Fault isolation technique for decentralized survivable communication network systems via regions and paths

Nethravathi B[1], Kamalesh V N[2]
Department of ISE,JSS Academy of Technical Education, Visvesvaraya Technological University, India
Department of CSE, T John Institute of Technology, Visvesvaraya Technological University, India

Article Info

Article history:

Received May 3, 2019
Revised Jul 5, 2019
Accepted Jul19, 2019

Keywords:

Decentralized system
Fault isolation technique
Paths
Regions
Survivable communication network

ABSTRACT

The rapid continuous growth of communication networks in size, complexity and dependencies, makes them extremely challenging to maintain the survivability of the complete large network. The complexity may be due to advance voice and video services like IP TVs, IP telephony, video streaming which demands high reliability and survivability. Network management has become a great challenge as Faults are expected only in these complex networks. Once a failure is detected, the next step in the diagnosis is fault isolation which locates the source of that failure. The necessitate of decentralized diagnosis is justified by various applications, like spacecrafts. This article presents model based fault isolation technique using a graph theoretical concepts, regions and paths for decentralized communication networks.

Corresponding Author:

Nethravathi B,
Department of ISE,
JSS Academy of Technical Education,
Visvesvaraya Technological University,
Belgaum, Karnataka, India.
Email: nethravathi.sai@gmail.com

1. INTRODUCTION

Communication networks turn into highly developed because of their speed, size and complexity. Many communication systems are collection of interconnected subsystems, by looking in to various applications. Scaling is the main challenge in the design of large systems. It could be vertical scaling, where the hardware of a single one is upgraded or horizontal scaling, where more components are added. Scaling vertically suffers after a certain point. Horizontal scaling becomes much cheaper after a certain threshold. This made the evolution from centralized system to decentralized systems and then to Distributed systems.

All individual nodes in centralized systems are directly dependent on the central control to send and receive data and to be commanded. At present, centralized systems are the majority pervasive model for applications like Amazon, Facebook, Google and all other mainstream service which we utilize on the Internet. Distributed systems, computations are distributed across multiple nodes. The performance of every single subsystem is inclined by its individual state, and by the state of a maybe tiny subset of every the other subsystems. Decentralized Systems, none of the node is drilling any other node. The behaviour of every solo subsystem is prejudiced only by its local state, without any communication with any other subsystems. Many stacks such as Google have agreed to a distributed architecture inside to speed up computing, which shows a structure can be both centralized and distributed. The development of efficient IoT [1] and VOIP [2] structures has drawn every one’s interest now a days.The comparative analysis is shown in Table1.Structure of centralized, decentralized and distributed systems as shown in Figure 1.

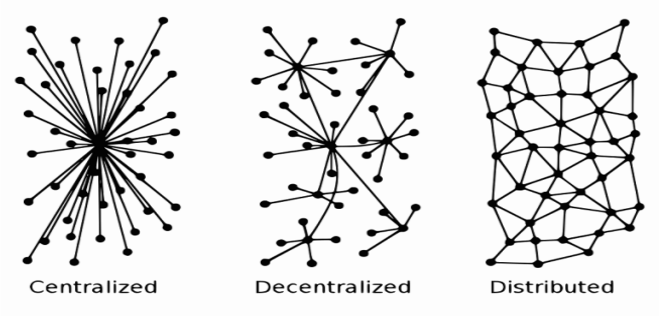

Figure 1. Structure of centralized, decentralized and distributed systems

Table 1.Comparative analysis of Centralized, Decentralized and Distributed Systems

Parameters	Centralized systems	Decentralized Systems	Distributed systems
Maintenance	Effortless	Complex but still fixed.	Complex
Fault Tolerance	Extremely unstable	Unstable	Very stable
Scalability	Low scalability	Moderate	Infinite
Ease of Creation	Fast	Moderate	Moderate
Evolution	Less diversity and progress slowly.	When the basic infrastructure is ready, remarkable evolution	Incredible

Easy scaling is not the only benefit from distributed and decentralized systems [3], fault tolerance is also similarly significant. A detail study of fault isolation techniques are discussed in “Analysis of Techniques for Isolation of Faults in Survivable Computer Networks” [4, 5]. The fault diagnosis has 3 steps. Fault detection : every indications of network disorder are confined using alarms. Fault isolation: analysis of sensible indication to conclude accurate starting place of failure. Testing: finding out actual faults using probable hypothesis.

Fault isolation is very important, since a minor fault might generate costly damages to whole network due to the spread of the fault [6]. In computer network, varieties of techniques are available to agree on accurate location of faults [7]. The most generally used methods are due to • Model traversing techniques • Alarm co-relation • Artificial intelligence techniques • Graph theoretical techniques [4].The requisite for security, reliability and availability is rising drastically with growing complexity of communication networks. The survivability of communication networks is possible due to fault detection and isolation [8], which is to be a key topic. In recent living, interconnected systems/networked systems are principally focused on the construction of distributed and decentralized design [9]. The decentralized fault detection and isolation situation for interconnected linear systems is proposed with a structural analysis scaffold, where the interconnections are accurately known [10].

This article is planned in the following way. Section 2 makes clear the concept of decentralized dialysis and inspire the work with high opinion to associated work. Section 3 presents the background theory and fundamentals about structural analysis. The region based fault detection and isolation algorithm for decentralized systems is offered in Section 4. The algorithm is illustrated for two different cases in Section 5. Then Section 6 compares the proposed algorithm with generic method and shows the experimental results. This paper is concluded in Section 7.

2. MOTIVATION AND RELATED WORKS

Topological design of centralized/decentralized computer communication networks is an intricate problem that is commonly cracked in two segments. The first phase: break up network nodes into groups and choose a concentrator location for every group so that every node in a cluster is allocated to the same concentrator. The second phase: decide topology of links that join network i) nodes to concentrator ii) concentrator to concentrator iii) concentrators to the central computer [11]. The necessitate for decentralized diagnosis is justified by numerous applications for example spacecrafts, where the architecture is planned into functional component, and constructed with decentralized system engineering [12].

In model-based diagnosis, detection and isolation of faults is done by considering model of the system during the design time or during reverse engineering procedure. The three kind of diagnosis architectures are centralized, decentralized and distributed which is shown in Figure 2.The concise details of 3 architectures are discussed in Table 2.

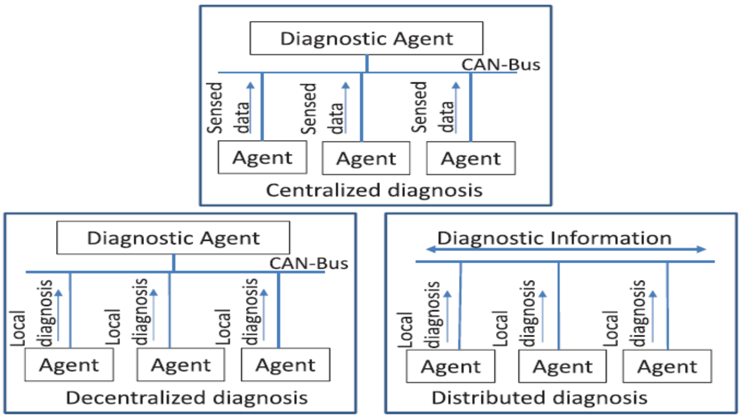

Figure 2. Diagnosis architecture of centralized, decentralized and distributed systems

Table 2. Analysis of Centralized, Decentralized and Distributed Systems Diagnosis Architecture

Diagnosis Architecture	Definition	Advantages	Disadvantages/ challenges
Centralized	No processing capabilities for local agents that convey facts to a centralized diagnoser which work out the (global) diagnosis.	simplicity	For large systems it is impractical.
Decentralized	consists of local diagnosers whose end results are coordinated by a supervisory diagnose	no communication at intra level	Disambiguate between local diagnosis result and supervisory diagnosis.
Distributed	Composed of a group of local diagnosers, the same in terms of function, with communication likely between any two of them.	A local diagnoser is self-governing	Globally consistent is the challenge.

Dealing with the fault tolerant manage crisis in a networked skeleton, Sauter et al. worked on fault isolability and fault detectability circumstances for centralized, distributed, and decentralized systems [13]. The “divide and conquer” principle is used for the decentralized diagnosis scheme [14]. The discrete event system is the popular method used for decentralized diagnosis [14-16]. Cordier and Grastien [14] explained a decentralized method in consideration of the size and to catch a tractable demonstration of the diagnose [15]. In recent times hybrid or continuous systems are the main targets for decentralized diagnosis. In [17] presented qualitative scaffold to a decentralized construction. The decomposition projected in [18] neglects pre existing restriction, that could be privacy-based, geographical or functional. In [19] showed the decentralized architecture plan, by making use of fault-driven residual generation scheme. In [20] presented two conditions to make sure the decentralized fault detection and isolation (FDI). Condition 1: every enablement situations of local required models must be fulfilled for some event of a series belonging to the global required behavior. Condition 2: any experience go against the global required behaviour, because of the happening of a fault, is noticed. Based on a set of local representations and a set of inter-local representations message events, all fault happenings can be detected and its related situate of liable candidates can be produced; system is decentrally FDI [20]. In [21] fully decentralized approach for analytical redundancy in detecting and isolating sensor faults of a wireless structural health monitoring are shown.From the survey of several literatures it is observed that fault isolation technique for decentralized system is the interesting and challenging.

3. PRELIMINARIES

3.1. Decentralized Diagnosis General Structure

The structural design is hierarchically scalable from the Figure 3. The local diagnosers depend on models of respective subsystems to appear at diagnosis. Since faults propagate between subsystems, might arise ambiguities. These ambiguities are resolved at the higher level and provide diagnosis at a higher resolution. Based on the communication possibilities between diagnosers, the decentralization levels are defined. Level i diagnosers talk with their level i + 1 and i−1 in the hierarchy.

A device that connects a number of links with only one destination is called the node concentrators. Concentrator is often called a hub/MAU - Media Access Unit. Back bone and access network are the levels in generic network. The backbone is formed by the connection of all the concentrators, and the access network which is in below level make connection between concentrators and the terminals. The back bone network may be complete or connected by path, ring or tree. The access networks are less dense than backbone network, since survivability and quality are more critical. Figure 4, represent a structure of network having completely connected backbone and Star topological access network. 1 to 5 are node concentrators and every remaining node is connected to single concentrator. The shorted path defines the traffic among two nodes of network. The traffic between two nodes connected to one concentrator never enters the backbone network. So the traffic between any two nodes is the traffic on backbone links. The location of concentrators and connecting the remaining nodes to concentrators plays an important role in minimizing the total installation cost and routing the traffic costs. This problem is called as “QCL-C: Quadratic Capacitated concentrator Location Problem with Complete Routing” [22].

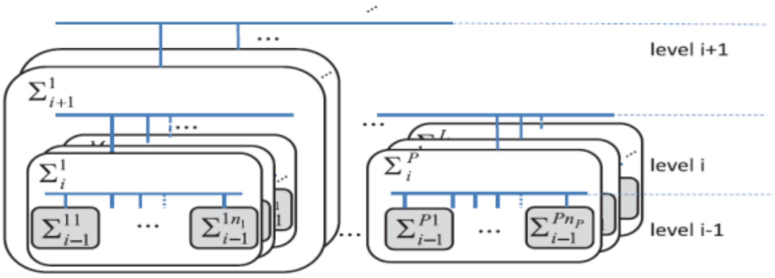

Figure 3. General architecture of decentralized diagnosis

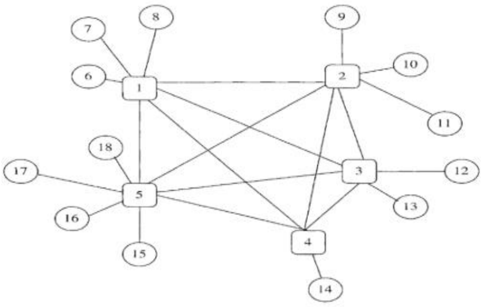

Figure 4. Complete backbone and Star access network

3.2. Region

The topology of the network can also be referred as *network geometry*. We can define region with respect tonetwork topology or network geometry. We consider geometry based region by Voronoi diagrams. Voronoi diagram definition : A Voronoi diagram divides the plane into n Voronoi regions for a given set of S points p_1, p_2,...,p_n in the plane having the following characteristics: • Every point p_i rests in accurately one region. • If a point r /∈ S rests in the similar region as p_i, then the Euclidian distance from p_i to r is smaller than the Euclidian distance from p_j to r, where p_j is any other point in S[23-25]. The Voronoi diagrams [24] are key data structure in computational geometry.

4. PROPOSED METHOD

Represent the given communication network by its corresponding network geometry M. divide the network geometry into regions by using Voronoi diagram process. Index the distinct regions so obtained by using first N natural numbers from left to right randomly. For each region assign the node concentrator. Further, the index of the region will be the index of the node concentrator for all calculation purpose.

Establish the links starting from region with number 1 to all adjacent regions so that, no cycle is formed. In the course, one should not consider the nodes (node concentrator) which are already visited. By this we get a list of nodes L. Sort the list L in increasing order.

Repeat the above step for all the node concentrator representatives of the regions sequentially. This process results in a tree T covering all the region of the network geometry with Node 1 as the root node. This forms a base framework for fault detection and isolation.

Pass a token message from root node to all its leaf nodes. If the token message reaches all the leaf nodes and when the leaf node send the acknowledgment to the root node then declare the network is fault free, else there is a fault.

Constructs two lists L1 and L2, where L1 consists of all those leaf nodes and other nodes of the tree T through which acknowledgment has been passed to the root node and L2 consisting of all those leaf nodes which have not send the acknowledgement and other nodes which lies in the paths connecting the root node and the leaf node((which have not send the acknowledgment).

Let L3=L2-L1. Where L3 consisting of nodes of the network where the fault exist. Divide set L3 faulty clusters F1, F2, F3… Fs such that L3= F1UF2UF3U…...UFs and F1∩F2∩….∩Fs=Null set.

Now consider the cluster F1, and pass the token message from starting node of F1 to the leaf node in F1. Identify the node f in F1, where the message is stopped. Declare f as a faulty node and the corresponding region. Repeat the process for the faulty clusters of L3.

5. ALGORITHM

Region based Fault Detection and Isolation Algorithm

```
Input: Network Geometry Graph N.
Output: Declare Network is Fault Tolerant or Faulty Region isolation/identification.
Step 1: Divide the given network geometry N in to n number of regions using Voronoi diagram.
Step 2: Name the each region by n natural numbers from left side to right side randomly.
Step 3: Starting from 1 establish a link to all adjacent nodes, (so that no cycle formed / should not consider visited nodes). List and sort all the adjacent nodes / visited nodes respectively.
Step 4: Repeat step 3 for all the nodes in the list until it covers all the regions of the graph sequentially.
Step 5: As a result a tree is formed covering all the regions in the network.
Step 6: Identify the paths in the tree constructed, so the leaf nodes.
Step 7:  Pass the message from node 1 to all the paths identified.
          If the message reaches all the leaf nodes, and when the leaf nodes sends alarm to the root (central monitoring concentrator) then the network is Fault free and Exit.
          Else declare fault is present in the Network and go to step 8.
Step 8: Identify the leaf nodes which sends the alarm to the root, and the respective paths.
          From the alarm received, make a fault free list FF={ Set of nodes, which are fault free }
Step 9: Identify the leaf nodes which does not send the alarm to the root, and the respective paths.
          If the nodes in the identified path are present in FF, remove those and declare them as fault free nodes.
          And declare the remaining set of nodes as faulty cluster for each path.
Step 10: Consider each faulty cluster, send the messagefrom starting node of faulty cluster. If message passes through, then declare those nodes as fault free. And wherever it stops, declare that node as Faulty node.Repeat step 10 for all the nodes in the each cluster.
Step 11: Exit.
```

6. ILLUSTRATION

Step 1: Divide the given network geometry N in to 34 regions using Voronoi diagram as shown in Figure 5. Regions numbered network as shown in Figure 6.

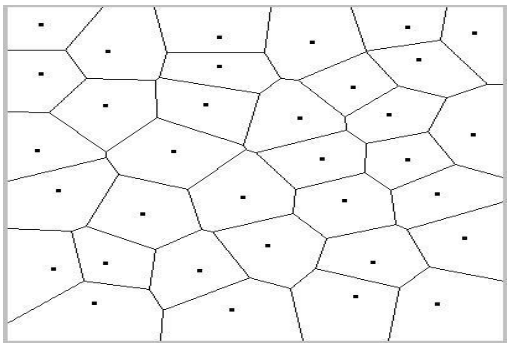

Figure 5. Division of network in to regions

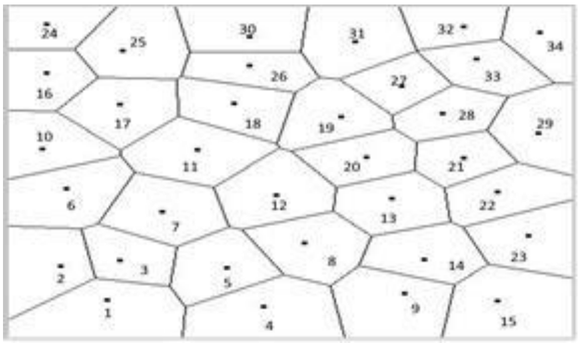

Figure 6. Regions numbered network

Step 2: Name the each region by natural numbers from left side to right side randomly as shown in Figure 6.
Step 3: Starting from 1 establish a link to all adjacent nodes. The adjacent nodes of 1 are 2,3,5,4
Establish links 1→2, 1→3, 1→5, 1→4
List and sort all the adjacent nodes / visited nodes. Sorted list is L= { 1, 2, 3, 4, 5}
Step 4: Repeat step 3 for all the nodes in the list until it covers all the regions of the graph sequentially.
Consider next element in the list, i.e., 2, the adjacent regions are 1, 3, 6. As 1 and 3 are already visited, establish link 2 →6 and add 6 to list L. Now the list L= {1. 2, 3,4,5,6}
Next element 3, the adjacent regions are 1, 2, 6,7, 5. As 1,2,6,5 are already visited, establish link 3→7 and add 7 to list L. Now the list L= {1. 2, 3, 4, 5,6, 7}
Next element 4, the adjacent regions are 1, 5,8,9. As 1 and 5 are already visited, establish link 4 →8, 4 →9 and add 8, 9 to list L. Now the list L= {1. 2, 3,4, 5,6, 7, 8, 9}.
Next element 5, the adjacent regions are 1, 3, 7, 12, 8, 4. As 1, 3, 7, 8, 4 are already visited, establish link 5 →12 and add 12 to list L. Now the list L= {1. 2, 3, 4, 5, 6, 7, 8, 9, 12}

Table 3.Establishing Link

Next element	Adjacent Regions	Visited Regions	Establish link	Updated List L
6	2,3,7,11,10	2,3,7	6 →10, 6 →11	{1,2, 3,4,5,6,7, 8, 9, 10,11,12}
7	3,5,6,11,12	3,5,6,11,12	--	{1,2, 3,4,5,6,7, 8, 9, 10,11,12}
8	4,5,12,13,14,9	4,5,12,9	8 →13, 8 →14	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14}
9	4,8,14,15	4,8,14	9 →15	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15}
10	6,17,16	6	10 →16, 10 →17	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17}
11	6,7,12,17,18	6,7,12,17	11 →18	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17,18}
12	5,7,8,11,13, 19,20	5,7,8,11,13	12 →19, 12 →20	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17,18, 19,20}
13	8,12,14,20, 21,22	8,12,14,20	13 →21, 13 →22	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17,18, 19,20, 21,22}
14	8,9,13,22,23,15	8,9,13,22,15	14 →23	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17,18, 19,20, 21,22,23}
15	9,14,23	9,14,23	--	---
16	10,17,24,25	10,17	16 →24 16 →25	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17,18, 19,20, 21,22,23,24,25}
17	10,16,25,18,11	10,16,25,18, 11	--	--
18	11,17,19,25,26	11,17,19,25	18 →26	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17,18, 19,20, 21,22,23,24,25,26}
19	18,12,20,26,27, 28,31	18,12,20,26, 28	19 →27 19 →31	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17,18, 19,20, 21,22,23,24,25,26,27,31}
20	12,13,19,28,21	12,13,19,21	20 →28	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17,18, 19,20, 21,22,23,24,25,26,27,28,31}
21	13,20,22,28,29	13,20,22,28	21 →29	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17,18, 19,20, 21,22,23,24,25,26,27,28,29,31}
22	13,14,21,23,29	13,14,21,23, 29	--	--
23	15,14,22	15,14,22	--	--
24	16,26	16,26	--	--
25	24,16,17,18,26, 30	24,16,17,18, 26	25 →30	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17,18, 19,20, 21,22,23,24,25,26,27,28,29,30,31}
26	18,25,30,19,31	18,25,30,19, 31	--	--
27	19,31,33,28	19,31,28	27 →33	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17,18, 19,20, 21,22,23,24,25,26,27,28,29,30,31,33}
28	19,20,21,27,29, 33	19,20,21,27, 29,33	--	--
29	21,22,28,33,34	21,22,28,33	29 →34	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17,18, 19,20, 21,22,23,24,25,26,27,28,29,30,31,33,34}
30	25,26,31	25,26,31		
31	26,27,30,32	26,27,30	31 →32	{1,2, 3,4,5,6,7, 8, 9, 10,11,12, 13,14,15,16,17,18, 19,20, 21,22,23,24,25,26, 27,28,29,30,31,32, 33, 34}
32	27,31,33,34	27,31,33,34	--	--
33	27,28,29,32,34	27,28,29,32, 34	--	--
34	32,33,29	32,33,29	--	--

Step 5: As a result a tree is formed covering all the regions in the network.

Constructed tree as shown in Figure 7. Fault free network as shown in Figure 8.

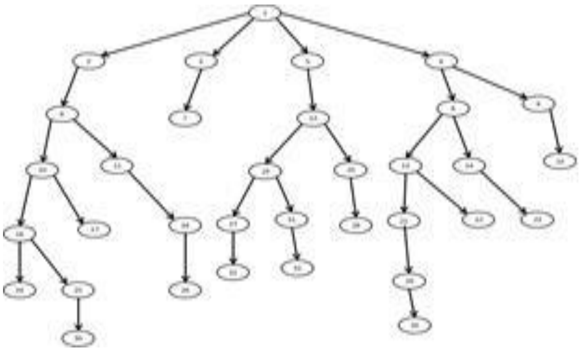

Figure 7. Constructed tree

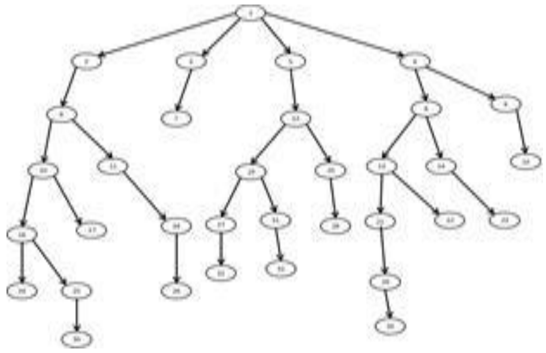

Figure 8. Fault free network

Step 6: Identify the paths in the tree constructed

P1 : 1 → 2 →6 → 10 →16 → 24
P2 : 1 → 2 →6 → 10 →16 → 25 → 30
P3 : 1 → 2 →6 → 10 → 17
P4 : 1 → 2 →6 →11 → 18 → 26
P5 : 1 →3 →7
P6 : 1 → 5 → 12 → 19 → 31→32
P7 : 1 → 5 → 12 → 19 →27 →33
P8 : 1 → 5 → 12 →20 →28
P9 : 1 → 4 →8 →13 → 21 → 29 →34
P10 : 1 → 4 →8 →13 →22
P11 : 1 → 4 →8 →14 →23
P12 : 1→4 →9 →15

The leaf nodes are { 24, 30, 17, 26, 7, 32, 33, 28, 34, 22, 23, 15 } of the respective identified paths.

Step 7: Pass the message from node 1 to all the paths identified. If the message reaches all the leaf nodes, and when the leaf nodes sends alarm to the root (central monitoring concentrator) then the network is fault free.

Example 1: Here the root (central monitoring concentrator) will receive the alarm from all the leaf nodes identified { 24, 30, 17, 26, 7, 32, 33, 28, 34, 22, 23, 15 }. So we can declare the network is Fault free.

Example 2: If the passed message doesn't reach all leaf nodes, then there exists faulty region or faulty regions. Now identification of faulty region is the challenge.

Faulty network as shown in Figure 9. Faulty clusters as shown in Figure 10.

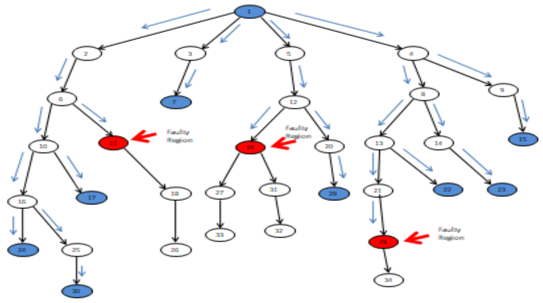

Figure 9. Faulty network

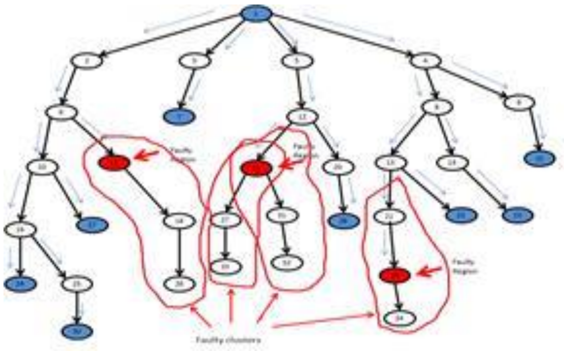

Figure 10. Faulty network : faulty clusters

Step 8: In Step 6, we have identified the paths and leaf nodes. The leaf nodes which sends alarm to root are fault free paths and the regions comes in that path are fault free regions. In the example 2 root will receive alarm from {24, 30, 17, 7, 28, 22, 23, 15} from the paths P1, P2, P3, P5, P8, P10, P11, P12. So the regions FF= { 2, 6, 10, 16, 24, 25, 30, 17, 3, 7, 5, 12, 20, 28, 4, 8, 13, 22, 14, 23, 9, 15} are fault free.

Step 9: Now we need to isolate or identify the faulty regions from the remaining paths. We have not received alarm from leaf nodes { 26, 33, 32, 34 }. sowe need to check the paths P4, P6, P7, P9 and the regions

{ 2, 6, 11, 18, 26 }, { 5, 12, 19, 27, 33 }, { 5, 12, 19, 31, 32 } and { 4, 8, 13, 21, 29, 34} respectively. It is observed from set FF, { 2, 6} in P4 are fault free, so the faulty cluster is { 11, 18, 26 }. In P6, { 5, 1 2} are Fault free, so the faulty cluster is { 19, 27, 33 }. In P7, { 5, 12 } are Fault free, so the faulty cluster is { 19, 31, 32 }. In P9, { 4, 8, 13} are fault free, so faulty cluster is { 21, 29, 34 }.
FC1= { 11, 18, 26 }, FC2= { 19, 27, 33 }, FC3= { 19, 31, 32 }, FC4= { 21, 29, 34 }
Step 10: Consider FC1, Send message from 11, it never travel since 11 is faulty region. So declare 11 as faulty region. Then send message from 18, it reaches leaf node 26, so declare 18 and 26 as faulty free regions. Consider FC2, Send message from 19, it never travel since 19 is faulty region. So declare 19 as faulty region. Then send message from 27, it reaches leaf node 33, so declare 27 and 33 as faulty free regions. Consider FC3, Since 19 is already considered as faulty region, Send message from 31, it reaches leaf node 32, so declare 31 and 32 as faulty free regions. Consider FC4, Send message from 21, it travels to 29, and stops. So declare 29 as Faulty Region. Declare 34 as fault free region. So now we conclude that the regions {11, 19, 29} are the faulty regions in the Example 2.

7. DISCUSSIONS AND EXPERIMENTAL RESULTS

Simulations are performed to evaluate the performance of the proposed algorithm for fault detection period. The results are compared with the frequently used general model based method of fault isolation technique [3, 19], where the alarm will be sent to central monitoring concentrator whenever a network changes its behaviour. The simulations are performed for multiple faults and the corresponding fault detection periods are recorded for up to100 nodes. Network size Vs fault detection period as shown in Figure 11.

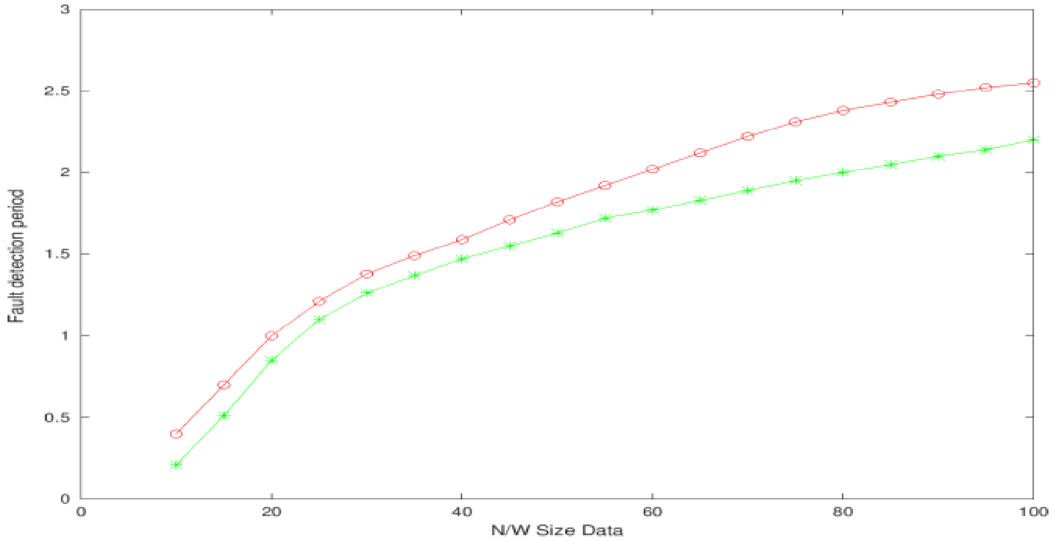

Figure 11. Network size Vsfault detection period

It is observed from the experimental results for different network size, the fault detection period of the proposed model is better compared to the general model considered.

8. CONCLUSION

The rapid growth of modern communication technologies made the detection and isolation of faults in the large-scale systems as important and complicated. In this research work, a model based: fault detection and isolation algorithm for decentralized survivable communication network systems is proposed. The method works for both single and multi faults as necessitate is justified by many applications. The proposed algorithm can isolate the fault more precisely because of its strong mathematical background. From the computer simulation results it is evident that the proposed algorithm is comparatively have better performance and easily adaptable.

REFERENCES

[1] SubrataChowdhury, P. Mayilvahanan, A survey on internet of things: privacy with security of sensors and wearable network ip/protocols, *International Journal of Engineering & Technology*, 7 (2.33) (2018) 200-205.

[2] C. Sharanya, M. Meena, M. Monisha and V. Rajendran, Development of efficient VoIP application using cognitive radio networks, *International Journal of Engineering & Technology*, 7 (2.33) (2018) 419-421.

[3] A. A. Tadjeddine, A. Chaker, M. Khiat, L. Abdelmalek, N. Khalfalah, A contribution to the control of voltage and power of the interconnection between two decentralized electrical grids with an optimal localization of the SVC devices in real-time, *International Journal of Power Electronics and Drive System*, 10(1) (2019) 170-177.

[4] Nethravathi B, Kamalesh V.N, Nidhi H Kulakarni, and Apsara, M. B. Analysis of Techniques for Isolation of Faults in Survivable Computer Networks, at IEEE Xplore on 26th June 2017, DOI: 10.1109/ICEECCOT.2016.7955235, Page no. 307-311.

[5] Nethravathi, B. & Kamalesh, V. N. Topological Design of Computer Communication Network Structures: A Comprehensive Review. *Indian. Journal of Science and Technology*, vol 9(7), (2016).

[6] Mukrimah Nawir, Amiza Amir, Naimah Yaakob, Ong Bi Lynn, Effective and efficient network anomaly detection system using machine learning algorithm, *Bulletin of Electrical Engineering and Informatics*, 8(1) (2019) 46-51.

[7] N. F. Fadzail, S. Mat Zali, Fault detection and classification in wind turbine by using artificial neural network, *International Journal of Power Electronics and Drive System* , vol 10(3), 2019.

[8] M. Thirumarimurugan, N. Bagyalakshmi and P. Paarkavi, Comparison of fault detection and isolation methods: A review, *IEEE,* 10th International Conference on Intelligent Systems and Control, 2016.

[9] Li-Wei Li1, Guang-Hong Yang. Decentralized fault detection and isolation of Markovian jump interconnected systems with unknown interconnections, International Journal Of Robust And Nonlinear Control, *Int. J. Robust Nonlinear Control (2017),* Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/rnc.3743

[10] Sauter D, Boukhobza T, Hamelin F. "*Decentralized and autonomous design for FDI/FTC of networked control systems*." Proceedings of the 6th IFAC Symposium Safeprocess 2006; 39(13):138–143. Beijing, China.

[11] HasanPirkul and VaidyanathanNagarajan Locating concentrators in centralized computer networks, J.C. Baltzer A.G. Scientific Publishing Company, Annals of Operations Research 36 (1992) PP 247-262

[12] ElodieChanthery, Louise Travé-Massuyès, and SaurabhIndra, Fault Isolation on Request Based on Decentralized Residual Generation, *IEEE Transactions On Systems, Man, And Cybernetics: Systems*.

[13] D. Sauter, T. Boukhobza, and F. Hamelin, Decentralized and autonomous design for FDI/FTC of networked control systems, Fault Detect. Supervision Safety Tech. Processes, vol. 6, no. 1, 2006, pp. 138–143.

[14] Y. Pencolé and M. Cordier, A formal framework for the decentralized diagnosis of large scale discrete event systems and its application to telecommunication networks, Artif. Intell., vol. 164, nos. 1–2, pp. 121–170, 2005.

[15] M. Cordier and A. Grastien, Exploiting independence in a decentralised and incremental approach of diagnosis, in Proc. 20th *Int. Joint Conf. Artif. Intell. (IJCAI),* 2007, pp. 292–297.

[16] Y. Wang, T.-S. Yoo, and S. Lafortune, Diagnosis of discrete event systems using decentralized architectures, Discrete Event Dyn. Syst., vol. 17, no. 2, pp. 233–263, Jun. 2007.

[17] Console L, Picardi C and Theseider D A Framework for Decentralized Qualitative Model-Based Diagnosis (20th int. joint conf. on artificial intelligence) 2007, p 286 – 291

[18] A. Bregon et al., "An event-based distributed diagnosis framework using structural model decomposition," Artif. Intell., vol. 210, pp. 1–35, May 2014.

[19] C G Pérez et. al., Journal of Physics: Conference Series 659, 2015 012054, doi:10.1088/1742-6596/659/1/012054.

[20] M. SayedMouchaweh, Decentralized Fault Detection and Isolation of Manufacturing Systems, 21st International Workshop on Principles of Diagnosis, 2010.

[21] Kay Smarslya, KinchoH.Lawb, Decentralized fault detection and isolation in wireless structural health monitoring systems using analytical redundancy, Advances in Engineering Software, Volume 73, July 2014, Pages 1-10.

[22] HandeYaman. Concentrator Location in Telecommunication Networks, Springer, Printed in United States of America, 2005.

[23] Diane Souvaine Michael Horn, Julie Weber (2004) Computational Geometry, Tufts University, Spring 2005.

[24] Aurenhammer, F. and Klein, R. Voronoi Diagrams. Ch. 5 in Handbook of Computational Geometry (Ed. J.-R. Sack and J. Urrutia). Amsterdam, Netherlands: North-Holland, pp. 201-290, 2000.

[25] de Berg, M.; van Kreveld, M.; Overmans, M.; and Schwarzkopf, O. Voronoi Diagrams: The Post Office Problem. Ch. 7 in Computational Geometry: Algorithms and Applications, 2nd rev. ed. Berlin: Springer-Verlag, pp. 147-163, 2000.