IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol-6, Special Issue-4, May 2018 E-ISSN: 2347-2693

A Framework for Fault Prevention and Detection in IoT using Smart Gateway

T. Saha[1*], R. Sunitha[2]

[1*]Dept. of Computer Science, School of Engineering & Technology, Pondicherry University, Puducherry, India
[2]Dept. of Computer Science, School of Engineering & Technology, Pondicherry University, Puducherry, India

[*]Corresponding Author: tapas.saha.99@gmail.com, Mobile no-9775357747

Available online at: www.ijcseonline.org

Abstract— An IoT network is rendered reliable when it is tolerant to faults. Fault management involves fault prevention, detection and recovery of which prevention is considered to be a significant phase. The main objective of this work is fault prevention and detection in IoT networks. The existing IoT fault prevention system finds only one reliable path identified using the goodness value of different available paths and no alternate solutions are provided in case of failure of the reliable path. The proposed system provides a solution by choosing the non-discarded path with the highest goodness value for transmission in case the chosen path fails. The proposed framework has a set of observer nodes connected to a smart gateway and a set of sensor nodes connected to the observers. The gateway acts as the interface between the network and the Internet. This framework comprises prevention and detection algorithms to prevent communication failures between sensor node and gateway, to provide alternative reliable paths for transmission and to detect node faults in early stages. The proposed gateway also provides a solution for hardware failure, software failure, connection failure and also manages the overall load balance of the network.

Keywords— IoT, Smart gateway, Fault-tolerant, Fault management system, Goodness value.

I. INTRODUCTION

The concept of Internet of Things was first proposed by MIT Auto-ID Labs in 1999 regardless of its investigation on object localization, using wireless sensor networks and radio frequency identification technologies for state recognition [1].

Internet of things(IoT) is an ever-growing technology in the world. A simple definition of IoT is to connect the smart devices (like a smart phone, wireless devices, sensor devices) to the internet. The smart devices are connected together through RFID (Radio Frequency Identification), ZigBee, Wi-Fi, Ethernet,3G/4G. Typically IoT is applicable for heterogeneous environment like-smart city, healthcare, smart home, environment monitoring, intelligent transportation, etc.

IoT gateway act as a bridge between sensor nodes and the Internet. It not only receives data from sensor nodes but also transmits data to the application domain. The main goal of gateway is to handle the heterogeneity between different endpoint networks and the Internet. An important issue for any technology is reliability, security, and privacy [2]. In IoT reliability (maturity, availability, fault tolerance and recoverability [3]) is a big challenge, as for example health care system, the health care system should be reliable because if any fault occurs in health data, it can be life-threatening for the patient. Managing fault-tolerance is one of the most promising control strategies in IoT to make the system robust.

This paper presents, Fault prevention and Detection algorithm with Architecture, Section II discuss the Literature Survey of non IoT and IoT fault tolerance system. Section III contain Motivation of this work, Section IV contain objectives of this paper. section V contain the Problem Definition Section. VI explain the Proposed System with Architecture. Section VII contains the Fault Prevention and Detection Algorithm and Section VIII concludes of this work.

Fault occurs in a situation such as malfunction of sink node hardware and traffic bottleneck at a node due to a high receiving data rate. The type of fault in IoT are- node connection failure or link failure, system failure (h/w malfunctions, s/w bugs, power shortages or environmental hazard), gateway failure.

II. LITERATURE SURVEY

In WSN and MANET most fault tolerant solutions [4] are based on the multipath routing paradigm, which provides each sensor with alternative path. The node-disjoint paths are another alternative solution, which is considered as the most reliable. The existing methods for tolerating faults in a non IoT system are defined in section A, while in section B the methods for tolerating faults in an-IoT system is discussed. Table 1 represents comparison of some IoT fault-tolerant solutions related research papers.

A. Fault tolerance for non IoT system

1) Distributed environment: Chen et al.[5] proposed three fault-tolerant task clustering algorithms; selective re-clustering (SR), dynamic re-clustering (DR), and vertical re-clustering (VR), in order to enhance available task clustering techniques for execution in a faulty distributed environment. Firstly, the selective re-clustering technique combines only failed tasks with a new clustered job. It does not assess the clustering size. Secondly, The Dynamic re-clustering uses a self-adjusted approach, which helps to minimize the cluster size to the number of failed tasks. The number of failed tasks can be larger or smaller than the actual optimal size. It merges failed tasks into new clustered jobs. The third technique is vertical re-clustering, which is almost identical to selective re-clustering. It is only retrying failed or uncompleted tasks.

Castro-León et.al [4] proposed an automatic and scalable fault tolerant system. The system model is competent to detecting a faulty node which is the offender for some failures in the communication socket. The faulty processes are recovered by a healthy node and without losing any data connection are reestablished. The Redundant Array of Distributed Independent Control (RADIC) architecture assures that the message will successfully deliver if the system is suffering a node failure. The RADIC architecture defines four models: protection, detection, recovery, and masking.

a) Protection model: Every node has one protector and one observer. The protector takes care of checkpoints and log messages.

b) Detection model: The fault detection model is able to provide a successful transition during failure-free execution. This model uses heartbeat/watchdog protocol. Each protector has charged to detect his neighbor node failure.

Table 1. Comparison chart for fault-tolerant IoT systems.

Title	Domain	Method	Type of fault	Advantage	Disadvantage	Complexity
Fault Tolerant and Scalable IoT-based Architecture for Health Monitoring	Healthcare	Backup routing	Node connection failure	i)A complete architecture which support high data rate bio-signals. ii)Customized tunnelling gateway for routing packet	Not specified	Not specified
An Adaptive Learning Approach for Fault-Tolerant Routing in Internet of Things	Not specified	Cross-Layer-Based Adaptive Fault-Tolerant Routing Algorithm	Node connection failure	Less energy consumption	Provides a single reliable path	Not specified
A reliable IoT system for Personal Healthcare Devices	Healthcare	Fault-tolerant algorithm using daisy chains	Gateway failure	Gateway store backup copy of previous gateway and also, parity data	At most 2 gateway fault occurrences can be recovered	Not specified
Leveraging Solution-Specific Gateways for Cost-Effective and Fault-Tolerant IoT Networking	Smart city	Network Intersection based Candidate Gateway Location Selection algorithm called NewIoTGateway-Select algorithm	Link failure and Gateway failure	Minimize the total number of gateway, it's help to reduce the cost	Not specified	Polynomial time.

a) Recovery model: After a failure of any process, the protector will roll back the process and restart from the last checkpoint. In this model, three different rollback protocols are introduced.

b) Masking model: The protector is responsible for communicating to the affected observers and update the new address in the Radic Table.

2) *Wireless sensor networks:* Y. Challal et al. [6] authors present a new intrusion-fault tolerant routing method which is able to provide a high level of reliability through a secure multipath routing construction. Main contributions of this paper are:

• SMRP (Sub-branch Multipath Routing Protocol), which is a new approach of multipath routing. It is derived from node-disjoint paths which amplify significantly the network lifetime compared to the existing solutions.

•SEIF (Secure and Efficient Intrusion-Fault tolerant protocol) which is also a developed an efficient and lightweight security scheme based on the above multipath protocol. SEIF differs from existing intrusion-fault tolerant solutions by providing a totally distributed and in-network execution, which does not require referring to the base station for both route building and security checks.

B. Fault Tolerant for IoT system

Gia et al. [7] provide customized 6LoWPAN health care Architecture with backup routing between nodes. This architecture provides an advance service which maintains connection failure between the node and the sink node. Also, they introduced "A method for extending the number of medical sensing nodes at a single gateway is presented". The complete system comprising bio-signal acquisition such as ECG, EEG and EMG and plotting graphical waveforms of these gathered bio-signals to a remote system.

Mishra et al. [8] proposed method which is based on mixed cross layered and learning automata (LA), for managing fault tolerant problem, which ensures successful delivery between two nodes even in the presence of faults. It is highly scalable and it is claimed to be able to deliver high degrees of performance in a heterogeneous environment. The concept of cross-layering is the help to optimize the energy even faults happen at the same time. LA is an adaptive approach which can takes decision based on the feedback from the environment, along with inherent intelligence.

Woo Woo, et al. [9] presented a fault-tolerant algorithm using daisy chain for the reliable IoT system, where gateways on the same layer in the system are linked to form a daisy chain for fault tolerance at the level, and the gateway also stores the backup copy. It avails to handle as many as two gateway faults at the same time. The fault tolerant algorithm has two phases - back up and fault recovery. The gateway stores backup data into the previous gateway positioned immediately ahead of the gateway in the daisy chain. The last gateway backup copy is stored in the upper layered gateway.

Karthikeya, et al. [10] proposed a Network Intersection based Candidate Gateway Location Selection algorithm called NewIoTGateway-Select algorithm which determines the minimum number of gateway in smart city scenario. There are three main purposes of the proposed algorithms:

1) Candidate location for gateway placement.
2) Selecting the optimal location for SSGW's and IGW's from candidate location.
3) Handle gateway failure and link failure provide a k-coverage-connectivity based fault tolerance scheme.

III. MOTIVATION

As the days passes IoT is knitting itself with our daily life more and more. With the advancement of technology, from homes to factories, everything is going digital. To achieve this digital world, systems should be connected to the internet and this is where IoT comes into play. But using a system with faults makes it non-reliable and vulnerable to attacks upon which the world cannot depend. So, handling the fault is a major aspect in IoT.

For fault-tolerance, there are many existing solutions in the distributed system and wireless sensor network. Very less number of solution exists in IoT fault tolerant. The existing work gives solution for the source node to destination node with one reliable path. But they didn't provide other alternative reliable paths. The gateway is also busy to monitor node connection state. The main motivation of this system is to handle hardware faults, avoid unnecessary traffic and maintain the overall load balance of the system and to provide multiple reliable path and node failure detection mechanisms.

IV. OBJECTIVES

Faults detection, prevention, and recovery of these three strategies, prevention is the most important after detection and recovery. Our objectives would be to:

- Prevent sensor node to the gateway communication failure.
- Provide more than one reliable path for the source node to destination node data transmission.
- Use of efficient heartbeat/watchdog protocol for detecting the node/observer fault at an early stage.
- Overcome from hardware failure and maintain the system overall load balance. Reduce unnecessary traffic to a gateway.

V. PROBLEM DEFINITION

In the case of reliability, fault occurrence is a major issue. To resist a fault, there are three different methods- Detection, Prevention, and Recovery. Prevention is very important because better prevention mechanism reduces the chances of fault occurrences more.

- To design a smart gateway that can detect and prevent node/link failures at early stages and handle hardware failures and also manage the unnecessary traffic. The aim would be to minimize fault occurrences during data transmission between a node and gateway in IoT.

VI. PROPOSED SYSTEM

The smart Gateway will handle hardware failures, software failures, connection failures, and give alternative reliable paths. It should also observe the connection status of the node and observer, also identify the faulty node/observer. The system supports load balance mechanism in observer failure situation. When any observer node fails, the corresponding child nodes get automatically connected to neighbour observer node. At this time, the system does not connect all nodes to one observer, it distributes the nodes to neighbour observer nodes according to the overall load balance of the system.

A. System Architecture

Figure 1 illustrates the architecture of the assumed IoT network. The network comprises of a smart gateway and a set of Observer nodes connected to the gateway. On the other end the gateway is connected to a remote administrator the controls the whole system through the gateway. Each Observer node has approximately equal number of nodes connected to it. The smart gateway is responsible for handling the overall system. Observers are mainly responsible for handling the corresponding sensor nodes and periodically update their status to the gateway.

The smart gateway is mainly responsible for tolerating faults in the whole network using prevention and detection mechanism. Prevention mechanism prevents the fault in node to gateway communication and also in case of the gateway to node communication. In Fault Prevention mechanism during node to gateway communication, when a node sends a message to the gateway, it uses different paths to sends this message. Gateway to node prevention mechanism works in two-phases. First phase is reliable path selection, for each transmission from the smart gateway, reliable paths are found according to a precomputed goodness value.

After getting reliable paths, in the second phases of this mechanism, gateway will communicate with destination node using the most reliable path. If this path results in a failure then gateway selects the next reliable path according to goodness value, likewise continue this process. In detection algorithm, observer node monitors corresponding child node and updates it to the gateway. When observer does not get any response from the child node then it informs the gateway. The gateway checks the connection status of this node and broadcasts all observer the information according to connection status. The gateway also mentions connection status of observer node and also informs it to others observers if any observer is faulty.

VII. FAULT PREVENTION AND DETECTION ALGORITHM

A. Fault Prevention

Prevention is the most important way as the chances of fault is less if prevention is applied. The proposed fault prevention strategy works in two modes as follows.

a) Fault Prevention Algorithm during node to gateway communication: When a node needs to send a message to another node in the network, the sender node generates the message with a unique number. The system has designed so as to transmit the message from the source to destination in different paths there are two techniques. The first one being through the Observer node to which it is attached to the gateway. The second technique transmit the message to the gateway through the cousin nodes of the source.

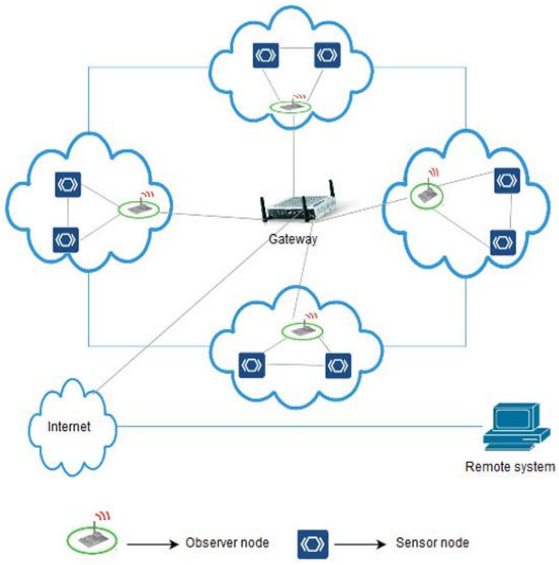

Figure 1. Architecture of the Proposed IoT network

Algorithm1: Fault Prevention Algorithm during node to gateway communication

INPUT-
G-Gateway.
O-Set of Observers. $O = \{O_1, O_2, O_3 \ldots\ldots O_n\}$ where n = no. of observer
N-Set of 'm' nodes for each observer, O_i . where $1 \le i \le n$.
$S_{a,b}$ -Source node. $N_{t,u}$ - Any arbitrary node.
where, $S_{a,b}$, $N_{t,u} \in N$ and *a*,*t*,=observer number, *b*,*u*,=node number
M=Set of Message received by *G*, $M = \{M_x, M_{x+1} \ldots\ldots M_x\}$ where x=Message id.
M_c =New Received Message by *G*
$ON_{n \times m=}$ Observer node adjacency matrix , where n= no. of Observer, m= no. of node assigned to observer
$NN_{m \times m}$ =Node to Node adjacency matrix, where m=no. of node.

OUTPUT- Ack message

Procedure Gateway communication
1. $S_{a,b}$ send message M_c to O_a ;
2. O_a sends M_c to G;
3. *VN*=viable_nodes **($S_{a,b}$);**
4. count = |VN|;
5. for k=1 to count do
6. $S_{a,b}$ sends message M_c to *VN* (k);
7. *VN* (k)) sends message M_c to O[*VN* (k)];
8. O[*VN* (k)] sends message M_c to G
9. end for
10 G Checks, if received message $M_c \notin M_x$.
11. *G* sends ACK massage to earliest O_i
12. O_i sends ACK massage to $S_{a,b}$ (via *VN* (k) if any)
13.end **procedure**
14. **Viable_nodes ($N_{a,b}$)**
15. for i=1 to n do
16. for j=1 to m do
17. if $N_{a,b}$ is adjacent to $N_{i,j}$ && a!=i $N_{a,b}$ is adjacent to $N_{i,j}$ && a==i
18. add $N_{i,j}$ to *VN* **;**
19. else if $N_{a,b}$ is adjacent to $N_{i,j}$ && a!=i
20. **Viable_node ($N_{i,j}$);**
21. endif
22. endfor
23. endfor
24. return *VN*;

Algorithm1 describes the prevention of faults during node to gateway communication. The source nodes transmits its message through the Observer, the Observer in turn transmits the message to the Gateway as illustrated in figure 2. When the gateway receives a message, it checks its register if the message has been received previously or not using the id of the received message. If a match occurs with the register contents, then the gateway simply ignores the message considering it as a duplicate. Otherwise the gateway stores the message details in the register and forward the message to the corresponding node through a reliable path obtained using the path finding algorithm discussed in algorithm 2. In the former case the message would have been transmitted through the cousin nodes of the source node which would repeat the same process of transmission through their Observer to the gateway. In this way by passing the message through three different paths, any message loss due to path failure is prevented.

2) Path Fault Prevention Algorithm: The path fault algorithm Where the first part finds all possible reliable paths depending on the goodness value, the second part uses these paths to provide a reliable path for transmission with the highest goodness value and continues to do so if the previous path fails.

Algorithm2.1 finds the possible path from the gateway to a destination node. The destination node id is known by the gateway, so it finds the corresponding observer id of the destination node using adjacency matrix and considers this as a path from destination node to gateway. Then the goodness value of the path is calculated. Since the path from destination node to gateway is available now, simply reversing this path gives a path from gateway to destination node. After that, it finds adjacent nodes of the destination node and check whether it is under the same observer or not, if it is under other observer then consider this as a new path: destination node adjacent node (under other observer) observer gateway, then by reversing this path; a path is found from the gateway to the destination node and the goodness value of this path is also calculated. However, if the adjacent nodes of the destination node are under the same observer then for every adjacent node, a path is found through the same observer node. In case the destination node is surrounded by other nodes under the same observer node then adjacent node of the destination node is found until the adjacent node is under some other observer node, then a path is found through this other observer node.

Initially, all links to success or failure of transmission, goodness values will change dynamically [11]. Each link's goodness value is maintained by the smart gateway. For any data transmission, when the smart gateway finds the destination node, the overall goodness value of a path is calculated using the goodness value between the intermediary links and the total hop count. Equation 1 depicts the calculation of the overall goodness value between two nodes i and j.

$$\textit{Overall Goodness value } G_{ij} = \frac{\sum g_{mn}}{h_{ij}} \textit{ (0≤G≤1)} \qquad (1)$$

where gmn denotes the goodness value between two intermediary nodes m,n in the path between the nodes i and j and hij denotes hop count between two nodes i,j.

Algorithm 2.2 describes how to prevent path failure. The gateway sorts all possible paths using the goodness value and selects the path with the highest goodness value (Figure-3). Initial goodness values of all links are 0.2. After a successful transmission, the goodness value will increase after receiving an acknowledgement. If by any chance, a reliable path fails three times i.e., after not receiving acknowledgement for three tries, the gateway gives another alternative reliable path.

Algorithm3 illustrates how to detect a node fault or observer fault. Observer node monitors the corresponding leaf nodes. The observer sends wake up message to the corresponding node and waits for ACK. If ACK is received then update the status to the gateway. If an acknowledgment is not received within the waiting threshold, observer node resends the same message three times at random time intervals, if this node is not response then update the status to the gateway. The gateway checks the connection status of this node. If the status is enabled then gateway broadcasts this node is failure to all observers, otherwise gateway broadcasts that this node is switched off to all observers. Meanwhile if any node wants to send data to this faulty node, then the corresponding observer node rejects this request message. Same procedure helps the gateway to detect observer fault.

Algorithm2.1: Possible Path finding Algorithm from Gateway to Node

INPUT-
G-Gateway.
O-Set of Observers. $O = \{O_1, O_2, O_3 \ldots\ldots O_n\}$ where n = no. of observer
N-Set of 'm' nodes for each observer, O_i. where $1 \le i \le n$.
$D_{i,j}$ - Destination node. $N_{t,u}$ - Any arbitrary node, where, $D_{i,j}, N_{t,u} \in N$ and *i*,*t*=observer number, *j*,*u*=node number
$ON_{n\times m=}$ Observer node adjacency matrix , where n= no. of Observer. m= no. of node assigned to observer
$NN_{m\times m}$ =Node to Node adjacency matrix, where m=no. of node.
P_k -Path, where k=path no
HGV-Goodness value of each hop, *GV*- Overall Goodness value
HC_{P_k} -Total hop count.

OUTPUT- Reliable paths according to goodness value.

Procedure Possible Path
1 hop count=0;*k =0, P'$_k$=0 && HGV'=0;*
2. for x=1 to n do;
3. if $D_{i,j}$ is adjacent of O_x
4. *k++;*
5. Add (O_x, $D_{i,j}$) to P_k ;
6. Add(O_x, G) to P_k ;
7. hop count=+2;
8. HC_{P_k}= hop count;
9. break;
10 end if
11. *HGV = HGV of $D_{i,j}$ to O_x +HGV of O_x to G;*
12. $GV_{P_k} = HGV/HC_{P_k}$;
13. end for
14.*AP*=**Alternative_Path($D_{i,j}$)**
15. P_k =*AP*;
16.end **Procedure**
17. **Alternative_Path($N_{x,y}$)**
18. hop count=0;
19. for i=1 to n do
20. for j=1 to m do
21. if N_{ij} is adjacent to N_{xy} && x==i
22. *k++ &&* hop count *++;*
23. Add (N_{ij} ,N_{xy}) to P_k ;
24. *P'$_k$= P_k ;*
25. *HGV = HGV of $N_{i,j}$ to $N_{x,y}$;*
26. *HGV'=HGV;*
27. Add (O_i , $N_{i,j}$) to P_k ;
28 hop count ++;
29. *HGV =HGV of O_i to $N_{i,j}$ + HGV*;
30. Add (G, O_i) to P_k ;
31. hop count++;
32. HC_{P_k}= hop count;
33. *HGV =HGV of G to O_i + HGV*;
34. $GV_{P_k} = HGV/HC_{P_k}$;
35. **Alternative_Path($N_{i,j}$)**;
36. else if N_{ij} is adjacent to N_{xy} && x!=i
37. k++ && hop count++;
38. if P'$_k$!=0;
39. Add (N_{ij} , N_{xy}) && P'$_k$ to P_k ;
40. *HGV = HGV of $N_{i,j}$ to $N_{x,y}$* + HGV';
41. hop count++;
42. else
43. Add (N_{ij} , N_{xy}) to P_k ;
44. *HGV = HGV of $N_{i,j}$ to $N_{x,y}$;*
45. end if

46. Add (O_i ,$N_{i,j}$) to P_k ;
47 hop count++;
48. *HGV =HGV of O_i to $N_{i,j}$ + HGV*;
49. Add (G , O_i) to P_k ;
50. hop count++;
51. HC_{P_k}= hop count;
52. *HGV =HGV of G to O_i + HGV*;
53. $GV_{P_k} = HGV/HC_{P_k}$;
54. endif
55. endfor
56.endfor
57.return P_k ;

Algorithm2.2: Path Fault Prevention Algorithm

INPUT- Goodness value (GV_{P_k}), where P_k = k^{th} path.
Path (P_k)= (P_k $P_{k+1...}$ P_k) where k= path number.
M_c -Current message. Threshold time (T). Waiting Time(WT).
Destination node ($D_{t,u}$), where t=observer no. u=node no.

OUTPUT- Ack from Destination node.

1. Gateway sorts all paths according to goodness value GV
2. for k=1 to n
3. If all resulted path goodness value same then
4. select random path P_k ;
5. set count=3;
6. else
7. select highest goodness value path P_k ;
8. set count=3;
9. end if
10.end for
11.send message M_c to destination node use P_k ;
12. if(count>0)
13. if T< WT && gateway received ACK then
14. increase good ness value of each hop by 0.1 in the path P_k ;
15 count =-1;
16. break.
17. count --;
18. goto 10;
19. end if
20. end if
21. if count=0
22. decrease each good ness of each hop by 0.1 in the path P_k ;
23. end for

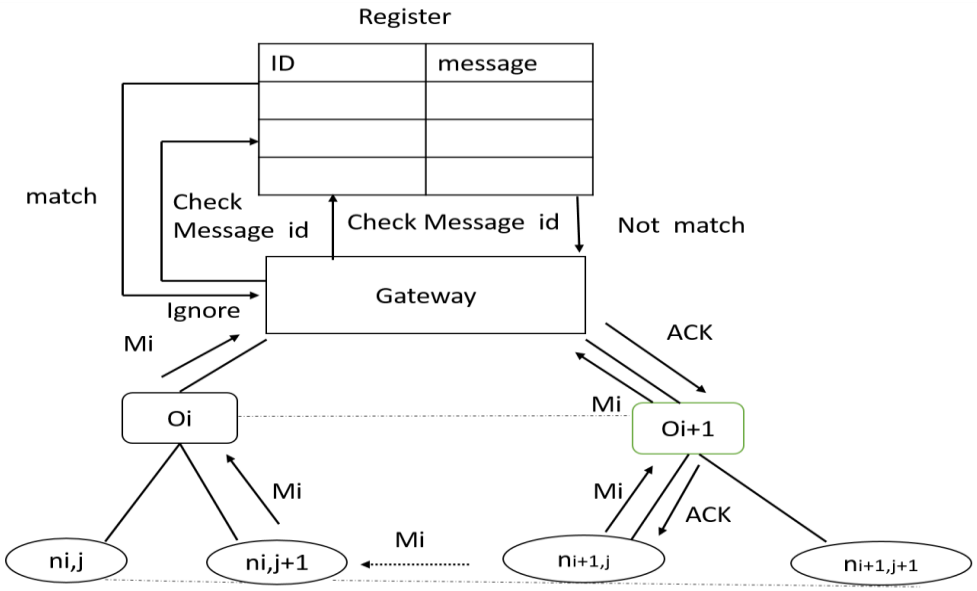

Figure. **2.** Illustration of Fault Prevention Algorithm during node to gateway communication

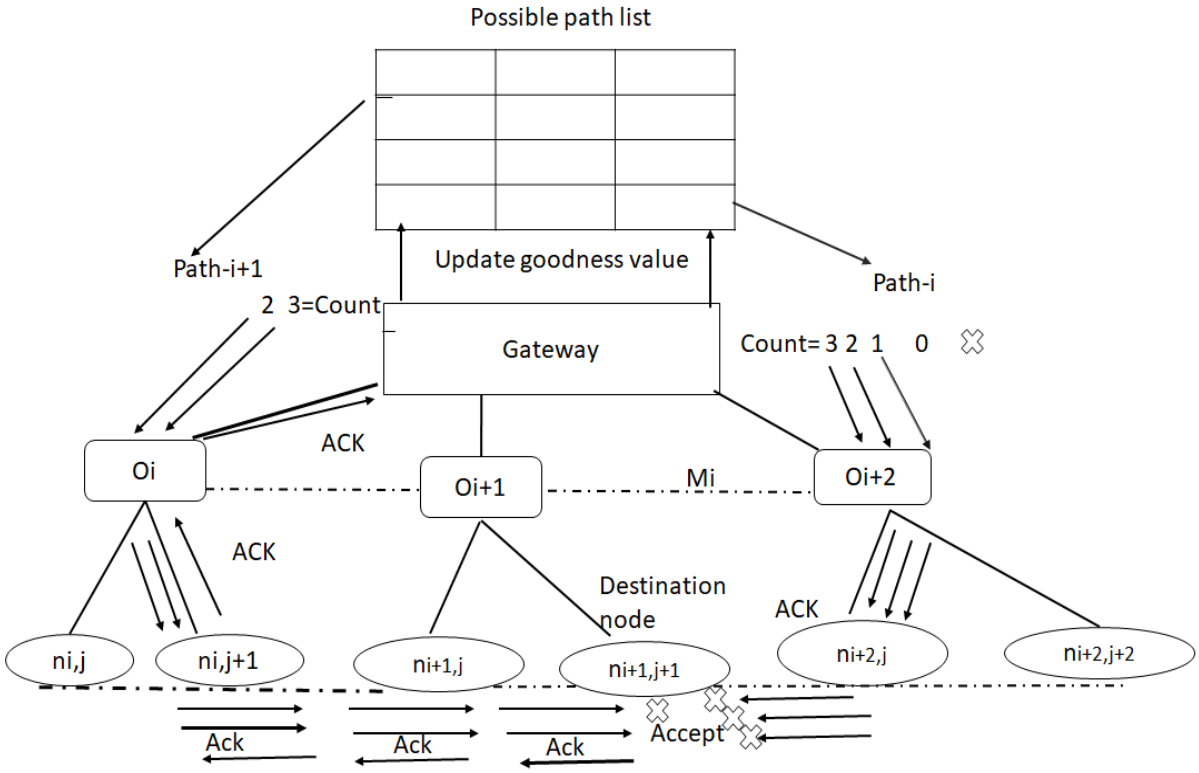

Figure. **3.** Illustration of Path fault prevention algorithm

Algorithm3: Fault Detection algorithm

INPUT-
G-Gateway.
O-Set of Observers. $O = \{O_1, O_2, O_3 \ldots\ldots O_n\}$ where n = no. of observer
N-Set of 'm' nodes for each observer, O_i where $1 \le i \le n$.
$N_{t,u}$ - Any arbitrary node. where, $N_{t,u} \in N$ and *t*=observer number, *u*=node number
$ON_{n \times m=}$ Observer node adjacency matrix ,where n= no. of Observer.
m= no. of node assigned to observer
Threshold Time=T, Random Time interval-RT, Node Response time=NRT
Observer Response Time=ORT, waiting time=WT

OUTPUT- Fault node id/Observer id

1.Repeat – After T time
2. for i=1 to k do
3. for j=1 to m do
4. count=3;
5. O_i sent wake up message to $N_{i,j}$
6. if observer received ACK from $N_{i,j}$ && NRT<WT then
7. if count >0
8. set wake up status=1.
9. count--.
10. O_i update wake up status of $N_{i,j}$ to Gateway.
11. end if.
12. while (NRT>WT) do
13. if count >0
14. count - -
15. After RT goto step-6.
16 end if
17. set wake up status=0
18. O_i update wake up status of $N_{i,j}$ to Gateway.
19. if Gateway received $N_{i,j}$ wakeup status=0 then
20. if $N_{i,j}$ connection status=1 then
21. send message to all O_i , $N_{i,j}$ is failure.
22. else
23. send message to all O_i, $N_{i,j}$ is switch off.
24. end if
25. end if
26. end while
27. end for
28. if O_i connection status=1 && ORT>T then
29. count=3;
30. gateway send wakeup message to O_i ;
31. if count >0
32. if gateway received ACK + Update && ORT>WT then
33. count= -1
34. exit.


```
35                    count- -;
36.                   After RT goto step-30;
37.                 end if
38.       end if
39.       if count=0 then.
40.           gateway send all Observer, O_i is failure;
41. end for
```

VIII. CONCLUSION

In an IoT system reliability has an important relation to robustness. A system which is more fault tolerant is more reliable and thus more robust. The existing system discusses various fault tolerant handling mechanisms in IoT domain as well as non-IoT domain. However, there is no solution for single reliable path failure for a node to node communication. the proposed framework has a total of four algorithms and a goodness value matrix, which is help find the reliable path in prevention mechanism. In prevention mechanism, the fault prevention algorithm during node to gateway communication prevented node to gateway communication failure and the path fault algorithm has two-part, which prevented gateway to node communication failure. The first part finds all possible reliable paths depending on the goodness value, the second part uses these paths to provide a reliable path for transmission with the highest goodness value and continues to do so if the previous path fails. The detection algorithm detects the node faults and observers fault at an early stage and also manage the overall load balance after failures. The observers are handling the unnecessary gateway traffic. In addition, it also handles hardware failure, software failure, and connection failure.

REFERENCES

[1] Q. Zhu, R. Wang,Q. Chen, Y. Liu & W. Qin, "Iot gateway: Bridgingwireless sensor networks into internet of things", Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on. IEEE, 2010

[2] H. Suo, J. Wan, C. Zou, & J. Liu, "Security in the internet of things: a review", In Computer Science and Electronics Engineering (ICCSEE), 2012 international conference on (Vol. 3, pp. 648-651). IEEE

[3] G.White,V. Nallur, & S. Clarke," Quality of Service Approaches in IoT": A Systematic Mapping", Journal of Systems and Software, 132, pp.186-203 ,2017

[4] M.C. León, H. Meyer, D. Rexachs, & E. Luque, "Fault tolerance at system level based on RADIC architecture", Journal of Parallel and Distributed Computing, 86, 98-111, 2015.

[5] W. Chen, R.F. da Silva, E. Deelman, & T. Fahringer," Dynamic and fault-tolerant clustering for scientific workflows", IEEE Transactions on Cloud Computing, 4(1), 49-62,2016.

[6] Y. Challal, A. Ouadjaout., N. Lasla, M. Bagaa & A. Hadjidj, "Secure and efficient disjoint multipath construction for fault tolerant routing in wireless sensor networks", Journal of network and computer applications, 34(4), pp.1380-1397, 2011.

[7] T.N. Gia, A.M. Rahmani, T. Westerlund, P. Liljeberg, & H. Tenhunen, "Fault tolerant and scalable IoT-based architecture for health monitoring", In Sensors Applications Symposium (SAS), 2015 IEEE (pp. 1-6). IEEE.

[8] S. Misra, A. Gupta, P.V. Krishna, H. Agarwal, & M.S. Obaidat, "An adaptive learning approach for fault-tolerant routing in Internet of things", In Wireless Communications and Networking Conference (WCNC), 2012 IEEE (pp. 815-819). IEEE.K.

[9] M.W. Woo, J. Lee, & K. Park, "A reliable IoT system for Personal Healthcare Devices. Future Generation Computer Systems" , *78*, pp.626-640 ,2017.

[10] S.A. Karthikeya, Vijeth, J. K., & C.S.R. Murthy,"Leveraging Solution-Specific Gateways for cost-effective and fault-tolerant IoT networking" In Wireless Communications and Networking Conference (WCNC), 2016 IEEE (pp. 1-6). IEEE**.**

[11] S. Misra, P.V. Krishna,A. Bhiwal, A.S. Chawala, B.E. Wolfinger, & C. Lee. "A learning automata-based fault-tolerant routing algorithm for mobile ad hoc networks." The Journal of Supercomputing 1-20,2012.

Authors Profile

Mr. Tapas Saha pursed Bachelor of Science from Visva-Bharti University, India in 2013 and Master of Science also from Visva-Bharti University in year 2015. He is currently pursuing M.Tech. degree from Pondicherry University.

R.Sunitha is an Assistant Professor in Department of Computer Science, Pondicherry University. She has received her Ph.D. in Computer Science and Engineering in the year 2014. Her research includes Knowledge Engineering, E-learning. She has published more than 20 papers in various reputed Journals and Conferences. She is a recipient of the 'Best Teacher 'award from Pondicherry University.

