the network.

Table 1. Implementations in the Sample IoT network until the services layer.

Layer	Innovation implemented	Reference
FTA computation Model	A hybrid model that helps compute the FTA considering a Linier and probability model	[53]
Device Layer	Predicting the occurrence of a Power fault and mitigating it through isolation procedure Implementing a crossbar network in the device layer	[54]
Controller Layer	Develop a parallel peer-to-Peer communication system to connect the cluster heads to the 1st Base station. Develop a redundant network using different topologies connecting the 2nd Base station and cluster heads. A new algorithm to find the shortest path to connect a cluster head to a 2nd base station	[55]
	Networking Controllers and implementing load balancing within the controllers	[56]
Service Layer	Connecting the controller to the servers of the service through a crossbar network Implementing a machine learning-based missing data estimation model	[57]

The devices in a cluster are linearized and connected to a cluster head free from sensing or actuating function. A crossbar network connects the linearized clusters’ outputs to the cluster heads. The cluster heads are connected to the two base stations, which are widely situated for transmitting data from sensors into base stations. Both the base stations are interfaced with three microcontrollers in peer-to-peer mode. The microcontrollers are interconnected through an I2C network which caters for connectivity failures of the controllers to the crossbar network. The controllers are connected to service servers using a crossbar network. The services server receives the requests from the devices or users, executes the service-related code, and transmits the results to controllers or the user. A new service is implemented in the service server to predict if the missing data is not transmitted due to a device’s failure in the layer.

For developing a Fault tree of the sample IoT network, the crossbar networks in the device and controller layer have been replaced by a single device whose success rate is computed using probability models related to crossbar networks. The connectivity between the services servers, routers and gateways is complicated. Voluminous data flow happens in this layer, and failure of devices in this layer hampers the fault tolerance capability of the IoT network quite drastically. The fault tree of the residual network and a table showing the success rate computations are generated through the algorithms presented by Bhupati et al. [53].

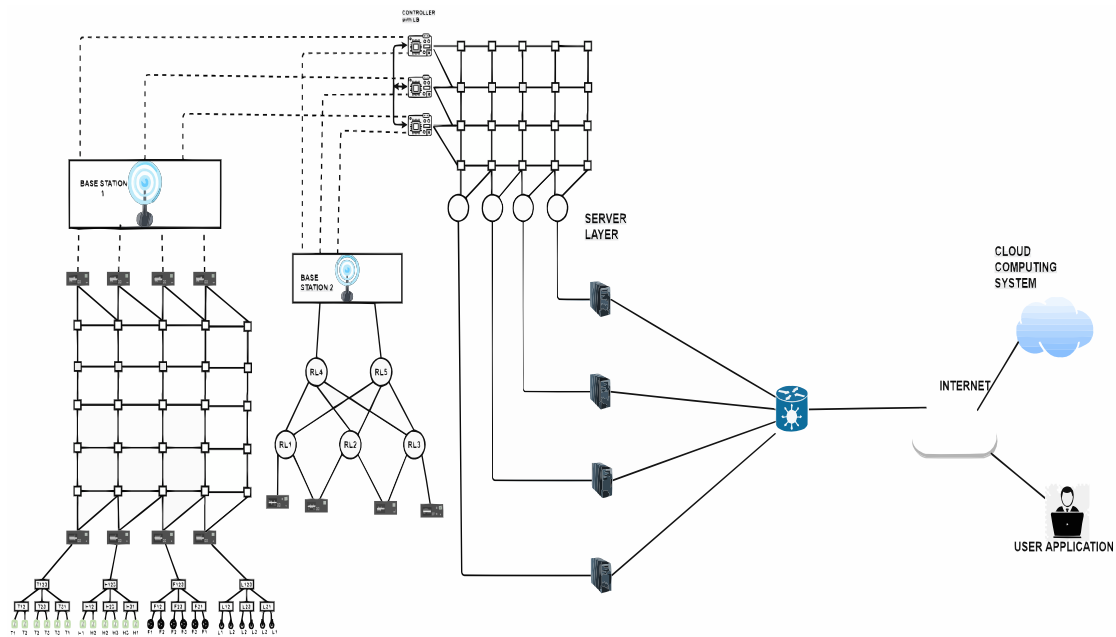


Figure 2. Example IoT network.

5. Methods and Techniques

5.1. Success rate computation method for Rectangular and Interstitial Mesh network

In a mesh network, all nodes are pressing nodes. The network can be formed using the services servers, routers, and Gateways. All the non-boundary nodes have 4 four incident links. To send a message from a node to a node that is not a neighbour, a path from the node form where the message is initiated to a destination node must be identified. The message must be forwarded involving the intermediate nodes along the path. A mesh network loses its property that there should be four links to communicate from a node when it breaks down for any reason. To provide tolerance, redundant nodes are added. The redundant nodes are switched in when any of their neighbouring nodes fails.

A mesh network involving 4 service servers, 4 gateways and 10 Routers is shown in Figure 3. 4 Routers are spare, and they will come into force only when its neighbouring node fails.

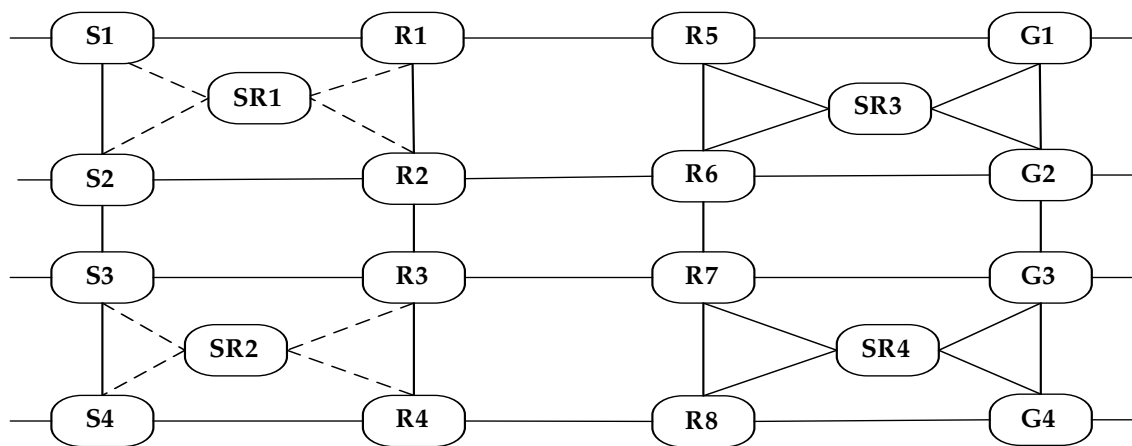


Figure 3. Mesh network for connecting services servers to the Gateway.

The reliability of a mesh network is the probability that the mesh property is retained. The fault tolerance of such a network is computed using equation (1).

R = is the reliability or the success rate of servers, routers, or gateways = 0.98

N = number of rows in the network = 4

M = number of columns in the network = 4

C = number of clusters in the network = $(N*M)/4$ with each cluster formed using 4 primary nodes and a spare node = 4

n = number of sub-meshes row wise = 2

m = number of sub-meshes column wise = 2

k = number of possible allocations = $N/(n) * M/m = 4$ which lead to 1-of- k model.

The reliability of such a model can be computed using equation (1) S = probability that an $n * m$ sub mesh can be allocated

$$= 1 - [1 - R^{n*m}]^k = 1 - [1 - 0.98^4]^4 = 0.99 \quad (1)$$

5.2. Revised IoT network

The following changes have been made in the Services Layer of the network.

1. A mesh network has been established connecting the Services servers, routers and gateways.
2. One spare router is included for every four processing nodes.
3. All the gateways are connected to the cloud through the Internet.

The revised network is shown in Figure 4.

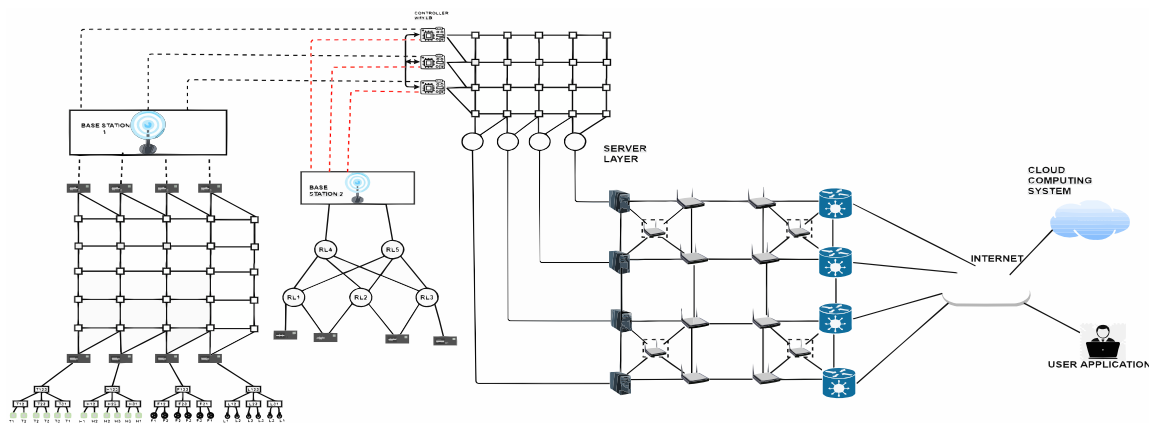


Figure 4. Revised IoT network with revisions in the Gateway Level

6. Experimentation and Results

6.1. FTA, Example, IoT network

Developing Fault tree for the revised IoT network

The crossbar network between the controllers and services servers is replaced by a single device assigned with a success rate that is the same as the crossbar network's success rate using its related probability model.

AN FTA diagram is generated using an algorithm described in [53]. After replacing networking topologies with a single device, the devices left in the diagram are connected using AND/OR conditions based on the precedence and data flow designed to build redundancy in the network. The modified FT diagram related to the revised IoT diagram is shown in Figure 5.

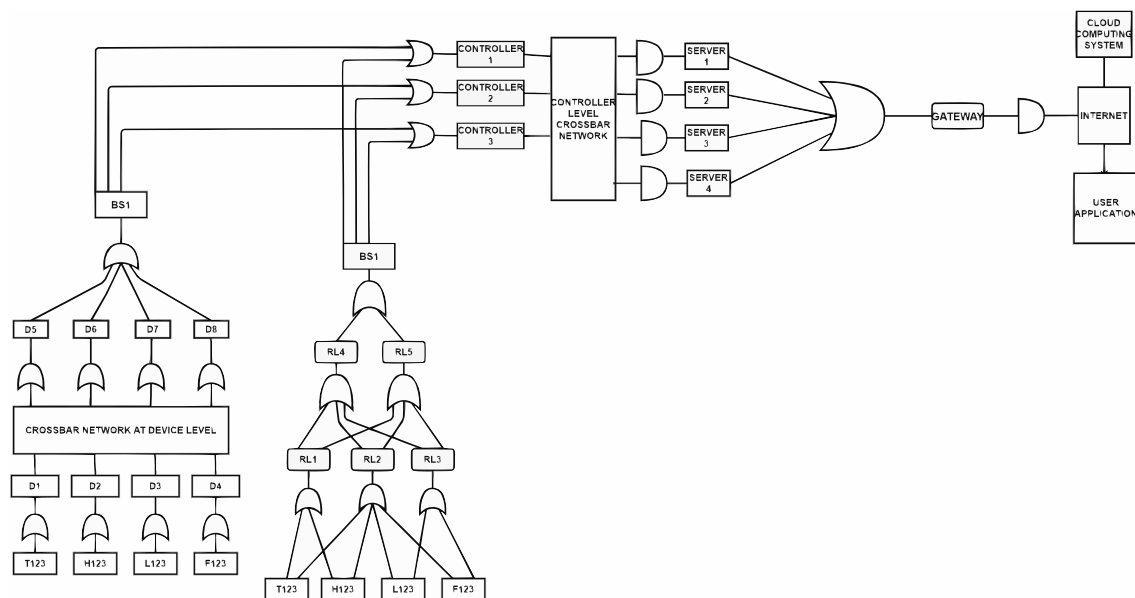


Figure 5. FTA Diagram for the revised IoT network

6.2. Success rate computation of Example IoT network

The success rates of every device are computed using its precedence relationships with other devices. The algorithm Bhupathi et al.[54] presented has been used to generate the success table, given the FTA diagram as the input. The generated Success table is presented in Table 2. From the table, it can be observed that the success rate of the revised IoT network is 0.980.

Table 2. success rate computations of an Example IoT network.

Sl.no	Device	Success Rate	Gates used For connection	Device Name D1 Success Rate S1	Preceding Devices Device Name D2 Success Rate S2	Device Name D3 Success Rate S3	Device Name D4 Success Rate S4	Combined Success Rate
1	Cluster Head1	0.950						0.950
2	Cluster Head2	0.950						0.950
3	Cluster Head3	0.950						0.950
4	Cluster Head4	0.950						0.950
5	D1	0.950	OR	Cluster Head1 0.950				0.950
6	D2	0.950	OR	Cluster Head2 0.950				0.950
7	D3	0.950	OR	Cluster Head3 0.950				0.950
8	D4	0.950	OR	Cluster Head4 0.950				0.950
9	Device Level Crossbar NW (DLCB)	0.987	OR	D1 0.950				0.987
10	Device Level Crossbar NW (DLCB)	0.987	OR	D2 0.950				0.987
11	Device Level Crossbar NW (DLCB)	0.987	OR	D3 0.950				0.987
12	Device Level Crossbar NW (DLCB)	0.987	OR	D4 0.950				0.987
13	D5	0.950	OR	DLCB 0.987				0.987
14	D6	0.950	OR	DLCB 0.987				0.987
15	D7	0.950	OR	DLCB 0.987				0.987
16	D8	0.950	OR	DLCB 0.987				0.987
17	BS1	0.950	OR	D5 0.987	D6 0.987	D7 0.987	D8 0.987	0.987
18	RL1	0.950	OR	Cluster Head1 0.950	Cluster Head2 0.950			0.950
19	RL2	0.950	OR	Cluster Head2 0.950	Cluster Head3 0.950			0.950
20	RL3	0.950	OR	Cluster Head3 0.950	Cluster Head4 0.950			0.950
21	RL4	0.950	OR	RL1 0.950	RL2 0.950			0.950
22	RL5	0.950	OR	RL1 0.950	RL2 0.950			0.950
23	BS2	0.950	OR	RL4 0.950	RL5 0.950			0.950
24	CONTROLLER 1	0.979	OR	BS1 0.987	BS2 0.950			0.987
25	CONTROLLER 2	0.979	OR	BS1 0.987	BS2 0.950			0.987
26	CONTROLLER 3	0.979	OR	BS1 0.987	BS2 0.950			0.987
27	CONTROLLER LEVEL CROSSBAR NW	0.970	CROSSBAR NW	CONTROLLER 1 0.987	CONTROLLER 2 0.987	CONTROLLER 3 0.987		0.987
28	SERVER 1	0.980	AND	CONTROLLER LEVEL CROSSBAR NW 0.987				0.967
29	SERVER 2	0.980	AND	CONTROLLER LEVEL CROSSBAR NW 0.987				0.967
30	SERVER 3	0.980	AND	CONTROLLER LEVEL CROSSBAR NW 0.987				0.967
31	SERVER 4	0.980	AND	CONTROLLER LEVEL CROSSBAR NW 0.987				0.967
32	GATEWAY	0.980	OR	SER 1 0.967	SER 2 0.967	SER 3 0.967	SER 4 0.967	0.980
33	INTERNET	0.980	OR	GATEWAY 0.980				0.980

6.3. FTA for revised IoT network

Developing Fault tree for the revised IoT network

The crossbar network between the controllers and services servers is replaced by a single device assigned with a success rate that is the same as the crossbar network's success rate using its related probability model.

AN FTA diagram is generated using an algorithm described in [53]. After replacing networking topologies with a single device, the devices left in the diagram are connected using AND/OR conditions based on the precedence and data flow designed to build redundancy in the network. The modified FT diagram related to the revised IoT diagram is shown in Figure 6.

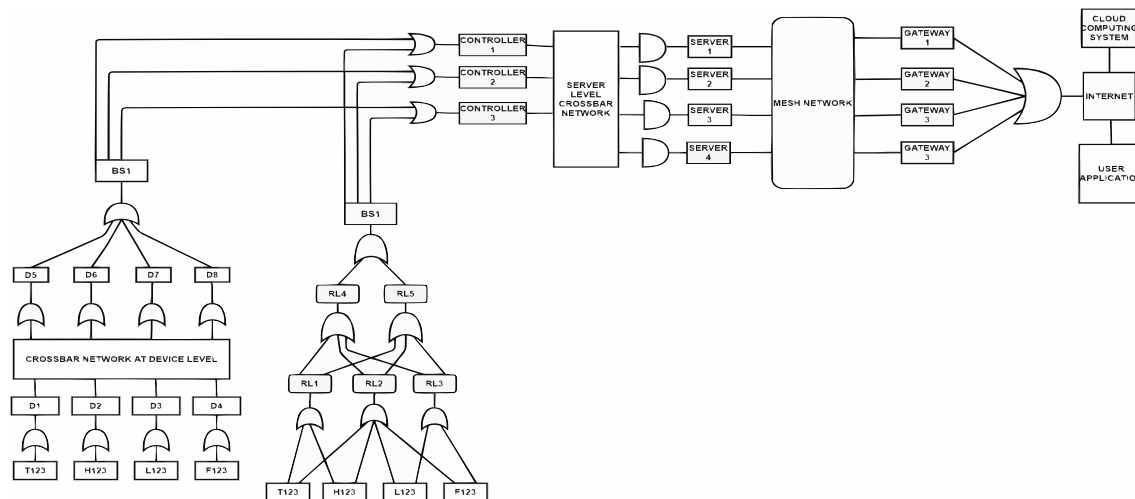


Figure 6. Revised FTA diagram

The algorithm presented by Bhupathi et al. [44] has been used to generate an FIA diagram for the revised IoT network. An algorithm they presented generates the success table, given the FTA diagram as the input. The generated Success table is presented in Table 3. The success rates of every device are computed using its precedence relationships with other devices. From the table, it can be observed that the success rate of the revised IoT network is 0.990.

Table 3. success rate computations of an Example IoT network.

Sl.no	Device	Success Rate	Gates used For connection	Preceding Devices				Combined Success Rate
				Device Name D1 Success Rate S1	Device Name D2 Success Rate S2	Device Name D3 Success Rate S3	Device Name D4 Success Rate S4	
1	Cluster Head1	0.950						0.950
2	Cluster Head2	0.950						0.950
3	Cluster Head3	0.950						0.950
4	Cluster Head4	0.950						0.950
5	D1	0.950	OR	Cluster Head1 0.950				0.950
6	D2	0.950	OR	Cluster Head2 0.950				0.950
7	D3	0.950	OR	Cluster Head3 0.950				0.950
8	D4	0.950	OR	Cluster Head4 0.950				0.950
9	Device Level Crossbar NW (DLCB)	0.987	OR	D1 0.950				0.987
10	Device Level Crossbar NW (DLCB)	0.987	OR	D2 0.950				0.987
11	Device Level Crossbar NW (DLCB)	0.987	OR	D3 0.950				0.987

Table 3. Cont.

Sl.no	Device	Success Rate	Gates used For connection	Device Name D1 Success Rate S1	Preceding Devices Device Name D2 Success Rate S2	Device Name D3 Success Rate S3	Device Name D4 Success Rate S4	Combined Success Rate
12	Device Level Crossbar NW (DLCB)	0.987	OR	D4 0.950				0.987
13	D5	0.950	OR	DLCB 0.987				0.987
14	D6	0.950	OR	DLCB 0.987				0.987
15	D7	0.950	OR	DLCB 0.987				0.987
16	D8	0.950	OR	DLCB 0.987				0.987
17	BS1	0.950	OR	D5 0.987	D6 0.987	D7 0.987	D8 0.987	0.987
18	RL1	0.950	OR	Cluster Head1 0.950	Cluster Head2 0.950			0.950
19	RL2	0.950	OR	Cluster Head2 0.950	Cluster Head3 0.950			0.950
20	RL3	0.950	OR	Cluster Head3 0.950	Cluster Head4 0.950			0.950
21	RL4	0.950	OR	RL1 0.950	RL2 0.950			0.950
22	RL5	0.950	OR	RL1 0.950	RL2 0.950			0.950
23	BS2	0.950	OR	RL4 0.950	RL5 0.950			0.950
24	CONTROLLER 1	0.979	OR	BS1 0.987	BS2 0.950			0.987
25	CONTROLLER 2	0.979	OR	BS1 0.987	BS2 0.950			0.987
26	CONTROLLER 3	0.979	OR	BS1 0.987	BS2 0.950			0.987
27	CONTROLLER LEVEL CROSSBAR NW	0.970	CROSSBAR NW	CONTROLLER 1 0.987	CONTROLLER 2 0.987	CONTROLLER 3 0.987		0.987
28	SERVER 1	0.980	AND	CONTROLLER LEVEL CROSSBAR NW 0.987				0.967
29	SERVER 2	0.980	AND	CONTROLLER LEVEL CROSSBAR NW 0.987				0.967
30	SERVER 3	0.980	AND	CONTROLLER LEVEL CROSSBAR NW 0.987				0.967
31	SERVER 4	0.980	AND	CONTROLLER LEVEL CROSSBAR NW 0.987				0.967
32	SERVER LEVEL MESH NW	0.980	MESH NW	SER 1 0.967	SER 2 0.967	SER 3 0.967	SER 4 0.967	0.999
33	GATEWAY 1	0.950	OR	SERVER LEVEL MESS NW 0.999				0.999
34	GATEWAY 2	0.950	OR	SERVER LEVEL MESS NW 0.999				0.999
35	GATEWAY 3	0.950	OR	SERVER LEVEL MESS NW 0.999				0.999
36	GATEWAY 4	0.950	OR	SERVER LEVEL MESS NW 0.999				0.999
37	INTERNET	0.980	MESH NW	GATEWAY 1 0.999	GATEWAY 2 0.999	GATEWAY 3 0.999	GATEWAY 4 0.999	0.999

7. Discussions

Mesh networks with built-in selected redundancy will provide very high-level fault tolerance of the IoT network. A probability model derived for the Mesh network will help compute the service layer's fault tolerance. Improving fault tolerance in the services layer will help achieve a very high fault tolerance considering the entire network. The services layer of the IoT network is the most crucial, as most computing is done in this layer. A stage-wise Fault tolerance improvement analysis, is provided in Table 4. The table shows that the IoT network's fault tolerance improved by 11% when a mesh network was introduced in the services layer (FTA changed from 0.980 to 0.990).

Table 4. Comparative Analysis of enhancements of fault tolerance in different layers of the IoT network.

Serial Number	Type of Network	Fault Tree Value
1	Prototype network [54]	0.717
2	Prototype with Changes Made in the device Levels – [55] (Fault prediction, mitigation and crossbar network implemented)	0.827
3	Prototype with Changes Made in the device Level and Base Station level (Introduction of dual networks to connect to two base stations and finding the shortest for the communication through 2nd base station) [55]	0.948
4	Prototype with Changes Made in the device Level and Base Station level with load balanced at the controller Layer (Controller interconnected through an I2C network and implementing middleware within the controllers) and connecting the controllers to the service's servers through a crossbar network [56]	0.980
5	Prototype with Changes Made in the device Level and Base Station level with load balanced at the controller Layer (Controller interconnected through an I2C network and implementing middleware within the controllers) and connecting the controllers to the service's servers through a crossbar network and implementing the prediction model for predicting the missing data [57]	0.980
6	Prototype with Changes Made in the device Level and Base Station level with load balanced at the controller Layer (Controller interconnected through an I2C network and implementing middleware within the controllers), connecting the controllers to the servers of the service through a crossbar network and implementing the prediction and estimating the to missing data and implementing a Mesh network in the services layer to connect the servers of the service to the gateways.	0.999

The fault tolerance capacity of the revised network and the mesh network included in the Services layer is compared with the fault tolerance computed based on node failure. The number of paths available for communication gets reduced in respect of NVD and HVD as the failure of the number of nodes increases. The fault tolerance of the MESH network implemented based on NVD and HVD is zero when there is a situation that 4 nodes fail at the same time. The fault tolerance of Mesh with Interstitial Mesh network remains the same even if 4 nodes fail due to the reasons of availability of the redundant nodes. Moreover, there is no empirical formulation for computing fault tolerance of the mesh network when NVD or HVD is used. At best, the NVD and HVD methods can compute the fault tolerance of the service layer, whereas the Mesh with Interstitial Mesh-based method helps to compute the fault tolerance of the entire network.

8. Conclusions

1. The services layer is the most critical in an IoT network, as most computing is done in that layer before the data is moved to the cloud through gateways.

2. Adding $\frac{1}{4}$ of nodes of the total nodes as standby or backup nodes will help the IoT network to be fully fault tolerant despite the failure of 4 Nodes at a time.

3. The Fault tolerance of the IoT network improved by 11% when a Mesh with an Interstitial Mesh network was implemented in the services layer.

4. An empirical formulation is developed, which helps compute the fault tolerance of the IoT network when a Mesh with Interstitial Mesh network is implemented in the services layer. At the same time, that is not the case with other models.

5. The Mesh with Interstitial Mesh network implemented in the services layer can be easily interconnected with networking topologies used in other network layers.

Table 5. A comparison of Rectangular and Interstitial Mesh with NVD and HVD model in the Gateway Layer.

Parameter	Mesh with Interstitial Mesh	NVD [21]	HVD [51]
Existence of empirical formulation for computing FTA	Yes	No	No
Total number of nodes in the network	24	16	16
Number of paths exiting in the network	16	16	16
Number of paths available when a node fails	16	12	12
Number of paths available when two nodes fail	16	8	8
Number of paths available when three nodes fail	16	4	4
Number of paths available when 4 nodes fail	16	0	0
FTA of Network when 4 Number of nodes fails	0.999	0.000	0.00

References

1. K. F. Khawaja and R. H. Bokhari, "Exploring the Factors Associated With Website Quality," Department of Technology Management, International Islamic University, Islamabad, Pakistan, vol. 10, pp. 37-45, 2010.
2. J. K. R. Sastry and T. S. Lalitha, "A framework for assessing the quality of a WEB SITE, PONTE," International Journal of Science and Research, vol. 73, 2017.
3. V. K. Mantri, "An Introspection of Web Portals Quality Evaluation," International Journal of Advanced Information Science and Technology, vol. 5, pp. 33-38, 2016.
4. V S Moustakis, "Website quality assessment criteria," Proceedings of the Ninth International Conference on Information Quality, 2004.
5. Nielsen, J. (2020), "10 usability heuristics for user interface design", Nielsen Norman Group, available at: <https://www.nngroup.com/articles/ux-research-cheat-sheet/> (accessed 3 May 2021).
6. Tognazzi, B. (2014), "First principles of interaction design (revised and expanded)", asking.
7. Shneiderman, B. (2016), "The eight golden rules of interface design", Department of Computer Science, University of Maryland
8. Law, R., Qi, S. and Buhalis, D. (2010), "Progress in tourism management: a review of website evaluation in tourism research", Tourism Management, Vol. 31 No. 3, pp. 297-313.
9. Shneiderman, B., Plaisant, C., Cohen, M.S., Jacobs, S., Elmqvist, N. and Diakopoulos, N. (2016), Designing the User Interface: Strategies for Effective Human-Computer Interaction, 6th ed., Pearson Higher Education, Essex.
10. Morales-Vargas, A., Pedraza-Jimenez, R. and Codina, L. (2020), "Website quality: an analysis of scientific production", Profesional de la Information, Vol. 29 No. 5, p. e290508.
11. Law, R. (2019), "Evaluation of hotel websites: progress and future developments", International Journal of Hospitality Management, Vol. 76, pp. 2-9.
12. Ecer, F. (2014), "A hybrid banking websites quality evaluation model using AHP and COPRAS-G: a Turkey case", Technological and Economic Development of Economy, Vol. 20 No. 4, pp. 758-782.
13. Leung, D., Law, R. and Lee, H.A. (2016), "A modified model for hotel website functionality evaluation", Journal of Travel and Tourism Marketing, Vol. 33 No. 9, pp. 1268-1285.
14. Maia, C.L.B. and FUiado, E.S. (2016), "A systematic review about user experience evaluation", in Marcus, A. (Ed.), Design, User Experience, and Usability: Design Thinking and Methods, Springer International Publishing, Cham, pp. 445-455.

15. Sanabre, C., Pedraza-Jimenez, R. and Vinyals-Mirabent, S. (2020), "Double-entry analysis system (DEAS) for comprehensive quality evaluation of websites: case study in the tourism sector", *Profesional de la Información*, Vol. 29 No. 4, pp. 1-17, e290432.
16. Bevan, N., Carter, J. and Harker, S. (2015), "ISO 9241-11 Revised: what have we learnt about usability since 1998?", in Kurosu, M. (Ed.), *Human-Computer Interaction: Design and Evaluation*, Springer International Publishing, Cham, pp. 143-151.
17. Rosala, M. and Krause, R. (2020), *User Experience Careers: If that a Career in UX Looks Like Today*, Fremont, CA
18. Jainari, MH., Baharum, A., Deris, F.D., Mat Noor, N.A., Ismail, R. and Mat Zain, NH. (2022), "A standard content for university websites using heuristic evaluation", in Arai, K. (Ed.), *Intelligent Computing. SA! 2022. Lecture Notes in Networks and Systems*, Springer, Cham, Vol. 506.
19. Jayanthi, B. and Krishnakumari, P. (2016), "An intelligent method to assess webpage quality using extreme learning machine", *International Journal of Computer Science and Network Security*, Vol. 16 No. 9, pp. 81-85.
20. Nikolic, N., Grljevic, O. and Kovacevic, A. (2020), "Aspect-based sentiment analysis of reviews in the domain of higher education", *Electronic Library*, Vol. 38, pp. 44-64.
21. Morales-Vargas, A., Pedraza-Jimenez, R. and Codina, L. (2023), "Website quality evaluation: a model for developing comprehensive assessment instruments based on key quality factors", *Journal of Documentation*
22. K. F. Khawaja and R. H. Bokhari, "Exploring the Factors Associated With Quality of Website," Department of Technology Management, International Islamic University, Islamabad, Pakistan, vol. 10, pp. 37-45, 2010.
23. J. K. R. Sastry and T. S. Lalitha, "A framework for assessing the quality of a WEB SITE, PONTE," *International Journal of Science and Research*, vol. 73, 2017.
24. V. K. Mantri, "An Introspection of Web Portals Quality Evaluation," *International Journal of Advanced Information Science and Technology*, vol. 5, pp. 33-38, 2016.
25. V S Moustakis, "Website quality assessment criteria," *Proceedings of the Ninth International Conference on Information Quality*, 2004.
26. A. Graniü, et al., "Usability Evaluation of Web Portals," *Proceedings of the ITI 2008, 30th Int. Conf. on Information Technology Interfaces*, 2008.
27. T. Singh, et al., "E-Commerce Website Quality Assessment based on Usability," Department of Computer Science & Engineering, Amity University, Uttar Pradesh, Noida, India, pp. 101-105.
28. R. Anusha, "A Study on Website Quality Models," Department of Information Systems Management, MOP Vaishnav College for Women (Autonomous), Chennai, vol. 4, pp. 1-5, 2014.
29. F. Ricca and P. Tonella, "Analysis and Testing of Web Applications," *Centro per la Ricerca Scientifica e Tecnologica, I-38050 Povo (Trento), Italy*, 2001.
30. S. Alwahaishi and V. Snášel, "Assessing the LCC Websites Quality," Springer-Verlag Berlin Heidelberg, in F. Zavoral NDT 2010, SI, CCIS 87, pp. 556-565, 2010.
31. L. Hasan and E. Abuelrub, "Assessing the Quality of Web Sites," *Applied Computing and Informatics*, vol. 9, pp. 11-29, 2011.
32. Singh KK et al., "Implementation of a Model for Websites Quality Evaluation – DU Website," *International Journal of Innovations & Advancement in Computer Science*, vol. 3, 2014.
33. L. S. Chen and P. Chung, "Identifying Crucial Website Quality Factors of Virtual Communities," *Proceedings of the International Multi-Conference of Engineers and computer scientists, IMECS*, vol. 1, 2010.
34. N. L. Wah, "An Improved Approach for Web Page Quality Assessment," *IEEE Student Conference on Research and Development*, 2011.
35. Sastry J. K. R et al., "Quantifying quality of WEB sites based on content," *International Journal of Engineering and Technology*, vol. 7, pp. 138-141, 2018.
36. Sastry J. K. R et al., "Quantifying quality of websites based on usability," *International Journal of Engineering and Technology*, vol. 7, pp. 320-322, 2018.
37. Sastry J. K. R., et al., "Structure-based assessment of the quality of WEB sites," *International Journal of Engineering and Technology*, vol. 7, pp. 980-983, 2018.
38. Sastry J. K. R., et al., "Evaluating quality of navigation designed for a WEB site," *International Journal of Engineering and Technology*, vol. 7, pp. 1004-1007, 2018.

39. N. P. Kolla et al., "Assessing the quality of WEB sites based on Multimedia content," *International Journal of Engineering and Technology*, vol. 7, pp. 1040-1044, 2018.
40. J. S. Babu et al., "Optimizing webpage relevancy using page ranking and content-based ranking," *International Journal of Engineering and Technology (UAE)*, vol. 7, pp. 1025-1029, 2018.
41. K. S. Prasad et al., "An integrated approach towards vulnerability assessment & penetration testing for a web application," *International Journal of Engineering and Technology (UAE)*, vol. 7, pp. 431-435, 2018.
42. M. V. Krishna et al., "A framework for assessing the quality of a website," *International Journal of Engineering and Technology (UAE)*, vol. 7, pp. 82-85, 2018.
43. R. B. Babu et al., "Analysis on visual design principles of a webpage," *International Journal of Engineering and Technology (UAE)*, vol. 7, pp. 48-50, 2018.
44. S. S. Pawar and Y. Prasanth, "Multi-Objective Optimization Model for QoS-Enabled Web Service Selection in Service-Based Systems," *New Review of Information Networking*, vol. 22, pp. 34-53, 2017.
45. B. Bhavani et al., "Review on techniques and applications involved in web usage mining," *International Journal of Applied Engineering Research*, vol. 12, pp. 15994-15998, 2017.
46. K. K. Durga and V. R. Krishna, "Automatic detection of illegitimate websites with mutual clustering," *International Journal of Electrical and Computer Engineering*, vol. 6(3), pp. 995-1001, 2016.
47. T. Y. Satya and Pradeepini G., "Harvesting deep web extractions based on hybrid classification procedures," *Asian Journal of Information Technology*, vol. 15, pp. 3551-3555, 2016.
48. S. J. S. Bhanu, et al., "Implementing dynamically evolvable communication with embedded systems through WEB services," *International Journal of Electrical and Computer Engineering*, vol. 6(1), pp. 381-398, 2016.
49. L. Prasanna et al., "Profile-based personalized web search using Greedy Algorithms," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, pp. 5921-5925, 2016.
50. Vassilis S. Moustakis, Charalambos Litos, Andreas Dalivigas, and Loukas Tsironis, *WEB site Quality Assessment Criteria*, *Proceedings of the Ninth International Conference on Information Quality (ICIQ-04)*, pp. 59-73
51. oon-itt, S. Quality of health websites and their influence on perceived usefulness, trust and intention to use: an analysis from Thailand. *J Innov Entrep* 8, 4 (2019). <https://doi.org/10.1186/s13731-018-0100-9>
52. Allison R, Hayes C, McNulty CAM, Young V. A Comprehensive Framework to Evaluate Websites: Literature Review and Development of GoodWeb. *JMIR Form Res*. 2019 Oct 24;3(4):e14372. doi: 10.2196/14372. PMID: 31651406; PMCID: PMC6914275.
53. Barnes, Stuart & Vidgen, Richard. (2000). *WebQual: An Exploration of Website Quality*. 298-305.
54. Layla, Hasan Emad Abuelrub, *Assessing the Quality of the web sites*, *Applied Computing, and Informatics* (2011), 9, 11-29
55. J. Sasi Bhanu, DBK Kamesh, JKR Sastry, *Assessing Completeness of a WEB site from Quality Perspective*, *International Journal of Electrical and Computer Engineering (IJECE)* Vol. 9, No. 6, December 2019, pp. 5596 5603 ISSN: 2088-8708, DOI: 10.11591/piece.v9i6.pp5596-5603
56. Rim Rekik*, Ilhem Kallel, Jorge Casillas, Adel M. Alimi, *International Journal of Information Management* 38 (2018) 201–216
57. Lin, H.-F. (2010). An application of fuzzy AHP for evaluating course website quality, *Computers and Education*, 54, 877–888. (Fuzzy hierarchical)
58. Heradio, R., Cabrerizo, F. J., Fernández-Amorós, D., Herrera, M., & Herrera-Viedma, E. (2013). A fuzzy linguistic model to evaluate the Quality of Library, *International Journal of Information Management*, 33, 642–654. (Fuzzy Linguistics)
59. Esteban, B., Tejeda-Lorente Á, Porcel, C., Moral-Muñoz, J. A., & Herrera-Viedma, E. (2014). Aiding in the treatment of low back pain by a fuzzy linguistic Web system. *Rough sets and current trends in computing, lecture notes in computer science (Including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*, (Fuzzy Linguistics)
60. Cobos, C., Mendoza, M., Manic, M., León, E., & Herrera-Viedma, E. (2013). Clustering of web search results based on an iterative fuzzy C-means algorithm and Bayesian information criterion. 2013 joint IFSA world congress and NAFIPS annual meeting, IFSA/ NAFIPS 2013, 507–512. (Fuzzy e-means)
61. Dhiman, P., & Anjali (2014). Empirical validation of website quality using statistical and machine learning methods. *Proceedings of the 5th international conference on confluence 2014: The next generation Information technology summit*, 286–291. (Bayesian)

62. Liu, H., & Krasnoprosin, V. V. (2014). Quality evaluation of E-commerce sites based on adaptive neural fuzzy inference system. *neural networks and artificial intelligence, communications in computer and information science*, 87–97. (Fuzy Neutral)
63. Vosecky, J., Leung, K. W.-T., & Ng, W. (2012). Searching for quality microblog posts: Filtering and ranking based on content analysis and implicit links. *Database systems for advanced applications, lecture notes in computer science (Including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)*, 397–413. (SVM)
64. Hu, Y.-C. (2009). Fuzzy multiple-criteria decision-making in the determination of critical criteria for assessing service quality of travel websites. *Expert Systems with Applications*, 36, 6439–6445. (Genetic Algorithms)
65. Michal Kakol, Radoslaw Nielek, Adam Wierzbicki, Understanding and predicting Web content credibility using the Content Credibility Corpus, *Information Processing and Management* 53 (2017) 1043–1061
66. Jayanthi, B. and Krishnakumari, P. (2016), "An intelligent method to assess webpage quality sing extreme learning machine", *International Journal of Computer Science and Network Security*, Vol. 16 No. 9, pp. 81-85.
67. Huang, Guang-Bin, Xiaojian Ding, and Hongming Zhou. "Optimization method based extreme learning machine for classification." *Neurocomputing* 74, no. 1 (2010): 155-163.
68. Huang, Guang-Bin, Hongming Zhou, Xiaojian Ding, and Rui Zhang. "Extreme learning machine for regression and multiclass classification." *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 42, no. 2 (2012): 513-529.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.