Towards Formalism of Link Failure Detection Algorithm for Wireless Sensor and Actor Networks

Umar Draz
CS. Department
(CUI) Sahiwal, Pakistan
sheikhumar520@gmail.com

Tariq Ali
CS. Department
(CUI) Sahiwal, Pakistan
tariqali@cuisahiwal.edu.pk

Sana Yasin
CS. Department
(CUI) Sahiwal, Pakistan
sanayaseen42@yahoo.com

Umair Waqas
CS. Department
LGU, Lahore, Pakistan.
umair.waqass@lgu.edu.pk

Usman Rafiq
CS. Department
(CUI) Sahiwal, Pakistan
usmansh32@gmail.com

***Abstract*— The merger of actors and sensors in a wireless network has evolved those opportunities that we can't think before some years. This merger has captured the interest of researchers from around the globe. In the last decade, wireless networks have become much stronger, reliable and secure but as the technology evolves, it carries their own problems. Researchers are targeting different problem according to their own interest like power consumption, network security, link failure etc., so we are also working on link failure detection that can be caused by power breakage, network delay or traffic overload. This model detects the link failure in the network through Link Failure Detection Algorithm (LFDA) and provide recovery mechanism against failure using cluster-based approach. Our purposed model detects network link failure accuratly and precisely solve the problem by creating alternative virtual route for data packets. Our technique can detect current failures also it can detect weaker links, those might be the cause for future failures. Most of the literature have their proof of correctness as simulation but no technique is there which is formally verified, therefore we have presented our idea including its formal verification and validation with the help of formal methods tool box and its formal specification language like (VDM-SL).**

Keywords—WSAN; Link Failure; Link Recovery; Gateways Node, Cluster head, Virtual Links; Verification & Validation

I. INTRODUCTION

In last decades, there have been mounting attentions in the field of Wireless Sensor and Actors Networks (WSAN), due to addition of actors. It consists of a various type of nodes that are deployed in any areas like disasters and surveillance's, to perform a specific task that doesn't perfume by the human being. These nodes perform self-scheduling (e.g; MANET) which causes many issues in the network. One of them is Link Failure (LF) that is a major issue of WSANs. LF happened at any part of the network like between the nodes and clusters, source and sink. It causes several other issues in the network like network failure (due to constantly failed the links between the nodes), network congestion and breakdown. In WSAN actor and sensor coordinate with each other and make their task complete in a well-disciplined manner [1]. The deployment of WSAN in harsh environment that remotely control with the help of actor and sensor co-ordinations is a major task. A link failure causes the inert-connection failure which further leads to the network failure. There are many types of approaches that deal this issue like centralize, distributed and hybrid. Among these approaches, the distributed approach is proactive in nature, that deal the LF issue efficiently [2]. Network connectivity gains too much attention in WSAN because lot of important parameters of the network depends upon the network connectivity; no matter it is connected through wired connection or wireless. All these parameters depend on the energy aptitude and load correspondence. The major problem in network connectivity is topology of the network. The topology is overall matter, to deal these types of issue. Overall representation of the WSAN in the form of clusters is shows in figure 1.

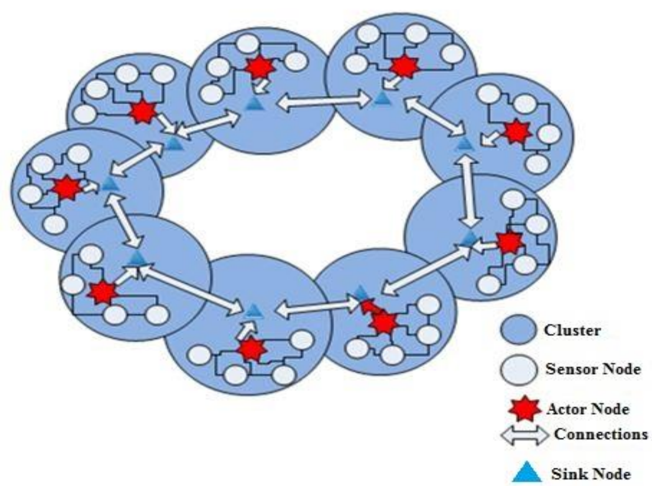

Fig. 1: Cluster based wireless sensor and actor-network

Bassically its need to design a distinctive model that solve the LF issue efficiently. In this paper, our focus is to detect the LF issue from the network by adopt the cluster-based approach (as display in fig 1). Bassically central node of the network utilize more power due to high transmission and turn out to be dead early than the other node in the network. LF occurs according to the node failure position, for example; if a node fails in between the network than LF will take place in between the network or if it exits insides the leaf node than LF will occur at the boundaries of the networks [3]. Anyhow many other possibilities are also present except above.

Most of the research have been done on the failure issue but today research still demand an efficient solution like in our previous work that is done through proactive approach [4, 9-10]. Link Failure Detection Algorithm (LFDA) is designed to solve the LF problem from a new prospect. LFDA algorithm against LF issue is to resolves this vulnerable issues inside the network. It firstly detects the links that become failed and are going to fail, and then produce alternative solution against them.

Rest of the paper is managed as: the related work and purposed model is described in Section II and III. The LF algorithm and its formal specification is described in section IV and V. The model analysis and conclusion is discussed in section VI and VII.

II. RELATED WORK

Existing technologies against LF issue are routing-inspired technique, energy-aware distributed routing algorithm, Novel Duel Separate Path Algorithm, supervisory routing control in low data rate, distributed cooperative target detection in decentralized wireless networks [3, 7]. The Distributed Actor Recovery Algorithm (DARA) purposed in [8] for a large network that is affected by actor link failure. In [6] the survey is done to highlight the silent features of the wireless sensor network and its connectivity. The purpose of this study is to diagnose the link failure proficiently in the network. In [2] anomaly detection using Support Vector Machine (SVM) is done in a medical wireless sensor network that based on the Linear Regression Model (LRM). This model is able to explore the anomalies in the network. A sequence-based approach is used in [11] that is dynamic in nature and detects a failure in the network. In [5, 13] issue of links failure is resolved through the topology. Topology management is done to tolerate the links failure in the WSANs. Network outlier detection is done in [8] that are carried out at the central processing links. It ignores the problems associated with fractional message loss. Detection of the targeted movements through link tracking is done in [10] that maintain the operation of the node in manner able fashion by turning their services off and enabling only the group of the node for the functional targets. A lightweight secure scheme is presented in [12] that detects provenance frogary and packet drop attack in WSN. It initiates multipath transmittingto build the dissemination tree around the cooperatednode. Adaptive routing is done in [7] that is based on the QoS optimization for enriched application performance. Effective key management also playsan energetic role in link failure detection. It captures any change in the network like humidity, temperature change, solar effects, joining or leaving the new sensor node in the network and report to the base station about all these changes. A probabilistic fault detector is used to detect all the failure in the network that is occurring due to the hardware. If any packets are loss or divert from the actual path is efficiently detected through this fault detector. Mobile Target Tracking (MTT) that is based on local area prediction is done in the WSANs. This technique is specially designed for the dense and sparse network in which high resilient, energy efficient multipath routing is presented. All the work is done on the disjoint path that provides the independence to the network. Most of the work is done by using formal specification and simulation like cloud-based watchman inlets and mobile crowd sensing in WSAN are also verified by using VDM-SL as [14, 5-6].

III. PURPOSED MODEL OF LINK FAILURE

Due to the distributed environment of the network LF till requires being focused more appropriately to overwhelm the issue. Existing work in the literature does not deal efficiently to resolve this issue, because most of the work is based upon simulation based. So link failure detection (LFD) and recovery model is necessary to resolve the LF issue from a new skyline, for example its need to formally verify the technique first by using ant Formal Method Tool Box like VDM-SL. The purposed model firstly detects and then resolves the LF issue in the network. It also reduces the chances of LF occurs in the network. In this approach, we assume the dynamic 2D-WSAN environment that consists of 'N' numbers of nodes that are arbitrarily positioned in the network. Figure 2 shows the link failure between two nodes like source 'A' want to send a message to destination 'R'. Firstly 'A' send s a RREQ to the path in which the destination 'R' present. Two paths acknowledgements are received like [AEFMNQR] and [ADCIJTSR]. If any node does not make a RREP inside the dedicated path then there must be a link failed between the nodes. For example; all nodes make a RREP for the path [AEFMNQR] but for the second path [ADCIJTSR] the link is failed 'J' and 'T', so the communication can be stop and not be forwarded towards 'S' and then finally 'R'. in this case the common practice of the node is that the route RREP only be generate completely when the link is stable throughout the destinations. The node 'T' 'S' and 'R' would not make a response towards the source 'A'. By and large; the proposed link failure strategy is working in a well-disciplined manners when the link is break between the nodes etc. Due to deployment of gateways nodes, the ultimate information about the failure of links is send and proper action is act upon the network.

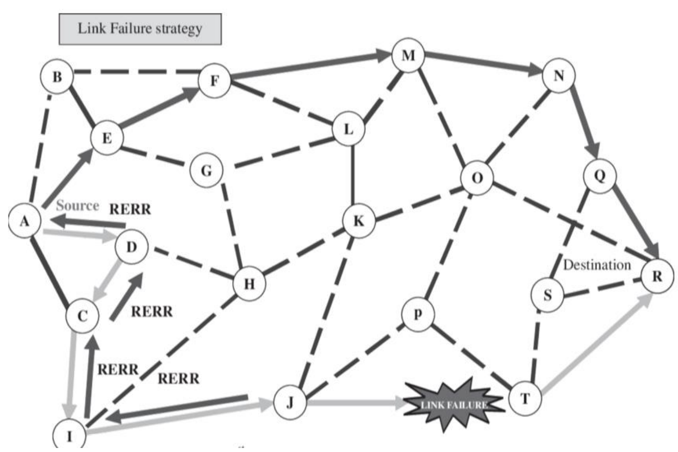

Figure 2: Representation of Link Failure in WSAN between source and destination

IV. PURPOSED ALGORITHM FOR LINK FAILURE DETECTION

This section presents the purposed algorithm for link failure problem called Link Failure Detection Algorithm (LFDA). The high level pseudo code of purposed model is presented in figure 4. The whole LFDA consist of further two segments by mean link failure detection and recovery. First segment deal the detection of link failed inside the cluster of two nodes while the other segment represents the recovery mechanism.

LFDA ()

Segment :01

1. WSAN is deployed in the form of actors, sensors and Gateways in the form of Clusters
2. The Sensors and Actors nodes are coordinate each other.
3. Sink Node check the status source and destination and its relevant path
4. Check the status of each node with respect to energy
5. RREQ forwarded against whole network nodes (including neighboring nodes)
6. Receive RREP against possible RREQ
7. If RREP does not receive then message broadcasting to Gateway's and Gateway inform to nearby Actor Node
8. Actor Node look after the missing RREP path and detect the any link fail between the nodes
9. The whole information send towards the sink inside the cluster
10. For a time being; Sink utilize the alternative path for data transmission until unless the recovery strategy for link failure act upon

Segment :02

11. After the detection of LF inside the cluster; the Actor Node place itself with replacing the failure Node in order to make the Virtual blocks (this is the only consideration for this research work)
12. Virtual Blocks Takes the data and the previous RREQ from the source
13. Against the RREQ Virtual links forwarded the RREP towards the destination
14. Furthermore Virtual blocks are used to store the link failure locations and the data of corresponding nodes
15. All virtual links are coped with failure links
16. Gateway is informed link failure than communicates with nearby Gateway in order to set the virtual links value
17. This process is continue until unless the temporary Virtual links replace with original communication links
18. The link recovery process is continue until the network become stable

Fig. 4: High level pseudo code of link failure detection and its recovery

V. FORMAL SPECIFICATION OF LINK FAILURE DETECTION ALGORITHM

This section represents the formal specification of the LFDA for WSAN sparsely connected through sensors, actors and gateways nodes. The scattered topology is presented in the form of composite objects developed through Vienna development language in the VDM-SL toolbox.

```
types
Location::
X coordinate: N
Y coordinate: N
Z coordinate: N

Energy= <high> | <Average>| <low>;
State-owned = <Failure_Occur>| <Failure_ Not_ Occur>;

Record = token;
Link_Connectivity = <Connected_Network> |
<Disconnected_Network>;
Critical_points = <Critical_Situation >|
<Non_Critical_Situation>
Network_Range = <Short_Range>| <Long_Range>;
Links:: sensor_node: Sensors
Actor_node: Actors
Gateway_links: Gateway

location: Location
power: Power
state-owned: State
Failure_information: set of Data
criticality: Criticality
connection: Connected

Link = network_links*network_links
InvNetwork_link== let mk_ (network_links1,network_links2)
= linkin lnetwork_links1 <>network_links2;
network_links= set ofnetwork_link
InvNetwork_links== Forall mk_
(network_links1,networklinks2) In Setnetwork_link&
mk_ (network_links2, network_links1) In Setnetwork_links;

Network_cluster::set of cluster:network
network_node:set ofnetwork_Links
network_links: aset ofnetwork_links
network_radio_frequency:nat
sensor_node:set of sensor_node
actors_node:set of actors_node
gateway_node:set of Gateway_node
infor: node_Information
adjacent_node:Adjacent

Invmk_network_cluster(network_cluster,network_node,
network_links,-,sensor_node,actor_node, gatway_node,-
,-) == forall sensorin setnetwork_cluster&
(card sensor.node>=3) and
forall R in set network_node&
(Exists _Sensor_nodes in set sensor_Nodes
& (Exists actor_Node in set_Actor_Nodes &
(Exists gateways_Node in set
Gateways & ({N} = Sensor_Node .node and
{R} = actors.node And
{R} = Gateway_Node.node)))) And
Forall sensor_node1, sensor_node2 in Set Sensors &
(Exists l in set Links &
```

(sensor_node1.sifinder= l node.links1 ***And***
sensor_Node2.sifinder = lnode.links2)) ***And***
actors_node.aifinder=lnode.links2))) ***And***
actor_node2.aifinder=lnode.links2)) ***And***

Forall *Sensor_node* ***In Set*** *Sensors* ***&***
Exists *Gateway_node* ***In Set*** *Gateways* ***&***
Gateway_Node.Node_Information.
Receive*=Sensor_node.Information.detected****and***
Forall *Sensor_Node* ***In Set*** *Sensors_Node* ***&***
(exists actor_Node ***In Set*** *actor_node****&***
(Sensor_node.radio_Frequency<
Actor_Node.radio_Frequency)) ***and***
Forall actor_node ***in Set*** *actors****&***
(Exists gateway _node ***inSet*** *Gateways* ***&***
(Actors.network_Radio_Frequency
<gateway_node.Network_rdFreq;

There are two blocks, *virtual* and *link-failure* in this specification to make the link failure mechanism transparent. A *virtual block*is used to store virtual links in the network. *Link-failure block* store the failed link in the network. Some limitations are represented below in this model that have to be satisfied the basic requirements of the specification during the designed model.

Invariant :*(1)* The number of clusters must be smaller than the threshold value. It cannot be exceeded to the pre-defined limit. *(2)* Wireless link should be exists between the clusters and their actor, sensor, Gateways and destination node.

Initializations: Clusters must be vacant before starting the modeling. Before modeling, no link will be available in the network.

state ***WSAN*** *of*
*network_cluster: a S****et of****network_clusters*
network_links_edges: a ***set of****network_links*
*sensors_node:****set of*** *Sensor_node*
*actor_node:****set of****actors_nodes*
*Sink_nodes:****set of****sink_node*

*Failure_node:****set of*** *failednode*
*Virtual_nodes Block:****Set of*** *Virtual links*
Path_way: ***seq of****links*

inv mk*_WSAN(cluster)== Card_ cluster<=****limit***
inv mk*_WSAN(cluster,links,sensors_nodes,actors_nodes, sink,message_pathway)*
== Forall ***LINK in set****links****&****links1* ***inSet*** *sensor_nodes_links And*
Edge.Links2 ***in set*** *Sensor_Node* ***And*** *forall kk* ***inSet*** *sensor_node*
init mk *WSAN(cluster)==cluster={}*
END

The creation of cluster function takes all the links ID's as an input function than based on these links all the nodes are added to the clusters.
Some preconditions before the formation of the cluster and *sink _links* are mentioned below.

Pre-conditions :*(1)* The new formatted clusters should not exist in the cluster block before cluster formation and it must be smaller than the threshold limit. (2)The links that belong to the sink nodes should also does not in the cluster block before formation.

Functions

Cluster_Formation(Network_Id:Links)
*Ext cluster:****Set of****Node*
pre-Network_In ***Not in Set*** *cluster****and Card*** *Cluster<****limit***
Post cluster= cluster~ union ***{NettIn};***

create_sinklinks(LinksId:Links)
ext wr cluster: ***set of****Node*
ext wr sink: ***set of****Node*

*pre LinksId****not in set****clusterand* ***not in set*** *sink*
post cluster= cluster~ union ***{LinksIn}***
Post Sink= Sink~ Union ***{LinksIn};***

Actor node plays ahead links role in clustering model. The nodes in the clusters are responsible to resolve the link failure issue in the network. Actor nodes uniquely collaborates to each other to finds the link failure in WSN. There are lots of functions in this modeling. The actor creation function takes the link id as input and writes clause the nodes in to the clusters block. Sensor nodes are the basic unit of the network. These nodes does not play any management role in the network. These nodes have only responsibility to sense from the environment and done the basic functionalities of the network. Every cluster block in the network contains some sensor nodes to perform sensing task, actors nodes to play the management role and some source and sink node to send and receive the traffic in the network. Links also plays a major role in the network. Links is also created between sensor actors and sink nodes. There are some pre and post-conditions for these modules.

Pre-conditions*: (1) The links between the clusters must not exist in the cluster and actors field before formation. The cardinality of the links must be less than the threshold limit. (2)* The links between the sensors nodes should not be exist before formation.

Create _actor_node_links(LinksId:Links)
*Ext or cluster:****a Set of****Node*
*Ext or actors:****a Set of*** *actor_node*

pre-LinksIn ***not inSet****cluster****and Card****cluster<****limit*** *And*
LinksIn ***not in Set*** *actors and* ***card*** *actors<limit*

Post cluster= cluster~ Union ***{LinksIn}***
Post actors= actors_Node~ Union ***{LinksIn};***

Create_sensorlinks(LinksId:Links)
*Ext or cluster:****a set of****Node*
*Ext or Sensors_Node:****a set of****Node*

Pre-LinksIn ***not in set*** *sensors* ***and*** *card sensors <****LIMIT***

*post cluster= cluster~ Union **{LinksIn};***
*Post sensor_node= Sensors_Node~ Union **{LinksIn};***

Virtual and Dead blocks are the two major blocks of the modeling. These blocks provide the support to the nodes that play the major responsibility of the network. When network suffer from the link failure issue the transmission system become stop and data cannot be transmitted between the nodes. Actor nodes play the head node responsibility in the network. It detects the link failure in the cluster and outside the cluster. Virtual block replace the dying node links with the virtual links to provide the stability to the network. Dead block contains the all the dead links from the network.

Pre Conditions :*(1)virtual block link should not be belongs to the virtual block before formation of the block.*

(2) dead block link should not be belongs to the virtual block before formation of the block.

*Create_Virtual_Block**(Virtual_ID:Links)***
*Ext or Virtual_Block:**a Set of**Node*

*Pre-V_Id **not in set** Virtual_Block **and** card Virtual_Block **<limit***
*post Virtual_Block= Virtual_Block~ Union **{LinksIn};***

Create_Dead
*-Block**(N_Id:Links)***
*Ext wr Dead_Block: **Set of**Node*

*Pre N_Id **not in set** Dead_Block **and** Card Dead_Block **<limit***
*Post Dead_Block= Dead_Block~ Union **{Node_Id};***

To detect the active links in the network cluster active node function is created in the network.This function takes thhe links id as an inputv and checks that this links is active or not.To remove the dead links from the networks remove dead link nfunction is created that takes the link id asa input and if it is dead link than remove that link from the network.

Pre-Condition*:(1) The links that have to remove from the network must be exists in the network.*

Post-Condition*:(1)*both functions active cluster and remove dead node must returns the active and dead links from the networks.

*Cluster_active_Node(S_Id:Links)Output:**Set of**Node*
*Ext rd cluster: **Set of**Node*

*Pre **true***
*Post output=**cluster;***
*remove_failed_links**(LinksIn:Links)***

*Ext wr cluster: **Set of**Node*
*Pre LinksInIn **in set**cluster*
Post cluster = cluster~ \ {Network_tIn}
Post Dead_Block= Dead_Block~ Union {LinksIn};

To returns totall number of cluster from the network totalcluster function is formatted. This function returns total number of clusters in the network and it is very effective in the network implementation. No input is requiring for this function. It just returns the number of clusters. There are some invariants for this module:

Invariants: Only same types of nodes have permission to create the links. Like actor node only connected with the actor nodes and sensor node will be connected with the sensor nodes.

*Totalclusters()Total:**Nat***
*Ext Rd cluster: **Set of**Node*
*Pre **True***
Post Total= Card cluster;

Network_topology(Sensor_node1:Sensors,actor_node1:actors node,sink_node1:sink_node)
*Create_Communication_cooridoor::**sensor_node1:sensor_node***
*actor_node1:actors_node,sink_node1:**sink_node;***

***Inv mk**_edge(Sensor_Node1, Actor_node1,sink_node1) ==*
Sensor_node1 <> Actor_node1<>sink_node1;

*The major function of network modeling is the Link failure Detection function ().*Two major input parameters are require for this function. The first one is *links_id* and the second one is dead id .Both these parameters helps to detect the failure in the network.

*Link**_failure_detection**(Links_id:Links,DeadIn:Links)Query:**Bool***
*Ext rd Dead_Block: **Set of**Node*
*Ext rd Actors_Block: **Set of**Node*
*pre **True***
*post Query <=> DeadIn **in Set** Dead_Block;*

The solution to the link failure problems is the virtual and dead block. Both these blocks read out the cluster and detects the dead links in the cluster . If dead link is find out in the network than link failure solution function replace the *dead links id* with new *virtual links _id* and adds the dead links into the die block and removes this links from the cluster. A *virtual _links* is also removed from the virtual block and add to the cluster with dead links id. There is some limitation of this module like:

Pre-condition:For link failure solution, the dead links must exist in the cluster.

Post-condition: Dead links is replaced by virtual links in the cluster and the dead links is added to the die block. A virtual linkis removed from the virtual block and adds to the cluster.

RecoveryMechnism(Linksid:Links,LinksIn:Links,V_Id:Links)

*Ext rd failure node: a **Set of**Node*
*Ext or cluster: a**set of**Node*

*Pre LinksIn exist in cluster **And** link failure*
*Post cluster= cluster~\ **{LinksIn};***

*post cluster= cluster~union **{V_Id};***
*post virtualblock= virtualblock~\ **{V_Id};***

VI. RESULT AND ANALYSIS

The model analysis of formal specification of Link Failure Detection Algorithm (LFDA) is provided using VDM-SL toolbox. Now-a-days the trend to convert the real time application to formal specification by using VDM-SL is increased. A snapshot of the model analysis that is dynamically checks all the invariants is presented in figure 5. Invariants, pre/post conditions, operation and possible attributes of the purposed LFDA are defined in integrity checker and pretty checker is display in table 1 that describe the possible model analysis of the whole algorithm. It is observed that through syntax and semantic check there is no error is finding throughout the whole model and its specification.

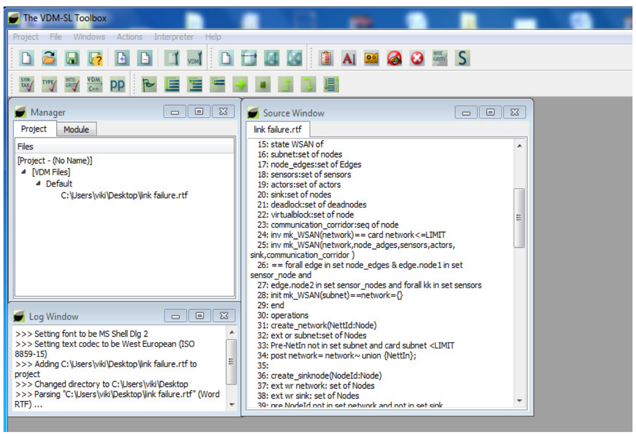

Fig. 5: Proof of correctness of LFDA

TABLE 1. Model Analysis of LFDA

Different attributes and possible manipulates invariants	***Syntax-Wise***	***Type-Check***	***Pretty-Printer***	***Integrity Check***
Object	ok	ok	ok	Y
Link failure	ok	ok	ok	Y
Link detection	ok	ok	ok	Y
Virtual Links	ok	ok	ok	Y
Nodes & actors	ok	ok	ok	Y
Link failure	ok	ok	ok	-
Alternative failure	ok	ok	ok	-
Link recovery	ok	ok	ok	-
Actor link	ok	ok	ok	ok
Sensor link	ok	ok	ok	ok
Execution	ok	ok	ok	ok
Pre/post conditions	ok	ok	ok	-
Validation and verification	ok	ok	ok	-

VII. CONCLUSION

In this paper, the Link Failure Detection Algorithm and technique for recovery from the failure is proposed. The main focus of the technique is actors and sensors networks. The purpose algorithm can proactively detect link failures from the network and provides a recovery mechanism against this issue. In result of this recovery mechanism, virtual links will be formed, and the network will not face traffic overload. Further, it minimizes the chance of link failure in the future. When any link failure occurs only actor node detects it and resolve the issue at cluster level. The purposed LFDA is verified and validated through Vienna development method using specification language and its toolbox. Previously simulations are used as a proof of correctness moreover graph theory has also unique relation with networks and widely used for describing the ideas. but these techniques can't give formal verification and validation, so we are using formal methods that is very helpful in this regard. The beauty of formal methods and its specification is, its formal proof of correctness and its model analysis mechanism, in which syntax including their semantics and with different invariants (initial conditions) can be easily verified. The purposed approach that is verified can be easily integrated with any system or real-time application.

VIII. REFERENCES

1. Rawat, P., et al., Wireless sensor networks: a survey on recent developments and potential synergies. The Journal of supercomputing, 2014. **68**(1): p. 1-48..
2. Chanak, P., I. Banerjee, and R.S. Sherratt, Energy-aware distributed routing algorithm to tolerate network failure in wireless sensor networks. Ad Hoc Networks, 2017.
3. Gupta, G. and M. Younis. Fault-tolerant clustering of wireless sensor networks. in Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE. 2003. IEEE.
4. Ali, T., Jung, L. T., & Faye, I. (2014, June). Three hops reliability model for Underwater Wireless Sensor Network. In Computer and Information Sciences (ICCOINS), 2014 International Conference on (pp. 1-6). IEEE.
5. Yasin, S., Ali, T., Draz, U., & Rasheed, A. (2018). Simulation-Based Battery Life Prediction Technique in Wireless Sensor Networks. NFC IEFR Journal of Engineering and Scientific Research, 6, 166-172.
6. Ali, T., Noureen, J., Draz, U., Shaf, A., Yasin, S., & Ayaz, M. (2018). Participants Ranking Algorithm for Crowdsensing in Mobile Communication. EAI ENDORSED TRANSACTIONS ON SCALABLE INFORMATION SYSTEMS, 4(16)..
7. Kamal, A.R.M., C.J. Bleakley, and S. Dobson, Failure detection in wireless sensor networks: A sequence-based dynamic approach. ACM Transactions on Sensor Networks (TOSN), 2014. **10**(2): p. 35.
8. Zhang, Z., et al., A short survey on fault diagnosis in wireless sensor networks. 2017..
9. Draz, U., Ali, T., Yasin, S., & Shaf, A. (2018, March). Evaluation based analysis of packet delivery ratio for AODV and DSR under UDP and TCP environment. In Computing, Mathematics and Engineering Technologies (iCoMET), 2018 International Conference on (pp. 1-7).
10. Draz, U., Ali, T., Khan, J. A., Majid, M., & Yasin, S. (2017, November). A real-time smart dumpsters monitoring and garbage collection system. In Aerospace Science & Engineering (ICASE), 2017 Fifth International Conference on(pp. 1-8). IEEE.
11. Ali, T., Ayaz, M., Jung, L. T., Draz, U., & Shaf, A. (2017). Upward and Diagonal Data Packet Forwarding in Underwater Communication.
12. Ali, T., Noureen, J., Draz, U., Shaf, A., Yasin, S., & Ayaz, M. (2018). Participants Ranking Algorithm for Crowdsensing in Mobile Communication. EAI ENDORSED TRANSACTIONS ON SCALABLE INFORMATION SYSTEMS, 4(16).
13. A.-S. K. Pathan, *Security of self-organizing networks: MANET, WSN, WMN, VANET*: CRC press, 2016.
14. Draz, U., Ali, T., & Yasin, S. (2018, November). Cloud Based Watchman Inlets for Flood Recovery System Using Wireless Sensor and Actor Networks. In 2018 IEEE 21st International Multi-Topic Conference (INMIC) (pp. 1-6). IEEE.