

# EFFICIENT ACTOR RECOVERY PARADIGM FOR WIRELESS SENSOR AND ACTOR NETWORKS

Reem Khalid Mahjoub

Under the Supervision of Dr. Khaled Elleithy

DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

AND ENGINEERING

THE SCHOOL OF ENGINEERING

UNIVERSITY OF BRIDGEPORT

CONNECTICUT

December, 2017

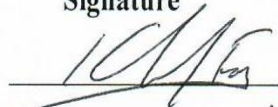
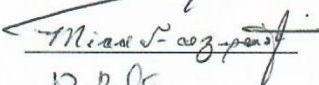
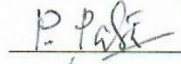
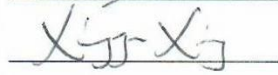

# EFFICIENT ACTOR RECOVERY PARADIGM FOR WIRELESS SENSOR AND ACTOR NETWORKS

Reem Mahjoub

Under the Supervision of Dr. Khaled Elleithy

## Approvals

### Committee Members

Name	Signature	Date
Dr. Khaled Elleithy		11/17/17
Dr. Miad Faezipour		11/17, 2017
Dr. Prabir Patra		11, 17, 2017
Dr. Xingguo Xiong		11/17/17
Dr. Eman Abdelfattah		11/17/ 2017

### Ph.D. Program Coordinator

Dr. Khaled M. Elleithy		11/17/17
------------------------	--	----------

### Chairman, Computer Science and Engineering Department

Dr. Ausif Mahmood		11/17/2017
-------------------	--	------------

### Dean, School of Engineering

Dr. Tarek M. Sobh		11/30/2017
-------------------	--	------------

# EFFICIENT ACTOR RECOVERY PARADIGM FOR WIRELESS SENSOR AND ACTOR NETWORKS

© Copyright by Reem Khalid Mahjoub 2017

# EFFICIENT ACTOR RECOVERY PARADIGM FOR WIRELESS SENSOR AND ACTOR NETWORKS

## ABSTRACT

Wireless sensor networks (WSNs) are becoming widely used worldwide. Wireless Sensor and Actor Networks (WSANs) represent a special category of WSNs wherein actors and sensors collaborate to perform specific tasks. WSANs have become one of the most preeminent emerging type of WSNs. Sensors with nodes having limited power resources are responsible for sensing and transmitting events to actor nodes. Actors are high-performance nodes equipped with rich resources that have the ability to collect, process, transmit data and perform various actions. WSANs have a unique architecture that distinguishes them from WSNs. Due to the characteristics of WSANs, numerous challenges arise. Determining the importance of factors usually depends on the application requirements.

The actor nodes are the spine of WSANs that collaborate to perform the specific tasks in an unsubstantiated and uneven environment. Thus, there is a possibility of high failure rate in such unfriendly scenarios due to several factors such as power fatigue of devices, electronic circuit failure, software errors in nodes or physical impairment of the actor nodes and inter-actor connectivity problem. It is essential to keep inter-actor connectivity in order to insure network connectivity. Thus, it is extremely important to

discover the failure of a cut-vertex actor and network-disjoint in order to improve the Quality-of-Service (QoS). For network recovery process from actor node failure, optimal re-localization and coordination techniques should take place.

In this work, we propose an efficient actor recovery (EAR) paradigm to guarantee the contention-free traffic-forwarding capacity. The EAR paradigm consists of Node Monitoring and Critical Node Detection (NMCND) algorithm that monitors the activities of the nodes to determine the critical node. In addition, it replaces the critical node with backup node prior to complete node-failure which helps balances the network performance. The packet is handled using Network Integration and Message Forwarding (NIMF) algorithm that determines the source of forwarding the packets (Either from actor or sensor). This decision-making capability of the algorithm controls the packet forwarding rate to maintain the network for longer time. Furthermore, for handling the proper routing strategy, Priority-Based Routing for Node Failure Avoidance (PRNFA) algorithm is deployed to decide the priority of the packets to be forwarded based on the significance of information available in the packet. To validate the effectiveness of the proposed EAR paradigm, we compare the performance of our proposed work with state-of the art localization algorithms. Our experimental results show superior performance in regards to network life, residual energy, reliability, sensor/ actor recovery time and data recovery.

## DEDICATION

*To my beloved family:*

*my father (Khalid Mahjoub),*

*my mother (Sabah Mahjoub),*

*my brother and sisters*

*Thank you for your endless love and support.*

*I love you!*

## **ACKNOWLEDGEMENTS**

In the name of Allah most gracious most merciful. My thanks are wholly devoted to Allah who has helped me all the way to complete this work successfully.

My most sincere appreciation goes to my beloved parents Khalid Mahjoub and Sabah Mahjoub for their understanding, prayers, support, and encouragement. Words cannot express how grateful I am for your support; your faith in me, enthusiasm, assistance, prayers and encouragement sustained me throughout my educational journey.

I would like to express my appreciation to my lovely brother, sisters and mentor who I wouldn't be able to proceed without their endless love, encouragement, prayers, and support. Also, special dedication to the soul and memory of my grandparents.

I am honored that my work was supervised by Dr. Khaled Elleithy. His advice, thoughtful comments, guidance, and support guided me throughout my dissertation.

Also, I would like to thank the dissertation committee members for their valuable comments and suggestions. I would also like to extend my appreciation to Dr. Maid Faezipour for her assistance and insightful comments about my work.

Furthermore, I would like to acknowledge and dedicate a special thanks to Dr. Elif Kongar for her encouragement, help, and support.

Finally, many thanks the Saudi Arabian Ministry of Education represented in the Saudi Arabian Cultural Mission in the USA for offering me the PhD scholarship.

# TABLE OF CONTENTS

ABSTRACT .....	iv
DEDICATION .....	vi
ACKNOWLEDGEMENTS .....	vii
TABLE OF CONTENTS .....	viii
LIST OF TABLES .....	xi
CHAPTER 1: INRTODUCTION .....	1
1.1. Research Problem and Scope .....	3
1.2. Motivation Behind the Research .....	6
1.3. Potential Contributions of the Proposed Research.....	7
CHAPTER 2: LITERATURE SURVEY ON WSAN .....	10
2.1. WSAN Features and Design Factors .....	11
2.2. WSAN Architecture .....	12
2.3. WSAN Applications.....	15
2.3.1. Environmental Monitoring and Precision Agriculture.....	16
2.3.2. Industrial Monitoring .....	18
2.3.3. Medicine .....	20
2.3.4 Microclimate and Lightning Control in Buildings.....	21
2.3.5. Military and Battlefield Surveillance.....	22
2.3.6. WSANs in Internet of Things (IoT) and Cloud Computing.....	22
2.3.7. Summary of Applications in WSAN .....	23



2.4. WSAN Challenges .....	24
2.4.1. Data Dissemination in Delay-Tolerant WSANs .....	26
2.4.2. Coordination and Localization .....	27
2.4.3. Routing.....	31
2.4.4. Actor Failure and Fault Recovery .....	32
2.4.5. Quality of Service (QoS).....	32
2.5. Actor Failure in WSAN.....	33
2.5.1. Distributed Connectivity Recovery Algorithm (DARA)[99].....	40
2.5.2. Recovery Through Inward Motion (RIM) [124].....	41
2.5.3. Actor Critical Recovery (ACR).....	42
2.5.4. Nearest Non-critical Neighbor (NNN)[100].....	43
2.5.5. Detection and Connectivity Restoration (DCR)[146].....	44
2.5.6. Recovering from Node Failure (RNF) Based on the Least-Disruptive Topology Repair (LeDiR) [155] .....	46
2.5.7. Delay and Throughput Performance Improvement in Wireless Sensor and Actor Networks[153] .....	47
2.5.8. Distributed Prioritized Connectivity Restoration Algorithm (DPCRA) [21] .....	47
2.5.9. Recovery of Lost Connectivity in Wireless Sensor and Actor Networks using Static Sensors as Bridge Routers (ACRA) [20].....	48
2.5.10. Algorithms Analysis and Evaluation .....	49
<b>CHAPTER 3 EFFICIENT ACTOR RECOVERY PARADIGM FOR WIRELESS SENSOR AND ACTOR NETWORKS .....</b>	<b>54</b>
3.1. Mathematical Model .....	56
3.2. Optimized Deterministic Actor Recovery System Model .....	62
3.2.1. Node Monitoring and Critical Node Detection (NMCND) .....	67
3.2.2. Network Integration and Message Forwarding Process.....	71
3.2.3. Priority-Based Routing for Node Failure Avoidance Process.....	75

CHAPTER 4 TEST PLAN AND SIMULATION SETUP.....	76
4.1. Number of Alive Days .....	80
4.2. Residual Energy .....	83
4.3. Actor/Sensor Recovery Time .....	86
4.4. Data Recovery .....	89
4.5. Time Complexity .....	91
4.5. Reliability.....	94
4.6. Overall Performance of EAR .....	97
CHAPTER 5 CONCLUSION .....	101
REFERENCES .....	104

## **LIST OF TABLES**

Table 2.1	WSAN applications	24
Table 2.2	Comparison of localization algorithms	30
Table 2.3	Actor Failure and Recovery Algorithms Analysis	50
Table 2.4	Failure Detection\recovery Algorithms advantages and limitations	51
Table 4.1	ACR, RNF, DPCRA, ACRA Algorithms analysis	77
Table 4.2	Summarized simulation parameters for the proposed EAR	79
Table 4.3	Residual energy notations and descriptions	84
Table 4.4	Mathematical proof of time complexity for EAR	93
Table 4.5	Time Complexity of EAR, RNF, DPCRA, ACR, and ACRA using O Big operation	94
Table 4.6	Improvement of EAR in percentile as compared to competing approaches: RNF, DPCRA, ACR, and ACRA	99

## LIST OF FIGURES

Figure 1.1	Critical actor node in WSAN	4
Figure 2.1	WSAN Architecture	11
Figure 2.2	Semi-automated WSAN	13
Figure 2.3	Automated WSAN	14
Figure 2.4	WSAN Applications	16
Figure 2.5	Localization Algorithm Classification	28
Figure 2.6	Proactive Fault Detection Mechanism Schema	34
Figure 2.7	Reactive Fault Detection Mechanism Schema	35
Figure 2.8	Failure Detection	36
Figure 2.9	DARA Algorithm	41
Figure 2.10	ACR failure recovery procedure	43
Figure 2.11	NNN failure recovery procedure	44
Figure 2.12	DCR Algorithm failure recovery procedure	45
Figure 3.1	Node Types in EAR	62
Figure 3.2	Optimized deterministic proposed system model for traffic monitoring using EAR model	65
Figure 3.3	Efficient Actor Recovery system model implementation	71
Figure 3.4	Graph representation of NIM	72
Figure 4.1	Number of alive nodes after completion of 12 events with $1200 \times 1200$ m <sup>2</sup> network topology (Results obtained from Scenario-1)	81

Figure 4.2	Number of alive nodes after completion of 12 events with $1000 \times 1000 \text{ m}^2$	81
Figure 4.3	Number of alive nodes after completion of 12 events with $1400 \times 1400 \text{ m}^2$ network topology (Results obtained from Scenario-3)	82
Figure 4.4	The residual energy of EAR and other competing approaches based on 9 event-monitoring	85
Figure 4.5	The residual energy of EAR and other competing approaches: RNF, DPCRA, ACR, and ACRA based on 18 event-monitoring	85
Figure 4.6	The residual energy of EAR and other competing approaches: RNF, DPCRA, ACR, and ACRA based on 27 event-monitoring	86
Figure 4.7	Number of failure actors/Sensors and required actor recovery time for EAR, RNF, DPCRA, ACR, and ACRA approaches with $1200 \times 1200 \text{ m}^2$	88
Figure 4.8	Number of failure actors/Sensors and required actor recovery time for EAR, RNF, DPCRA, ACR, and ACRA approaches with $1400 \times 1400 \text{ m}^2$	88
Figure 4.9	Data lost vs. Data recovery during 10 events	90
Figure 4.10	Data lost vs. Data recovery during 20 events	90

Figure 4.11	Big-O Complexity Chart of EAR, RNF, DPCRA, ACR, ACRA	93
Figure 4.12	Reliability for EAR, RNF, DPCRA, ACR, and ACRA approaches with 600 X 600 m <sup>2</sup>	95
Figure 4.13	Reliability for EAR, RNF, DPCRA, ACR, and ACRA approaches with 1400 X 1400 m <sup>2</sup>	96
Figure 4.14	Reliability for EAR, RNF, DPCRA, ACR, and ACRA approaches in regards to the size of deployment area	96

# CHAPTER 1: INTRODUCTION

Wireless Sensor Actor Networks (WSANs) comprise of actor nodes with powerful resources and sensor nodes with limited computation, power, and communication capabilities. WSANs utilize the feedback methodology that has been considered the core component of control systems. Therefore, the introduction of WSANs has helped in promoting revolutionarily the current control systems. It is easy to consider also that the WSANs will become the cornerstone of various control systems, which in turn enables an exceptional level of distributed control. The adoption of WSANs in control systems has various benefits matched up to wired solutions or WSNs, which are dominant current. For example, WSANs enable more flexible maintenance and installation, total mobile operation, and control and monitoring of appliances in formerly difficult-to-access and dangerous environments [1]. Another critical element that activates the deployment of WSANs is their comparatively low costs.

Irrespective of various advantages, WSANs do not go without some challenges and barriers for control applications. WSANs are recognized as disreputably volatile and intrinsically unreliable. This is mainly true in an event of low-power communications and when there is node mobility. With such features, the quality of service (QoS) of WSANs may not often be certain [2]. The actual outcome is that control applications will negatively undergo packet loss and time-varying delay, both that greatly reduce the

control performance, or even bring about system instability. As a result, WSAWs have to be effectively designed prior to deploying to control applications.

Also, because of the following WSAW's unique features and its differences with WSNs, there are various challenges to ensure effective communication requirements among actors and sensors in WSAWs.

- Node heterogeneity: as aforementioned, WSAW constitute several sensors and actors, where sensors are cheaper and smaller devices with limited wireless communication, computation, and limited sensing abilities. Nonetheless, because acting mechanism is energy consuming and significantly complicated action compared to sensing mechanism, actors are more resource-rich with superior processing abilities, longer battery life, and stronger transmission powers [3].
- Real-time requirement: based on the application in WSAWs, there can be a necessity to timely react to sensor input. For example, in a fire event aforementioned, actions need to be commenced on the event location immediately. Additionally, to initiate appropriate actions, sensor data should still be applicable at the time of acting. As a result, real-time communication is essential in WSAWs because actions are conducted on the environment immediately once the sensing happens. This means that one key goal of WSAWs is to reduce the communication wait between sensing and actual acting [4].
- Deployment: there are hundreds or thousands of deployed sensor nodes in analyzing the phenomenon, but this massive deployment is not required in actor nodes because of the different physical interaction methods and coverage requirements of acting



activity. Therefore, there is much less actors than sensors in the WSN [4]. The trade-off between acting coverage and connectivity is essential and need to be taken into consideration during the deployment of actors.

- Coordination: whilst the key communication challenge is mostly between the sink and sensor in WSNs, in WSNs, the key communication challenge can be sensor-actor communication [5]. Additionally, in some cases, actor-actor communication can also be needed to attain the entire application goal. Therefore, to give efficient acting and sensing, a distributed local coordination phenomenon is essential for actors and sensors.
- Mobility: nodes (mainly actors) may be mobile in WSNs. For instance, robots adopted in the battlefield or in distributed robotics applications are often mobile. Thus, protocols designed for WSNs need to facilitate the mobility of nodes [6].

### **1.1. Research Problem and Scope**

The Sensors and actors in WSNs collaborate together to monitor and respond to the surrounding world. The WSNs can be applied to wide range of applications, like health, environmental monitoring, chemical attack detection, battlefield surveillance, space missions, intrusion detection etc. However, the WSNs are greatly affected due to environmental change, frequent change in event-detection, actor mobility and actor failure. The failure of an actor node can result in a partitioning of the WSN and may limit event detection and handling. Actors may fail due to hardware failure, node mobility, event handling, attacks, energy depletion, or communication link issues. Sensor

node failure may cause lost in the event detection of the assigned environment covered by the sensor. The probability of actor failure is less than that of sensor failure and can be controlled through the relocation of mobile nodes due to their powerful characteristics; however, actor failure can cause more damage than can sensor failure. Actor failure can cause a loss of coordination and connectivity between nodes, limitation in event handling, and can leads to a disjoint WSN.

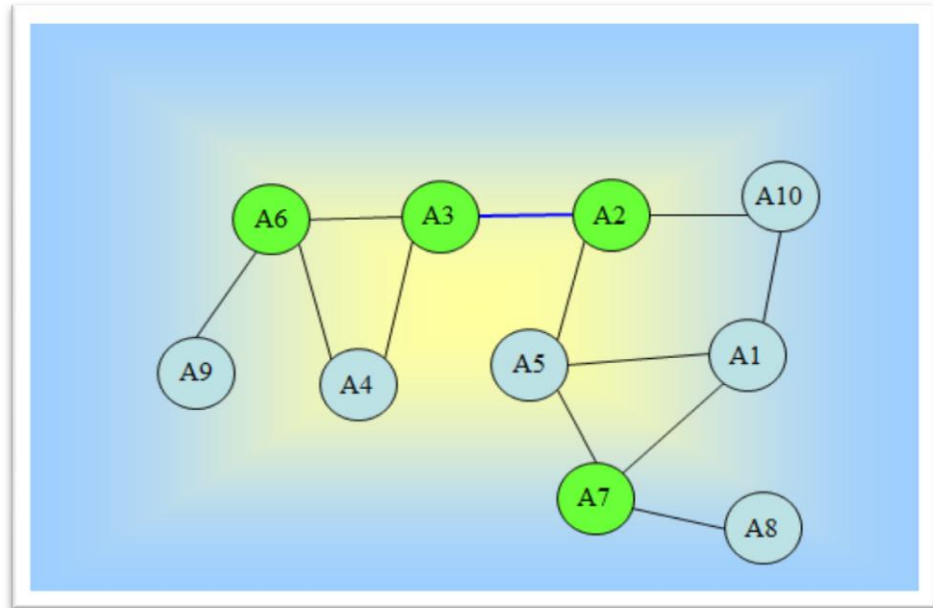


Figure 1.1. Critical actor node in WSN

The actor failure occurrence is very critical that degrades the network performance. In particular, the failure of critical actor may cause high impact for the whole network. Critical actor nodes refer to actors which their failure cause network partitioning. Figure 1.1 illustrates the concept of critical actor nodes. Assume that actor A3 failed. Its failure will cause the network to disjoint. Thus, A3 is a critical node. Actor nodes A2, A6, and A7 are critical nodes as well. Most of the existing approaches

attempted to replace the critical node with another backup node, but they failed to maintain the QoS parameters and energy consumption. Furthermore, due to the fact that WSNs are deployed in harsh area and requires long term monitoring/acting process, proposed methods should offer robust self-healing failure detection/ recovery techniques which ensures that network lifetime is maximized as much as possible while maintaining QoS.

Critical actors are key nodes in WSN. Their failure cause high impact to the WSN. Critical actor failure may cause network partitioning and high impact to the overall network. Thus, it is essential to implement a robust WSN which has the support of following features:

- It is essential that develop techniques which support the heterogeneity of WSN. The techniques should support self-monitoring, and self-healing.
- Failure detection and recovery mechanisms are important factors. Hence, failure detection and recovery should be efficient.
- In addition, not only essential to have a robust failure detection/recovery mechanism, it is essential to implement robust failure detection/recovery technique while maintain network quality of service.
- Moreover, WSN should be supported with mechanisms that reduce and prevent packet lost; as well as failure. Even if failure occurs, their impact should be minimized.

## 1.2. Motivation Behind the Research

WSANs are currently adopted in various civilian and industrial applications. WSANs have a unique architecture that distinguishes them from WSNs. Sensor nodes having limited power resources are responsible for sensing and transmitting events to actor nodes. Actors are high-performance nodes equipped with rich resources that have the ability to collect, process, and transmit data and perform various actions. The adoption of WSANs in control systems has various benefits matched up to wired solutions or WSNs. Irrespective of various advantages, WSANs do not go without some challenges and barriers for control applications. Therefore, studies should focus more on improving and overcoming barriers of WSANs for monitoring and surveillance purposes. WSAN consist of sensor, actor nodes, and base station. Since actor nodes are the main characteristic in WSAN that overcome the limitation of WSN, special attention should be giving to those nodes. The efficient usage and managing of actor node can improve/degrade the overall network performance.

Thus, the actor node is a major component of WSAN. The failure of an actor node can degrade network performance. Furthermore, the failure of an actor node may result in a partitioning of the WSAN and may limit event detection and handling [7].

In WSAN, the actor nodes involve high features that increase the power capability and network usage. Thus, maintaining the inter-actor connectivity is indispensable in WSANs. A failure of an actor may cause loss of communication or a network disconnect. Thus, actors must communicate with each other to guarantee the entire network coverage and to harmonize their actions for best response. Therefore, to give efficient acting, actor

nodes featured should be efficiently utilized, and robust. In addition, WSAN should provide proficient network monitoring, failure detection, and handling capabilities. Accordingly, a new method is proposed for efficient actor recovery paradigm (EAR) which guarantee the contention-free traffic-forwarding capacity. Unlike previous studies, EAR craves for providing efficient failure detection and recovery mechanism while harvesting maintains the QoS.

### **1.3. Potential Contributions of the Proposed Research**

The main contributions of this work is to build a framework that can be used in Wireless Sensor and Actor Network that can act robustly and for various scenarios in different environments.

The proposed model should address the following requirements:

- Fault Tolerance
- Actor node Recovery
- Energy Efficiency
- Reliability
- Network Quality of Service (QoS)

In WSAN, most of the node localization and routing approaches depend on the hop count information rather than table-based routing protocols. With these motivation, this study introduces a mathematical model for determining actor forwarding capacity in WSAN using RSSI message information. The model aims at guaranteeing contention-free forwarding capacity. The EAR contributes to the literature by providing the best

RSSI value to improve the traffic forwarding process. The state-of-the-art research is to provide the best node failure recovery process which handle to manage network resources in order to extend the network lifetime.

Most existing actor failure detection/ recovery mechanisms manage to handle the failure detecting and recovery but failed to manage the overhead from such event. These mechanisms may degrade the overall quality of service. However, to overcome those limitations, EAR contributes three novel algorithms that provide effective critical actor failure detection/ recovery technique while maintaining QoS. Moreover, EAR helps to minimize critical actor failure occurrence by utilizing critical actor nodes resources.

The proposed paradigm contributes the RSSI mathematical model in addition to three novel algorithms as follows:

- Node Monitoring and Critical Node Detection (NMCND) algorithm that monitors the activities of the nodes to determine the nodes types and distinguish critical nodes. The NMCND algorithm checks the entire network to determine the critical node during network life time and pre-assign a backup node for each critical node; so incase the failure of critical node, this node takes place in order to improve and balances the network performance.
- Optimized RSSI model is introduced that selects the different power strengths for each beacon in order to ensure the proper delivery of the beacon to each node. This aims to reduce the latency and estimating the prediction of the node energy-level. As a result, QoS provisioning is maintained and extended the network lifetime.

- Network Integration and Message Forwarding (NIMF) improves the QoS by handling the packet forwarding process. NIMF works to reduce packet forwarding through critical nodes and enhances network lifetime. Moreover, NIMF has the capability to decide the source of the forwarding packet which enhances the packet forwarding flow. Thus, accurate packet forwarding process reduces the latency and bandwidth consumption.
- Priority-Based Routing for Node Failure Avoidance algorithm (PRNFA) handles the routing process. PRNFA analyzes and evaluates the information of the packet in order to route it to the next node. It determines the priority of the forwarded packets. In addition, PRNFA eliminates redundant data prior to routing the packets.

## **CHAPTER 2: LITERATURE SURVEY ON WSAN**

Wireless sensor networks (WSNs) are becoming widely used worldwide. Wireless Sensor and Actor Networks (WSANs) is a field where actors and sensors are aggregated together in order to perform certain tasks, or transmit and process information. A WSAN consists of sensors, actors, and a base station. The actor network is integrated with the sensor network to implement the Wireless Sensor and Actor Network[4]. Sensors are responsible for sensing and transmitting events to the actor nodes. Actors are high performance nodes that have the ability to collect, process, transmit data, and perform actions. Actors run on a high power source [8, 9]. In WSAN, an actor node can communicate with several sensors. . In this network, actor node's lifetime and coverage is higher than the sensor node. This type of network is brought up in various applications such as detection of forest fire, mobile robots which are used to monitor the battlefield, etc.. Communication in the WSAN can be classified as sensor-actor communication, actor-actor communication, or actor-sink communication. In WSAN, due to the architecture, the number of sensors can range up to thousands while the number of actor nodes is much lower [10]. WSAN can assist many applications such as smart energy grids, battlefield surveillance, and cloud computing, as well as, can be used in medical, industrial, and nuclear fields. Several parameters may affect the WSAN, including energy efficiency, transmission media, scalability, and environment.



Figure 2.1 illustrates the architecture of WSAN with both sensor and actor nodes in the network. Broadly, this architecture is classified into two types such as semi-automated and automated architecture [11], [12], [13], [14]. The architecture is classified based on its data passing and decision making. Furthermore, in WSAN sensor nodes are static and the actor nodes are either static or dynamic based on the application.

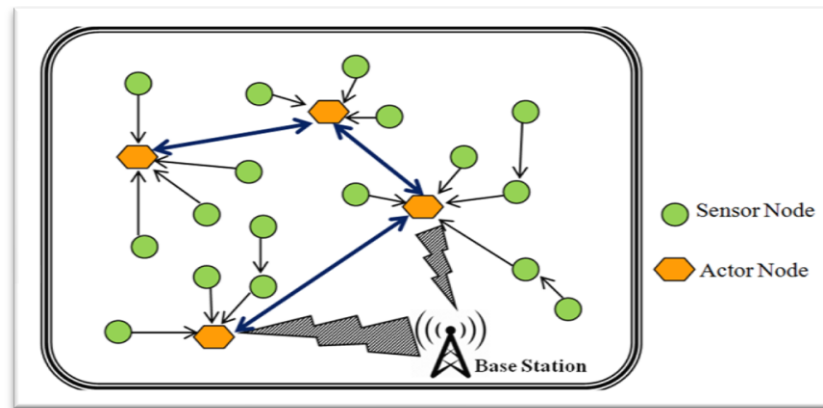


Figure 2.1. WSAN Architecture

The WSAN has many advantages over the regular WSN. One of the most notable differences is the high energy and low power consumption. The actor nodes may include high performance features which can increase the power and usage of the network in general. Maintaining the inter-actor connectivity is essential in WSAN.

## 2.1. WSAN Features and Design Factors

There is much difference communication features between WSNs and WASNs. Although, both handle the communication occurrence between the sink and sensors or sensor-sensor communication. whereas in WSANs, additional advanced networking communication link termed actor-actor or sink-actor communications can take place [1].

Moreover, event features are transmitted by the sensor-actor communication from sensor to actors. Communication between the actors occurs once they receive the event information with the aim of conducting the relevant action across the event area.

As a result of the use of both actors and sensors, WSANs utilize the feedback methodology that has been considered the core component of control systems. Therefore, the introduction of WSANs has helped in promoting revolutionarily the current control systems. It is easy to consider also that the WSANs will become the cornerstone of various control systems, which in turn enable an exceptional level of distributed control. The adoption of WSANs in control systems has various benefits matched up to wired solutions or WSNs, which are dominant current.

A major aspect of designing an efficient WSAN is to utilize and optimize the following network features and Factors. Factors includes: scalability, energy efficiency, reliability, transmission media, fault tolerance, environment, node Cost, and hardware Constrains. Most of factor optimization should be met in respond to application requirement.

## **2.2. WSAN Architecture**

The main components of WSANs are sensor nodes and actor nodes. The architecture of the WSAN plays a critical role in the network in general because this may affect different components such as the behavior of the sensing node and the energy usage. A WSAN contains sensors, actors, and a base node. In WSANs, sensors sense events, and actors receive information from sensors and communicate with each other to

facilitate decision making. A sensor node consists of a sensing unit, a power source, storage, a processing unit, a communication transceiver, an analog-to-digital converter (ADC), and an antenna. The actor node contains a more substantial power source and more capable processing and storage units. The sink node supervises the overall network.

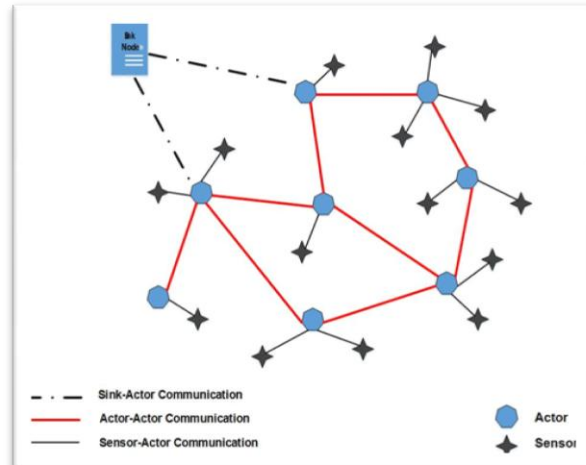


Figure 2.2. Semi-automated WSN

WSAN architectures can be classified into semi-automated and automated WSNs [4, 15-18]. Figure 2.2 illustrates a semi-automated WSN while Figure 2.3 illustrates an automated WSN. In automated WSNs, sensor nodes sense events and send sensed data to actor nodes. Actor nodes act as the base station in an automated WSN. Meanwhile, actor nodes receive, process, and communicate with other actors if needed and subsequently perform an action. On the other hand, in semi-automated WSNs, sensor nodes sense data and transfer those data to the sink node while the sink node processes the data and communicates with the actors to perform the needed task. Then, the sink node sends commands to the assigned actor [19]. Coordination

mechanisms are essential in WSANs. Coordination can occur within sensor-actor communication, sensor-sensor communication, and actor-actor communication. Sensor-actor and sensor-sensor communication can be affected by the limitations of sensor nodes. Thus, it is better to rely more on actor-actor communication to maximize the network efficiency. In terms of sensor-actor communication, WSANs can be Multi-Actor (MA) or Single-Actor (SA) WSANs. More than one actor can receive a sensed event from a particular sensor node in MA WSANs, whereas in SA WSANs, the sensor node can only send to a specific actor node.

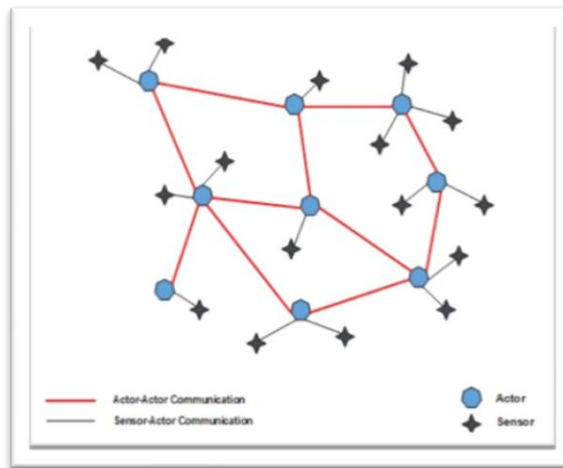


Figure 2.3. Automated WSAN

In WSAN, sensor and actor node may be mobile. Moreover, nodes might experience failures. Thus, network topology can be changeable during the life of the network. Thus, topology management is essential. Topology management is referred to maintaining the organization and connectivity of nodes network while maintaining coverage and managing power resources. Topology management techniques in WSAN can be conducted using sleep cycle management, movement control, power control, node

discovery, node repositioning, and clustering. Some researchers focused on implementing using one feature while others combined more than one [20] , [21], [22], [23], [24-27], [28], [29], [30], [5]. Topology management can play important role in fault management. Nodes failure can cause network partitioning or missing sensing areas. Thus, topology management can be executed automatically for recovery purposes. We'll explore more about network topology management throughout while discussing actor failure and fault management.

### **2.3. WSAN Applications**

The main purpose of WSANs is to address a particular application. The development of WSANs was stimulated by military systems such as battlefield surveillance. . Applications can be as simple as environmental monitoring, event detection, information gathering, or measurement. WSANs are widely used in many applications, including industrial monitoring, medical applications, smart energy grids, climate control, nuclear and biological applications, military applications, and attack prevention as shown in Figure 2.4.

A significant number of parameters, such as energy efficiency, network topology, transmission media, scalability, environment, bandwidth, sensing frequency, and device deployments, may affect the WSANs. The selection of the parameters for optimization, such as transmission media, depends on the application [31]. The following subsections briefly discuss these applications.



Figure 2.4. WSA Applications

### 2.3.1. Environmental Monitoring and Precision Agriculture

These advanced systems of sensing and acting have shown much potential in various fields, such as environmental monitoring and agriculture. Environmental monitoring is particularly dependent on accurate position estimation with the aim of evaluating or processing the data gathered. Environmental monitoring applications are very important in enhancing and monitoring the lives of various animals and humans. Monitoring practices using WSA applications include security control, surveillance applications, and natural disaster detection. WSANs can also be used in specific areas to help in limiting the effects or entirely preventing the impact of some factor. WSANs are also critical in environmental monitoring because of the capabilities and sizes of the

sensors. Moreover, some environmental applications involve determining real-time performance, which may be achieved by using actors that can conduct certain actions and interact with the environment. These sensors are also important and useful in areas that are inaccessible to humans.

Greenhouses that utilize WSANs for climate parameter control and crop monitoring offer an efficient solution for preventing crop diseases and improving crop growth [32]. Crops and greenhouses require monitoring because they are affected by climate and environmental changes.

Irrigation and nutrition systems can perform crop parameter measurements to ensure crop health. WSANs can be employed to improve the monitoring of a crop's environment in response to climate changes or diseases. In agriculture, actor nodes are used to process sensed data and to initiate actions using attached devices or sensor nodes that enhance the control of CO<sub>2</sub>, ventilation, heating sprinklers, humidity, and lighting. Sensor nodes can also sense environmental parameters, including crop diseases, water level, humidity, and temperature [33]. These parameters are important factors in the selection of sensor types and transmission media.

For an application combining artificial intelligence and the current WSAN, Sabri et al. [34] recommended enhancing the performance of greenhouse climate control. Moreover, Sabri et al. [34], [35] recommended merging the Jennic5148 wireless network controller with the fusion of artificial intelligence (FIS) to create a new cognitive wireless sensor-actor network that supports environment sensing and responds with the finest performance without interference from a gardener. In summary, WSANs have been

shown to be potentially helpful in ensuring that specific variables, such as luminosity, humidity, and temperature, are controlled. This paper also proposes the merger between WSNs and artificial intelligence to improve accuracy and reliability.

The current research aims to extract the greatest benefit of WSNs in the environmental field by adding the Internet of Things (IoT) to the application layer of the designed WSN. By combining the features of WSNs and IoT, [36] designed a harvesting WSN for sensing temperature and controlling a radiator using the DASH7 protocol.

The WSN solution for greenhouses requires a full understanding of the greenhouse environmental variables and specifying the required application optimization parameters [37]. Sensors must support high humidity and temperatures, especially in greenhouses. A huge challenge in deploying WSNs is ensuring their resistance to changes in weather and the topography of crops. The topography may affect the signal range and network reliability [38, 39].

### **2.3.2. Industrial Monitoring**

In industrial sector, frequent monitoring of processing and production environments are very vital since it may be very challenging and nearly impossible for employees to work and interact in certain locations in industrial fields.

In the industrial sector, frequent monitoring of processing and production environments is vital, as it may be very challenging and nearly impossible for employees to work and interact in certain locations in industrial fields. This may include difficult-to-



transverse, high-temperature, and critical geographical locations. For instance, the extraction processes for oil, minerals, and gas require a massive amount of energy, which makes it difficult for employees to work in the surrounding environment. In addition, the extraction processes necessitate essential monitoring and actions. WSNs are compatible with such types of environments because sensor nodes can detect and gather events and transfer these data to actor nodes, which can act immediately without input from industrial workers. The application of WSNs to industrial settings enhances productivity and safety.

The authors in [40] suggested applying an information-centric networking approach to WSNs to improve bandwidth. The scheme used content-centric networking (CCN). The architecture meshes routers using the CCN protocol. The sensors also use the CCN protocol but do not cache sensed data. Sensors are connected to IEEE 802.15.4 routers. Two types of messages are used for communication in this architecture: interest messages and data packets. Interest messages are broadcast to the backbone network. The technique uses the naming scheme in [41], which contains two sections: 1) a category prefix for specifying the sensed data type and 2) the information identity, including the broadcast value. Therefore, when broadcasting an interest message, the response is based on the category prefix. This technique is believed to improve energy efficiency because only related sensors are involved in the communication. A data packet includes the content name, signature, signed information, and data. If security is not critical to the application, the signature field may be omitted to decrease the message overhead. When a control device requests sensing from the network, an interest message is routed to the

sensors. The corresponding sensor will reply with the sensing information to the control device. This information is cached in the routers, which act as actors in this scheme. Therefore, if another control device requests the same information, data will be forwarded from the router cache instead of via another interaction with the sensor node. Thus, network caching through the application of a CNN decreases the memory usage and sensing overhead in a WSN. Nevertheless, this method is not applicable to industrial fields, which require real-time sensing capabilities.

Potsch [42] proposed the adoption of a test-bed in an industrial WSN. This proposed system has an embedded host node, operating-data acquisition, real-time Ethernet system, and WSN. This system helps in monitoring the timing and energy restraints of nodes.

This subsection shows that it may be impossible to access some areas of an industrial company. Therefore, an advanced system such as a WSN is required because it is compatible with such lack of access and, in turn, can be used to monitor relevant data to help in controlling and decision-making. The use of a test-bed is also proposed to improve various elements such as timing.

### **2.3.3. Medicine**

The advancement in various communication technologies have led to the development of small size and low cost sensors that are capable of being even injected into human body. Such advancements have placed massive impact on medical healthcare and have become an alternative for traditional procedures and equipment with small-

sized applications. Various health monitoring and patient management systems are developed and applied daily within health care facilities [43]. Even though such advanced systems are potent systems, they also face some drawbacks and limitations. Since most of these systems are installed locally within the healthcare facilities, they inflict increased costs for patients. Therefore, this imposes a significant healthcare costs in that it requires more human resources. To resolve this problem, there is a necessity for healthcare applications to improve patient's quality of care, ensure effective services, and reduce costs.

Health monitoring applications by adopting WSANs improves the current patient and healthcare monitoring. Application such as infant and adult monitoring, fire-fighter vital sign, blood pressure monitoring, and alerting the deaf has been improved through WSAN application and implementation.

### **2.3.4 Microclimate and Lightning Control in Buildings.**

Some buildings have deployed sensing techniques with the aim of monitoring different parameters. For example, building Automated Systems (BASs) were designed to provide comfort, functionality, and control of indoor environments. These building control and monitoring systems need an action to be executed after an event occurs. Therefore, WSANs can be deployed to adopt optimal acting/sensing processes for controlled buildings. Moreover, sensor/actor nodes are critical in building security. WSANs can also be deployed to detect and manage fire within the buildings. In an event of fire, sensor nodes can transmit sensed events to the relevant sprinkler actor to limit fire

possibility of spreading and support the management of a fire. For instance, Li, [44] modeled a light control and monitoring systems, serving as a WSNs' case study. Thus, installing BASs using WSN applications enable resource optimization, security, and energy efficiency.

### **2.3.5. Military and Battlefield Surveillance**

Critical monitoring and acting systems are very vital for battlefield surveillance, where real-time tracking and monitoring are also important components. Due to its small size, sensors can be implanted into various appliances. Cars, humans, and animals can carry these sensors. WSNs can be deployed also for positioning applications to provide tracking and monitoring purposes for effective battlefield management [45]. An effective sensing can give the actor nodes a set of information for decision making and may also boost the speed at which certain relayed actions are executed.

It can be concluded that effectively deploy a WSN in a battlefield, reliable and accurate event positioning and detection schemes should be adopted.

### **2.3.6. WSNs in Internet of Things (IoT) and Cloud Computing**

Cloud computing technology is among the most modern, advanced internet-based technology system. Cloud computing is a platform that offers the end-users with the capability of virtually accessing resources, services, and storage over the online platform or Internet [46].

Rapid growth of IoT and cloud computing inspires the adoption of System of Systems. Most applications use WSANs as standalone system, but integrating cloud computing with WSANs can optimize the benefits of a network [47]. Sensor nodes are more beneficial due to their low-cost and can also be assigned IP addresses that enable them to be compatible with Internet-enabled devices and cloud computing. Moreover, actor nodes have network components that enable them to be integrated with cloud computing applications. In cloud computing, sensor nodes can enable sensing capabilities and transfer information to a cloud, which enable cloud to process and share the sensed information with a range of applications and end-users.

### **2.3.7. Summary of Applications in WSAN**

The development of WSANs was stimulated by military systems such as battlefield surveillance systems. Currently, WSANs are adopted in various civilian and industrial applications, such as information gathering, event detection, environmental monitoring, home automation, and chemical and biological attack detection [48]. The adoption of WSANs in control systems has various benefits similar to those of wired solutions or WSNs. Irrespective of the various advantages of WSANs, their use in control applications faces some challenges and barriers. Therefore, further studies should focus more on improving WSANs and overcoming barriers to the practical implementation of WSANs in various applications. Table 2.1 provides valuable references to researchers with respect to WSAN application usage, deployment and challenges.

Table 2.1. WSAN applications

<b>Application</b>	<b>Related research</b>
Environmental Monitoring	[38], [49], [37], [39], [36], [32], [50], [51], [52]
Industrial Monitoring	[53], [40], [41], [54]
Medicine	[55], [56], [43], [57], [58], [59], [59], [60], [61], [62], [63], [64]
Smart Energy Grid	[65-69], [70], [71]
Microclimate And Building Lightning	[72]
Military And Battlefield	[73], [74], [75], [76], [77]
Cloud Computing And IoT	[46, 78], [79], [6], [80], [81], [36], [82], [83], [84], [85], [86]

## 2.4. WSAN Challenges

Actor and sensor nodes must cooperate to determine the optimal route between them. Due to the unique layout of a WSAN, the sensor nodes, sensor-actor setup, network communication, coordination, timing and synchronization, and routing protocols may present challenges. For example, wireless channels have several features, such as half-duplex operations, Doppler shifts, adjacent channel interference, multipath fading, and path loss. WSANs are recognized as disreputably volatile and intrinsically unreliable. This is mainly true in the event of low-power communications and when there is node mobility. With such features, the QoS of WSANs may often be uncertain [31]. As a result, control applications will undergo packet loss and time-varying delay, both of which greatly reduce the control performance or cause system instability. As a result, WSANs must be effectively designed prior to being used for control applications.

Challenges can be classified as either sensor-actor challenges or actor-actor challenges. Sensor-actor challenges include event synchronization between actor and sensor nodes, the case in which only targeted actors are able to communicate with a specified sensor, the case in which sensor dependency is limited to lower a sensor's power consumption, the determination of whether semi-automated or automated features are more efficient given the application specification, the implementation of routing protocols that can manage an actor's selection of a specific sensor node, and the deployment of a protocol that can manage sensor-actor communication while improving the network performance and throughput.

In contrast, Actor-actor challenges include:

- Understanding how to define communication between actors.
- Determining if it is more suitable to use semi-automated or automated architectures.
- Selecting the appropriate actor that covers a specific field of sensors under SA architectures.
- Defining an algorithm for dividing actors among sensors, determining when the use of clusters improves the performance of a WSN.
- Applying actor-actor real-time synchronization to perform particular actions simultaneously.
- Calculating the minimum number of actors that must be connected to a sensor node in an automated architecture

- Creating an overall protocol model that can manage actor communication, and defining the protocol messages.

As a part of the WSN topology, actors need to ensure inter-actor communication to maintain and operate the network and for actor coverage and coordination. These factors are critical elements that can affect the network. Thus, these factors are important components of WSN challenges. Other factors include data dissemination in delay-tolerant WSNs, coordination and localization, routing, actor failure, real-time communication, mobility, energy efficiency, security, and QoS. In this section, we will discuss the most challenging factors.

#### **2.4.1. Data Dissemination in Delay-Tolerant WSNs**

WSN deployed in large geographical area. During the WSN life time, network can suffer from major changes due its topology change, node power exhaustion, or node mobility. In addition, in various types of WSNs, actors cannot communicate with each other. Therefore, actor-to-actor communication can be performed through sensor nodes. Moreover, sensors nodes sleep scheduling mechanism; some sensors may not loss the received data during its sleep time. Thus, nodes may cause data dissemination. Data dissemination is the process of transmitting, distrusting data to nodes. In WSN, data collection is done by sensors nodes. The data collection can be performed by single sensor node, within particular region, or collaboration between neighbors nodes. Various studies have attempted to address data dissemination [87], [88], [89], [90], [91], [92]. Overlay techniques have been used to overcome the data dissemination issue. Since WSN involves numerous data transmission between sensor and actor nodes, methods of



combining or aggregating into small datasets of is essential. In addition, energy saving methods should be conducted. Data dissemination is a challenging issue in WSN due to its dynamic environmental change and its huge number of nodes [93].

#### **2.4.2. Coordination and Localization**

Localization and coordination are important factors for measuring the effectiveness of a WSN. Location discovery is critical for event detection and routing. Moreover, some applications require location information along with the collected event [94]. Localization can be performed when determining the physical geographic location of the sensors and actors. Localization challenges might fall under the architecture of the WSN; such challenges may also concern computational factors that might be affected by path calculation and optimization and the respective power consumption of these processes. Other localization challenges include shadowing, multipath, sensor imperfections, failure of sensor/actor nodes, and computational challenges.

To design a robust WSN, one should build optimized localization and coordination techniques while considering various SA, SS, and AA coordination approaches. Other factors, such as the application requirements [95], autonomous operations, network and system HW, node mobility, equipment capabilities, and scalability, must be considered. Application requirements can play an important role in designing coordination protocols because different applications may focus on specific parameters that are considered as critical in that specific application.

The WSAN architecture plays an important role in applying the localization and coordination technique. Latency and power consumption can be decreased as a result of the direct communication between actors and sensors in automated WSANs but can be increased in Semi-automated WSANs. WSAN applications vary between having predefined locations for their nodes [96] and mobile nodes that require accurate location discovery.

Localization algorithms are classified into centralized and distributed algorithms, as shown in Figure 2.5 In centralized approaches, an assigned node, namely , a base node, is responsible for managing all the network node states and topologies and for applying recovery and relocation schemas when required [97]. Using a centralized approach may affect the network performance because nodes near the base node may experience increased power consumption due to the nature of this approach.

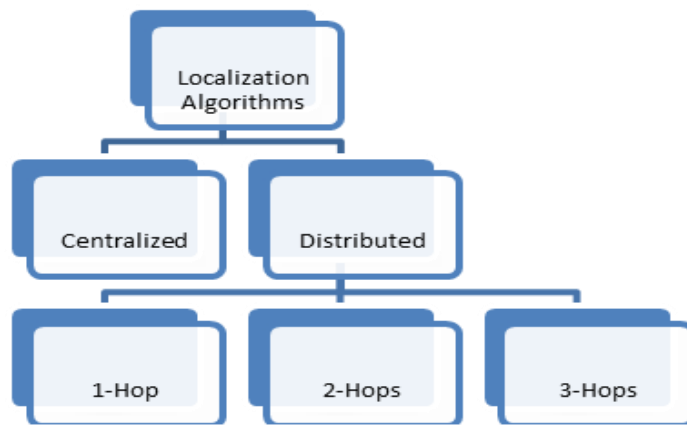


Figure 2.5. Localization Algorithm Classification

In distributed algorithms, nodes maintain limited network states by exchanging "hello" messages with reachable neighbors. Thus, each node must calculate its position. This approach has an advantage over centralized approaches in that it results in an evenly distributed power consumption between nodes and requires less computationally expensive position calculations when the topology changes. Distributed algorithms can be 1-hop, as in [98]; 2-hop, as in [99] and [100]; or 3-hop algorithms.

Table 2.2 summarizes the localization features of existing WSN localization algorithm and covers the architecture, mobility, coordination services, coordination decision, optimization parameter, localization algorithm type and routing.

Table 2.2. Comparison of localization algorithms												
		[101]	[102]	[103]	[104]	[105]	[106]	[99]	[107]	[108]	[100]	[109]
Architecture	sa	√	×	×	×	×	×	√		×		√
	at	√	√	√	√	√	√	√	√	√	√	√
Mobility	Sensor	fixed	ns	fixed	fixed	fixed	fixed	ns	ns	ns		mobile
	Actor	mobile	fixed	fixed	mobile	mobile	fixed	mobile	mobile	mobile	mobile	fixed
	Sink	fixed	na	na	na	na	na	fixed	×	×	ns	fixed
Coordination level	SA	√	√	√	√	√	√	√	√	√	ns	√
	SS	√	na	na	√	√	ns	√		×	ns	√
	AA	na	√	√	√	√	√	√	√	√	√	√
Services	Location	√		√	√	√	√	√	√	√	√	√
	Sync	na		√	ns	√	ns	ns	×	√	ns	ns
	Fusion	na		√	√	√	ns	ns	ns	ns	ns	ns
Coordination decision	CB	√	√	√	√	√	√	√	√	√	√	√
	CF	ns	na	√	√	√	√	√	ns	√	√	√
Localization algorithm	Centralized	√	×	×	×	√	×	√	√	×	√	√
	Distributed	√	√	√	√	√	√	√	√	√	√	√
na: not available, ns: not specified, consensus-based (cb), consensus-free (cf)												

### **2.4.3. Routing**

In sensors networks, routing techniques might differ in respond to the networks architecture and/or application requirements [110]. In regular WSNs, sensors and base stations are typically static, whereas in WSANs, sensors and actors are mobile. Thus, WSAN routing techniques should address sensor-actor communication and delay [95], and routing protocols must ensure reliable communication between nodes. WSANs can be distinguished from WSNs by their real-time communication and coordination [111].

Because semi-automated WSANs can obtain various benefits by using the same routing protocols designed for WSNs, they almost have the same layout as WSNs, where sensors send sensed events to a base station [112]. This common characteristic can be advantageous for the network when using or modifying existing WSN protocols [113]. Additionally, the transport layer protocols that are used for WSANs suffer from limitations [111].

In conclusion, the most challenging point for WSANs is creating a protocol that can manage actor-actor and sensor-actor communication, coordination, and optimization for all the above-mentioned challenges while improving the overall performance. Moreover, WSAN protocols should support real-time applications [114]. For example, B. P. Gerkey and M.J. Mataric addressed routing actor-actor coordination but did not focus on sensor-actor coordination [115].

#### **2.4.4. Actor Failure and Fault Recovery**

The failure of an actor node can result in a partitioning of the WSN and may limit event detection and handling. Actors may fail due to hardware failure, attacks, energy depletion, or communication link issues. Sensor node failure may cause lost event detection of the assigned environment covered by the sensor. The probability of actor failure is less than that of sensor failure and can be controlled through the relocation of mobile nodes due to their powerful characteristics; however, actor failure can cause more damage than can sensor failure. Actor failure can cause a loss of coordination and connectivity between nodes, leading to a disjoint WSN. Detailed information on actor failure is represented in section 2.5.

#### **2.4.5. Quality of Service (QoS)**

WSNs are used in critical applications. A network's QoS depends on the application requirements. QoS can be affected by reliability, node failure, robustness, energy efficiency, and security and should satisfy application requirements [116].

A WSN's resource constraints may affect its QoS. Sensor nodes are equipped with limited power and memory resources. This can cause the network to suffer from node unavailability, packet drops, or low signal ranges, thereby influencing the network QoS. To achieve an improved QoS in the WSN, communication protocols should support a balanced sensor/actor QoS.

Due to the mobility feature of WSANs, the network topology dynamically changes over time, therein causing nodes to enter or leave the network; thus, relocation techniques are used. This feature may affect the power consumption of nodes, thereby affecting the network QoS. Kim and Jorge addressed the issue of optimization based on minimizing actors' movements and sensors' radii and produced an optimal ILP solution [117]. To achieve improved QoS in a WSAN, communication protocols should support a balanced sensor/actor QoS.

## **2.5. Actor Failure in WSAN**

As previously mentioned, the failure of an actor node can result in a partitioning of the WSAN and may limit event detection and handling. Actors may fail due to hardware failure, attacks, energy depletion, or communication link issues. Sensor node failure may cause lost event detection of the assigned environment covered by the sensor. The probability of actor failure is less than that of sensor failure and can be controlled through the relocation of mobile nodes due to their powerful characteristics; however, actor failure can cause more damage than can sensor failure. Actor failure can cause a loss of coordination and connectivity between nodes, leading to a disjoint WSAN.

Actor failure can be due to their limited power, mobility, or topology change. The mobility feature can cause actors to become outside the communication range. Moreover, network topology may be affected by such behavior. Effective topology management techniques should be implemented. Purposed mechanisms were introduced in order to manage network failure in concern with topology management [5, 20-25, 27-30].

Fault tolerance is the ability of a network to preserve its services regardless of the occurrence of faults. In other words, fault tolerance is the ability of the network to stabilize in response to node failure [118].

Purposed fault tolerant techniques can manage one or more types of fault at one or more network layer. The general taxonomy of fault tolerance techniques used in distributed system. Fault prevention; fault detection; fault isolation; fault identification; and fault recovery. Some fault tolerance techniques focused on one fault tolerance portion while others used a combinations of portions. Moreover, Fault detection mechanisms are classified to proactive, reactive methods, or hybrid.

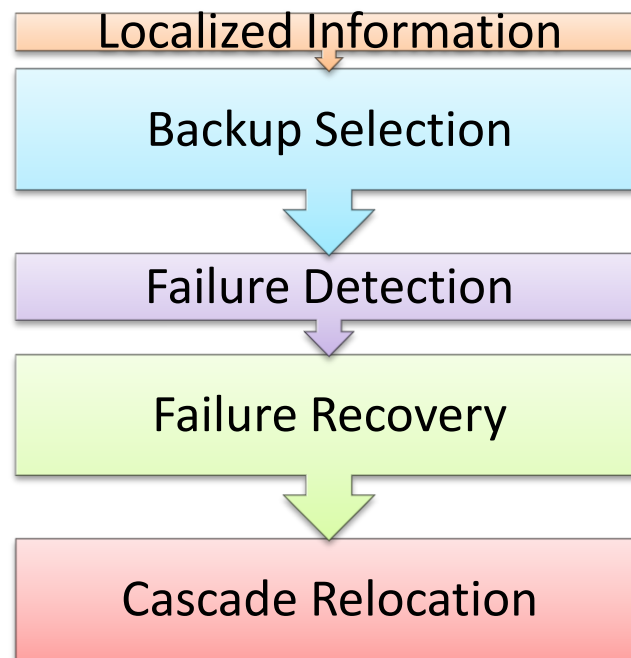


Figure 2.6. Proactive Fault Detection Mechanism Schema



In proactive methods, fault and restoration mechanisms addressed during the network setup, Figure 2.6. Various mechanisms implement a fault tolerance topologies in the network setup while other use redundant and backup nodes to insure fault tolerance [119].

On the other hand, reactive scheme pursue to utilize network resources and perform recovery dynamically through node repositioning, Figure 2.7. Reactive schemes require network monitoring in order to maintain nodes status. Network status, recovery algorithm, and recovery scope are important factors in reactive schemes.

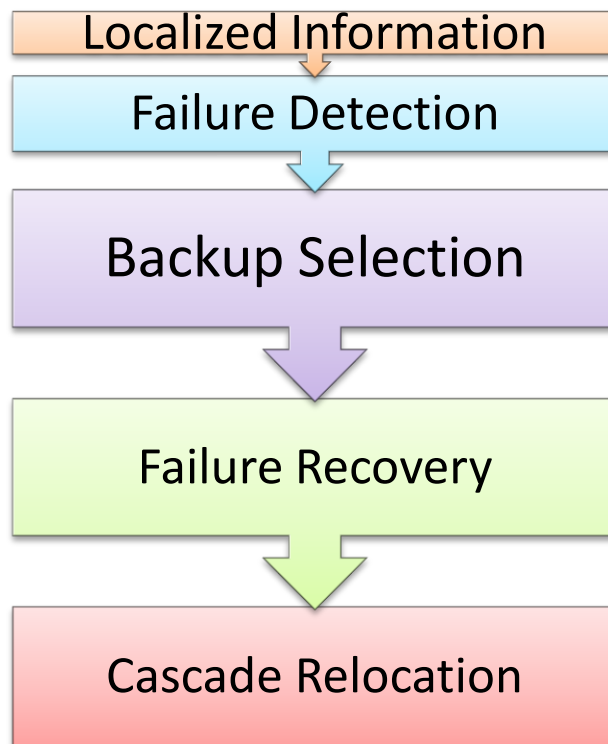


Figure 2.7. Reactive Fault Detection Mechanism Schema

Reactive recovery algorithms are classified into centralized algorithms such as [120] and distributed algorithm such as [99] [121] [29] [24, 25, 28, 29, 119, 122, 123]. Scope of recovery is referred to how many nodes are involved in the recovery. Some mechanisms require single node [99] [121] while others identifies a block of nodes for the recovery process [124] [125].

Due to the nature of WSNs and their important applications, WSNs should support self-reconfiguration in response to a failure. To ensure inner-actor connectivity, node failure detection and recovery techniques should be applied. All approaches attempt to either recover the failed actor or reduce the overhead [126] [127-129].

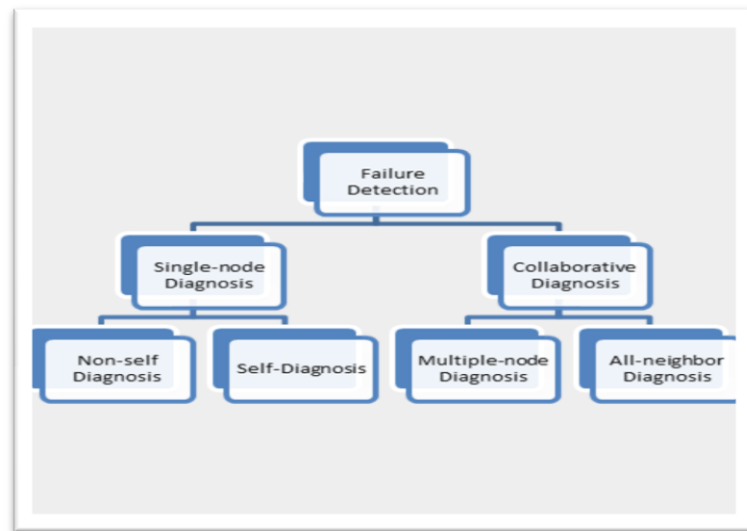


Figure 2.8. Failure Detection

Failure detection refers to the identification of the failed node. Figure 2.8 presents a model for fault detection classification. Fault detection can be classified into single-node diagnosis [21], which consist of self- or non-self-diagnosis, and collaborative

diagnosis, which consists of multiple-node or all-neighbor diagnosis [119]. While fault management techniques can be classified into one of the previous category [5, 20, 21, 120, 124, 125, 128, 130-141].

PADRA is an example of a non-self-single-node diagnosis wherein a distributed actor deployment algorithm is deployed at actors to detect failures by neighbors when their exchange messages are lost [142]. Akkaya and Senel proposed MDAPRA for single-node failure detection and recovery [126]. Then, they extended the approach to address multiple-node detection and recovery. Alfadhly, Baroudi, and Younis proposed the Simultaneous Failure Recovery Approach (SFRA), which overcomes multiple-actor failures in WSNs through ranking nodes applicable to reassigned root actors [143].

In [144, 145], the authors presented a multi-actor/multi-sensor (MAMS) fault tolerance model that ensures network connectivity by allowing the assignment of multiple-sensor connectivity with an actor node and multiple-actor connectivity with each sensor. Thus, multiple actors can receive the sensed event and act upon it, which requires redundant prevention techniques for ensuring real-time reliable actions.

Actors fault impact can vary depending on the node's importance and type. Some fault management detection and recovery procedures classify the actor node to critical nodes and non-critical nodes [5] [124] [5] [125]. Critical node referred to node which its failure causes network partitioning. Most algorithms define the critical nodes using 2-hop message exchange information [100] [127]. On the other hand, [146] used 1-hop message exchange to identify its critical actors. This is performed by calculating the distance from the actor to its adjacent nodes. If the distance is less than the neighbor's

communication range, the actor is defined as non-critical; otherwise, the actor is defined as critical. In [100], the failure of critical node would trigger the recovery process. In the recovery process, non-critical actor node with lowest degree and minimum distance is preferred. [146] prefer to choose a non-critical leaf node with the smallest degree as a backup. Table 3 analyzes existing fault tolerance algorithms in reference to their aim, hops count, and performance parameters.

Cluster-based node failure algorithms have been introduced in the area of WSN and WSNAN such in [147], [148], [149], [136], [150], [151]. In [147], an actor was repositioned in the cluster center to act as a cluster head. The repositioning of the actor is based on position location. A cluster center (CC) is calculated in reference to the area in which the actor node signal is overlapped with all sensors in that area. After sensor node deployment, the sensor nodes discover their one-hop sensors and form clusters. Some sensors act as CH using [152]. CH sensors create a direct link to sensors under their cluster and store their position information. Then, the CH sensor is mapped to an actor node based on their approximate distance to each other. After CH actor assignment, the CH sensor forwards its sensor neighbor information to the CH actor. Based on the forwarded information, the CH actor computes its CC position and then moves toward it. Then, the actor node's position is moved toward the CC. When the actor reaches its CC position, it initiates a direct communication with all sensor nodes in the CC and acts as a CH for them to perform the required action.

This technique attempted to position actor nodes to form a CH but failed to address an actor failure situation. Moreover, the technique consists of overhead connections, which affect sensor and overall network energy.

The approach in [143] relies on applying breadth-first search (BFS) with a pre-assigned root node to the network during deployment. Each node stores the information of its parent node and assigns a recovery weight depending on its distance to its leaf node; in addition, clustered nodes are defined. When a node failure occurs, one of the node's children will replace the parent and restore the connectivity of this sub-tree with the remainder of the network. If this child is not able to reconnect to the network after replacing its parent, the child will use the clustering node to reconnect to its cluster.

The objective of [136] was to minimize latency between two segments by balancing the traffic load between the mobile relay node using a star topology.

Grid-based WSAN has been introduced in many WSAN studies [69, 153], [154]. In [153], a grid-based actor repositioning mechanism was proposed whereby each grid is monitored by a single static actor. Each grid contains one static actor, sensor nodes that sense events, and mobile actor nodes. Static actors retain information of sensors locations along with grid information. Grid information consists of reporting region calculations. When an event occurs, sensor nodes forward sensed data to the static actor node; then, the mobile actor node is repositioned based on the grid mechanism calculation. The overhead of the technique increases when using one static actor node for multiple grid regions monitoring. The negative aspect increases when multiple reporting regions report simultaneously to the same static node.

WSANs are implemented in many applications. Due to the importance of ensuring inter-actor connectivity, various studies in the literature have attempted to address actor network failure and network re-connectivity.

In the following section, we will provide analysis and comparisons of the most well-known node failure recovery algorithms.

### **2.5.1. Distributed Connectivity Recovery Algorithm (DARA)[99]**

The network re-connectivity process is performed by replacing the failed actor node with one of its adjacent actors. In DARA, each actor stores the information of its two-hop neighbors via message exchange among actors. When an actor node fails, the candidate node takes its place. BC node selection is chosen in favor of node degree, distance to failed actor, and actor ID. The node degree is the number of neighbors connected to the actor; the smallest number of neighbors is preferred. The actor at the smallest distance to the failed actor is favored. When two or more actors have the same degree and distance, the actor with the highest ID is selected. After the BC actor is selected, the actor calculates its expected time required to complete the movement to the new location and reports to its connected neighbors by sending a moving message. The moving message contains the time needed for the actor to reach its new position along with information about the new location. When the BC actor reaches its new location, it sends a Recovered message to its neighbors. If a neighbor does not receive the recovered message by the time mentioned in the moving message, it will execute the DARA

scheme to connect itself to the network. Figure 2.9 illustrates the DARA algorithm procedure.

DARA optimization focuses on connectivity restoration. However, DARA is limited due to its limitation to 2-hop information, therein not considering application-level constraints and latency. Moreover, applying DARA to a node may cause the algorithm to execute recursively to adjacent nodes, which causes network recovery overhead. Furthermore, the recursive cascade relocation can negatively impact network resources.

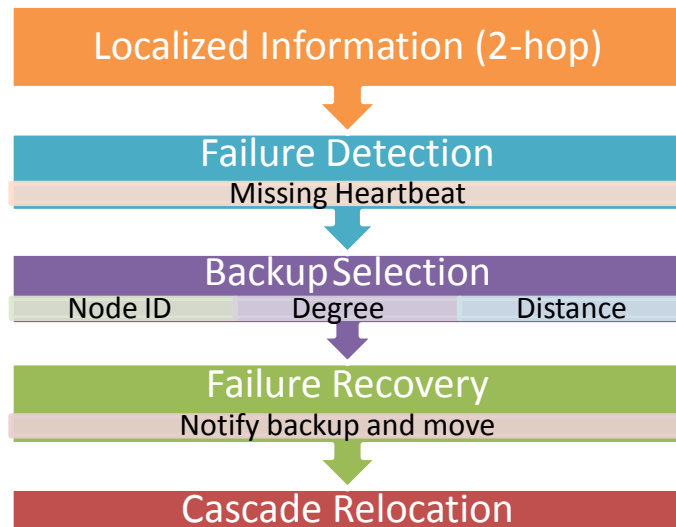


Figure 2.9. DARA Algorithm

### 2.5.2. Recovery Through Inward Motion (RIM) [124]

RIM is a distributed localization algorithm in which each actor node maintains its 1-hop neighbors information. At the network formulation phase, actor nodes exchange hello messages in order to build its 1-hop neighbor list and save it in a table. Table entry

contains each 1-hop node-ID along with its position. Neighbors list maintains updated during the network lifetime. Several actions are considered to maintain neighbors list updated such as exchanging heartbeat messages between 1-hop neighbors, actor node broadcast a hello message when it joins the network, and actor node informs its neighbor when it changes its position. Neighbors diagnose is achieved through the heartbeat messages. If heartbeat messages are lost with a neighbor, this neighbor is considered as failed and recovery process takes place. Then, the 1-hop neighbors of the failed node will start moving toward the failed node position until they connect with each other again. The recovery process is recursively performed further disconnected caused by neighbors movement.

Although RIM manage to minimize message overhead using its 1-hop neighbor list, but the scope of recovery increase due to higher number of nodes which are involved in the recovery process. Moving such nodes may increase the overall network overhead.

### **2.5.3. Actor Critical Recovery (ACR)**

ACR aims to minimize the delay and determining the primary backup node to satisfy application requirement[126]. In this study, Akkaya et al. proposed the distributed partition detection and recovery algorithm (PADRA) to handle cut-vertex node failure recovery. The main objective of the work is to minimize node movement distance during the recovery process. Cut-vertex node determination is done using 2-hop information. PADARA assign a failure handler (FH) node for each cut-vertex. FH is responsible for the network recovery when cut-vertex failure occur. FH chooses to replace the failed



node with the node which has closet distance to that failed node. Figure 2.10 illustrates ACR failure\recovery procedure. The main drawbacks in PADARA are the involved communication and calculation overheads, and FH recovery assignment criteria.

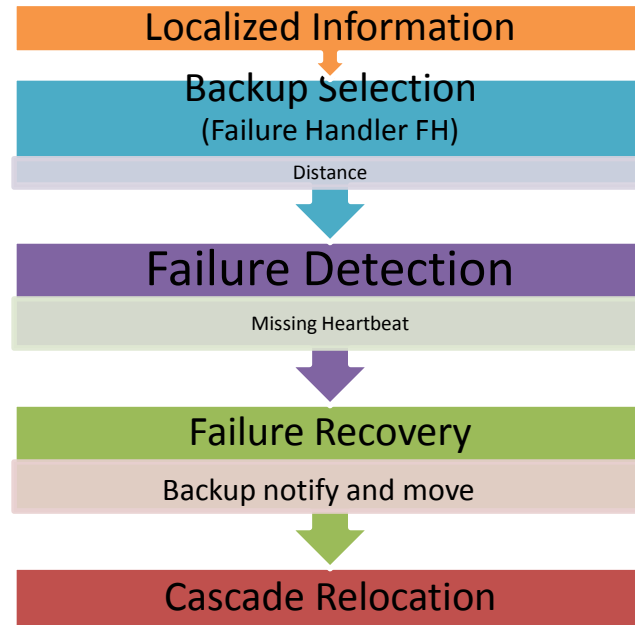


Figure 2.10. ACR failure recovery procedure

#### 2.5.4. Nearest Non-critical Neighbor (NNN)[100]

The NNN algorithm attempts to recover inter-actor connectivity and network partitions. The algorithm applies localized and distributed localization techniques. When the neighbors of a critical actor detect the failed actor, they initiate the reinstatement process. This process consists of the replacement of the critical node with the nearest non-critical actor to control any further splitting overhead. Figure 2.11 illustrates the NNN algorithm procedure. The overhead can occurs when a critical actor is selected for node replacement. Distinguishing between critical and non-critical actor nodes favors the

NNN procedure to have slighter recovery scope in comparison to DARA. However, the network is adversely affected due to spilt transposition overhead.

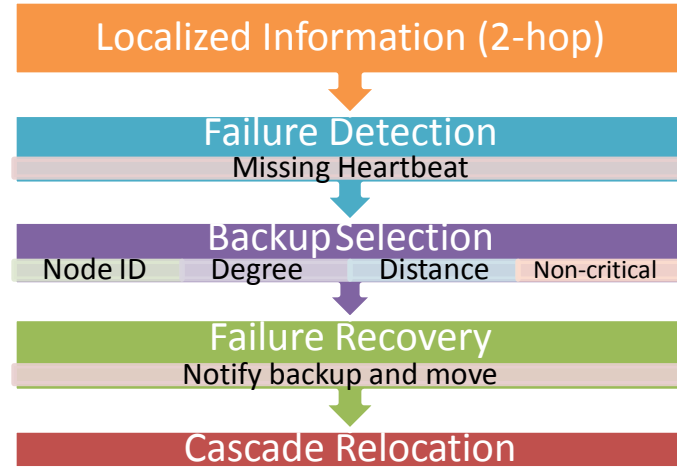


Figure 2.11. NNN Failure recovery procedure

### 2.5.5. Detection and Connectivity Restoration (DCR)[146]

In DCR, a backup actor is assigned to each critical actor node. DCR is an example of proactive fault tolerance algorithm. Critical actor identification is performed locally by each actor using information concerning its one-hop neighbor list. This is performed by calculating the distance from the actor to its adjacent nodes. If the distance is less than the neighbor's communication range, the actor is defined as non-critical; otherwise, the actor is defined as critical. Critical actor backup node assignment is based on node criticality and position. The non-critical leaf node with the smallest degree is preferred. If there are no leaf nodes, DCR chooses the non-critical node with the highest degree as a backup. If a non-critical actor is not available in the neighborhood, it will choose a critical node with the highest degree and smallest distance to the node as a backup. For example, critical

actor C will assign a non-critical backup actor B to itself using AssignBackup(C). Both actors B and C will exchange heartbeat messages to ensure availability. Backup actor B runs the recovery algorithm if it detects the failure of C or if actor C sends a moving notification message. Before moving, actor B will check its criticality status and ensure a backup node assignment in the case that it is a critical actor. Then, actor B will broadcast a movement message and move toward actor C's location. Figure 2.12 illustrates the DCR algorithm procedure. The DCR approach is said to be a hybrid approach by which it identifies the critical actor node in the network. The recovery process depends on neighboring actor status.

DCR has greater storage requirements for every actor node because they maintain a backup of other nodes' information. Similarly, the Partition detection and Connectivity Restoration (PCR) algorithm was proposed with a similar working procedure as the DCR algorithm.

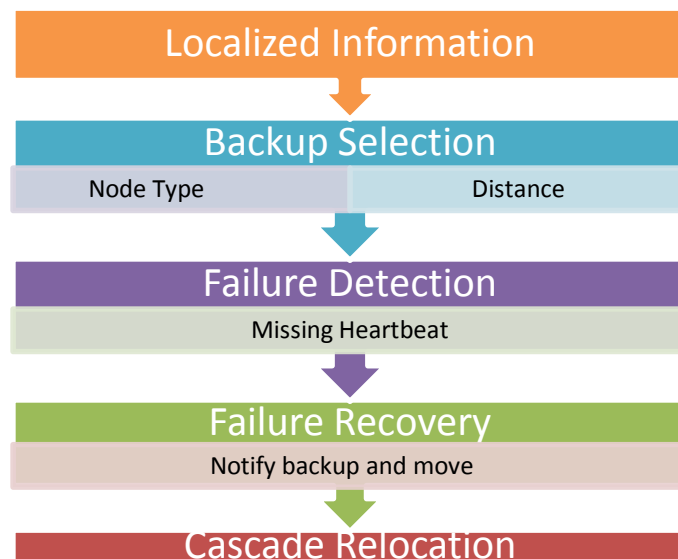


Figure 2.12. DCR Algorithm failure recovery procedure

### **2.5.6. Recovering from Node Failure (RNF) Based on the Least-Disruptive Topology Repair (LeDiR) [155]**

LeDiR is a localized and distributed localization algorithm that attempts to detect and manage cut-vertex node failure and perform recovery using path discovery and routing information. LeDiR attempts to manage the restoration process while ensuring that no path length is extended between nodes [155, 156].

In the case of node failure in LeDiR, neighboring nodes will re-compute their routing tables and derive their enrollment decisions for the recovery process. In response to the failure of a critical node, as shown in figure 1, the neighbor containing the smallest block replaces this node.

LeDiR assumes that each node calculates the shortest path to every other node and stores this information in its routing table. When a node fails, its 1-hop neighbors identify if the node is critical or non-critical using the shortest path routing table. Then, the smallest block is identified. Within the smallest block, a neighbor of a failed node is chosen as the Candidate node to replace the failed node. If more than one neighbor node is part of the smallest block, the neighbor nearest the failed actor is chosen to manage the block movement.

### **2.5.7. Delay and Throughput Performance Improvement in Wireless Sensor and Actor Networks[153]**

In [153], an algorithm is introduced based on positioning regions in WSN by which the actor node communicates with sensor nodes; the actor node calculates the reporting region. The repositioning region is calculated based on the count of the sensor reporting and average energy. The network consists of both static and dynamic actor nodes. The coverage area is split into one, two, three, four and five. Static sensors send the information to the static actor node, which then forwards the information to dynamic actor nodes. The static mobile node verifies the position (P) of the dynamic actor node ( $P_1 \neq P_2$ ) because both the static and dynamic actor nodes should not be in same position. This process achieves improvements in throughput and reductions in delay. On the other hand, even if the number of grids increases, only one positioning point is needed. This requires the positioning point to cover a larger range, which may decrease the algorithm's efficiency. Thus, this may increase the overall network load and affect performance.

### **2.5.8. Distributed Prioritized Connectivity Restoration Algorithm (DPCRA) [21]**

DPCRA is introduced to cover the partitions and reinstate the node connectivity by using small number of nodes. This aims to identify the negative effect of the actor on the partitions. Repairing process is done locally while storing minimum information in each node. The main focus of the work is to use multiple backup nodes for the partitioned recovery.

Even though, the algorithm failed to address backup node selection criteria which leads that those node may have higher probability of failure. Thus, this can affect the overall network performance; especially energy consumption. This leads to higher probability of nodes failure throughout the network.

#### **2.5.9. Recovery of Lost Connectivity in Wireless Sensor and Actor Networks using Static Sensors as Bridge Routers (ACRA) [20]**

Ranga, Dave and Verma introduced an Advanced-self-healing Connectivity Recovery Algorithm (ACRA) using GA. The ACRA determines the nature of the actor whether it is cut vertex or the connectivity of the node using depth-first search algorithm. This work is an extension to their work [157] where they proposed a clustered based coordination framework using GA for WSN. The extension was in considering the recovery from cut-vertex actor failure.

In ACRA, an actor node with high transmission power and higher coverage area is selected, and connectivity is recovered. ACRA is based on two point crossover GA to reconnect the partitioned network. Sensor and actor nodes are scattered randomly in the area of interest and form clusters. All nodes are equipped with a failure detection system and able to detect the failure of cut-vertex actor nodes. When a cut-vertex failure is detected, neighbor cluster head (CHs) broadcast recovery message to all its neighboring nodes toward sink node until next actor node or next CH is found for lost connectivity. Recovery Phase is performed through finding stable with high transmission power and

higher coverage. The stable sensor CHs (as per GA based criterion) among their neighbor nodes is chosen as bridging router for connecting disjoint network.

The main focus of the work is to measure total travel distance and number of the messages. Nonetheless, ACRA tend to consume more energy because the cluster head. Moreover, involving sensor node in the recovery impact sensors resources and made them more variable to failure.

#### **2.5.10. Algorithms Analysis and Evaluation**

Tables 2.3 and 2.4 summarize the most promising existing WSN actor failure and recovery algorithms. Multiple strategies are used to address network re-localization in response to actor failure. Those algorithms fail to enhance energy usage, resource utilization, and QoS.

Significant Considerations should be given while designing WSN algorithms and protocols which include: managing node re-localization, improving recovery from failures, reducing energy consumption, and optimizing parameters in reference to application requirements, improving network efficiency. Additionally, these algorithms must support QoS measures while managing the recovery process.

Thus, future work in WSNs should focus on improving network QoS by implementing optimal actor failure/recovery algorithms that can perform critical node failure restoration.

Table 2.3. Actor Failure and Recovery Algorithms Analysis

	<b>Centralized/ Distributed</b>	<b>Number of Hops</b>	<b>Address Single/  Multiple Node Failure</b>	<b>Aim</b>	<b>Selection Parameters</b>
[99]	Distributed	2-hop	Single node	Recovery while minimizing total travel distance	Node degree which reflects number of neighbors, and distance
[29]	Distributed	2-hop	Single node	Reduce total distance, minimize message overhead	Node degree and distance
[126]	Distributed	2-hop	Single node (PADRA) Multiple node (MDAPRA)	Localize the scope of the recovery, reduce message overhead and total distance	Nearness distance to failed node
[124]	Distributed	1-hop	Single node	Reduce total distance, minimize message overhead	Rank within neighbors
[120]	Centralized/ Distributed	1-hop	Multiple Node	Reduce total distance	Node_ID, Relative position
[124]	Distributed	-	Single Node	Maintaining path length	Distance to failed node and block size
[127]	Distributed	2-hop	Single Node	Reduce total distance	Favor non-critical node and distance to failed node
[158]	Distributed	-	Multiple Node	Reduce total distance	Network optimization parameters
[28]	Distributed	-	Single Node	Minimize number of relocated node and reduce total distance	Smallest block size
[146]	Distributed	1-hop	Multiple Node	Reduce recovery scope and total distance	Node status( critical/noncritical), degree and distance
[159]	Distributed	-	Multiple Node	Reduce total distance	Network optimization parameters



[123]	Distributed	1-hop	Single Node	reduce total distance, and awareness of message overhead	Node status( critical/noncritical), overlapped coverage, connectivity degree, task priority
[160]	Distributed	-	Multiple Node	Reduce total distance	Distance to the failed nodes along with node properties
[146]	Distributed	1-hop	Single Node	Localize the scope of recovery and minimize the movement overhead	Node status ( critical/ noncritical), position
[100]	Distributed	2-hop	Single Node	Reduce total distance, reduce cascade relocation overhead	Node status( critical/noncritical), degree and distance
[155]	Distributed	-	Single Node	Minimize scope using path length validation	Block movement, path length constrains

Table 2.4. Failure detection\recovery algorithms advantages and limitations

Algorithm	Features	Limitations
[99]	<ul style="list-style-type: none"> <li>• Distributed reactive method.</li> <li>• 2-hop neighbor list for failure detection and recovery.</li> </ul>	<ul style="list-style-type: none"> <li>• Recovery process may involve increasing path lengths among nodes.</li> <li>• No application-level or latency constraints.</li> <li>• Recursive overhead.</li> <li>• High energy consumption.</li> </ul>
[124]	<ul style="list-style-type: none"> <li>• Distributed Reactive method.</li> <li>• 1-hop neighbor list</li> <li>• The recovery process tends to move more than one node toward the failed actor in-order to reconnect the network.</li> </ul>	<ul style="list-style-type: none"> <li>• Moving more than one node increase the overall network overhead in sense of total travel and the recursively execution of RIM</li> </ul>
[126]	<ul style="list-style-type: none"> <li>• Distributed 2-hop</li> </ul>	<ul style="list-style-type: none"> <li>• Nearest distance to failed</li> </ul>

	<ul style="list-style-type: none"> <li>FH is responsible for the network recovery when critical node failure occur</li> </ul>	node which impact the overall network.
[146]	<ul style="list-style-type: none"> <li>Distributed Hybrid method</li> <li>Pre-assigned backup actor for cut-vertex nodes.</li> <li>1-hop neighbor list for identifying cut-vertex node.</li> <li>Reduces the total travel distance overhead and engages few nodes.</li> <li>Avoids the overhead of tracking 2-hop neighbors.</li> </ul>	<ul style="list-style-type: none"> <li>Does not consider the energy as a performance factor when selecting the backup node.</li> <li>Considers managing one failure at a time and that no other node fails during the recovery.</li> <li>The backup node restoration process does not consider further network connectivity failure caused by node movement.</li> </ul>
[100]	<ul style="list-style-type: none"> <li>Distributed Reactive method</li> <li>2-hop neighbor list for failure detection and recovery.</li> <li>Selecting nearest non-critical actor for replacing the failed node to improve performance over DARA.</li> </ul>	<ul style="list-style-type: none"> <li>Recovery process may involve path increments among nodes.</li> <li>Recovery process may involve the algorithm executing recursively, which leads to overhead.</li> </ul>
[155]	<ul style="list-style-type: none"> <li>Distributed Reactive methods</li> <li>Uses path discovery and routing information for failure detection and recovery.</li> <li>Technique avoids pre-failure communication overhead and 2-hop neighbor tracking.</li> <li>Recovery process reduces the extension of length among nodes.</li> </ul>	<ul style="list-style-type: none"> <li>Does not address multiple node failures. Can only address single-node failure recovery</li> <li>Overhead from Moving more than one node which has an impact to the overall network.</li> </ul>
[21]	<ul style="list-style-type: none"> <li>Is introduced to cover the partitions and reinstate the node connectivity by using</li> </ul>	<ul style="list-style-type: none"> <li>Failed to manage backup node selection criteria which leads to higher probability of</li> </ul>

	<p>small number of nodes. This aims to identify the negative effect of the actor on the partitions. Repairing process is done locally while storing minimum information in each node. The main focus of the work is to use multiple backup nodes for the partitioned recovery.</p>	<p>nodes failure throughout the network.</p>
[153]	<ul style="list-style-type: none"> <li>• Grid-based Method.</li> <li>• The network consists of both static and dynamic actor nodes.</li> <li>• The coverage area is split into one, two, three, four and five. Static sensor sends the information to the static actor node, which then forwards the information to dynamic actor nodes.</li> <li>• The static mobile node verifies the position of the dynamic actor.</li> </ul>	<ul style="list-style-type: none"> <li>• A single positioning point for more than one cell is not efficient</li> <li>• All the nodes approach the same point for transmission.</li> </ul>
[20]	<ul style="list-style-type: none"> <li>• Clustered based, where an actor node with high transmission power and higher coverage area is selected, and connectivity is recovered.</li> </ul>	<ul style="list-style-type: none"> <li>• Actor nodes include higher transmission power, but the process still consumes more energy.</li> </ul>

## **CHAPTER 3 EFFICIENT ACTOR RECOVERY PARADIGM FOR WIRELESS SENSOR AND ACTOR NETWORKS**

The WSANs comprise of actors with powerful resources and sensor nodes with limited computation, power, and communication capabilities. The Sensors and actors in WSANs collaborate together to monitor and respond to the surrounding world. The WSANs can be applied to wide range of applications, like health, environmental monitoring, chemical attack detection, battlefield surveillance, space missions, intrusion detection etc. However, the WSANs are greatly affected due to environmental change, frequent change in event-detection and actor failure process. The failure of an actor node can result in a partitioning of the WSAN and may limit event detection and handling. Actors may fail due to hardware failure, attacks, energy depletion, or communication link issues. Sensor node failure may cause lost event detection of the assigned environment covered by the sensor. The probability of actor failure is less than that of sensor failure and can be controlled through the relocation of mobile nodes due to their powerful characteristics; however, actor failure can cause more damage than can sensor failure. Actor failure can cause a loss of coordination and connectivity between nodes, limitation in event handling, and can leads to a disjoint WSAN.

The actor failure occurrence is very critical that degrades the network performance. The failure of critical actor may cause high impact for the whole network. Critical actor node refers to actor which its failure causes network partitioning. Most of the existing approaches attempted to replace the critical node with another backup node,

but they failed to maintain the QoS parameters and energy consumption. For instance, RNF manage to handle failure by moving a small block of neighbor actors toward the failed node in order to recover the communication among them. Even though this manages the recovery of the network but it enlarges the recovery scope and cascade relocation. Such behavior should be eliminated in recovery algorithms. In addition, due to the fact that WSAN deployed in harsh area and requires long term monitoring/acting process, proposed methods should offer robust self-healing failure detection/ recovery techniques which ensures that network lifetime is maximized as much as possible while maintaining QoS.

All existing approaches either attempt to recover the failure actor or try to reduce the overhead [123, 127, 128, 134, 161, 162]30-35]. We conclude that existing approaches attempted to replace the critical node with another backup node, but they failed to maintain the QoS parameters and energy consumption. Thus, a new method is proposed for efficient actor recovery paradigm (EAR) which guarantees the contention-free traffic-forwarding capacity [163]. Unlike previous studies, EAR craves for providing efficient failure detection and recovery mechanism while harvesting maintains the Quality of service.

For ensuring QoS in EAR algorithm, Node Monitoring and Critical Node Detection (NMCND) algorithm that monitors the activities of the nodes to determine the nodes types and distinguish critical nodes. Additionally, the proposed approach not only determines the critical node, but handles the packet forwarding process when primary node fail. To handle packet forwarding, Network Integration and Message Forwarding

(NIMF) message was introduced. In addition, process -Based Routing for Node Failure Avoidance algorithm (PRNFA) was developed in order to handle the routing process and to eliminate routing process of the redundant packets to other node in order to avoid the network congestion and reducing the latency. Therefore, the goal of this work is to improve the recovery node process while maintaining the QoS provisioning and power efficiency. Detailed description of the model is provided in the following sections.

### **3.1. Mathematical Model**

In WSANs, the nodes track their neighbors by using heart beat messages to avoid any possible interruption. Moreover, algorithms are used to define the critical nodes using 1-hop or 2-hop message exchange information [8,36,40]. In addition, they identify the actor node failure by the interaction of those heartbeat messages with this particular actor node. Thus, there is possibility of interruption due to losing the trail of heart beat messages. Monitoring actor failure detection using 2-hop neighbor list is efficient once it is combined with QoS measurement capabilities, i.e., packet delivery and forwarding techniques should support efficient packet handling and forwarding. Also, we should minimize the through critical actor nodes. Thus, in the proposed EAR model, we assume that each actor node stores the information up to 2-hops to keep the extended trail information. This helps determine the forwarding capability of the actor nodes. The model aims to ensure the contention-free forwarding capability that minimizes the loss of packets in case of node failure. To determine the actor's forwarding capability, each actor conveys the group of beacon messages using different power strengths. Furthermore, the neighbor of each actor listens and returns the value in response. After the neighbor actor

receives the message, it starts calculating its RSSI value and sends it back to the sender actor. The RSSI model is used to calculate the distance. It has also been combined with further techniques for better accuracy and to find the relative error. Equations (3.1)–(3.5) illustrate applying RSSI in actor nodes [55]. RSSI can also be used to determine the link quality measurement in wireless sensor networks [56]. The RSSI shows the relationship between the received energy of the wireless signals and transmitted energy and the required distance among the actor-sensor nodes. This process helps determine failure node recovery process given in Definition 1. The relationship is given by Equation (3.1):

$$E_r = E_t \times \left(\frac{1}{r}\right)^\beta \quad (3.1)$$

where  $E_r$ : Received energy of wireless signals,  $E_t$ : transmitted energy of wireless signals,  $r$ : Distance between forwarding and receiving node, and  $\beta$ : Path loss transmission factor whose value depends on the environment.

Taking the logarithm of Equation (1) provides:

$$10\log E_r = 10\log E_t - 10\beta \log r \quad (3.2)$$

where  $10\log E$ : Description of energy that could be converted into dBm.

Therefore, Equation (2) can be converted to its dBm form as:

$$E_r(\text{dBm}) = \gamma - 10\beta \log r \quad (3.3)$$

where  $\gamma$ : transmission parameter.

Here,  $\gamma$  and  $\beta$  represent the relationship between the strength of the received signals and the distance of the signal transmission among sensor-to-sensor, actor-to-sensor or actor-to-actor.

RSSI propagation models cover free-space model, log-normal shadow model and ground bidirectional [20]. In this study free space model is used due to following conditions.

- The transmission distance is larger than carrier wavelength and antenna size.
- There is no obstacle between forwarding actor and either receiving actor or sensor.

The transmission energy of the wireless signals and the energy of the received signals of sensor nodes located at distance of 'r' can be obtained by Equations (3.4) and (3.5):

$$\frac{E_t A_{gt} A_{gr} \lambda^2}{(4\pi)^2 r^2 \omega} \quad (3.4)$$

where  $\lambda$ : 1/Frequency of the actor node,  $A_{gt}$  &  $A_{gr}$ : An antenna gains,  $\omega$ : failure factor of the actor and r: distance of the node:

$$E\omega(dB) = 10 \log \frac{A_{gt}}{A_{gr}} = -10 \log \left[ \frac{\lambda^2}{(4\pi)^2 r^2} \right] \quad (3.5)$$

Equation (3.5) represents the signal attenuation using a logarithmic expression. Assume a field with k actor nodes  $a_1, a_2, a_3, \dots, a_k$ . The coordinates of the actor nodes



are  $(p_i, q_i)$  for  $i = 1, 2, \dots, k$ . The actor nodes transmit the information regarding their location with their signal strength to the sensor nodes  $\{s_1, s_2, \dots, s_n\}$ . The locations of the sensor nodes are unknown. The estimated distances of the actor nodes are calculated from the received signals.

In the proposed model, the actor nodes broadcast signals to all sensors. The actor nodes are also responsible for estimating the distances between them and sensor nodes. Let  $a_i$  be an actor node located at  $(y_i, z_i)$  and sensor node is located at  $(y, z)$ . Focusing on the relative error ' $r_e$ ' relating to  $a_i$ , suppose that the actor node reads a distance  $r_i$ , but the correct distance is  $r_j$ . Therefore, the relative error can be obtained by Equation (3.6):

$$r_e = \frac{r_i}{r_j} - 1 \in (-1, +\infty) \quad (3.6)$$

The relation between actual distance and the measured distance can be obtained by Equation (3.7):

$$r_i = r_j 10^{\frac{\pi}{10\beta}} - 1 \quad (3.7)$$

which can be reduced to Equation (3.8):

$$r_e = 10 \cdot \frac{\pi}{10\beta} - 1 \quad (3.8)$$

The probability distribution of the location of the actor node based on beacon messages is described in the following definition.

**Definition 1.** Let  $a_i$  be an actor node located at  $(y_i, z_i)$  that sends information to a sensor using RSSI model with standard deviation ' $\sigma$ ' and path loss ' $\beta$ '.

Let  $r_i$  be the calculated distance from the actor node  $a_i$  at the sensor node. The probability density function for correct location  $(y, z)$  of the sensor node is obtained by Equation (3.9):

$$\left\{ E_{Y,Z}^{(i)}(Y, Z) \frac{10\beta \exp \left( - \left( \log \beta \log \frac{r_i}{\sqrt{(y-y_i)^2 + (z-z_i)^2}} \right)^2 / 2\sigma^2 \right)}{2\sigma^2 \sqrt{2\pi \log(10) ((y-y_i)^2 + (z-z_i)^2)}} \right\} \quad (3.9)$$

Probability distribution can be simplified due to an actor  $a_i$  with Equation (3.10):

$$E_{Y,Z}^{(i)}(Y, Z) = \Psi_{a_i}(y, z) \quad (3.10)$$

This can further be extended by using finite set of actors

$A = \{a_1, a_2, a_3, \dots, a_k\}$  that produces definition 2.

**Definition 2.** Let  $A = \{a_1, a_2, a_3, \dots, a_k\}$  be the set of the actors sending information to the set of sensor nodes using RSSI model with path loss exponent ' $\beta$ '.

If the calculated distance from the actor node  $a_i$  at the sensor nodes (S)

$S = \{s_1, s_{12}, \dots, s_n\}$ , then the probability density function of correct location  $(y, z)$  of the sensor nodes can be obtained by:

$$\left\{ \Psi(A)(y, z) = \frac{\sum_{i=0}^k \Psi_{a_i}(y, z)}{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (\sum_{i=0}^k \Psi_{a_i}(y, z) dz dy)} \right\} \quad (3.11)$$

where  $\Psi_{a_i}(y, z)$ : probability distribution because of an actor  $a_i$ .

**Definition 3.** Assume an actor node  $a_i$  reads a sample distance  $R = \{r_1, r_2, r_3, \dots, r_n\}$  using beacon messages  $B = \{b_1, b_2, b_3, \dots, b_n\}$  that is modeled with RSSI with path loss  $\beta$  and standard deviation ' $\sigma$ '.

If ' $R$ ' provides the mean sample distances and  $\sigma_i$  is the mean standard deviation, then the square of the actual distance from actor to sensor using beacons can be determined as:

$$R = \frac{R^4}{R^2 + \sigma_i^2} \quad (3.12)$$

Furthermore, square standard deviation can be found as:

$$\sigma_i^2 = \frac{100\beta^2}{\log^2(10)} \log\left(1 + \frac{\sigma}{R}\right)^2 \quad (3.13)$$

The definition shows that the actual distance is greatly dependent on the distribution of the measured ranges.

Hence, our proposed formulas for RSSI-based wireless node location are optimized and modified. They are different from the original RSSI-based formulas. It was focused particularly on the energy consumed for transmission and receiving the data including determining the distance between actor-sensor nodes and error rate for finding location of the node that helps identifying the accurate position of the deployed actor nodes for events. Thus, the previous model is used by our proposed algorithm in order to identify the node locations during deployment in addition to during the network lifetime.

### 3.2. Optimized Deterministic Actor Recovery System Model

The network consists of multiple actors and sensor nodes that are structured with the hierarchical structure of the nodes. The hierarchical structure of the nodes provides an efficient, fast and logical packet forwarding patterns. It also determines the features of all nodes connected with WSANs. Another advantage of the hierarchical structure is that it helps to start with little multiplexing process for intra-domain routing. As the packets travel further from the source node the model helps to develop higher degree of multiplexing. The nodes of different categories in WSAN as depicted in Figure 3.1 possess the assorted nodes types. The network aims to use the resources efficiently for each packet forwarded by an actor node. In addition, it reduces the latency while keeping the network more stable.

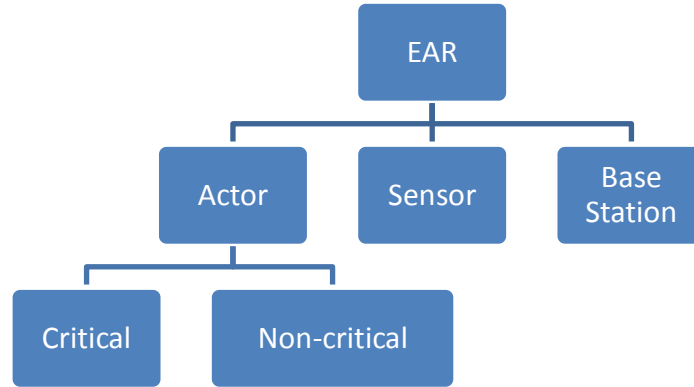


Figure 3.1. Node Type in EAR

**Definition 4.** Critical actor node is the actor node which its failure causes network partitioning.

**Definition 5.** Non-critical nodes (NCNs) are regular actor nodes.

**Definition 6.** Cut Vertex Nodes (CVNs) are nodes which have a cut-vertex link with a critical node, i.e., neighbors of critical node.

**Definition 7.** The Critical Backup Nodes (CBNs) are actor nodes that are assigned to be the backup nodes for a critical node.

The EAR consists of actor nodes, sensor nodes, and base station. Actor node can be critical or non-critical. Critical actor node is the actor node which its failure causes network partitioning. Non-critical nodes are regular actor node. Sensors node are used to monitor the network for event detection.

In this topology, Cut Vertex Nodes (CVNs) are responsible for the removal of the paths that lead to the critical nodes. When the actor node becomes a critical node, then it is necessary to redirect the traffic of the neighbor nodes to the non-critical nodes. Thus, this task is done by removing the vertex (a path leading to critical nodes) and redirecting the traffic, as it further helps improving the throughput performance and reduces the latency. The Critical Backup Nodes (CBNs) replace the actor nodes when the actor nodes become the critical nodes. We assume that the numbers of critical backup nodes are more than actor nodes in the network. If all the actor nodes become critical nodes, then replacement should be much easier to avoid any kind of interruption or data loss. There could a possibility of disconnecting the direct communication links of the actor nodes towards the backup nodes when the actor nodes start moving. Thus, we also assume that the links of the actor nodes lead to the backup nodes are always stable despite the mobility. Therefore, there is a high possibility to easily replace the critical nodes with CVNs. The actor node has a privilege to collect the data from event-monitoring nodes (sensor nodes), then it forwards the packets to either base station or sensor /actor nodes in the network. On the other hand, the least degree Non Critical Nodes (NCNs) are preferred to be labeled as backup nodes for event-monitoring nodes.

A neighbor node that is available in Node Distance range (ND) has a similar Cut Vertex Node Distance (CVND). This helps reduce the recovery time and overhead which is important for resource-constrained mission-critical applications.

In the network, each actor node maintains its 2-hop neighbors' information using heartbeat messages. This information helps to maintain the network state, defining critical actor; as well as assigning backup node for the critical actor. Each actor node saves its neighbors information which includes node ID, RSSI value, number of neighbors which is denoted by the degree of node, node criticality (critical actor/ non-critical actor), and node distance. Once a critical node is detected, the backup assignment process is executed in-order to assign a backup node for this critical node. The 2-hop node information is used in the process of backup node selection. Depending on RSSI value (extracted from mathematical model), the non-critical node with the least node degree is preferred to be chosen as a backup node. In case there is more than one neighbor with the same node degree, the neighbor with the least distance is preferred. For each critical actor, a pre-assigned backup actor node is selected which is called Critical Backup Nodes CBN. Consequently, CBN monitors its critical node through heartbeats messages and handles the backup process in case the failure of its critical node. Missing a number of successive heartbeats messages at CBN indicates the failure of the primary.

To illustrate node types, a proposed system model for traffic monitoring using WSN is used. The traffic-monitoring process using the proposed EAR system model is shown in Figure 3.2. We assume that the sensor and actor nodes get failure during the monitoring process. In the proposed scenario, the sensor nodes monitor the traffic, once

sensor node gets failure, and then NCN replaces the failed node in order to continue the monitoring process. If the actor node starts to be critical node (when actor node starts to run out energy or read the traffic wrongly), then it is of paramount significance to redirect the network traffic to neighbor nodes that are CBNs. Thus, the CVNs are responsible for the removal of the paths that lead to the critical nodes. Furthermore, CVNs help replace critical node by CBNs. This systematic process further helps improve the throughput performance and reduces the latency.

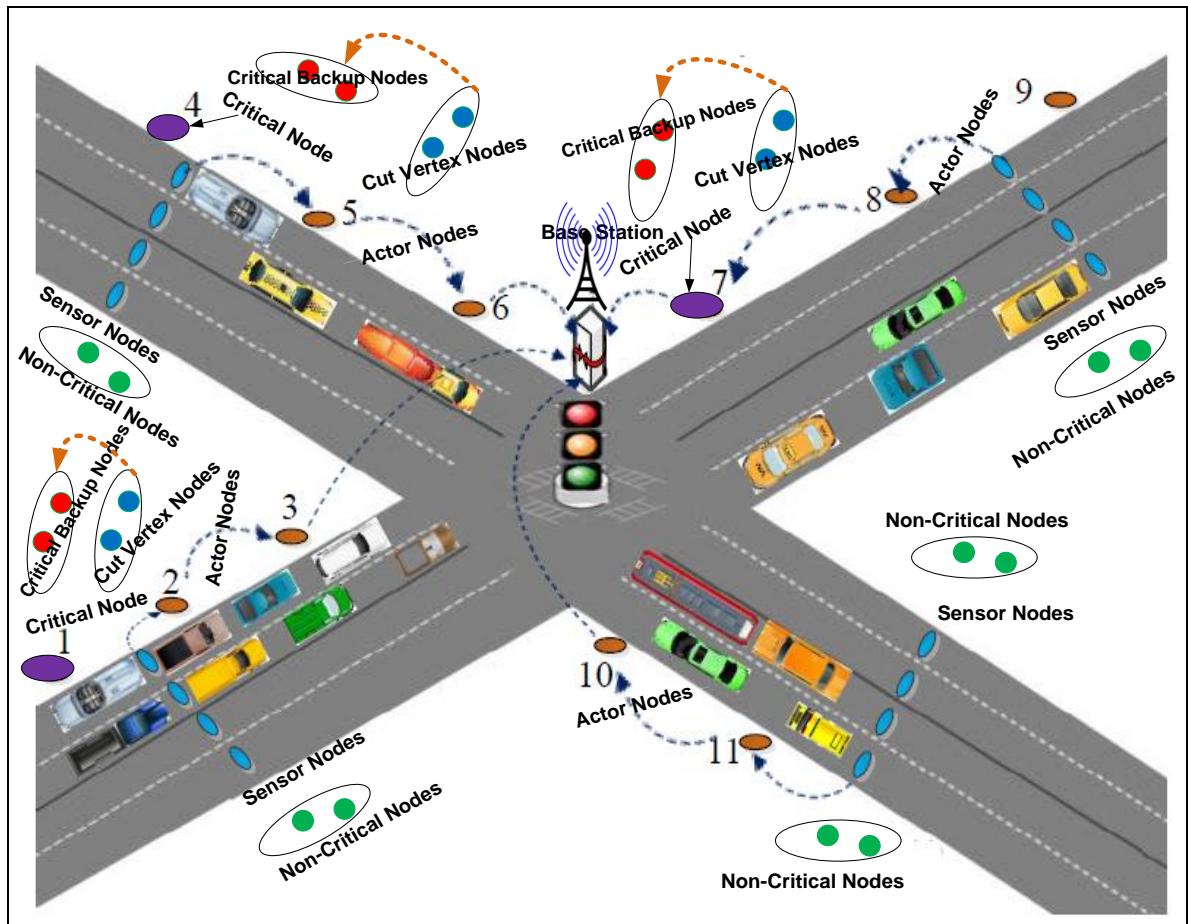


Figure 3.2. Optimized Deterministic proposed system for traffic monitoring using EAR

The topology of the WSN can be changed during the network lifetime due to the mobility feature of actors, actor node failure, or event handling. Backup nodes are subject to failure as well. Therefore, there are primary backup nodes that select other backup nodes in case of primary backup nodes fail or move beyond the range of ND. To ensure the effectiveness and availability of backup node, a novel backup node selection process is introduced in case of primary backup is either failed or in critical condition given in Algorithm 3.1.

Moreover, existing approaches attempted to replace the critical node with another backup node, but they failed to maintain the QoS parameters and energy consumption. To overcome existing approach limitations and ensuring QoS in our EAR algorithm, Node Monitoring and Critical Node Detection (NMCND) algorithm that monitors the activities of the nodes to determine the nodes types and distinguish critical nodes. Additionally, our proposed approach not only determines the critical node, but handles the packet forwarding process when primary node fail. To handle packet forwarding, Network Integration and Message Forwarding (NIMF) message was introduced. In addition, process -Based Routing for Node Failure Avoidance algorithm (PRNFA) was developed in order to handle the routing process and to eliminate routing process of the redundant packets to other node in order to avoid the network congestion and reducing the latency. Therefore, the goal of this work is to improve the network performance and recovery node process while maintaining the QoS provisioning and power efficiency. Detailed description of the algorithms presented in the following sections.



The proposed algorithms can be deployed in variety of applications e.g., smart systems, such as monitoring large-scale wind turbine, battlefield, airport surveillance system, grid-monitoring, and smart-house. As these applications consist of busy-traffic, thus, there is possibility of high failure rate in such unfavorable scenarios due to several factors such as power lassitude of actor nodes, electronic circuit failure, software counter error or physical damage and inter-actor connectivity problem. In these applications, actor nodes are fixed on some particular places to play a role as data-gathering and data-forwarding gateway. In case of the failure, entire communication process is severely affected and that leads to poor QoS provisioning. Thus, it is necessary to ensure connectivity and to have backup node for such applications in order to avoid any kind of possible interruption. As, our proposed algorithms help select the backup nodes and performing successful data forwarding process and determining the legitimacy of the actor nodes for data-privacy.

### **3.2.1. Node Monitoring and Critical Node Detection (NMCND)**

Node monitoring process is used in order to monitor the node pre-failure causes, the post-failure causes and allocates the recovery options. Once each critical actor node picks a suitable backup, then it is informed through regular heartbeat messages (Special signals are sent to neighbor node to play a role as backup node for critical node). Furthermore, the pre-designated backup initiates monitoring its primary actor node through heartbeats. If a number of consecutive heartbeats are missed from the primary actor, then it notifies that the primary actor failed. Thus, a backup node replacement process is started as given in algorithm 3.1.

---

**Algorithm 3.1:** Backup Node Selection Process

---

$P_b$ : Primary Backup;  $Pb_c$ : Critical Primary Backup;  $Pb_m$ : Moving Primary Backup;  $S_b$ : Secondary Backup;  $Sb_c$ : Critical Secondary Backup;  $Sb_m$ : Moving Secondary Backup;  $T_b$ : Tertiary Backup.

1. **Input:** ( $P_b$ )
  2. **Output:** ( $S_b, T_b$ )
  3. **If**  $P_b = Pb_c \parallel Pb_m$ //The condition of primary backup node is checked.
  4. Notify  $S_b$  and Set ( $S_b, P_b$ )//Secondary backup node is assigned as primary backup node
  5. **end if**
  6. **If**  $S_b = Sb_c \parallel Sb_m$ //The condition of Secondary backup node is checked.
  7. Notify  $S_b$  and Set ( $S_b, P_b$ )//Tertiary backup node is notified to play a role as primary backup node
  8. **end if**
- 

Algorithm 3.1 shows the backup selection process. In this process, the condition of primary backup  $P_b$  node is checked. If a primary backup node is in critical condition or ready to move, then a secondary backup node  $S_b$  is chosen. However, if a secondary backup node  $S_b$  is in critical condition or ready to move, then tertiary backup node  $T_b$  is notified to play a role as primary backup node. If the tertiary backup node is in critical condition then the backup assignment algorithm executes and a backup node is assigned.

Before substantiating the In the post failure process, we must ensure that connection is not interrupted because of the network. In addition, any redundant action of the network must be controlled to avoid any possible increase of the network overhead. The pre-failure backup node process is given in algorithm 3.2 which denoted as node monitoring and critical node detection (NMCND).

---

**Algorithm 3.2:** Node monitoring and critical node detection (NMCND) process

---

$\{F_{pr}$ : Pre-Failure;  $F_{po}$ : Post-Failure;  $R_{pr}$ : Recovery process;  $N_c$ : Critical node;  $N_b$ : Backup node;  $A_p$ : Primary actor;  $M_{hb}$ : Message heartbeat;  $N_s$ : Sink node}

1. **Input:**  $\{N_c; N_b; A_p\}$
  2. **Output:**  $\{R_{pr}; F_{pr}; F_{po}\}$
  3. **Set**  $A_p$ //Number of actor nodes are set as primary actors in the network
  4.  $N_s$  broadcasts  $M_{hb}$ //Sink node broadcasts the message to all primary actor nodes
  5. **If**  $A_p \neq F_{pr}$  then//
  6. Determine  $N_c$ //Initiate critical node discovery process
  7. **If**  $\forall A_p: A_p \in N_c$  then
  8.  $N_b$  assigns  $N_c$
  9. **Set**  $N_c = M_{hb}$
  10. **end if**
  11. **If**  $M_{hb}$  NotDelivered  $N_c$  then
  12. **Set**  $N_b$  for data delivery
  13. **end if**
  14. **Process**  $A_p \exists R_{pr}$ //Primary actor node recovery process is conducted
  15. **end if**
- 

As shown in algorithm 3.2,  $A_p$  represents the number of actor nodes in the network. Then, the sink node  $N_s$  broadcasts the message to all primary actor nodes to determine pre-failure actor nodes. If primary actor is not identified as pre-failure, then the process of determining the critical actor node will be started in order to choose the backup node. Next, the critical node discovery process is initiated. For each primary node, if the primary node is found as critical node then this node will be defined as critical node  $N_c$  and a backup node  $N_c$  selection is assigned. The critical node will broadcast a message to its neighbors which includes the information of its backup node. This information is stored by the neighbors and it is used when starting the network recovery process in case the neighbors detect the failure of their critical actor neighbor. If consecutive heartbeat messages are not received from the critical node, then back node starts replacing the critical node to avoid any kind of packet-forwarding delay. In

addition, neighbors of the critical node will use stored information to communicate with the backup node in order to restore connectivity. In conclusion, the recovery process is conducted.

After the node monitoring process is executed, the model proceeds further with checking the backup assignment and the critical back up assignment of the nodes. This allows maintaining the network connectivity without generating any disjoint procedure of the network. The possibility of the recovery depends on the cut vertex node. If the backup is a non-critical node, then it simply substitutes the primary actor node, and the recovery process is initiated to confirm the backup actor node. If the backup is also a critical node, then a cut vertex node replacement is completed. The pre-assigned backup actor node instantly activates a recovery process once it senses the failure of its primary actor node. The complete node monitoring including failure, recovery and replacement processes are depicted in Figure 3.3. In complete node monitoring process, first, the node identification process is initiated. The node is identified based on local neighbor information (LNI) that involves global data position, node property and node degree. The critical node selection process is decided using algorithms 3.1–3.2. Once a critical node is identified, it will be assigned a backup node; second, the backup node selection process is started if an actor node fails. The selection process is decided based on monitoring the algorithms explained earlier. Once the backup node selection process is complete, then the backup process starts working in case of node failure. If an actor node does not fail, then the node connectivity monitoring process is started and routing connectivity metrics are checked to ensure whether there is no problem of the router.

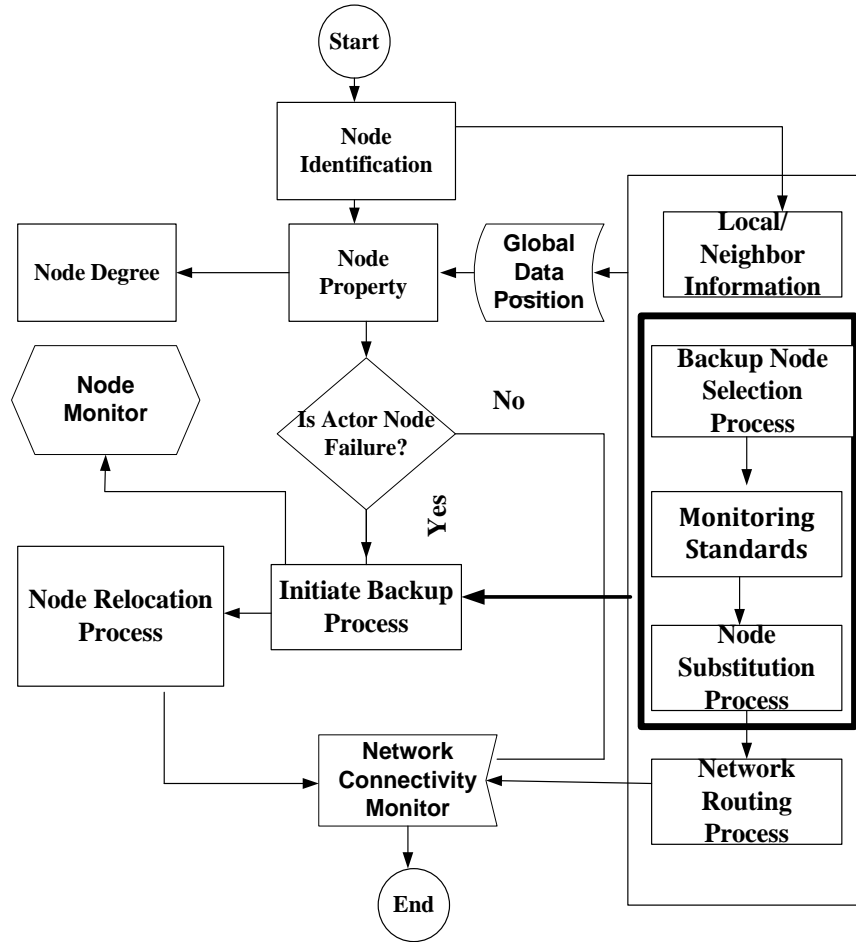


Figure 3.3. Efficient Actor Recovery system model implementation

### 3.2.2. Network Integration and Message Forwarding Process

After completion of the actor node failure and node assignment processes, the actor nodes should be linked to forward the collected data to the base station. In response, the base station sends its identity (ID) using network integration message (NIM).

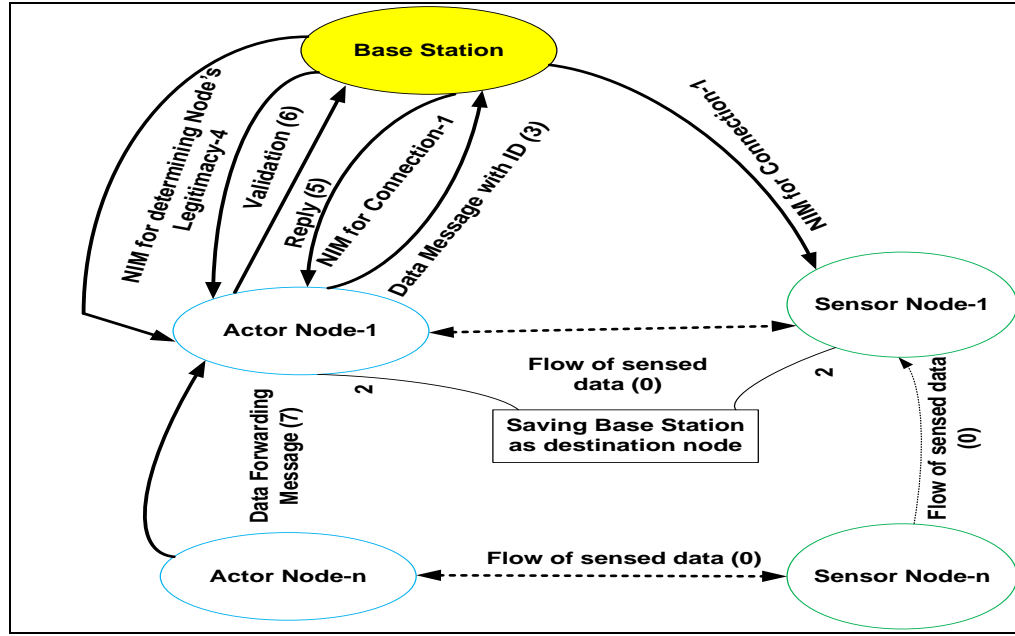


Figure 3.4. Graph representation of NIM

To illustrate NIM, the network is shown as undirected graph  $G = (V, E)$  where each edge  $e \in E$  that has capability to transmit the amount of traffic through  $e$ , Figure 3.4. The actor and sensor nodes in the 'V' denote for the entities that intend to send the traffic. In the network, sensor node represent the local entity. An edge denotes that there is either physical or logical connection between the sensor nodes. Hence the capability of the edge shows the exact capability of the connection. In addition, the model includes the sensor and actors' capabilities. The physical or logical connection between the sensor nodes denotes that there is possible change for those sensor nodes that are available on that edge 'e' to forward the traffic directly rather than neighbor nodes. The source node (Either sensor node or actor node) that has specific intention to route the traffic to the destination node (Base station or sink node). The source and destination nodes agree on the amount of traffic to be routed. In the network, if the edge capabilities are large

enough, then all the data still hold, unfortunately, if the edge capability is small enough, then there could be instability in the network and having a chance to fail the network scenario. Furthermore, the network integration message (NIM) determines the legal position of either sensor or actor node in the network. If the node is identified as legal through NIM, then sensed data is routed through that node to the next hop. Similarly, the multiple actor nodes are checked through NIM in order to validate their status (in the network. If illegitimate node is found erroneously as the legitimate node, then there is possibility of malicious attacks that lead to network failure. As, this situation happens once all the backup nodes get failure at the same time, then illegitimate node replaces the legitimate backup node and declares itself as legitimate node and gains the access to the confidential data. However, this condition can happen in worst-case when getting failure the all backup nodes.

When an actor node receives NIM from the base station, it saves the destination address of the base station for packet forwarding (PF). Subsequently, NIM is broadcasted in the network among all the actors. At least one actor node is within the range of the base station to avoid any bottlenecks. Otherwise, the base station receives the data through sensor nodes that could be the cause of packet delay and loss. The actor node saves the information of the first actor node from which it receives NIM to use PF process and further forwards NIM with its ID. If an actor gets NIM from multiple actors, then it stores the identity of additional actors in the buffer list. Identity of the saved actors is used in case of topology changes due to mobility or node failure. The detailed process of an actor node that receives NIM is presented in algorithm 3.3.

---

**Algorithm 3.3:** Network Integration and Message Forwarding Process

---

1.  $\{B_s$ : Base Station; NIM: Network integration message; PF: packet forwarding;  $N_{id}$ : Node identity;  $N_a$ : Actor node;  $N_b$ : Node buffer}
  2. **Input:**  $\{B_s, N_a\}$
  3. **Output:**  $\{NIM, PF\}$
  4. **Set** NIM//Network integration message is set to interconnect the entire network
  5. **Set** PF//it saves the destination address of the Base station for packet forwarding
  6. **If**  $PF = B_s$  then//If a base station is saved as the destination address
  7. Decline NIM//If base station is found, then NIM is declined
  8. **else if**  $NIM \in B_s$ //
  9. **Set**  $PF = N_{id}$ // Data forwarding packet is given ID transmitted to Base station
  10. Transmit NIM
  11. **else if**  $NIM \in N_a$  then//It will be considered that NIM is forwarded by an actor node
  12. **end if**
  13. **end else**
  14. **If**  $PF \in N_a$  //If it is validated that data packet is forwarded by an actor node, t
  15.  $N_a$  stores  $N_{id}$  into  $N_b$ //When actor node receives NIM from multiple actors
  16. **else Set**  $PF = N_{id}$ //Data forwarding packet is given ID
  17. Transmit NIM//NIM is transmitted by an actor node
  18. **End if**
  19. **end else**
  20. **else if**  $NIM \in B_s$  then//NIM message is broadcasted by the Base station.
  21. **If**  $PF \in N_a$  then//If data forwarding packets is from an actor
  22. Decline NIM//If actor node is found, then NIM is declined
  23. **else**  $PF = N_{id} \in N_a$ //Each forwarded data packet is given identify from an actor
  24. Transmit NIM//network integration message is transmitted by an actor node
  25. **end else**
  26. **end else**
  27. **end if**
- 

In conclusion, the protocol applies a simple algorithm to process the NIM. The actor node first transmits NIM to its higher hop neighbor actors/sensors. When it gets the first NIM from the higher hop actor/sensor, then it forwards to its lower-hop neighbor actors/sensors to ensure the transmission of NIMs in the entire network. All other NIMs are then dropped by the actor nodes. Therefore, if each actor is ensured to be in the communication range of at least one actor, then the NIMs should not require to be managed at sensor nodes.



### 3.2.3. Priority-Based Routing for Node Failure Avoidance

#### Process

Let us assume that an actor node transmits the number of bits in each packet ' $P_r$ ' that uses encoding mechanism to reduce the complexity of each packet. The sensor nodes monitor the events which check its contribution table that specify the important events. If events are of the significant interest, then the sensor nodes generate the packets and forward to the actor node.

The complete process of monitoring the events and forwarding the routing of the data packets is given in algorithm 3.4.

---

**Algorithm 3.4:** Priority-Based Routing for Node Failure Avoidance Process

---

{  $P_r$ : Packet rate;  $N_a$ : Actor node;  $S_i$ : Significant;  $F_i$ : First interest,  $N_{sc}$ : Sharing capacity of node;  $P$ : Packet;  $N_c$ : remaining output capacity of the node;  $E_r$ : Efficient packet rate;  $F_i$ : Flag of interest;  $F_u$ : Flag of uninterested;  $N_f$ : Node failure }

1. **Input:** {  $P_r$ ,  $P$  }
  2. **Output:** {  $N_f$ ,  $F_i$ ,  $F_u$  }
  3. **If**  $P$  received by  $N_a$  // If actor node receives the packet
  4. **If**  $P \in S_i$  then // If the received packet is of significant interest
  5. Set  $F_i$  // If condition in step-4 is satisfied, received first packet is considered as significant of interest.
  6. Set  $N_{sc} + 1$  & decrease  $N_c$  by  $P_r$  // Sharing capacity of the node is increased
  7. **end if**
  8. **end if**
  9. **If**  $N_c > 0$  then // Determine the power of node Forward  $P$  // Received packet is forwarded
  10. Set  $F_i$  // Flag of interest is set in the buffer
  11. **Else if**  $N_c < 0$  then
  12. Process  $P_r > E_r$  &  $P \in F_u$  // Showing that packet rate is higher than efficient packet rate
  13. Increase  $N_c$  by  $P_r$  // remaining capacity of the actor node is increased
  14. Reduce  $N_f$  // When capacity of the node is increased, less possibility of node failure
  15. **end else**
  16. **end if**
-

## CHAPTER 4 TEST PLAN AND SIMULATION SETUP

There are two processes running throughout the network's deployment and monitoring, the underlying process obtains individual node properties while the second monitors the network consistency. Our goal is to prolong the network lifetime while maintaining the minimum overhead and determining the nodes' failure causes. Efficient actor recovery protocol (EAR) along with RNF, DPCRA, ACR, and ACRA have been implemented and simulated in wireless sensors and actor networks. The simulation is conducted on OMNET++ simulator. The size of the network is up to  $1400 \times 1400$  square meters. Nodes are deployed randomly in the network. The main objective of simulation is to determine the performance of the proposed EAR algorithm in order to ensure the effectiveness of the protocol in presence of QoS parameters, energy efficiency when incident of node failure occurs. In addition, the performance of proposed EAR algorithm is compared with known similar type of schemes such as RNF, DPCRA, ACR, and ACRA.

RNF, DPCRA, ACR, and ACRA are state-of-the-art actor failure recovery algorithms. Detailed description was provided in Chapter 2. The proposed algorithms manages cut-vertex actor failure and recovery while they differ in their selection and objective obtained while recovery as given in Table 4.1. The similar parameters including properties have been used for testing purposes.

Table 4.1. ACR, RNF, DPCRA, ACRA Algorithms analysis

Algorithm	Actor Deployment	Actor Recovery Selection	Aim
<b>ACR</b>	Random	FH selection based on node distance to failed actor	Actor recovery while Reduce total travel distance
<b>RNF</b>	Random	Neighbors containing smallest block	Limit path extend between nodes.
<b>DPCRA</b>	Random	FH handles the recovery based on smallest block of nodes	Minimum number of recovery nodes having their minimum travel distance
<b>ACRA</b>	Randomly	Actor node with high transmission power and higher coverage area is selected, and connectivity is recovered	Recover network from cut-vertex actor failure by limiting actor movement and using sensors as connecting bridges

The simulation scenario consists of 400 nodes including 27–54 actor nodes and 173–356 sensor nodes with a transmission range of 70 m. The sensor/actor nodes are arbitrarily deployed in a mesh fashion. The initial energy of the actor nodes is set 20 J and sensor nodes have 4 J. The bandwidth of the actor node is 4 Mbps, and maximum power consumption of the sensor/actor node for receiving and transmitting the data is set to 13.5 Mw and 15.0 Mw respectively. Sensing and idle modes have 12.4 mW and 0.60 mW, respectively. The total simulation time is 36 min that is enough to determine the effectiveness of the proposed versus stat-of-the-art schemes. However, the simulation time could also be minimized or maximized, and the pause time is 20 s set to warm up the nodes before beginning of the simulation. The results demonstrate presented here are the average of 10 simulation runs. The simulation parameters are summed up in Table 4.2.

The simulation consists of three simulation scenarios that replicate the real wireless sensors and actor wireless sensor network environment. The obtained simulation results are equitably significant and indistinguishable to realistic tentative results.

- Scenario-I: Sensors-to-actor communications. In this scenario, the source nodes are set as the sensors, while the destination nodes are set as actors. The multiple connections are setup with one actor. Thus, the actor node acts as the sink of the communication. There is 86.5%:13.5% ration of sensor-to-actor, and 20% mobile sensor nodes are set. In this scenario, with verity of the network size ;  $1000 \times 1000 \text{ m}^2$ ,  $1200 \times 1200 \text{ m}^2$  and  $1400 \times 1400 \text{ m}^2$ .
- Scenario-II: Actor-to-actor communications. In this scenario, the distance between the two actors is 300 m. The distance is covered by less than 4 hops. This scenario involves multi-hop communication among the actors. In this scenario, a maximum 54 actors are used.
- Scenario-III: Actor-to-sensor communications. In this scenario, communication is done between actors and sensor. The distance between actor and sensor is set to 250 m. The number of hops are 5 and mobility of the nodes is 20% in this communication. There is 13.5%:86.5% ration of sensor-to-actor, and 20% mobile sensor nodes are set. In this scenario, sizes of the network;  $1000 \times 1000 \text{ m}^2$ ,  $1200 \times 1200 \text{ m}^2$  and  $1400 \times 1400 \text{ m}^2$ .

15–70 connections are set up among the nodes. The connections start working randomly during the warm up time. The source and destination nodes are randomly chosen in each scenario. Based on the simulation, we obtained interesting results including the following parameters:

- Number of alive Days.
- Residual Energy.

- Actor/Sensor Recovery time.
- Data Recovery.
- Time Complexity.
- Reliability.

Table 4.2. Summarized simulation parameters for the proposed EAR

Used Parameters	Parameters' Description
Transmission Range	70 m
Sensing Range of sensor node	35 m
Initial energy of a sensor node	4–10 J
Initial energy of an actor node	20–40 J
Sensing Range of an actor node	65 m
Bandwidth of sensor node	50 Kb/S
Bandwidth of an actor node	4 Mb/s
Simulation time	36 min
Maximum nodes	400
Number of sensors	173–346
Static Sensor	80%
Mobile Sensor	20%
Number of actors	27–54
Actor-Sensor Ratio	13.5:86.5
Network Size	$1000 \times 1000 \text{ m}^2$ , $200 \times 1200 \text{ m}^2$ , $1400 \times 1400 \text{ m}^2$
Number of hops in network	18 Maximum
Models	EAR, RNF, DPCRA, ACR, and ACRA
Buffering capacity at sensor and actor	50 & 300 Packets buffering capacity at each sensor and actor respectively
Mobility (Speed of the nodes)	0 m/s to 12 m/s
Data Packet size	512 bytes
Initial pause time	20 s
Rx energy	12.4 mW
Tx energy	0.60 mW
Power Intensity	–14 dBm to 13 dBm.
Total simulation time	36 min

## 4.1. Number of Alive Days

Extended network lifetime has a significant role in improving the performance of the applications. In Figures 4-1, 4.2 and 4.3, the performance of EAR is shown compared to RNF, DPCRA, ACR, and ACRA in form of number of alive nodes. In these experiments, we used the results of three scenarios with different network topologies. In scenario-1, we used  $1200 \times 1200$  m<sup>2</sup> network size with number of maximum 200 nodes that include 27 actor and 173 sensor nodes. Sixty five connections are established to cover the entire scenario. Based on the results, we observed that 24 nodes have 78 alive days in our approach and same days with compared approaches, but when the numbers of nodes increase up to maximum 200 nodes, then the number of alive days changes. Our approach has slight edge over other competing approaches. In our approach, the nodes are alive up to 367 days as compared with other approaches that have less alive days. RNF approach has 323 alive days, and ACRA has 362 alive days. In scenario-1, our proposed EAR has improvement over other competing approaches of 1.36–11.98%. In scenario-2, we used  $1000 \times 1000$  m<sup>2</sup> network size with number of maximum 54 actor nodes with 15 connections. Based on the results, we observed that actors are alive for 643 days in our approach, while other approaches have actor life of 512–592 days. ACR approach has less 512 alive days, and RNF has 588 alive days. In scenario-2, our proposed EAR has improvement over other competing approaches of 7.93–20.37%.

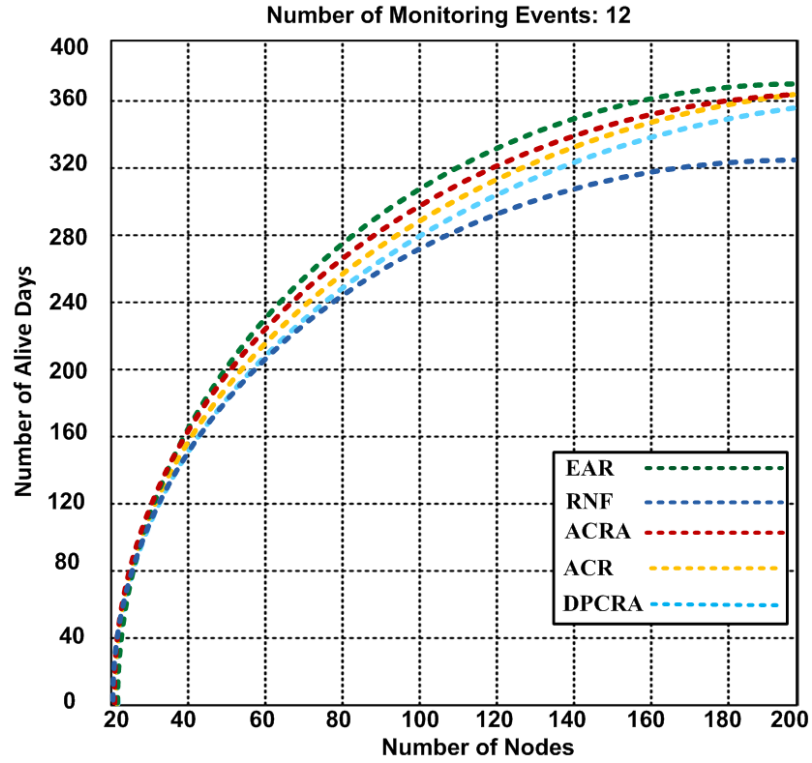


Figure 4.1. Number of alive nodes after completion of 12 events with  $1200 \times 1200 \text{ m}^2$  network topology (Results obtained from Scenario-1)

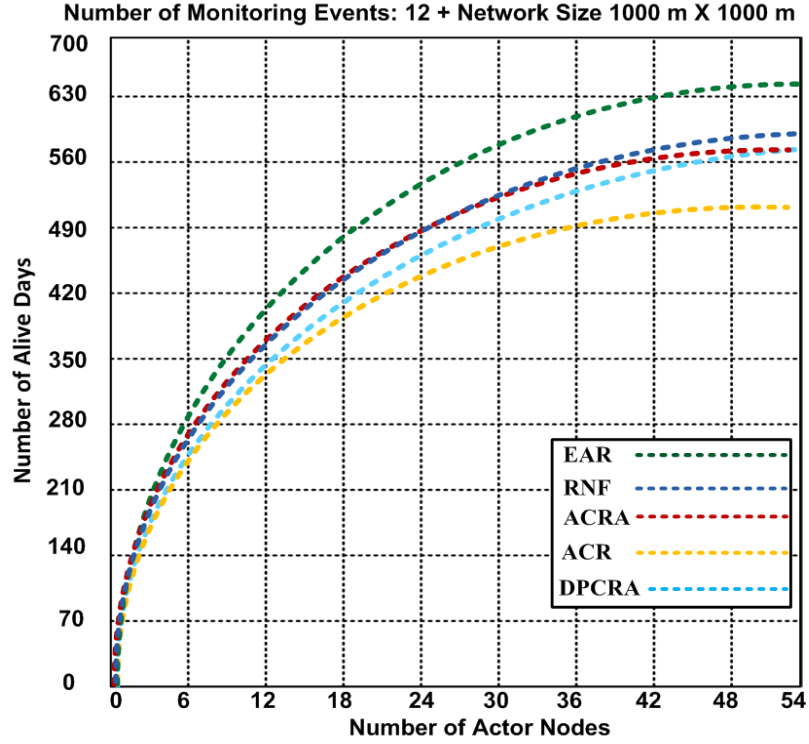


Figure 4.2. Number of alive nodes after completion of 12 events with  $1000 \times 1000 \text{ m}^2$  network topology (Results obtained from Scenario-2).

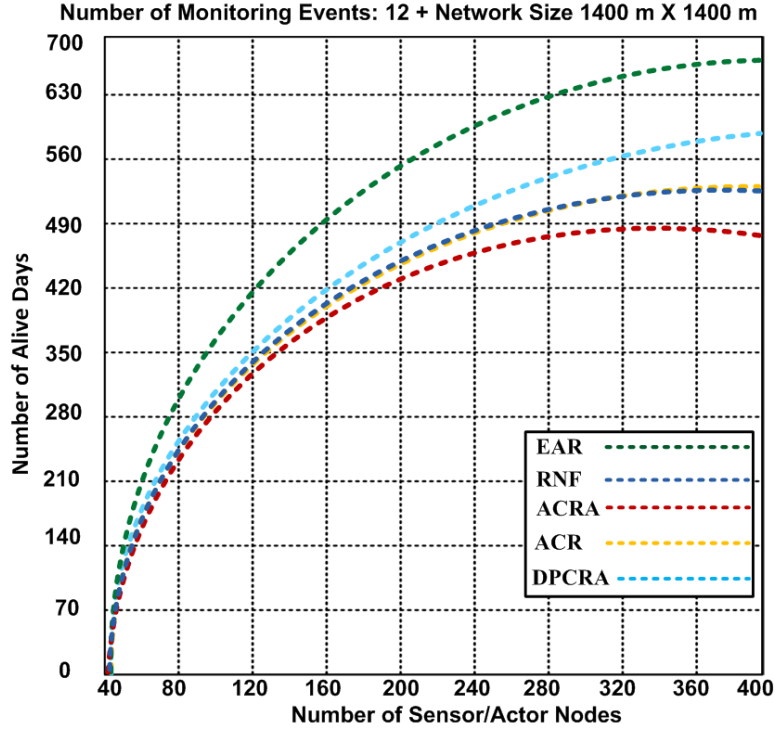


Figure 4.3. Number of alive nodes after completion of 12 events with  $1400 \times 1400 \text{ m}^2$  network topology (Results obtained from Scenario-3)

In scenario-3, we used  $1400 \times 1400 \text{ m}^2$  network size with number of maximum 400 nodes that include 54 actor and 346 sensor nodes. Seventy five connections are established to cover the entire network scenario. Based on the results as in Figure (4.3), we observed that nodes have lifetime 671 days in our proposed EAR approach, whereas other approaches have alive nodes 485–571 days. The performance of the network in ACRA is greatly affected which has minimum of 485 alive days. Therefore, our proposed EAR has improvement over other competing approaches of 14.9–27.71% in scenario-3.

The reason of the better stability of our approach is the usage of the RSSI model that helps to determine the proper distance between sensor-to-sensor, sensor-to-actor and actor-to-actor nodes. Furthermore, the network integration message process connects the



entire network. As a result, bottlenecks are avoided. In case of the node failure, the backup node discovery process is initiated, that does not only improve the throughput, but also extends the nodes' lifetime.

## 4.2. Residual Energy

The residual energy is the remaining energy level of the actor/sensor nodes when concluding the event(s). Here, we discuss an average residual energy level of the actor/sensor nodes after monitoring of different number of events. Figures 4.4, 4.5, and 4.6 compare the residual energy of EAR with those of the RNF, DPCRA, ACR, and ACRA at nine, 18 and 27 events respectively. The sensor/actor nodes have a higher residual energy with the EAR after completion of the events. In this experiment, the results are obtained based on three scenarios:

$$E_{res} = \left[ E_{in} - \left\{ \frac{n * (P_c * E_{red}) + n * (E_{amp} * P_c)}{2E_{ra}} + r^2 (N_n - 1) \right\} + \left\{ \frac{(\Delta C_p * E_{red}) + (E_{amp} * P_c)}{2E_{ra}} \right\} + \left\{ \frac{\{n * (P_c * E_{red}) + n * (E_{amp} * P_c)\}^2}{2E_{ra}} + h^2 (N_n - 1) \right\} \right] \quad (4.1)$$

In Figure 4.4, 70 connections are established for nine events. The actor-to-actor are 12 connections, actor-to-sensor are 32 connections and sensor-to-sensor are 26 connections. Each connection consumes different energy. However, we obtained an average of overall residual energy for the entire network based on the number of connections. We observed in Figure 4.4 that the residual energy of our proposed approach has 8.4 J with nine events as compared with other approaches have residual

energy ranging from 6.9–8.2 J. When we increased the events up to 18 in Figure 4.5, the residual energy of our approach marginally dropped and became 7.4 J and competing approaches have residual energy from 4.2–5.9 J. In Figures 4.4 and 4.5, RNF has less residual energy due to sending additional control message during the actor node failure process. In Figure 4.6, EAR has 6.7 J of residual energy whereas competing approaches have 3.4–5.2 J residual energy. However, ACR has the minimum residual energy due to the decision tree that is incorporated in its reactive routing protocol.

In fact, one of the main reasons that EAR approach has higher residual energy for all events is because the proposed model effectively determines the forwarding capacity of each sensor/actor node prior to transmission which helps to avoid the node failure. The residual energy of sensor/actor is calculated using Equation (4.1) and the description of the used notations is given in Table 4.3.

Table 4.3. Residual energy notations and descriptions

Notations	Descriptions
$n$	Number of the packets
$P_c$	Control packets
$E_{in}$	Initial energy
$E_{res}$	Residual energy
$E_{red}$	Energy consumed for the radio signal
$E_{amp}$	Energy consumed for amplifying the signal
$E_{ra}$	Mean Energy consumed for amplifying the signal and radio
$h$	Number of hops
$N_n$	Number of sensor/actor nodes

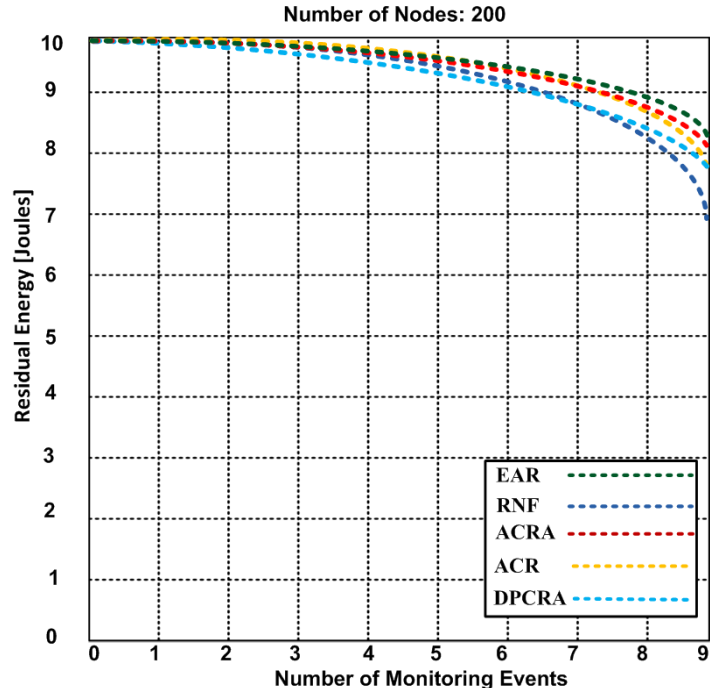


Figure 4.4. The residual energy of EAR and other competing approaches based on 9 event-monitoring.

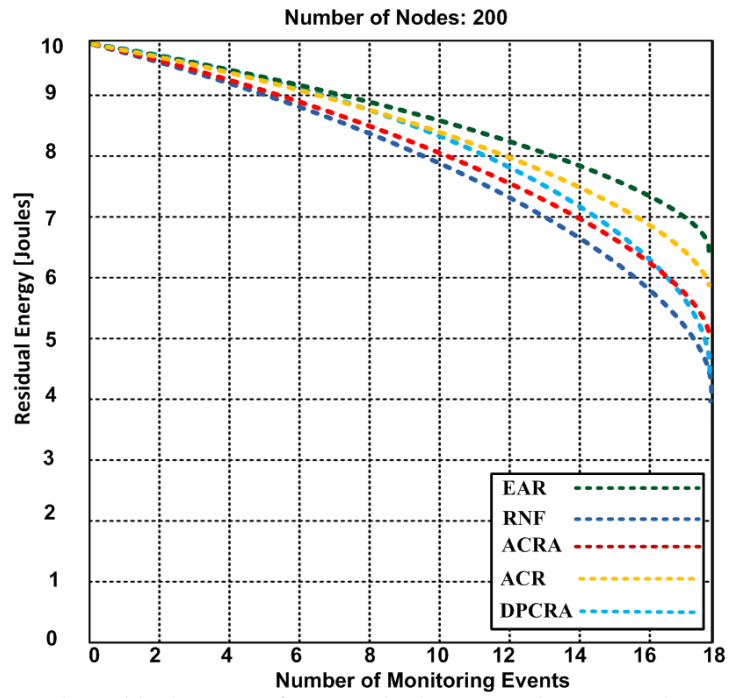


Figure 4.5. The residual energy of EAR and other competing approaches: RNF, DPCRA, ACR, and ACRA based on 18 event-monitoring

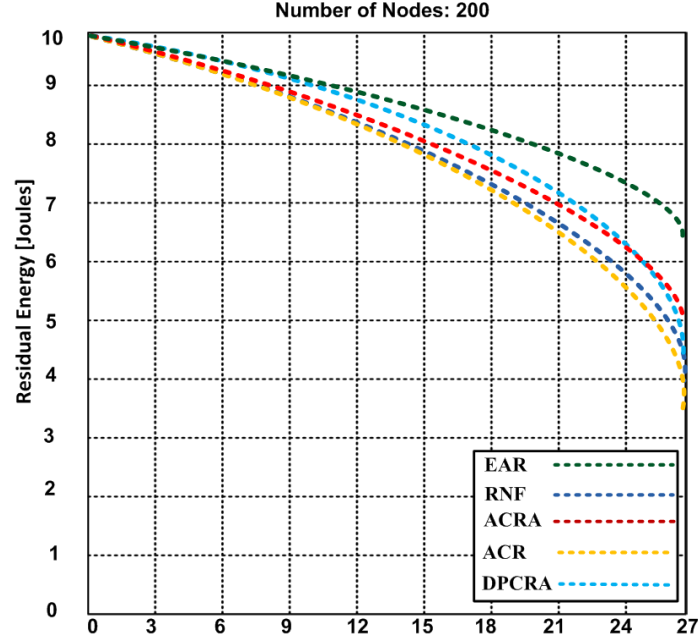


Figure 4.6. The residual energy of EAR and other competing approaches: RNF, DPCRA, ACR, and ACRA based on 27 event-monitoring

### 4.3. Actor/Sensor Recovery Time

The actor recovery time is of high significance for network improvement and running applications on it. When the actor fails, then it is important to initiate the prompt recovery process to avoid the reduction in the network performance. Figures 4.7 and 4.8 show the actor recovery time of the proposed EAR algorithm and other competing approaches: RNF, DPCRA, ACR, and ACRA. In these experiments, we used two different network topologies:  $1200 \times 1200 \text{ m}^2$  and  $1400 \times 1400 \text{ m}^2$ . In Figure 4.7, we used  $1200 \times 1200 \text{ m}^2$  network topology with 48 connections.

Based on the results, we observed that EAR has overall minimum actor/sensor recovery time. We determined an actor recovery time for maximum 27 failure nodes including 11 actors and 16 sensors nodes. At the maximum of 27 failure nodes, EAR has

3.25 s actor/sensor recovery time while other approaches have 3.6–4.7 s. The results show that EAR has 3.19–20% improvement over other competing approaches.

While in Figure 4.8, we used  $1400 \times 1400 \text{ m}^2$  network topology with 60 connections. Based on the results, we observed that EAR has overall minimum actor recovery time. We determined an actor/sensor recovery time for maximum 27 failure nodes including 11 actors and 16 sensors nodes. At the maximum 27 failure nodes, EAR has the same time of 3.25 s as obtained with  $1200 \times 1200 \text{ m}^2$  network topology with 60 connections. It is confirmed that the increase in the network topology does not affect the actor/sensor recovery time while other approaches have 3.62–4.75 s. The result show that EAR has 3.21–20.8% improvement over other competing approaches.

The results confirm the soundness EAR in terms of an actor recovery time due to contention-free forwarding capacity of the nodes. In addition, particular RSSI value is selected for traffic forwarding process that makes the process of actor recovery much easier. As all of the existing approaches either attempt to recover the failure actor or try to reduce the overhead, but our proposed approach reduces the power consumption and delivers the data without contention. Furthermore, it improves the backup node selection process in case of node failure or being disjoint. These characteristics of EAR help reduce the actor recovery time as compared with other approaches.

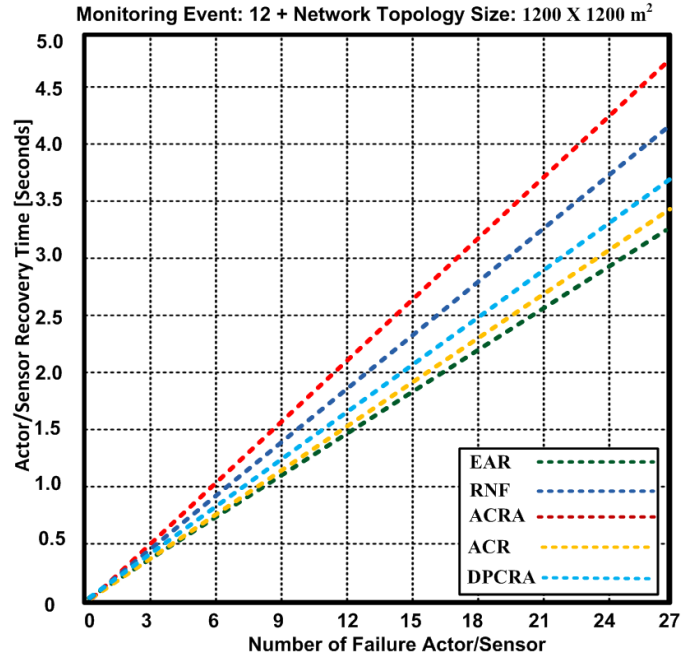


Figure 4.7. Number of failure actors/Sensors and required actor recovery time for EAR, RNF, DPCRA, ACR, and ACRA approaches with 1200 × 1200 m<sup>2</sup>

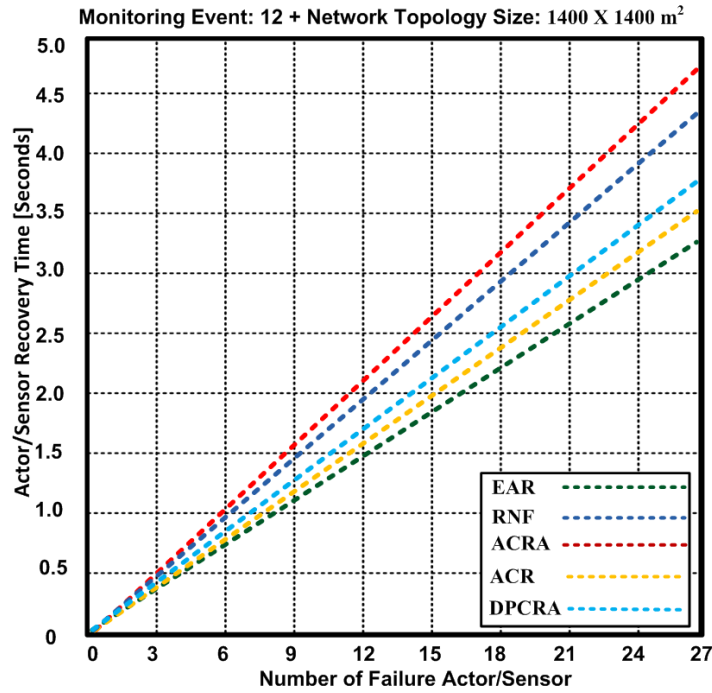


Figure 4.8. Number of failure actors/Sensors and required actor recovery time for EAR, RNF, DPCRA, ACR, and ACRA approaches with 1400 X 1400 m<sup>2</sup>

#### 4.4. Data Recovery

Although data loss is very critical issue, very little information is publically released even when substantial data is lost. A wide variety of failures can cause physical mutilation to the quality of service of the applications. To retain the lost data, the backup recovery approaches perform vital role. However, data recovery methods are not capable enough particularly in wireless sensor and actor networks. In our proposed approach, we have a node monitoring algorithm that monitor the status of the node prior to failure as well as post-failure.

As a result, backup nodes take the responsibility of storing the data. Figures 4.9 and 4.10 show data lost vs. recovered data with  $1400 \times 1400$  m<sup>2</sup> network topology using 72 connections. In Figure 4.9, the total data lost is 15 KB when monitoring 10 events. Based on the results, we observed that EAR lost 15 KB data and recovered 15 KB that shows our scheme of data recovery is fault-tolerant, whereas other approaches also lost the same amount of data, but recovered 11.1–13.5 KB data. It is confirmed that EAR has 10%–26% improvement over other competing approaches.

In Figure 4.10, total data lost is 30 KB with 20 events. As some of the events are not highly critical so that less amount of data is lost with 20 events. As EAR recovers 29.82 KB out of 30 KB that is quite better recovery as compared with other competing approaches. The other competing approaches are greatly affected due to the increase in events so that other approaches have recovery data from 23.8 to 29.1 KB. The least adaptable data recovery algorithm is DPCRA with 20 events. The results also validates that EAR has 2.41%–20.66% improvement over other competing approaches.

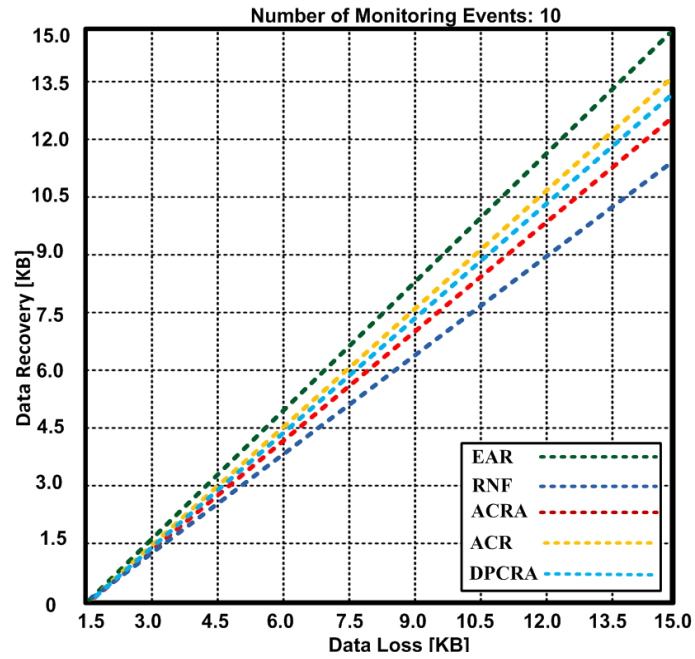


Figure 4.9. Data lost vs. Data recovery during 10 events

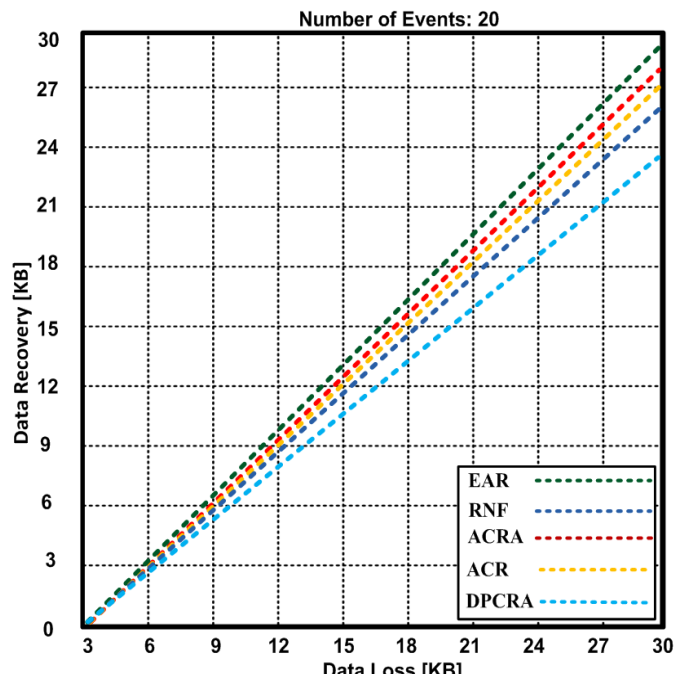


Figure 4.10. Data lost vs. Data recovery during 20 events



## 4.5. Time Complexity

The quality of the running applications depends on the time complexity of algorithm. The time complexity is normally measured by calculating the number of basic operations and time consumed for those operations performed by the algorithm. The algorithm that takes less time improves the performance of the running applications.

Based on the experimental results, we observed that EAR sent more input data in minimum time as compared with other competing algorithms. EAR sent maximum 54 KB input data within 0.065 s, whereas other protocols took 0.067–0.094 s in sending the same amount of data. EAR achieves minimum time because using the single operation for either pre-failure or post-failure recovery processes help reducing the time complexity.

To analyze the time complexity of EAR, let's determine the processes involved in the pre-failure and post failure process. In EAR, each critical actor node has a pre-assigned backup actor node which monitors its critical node. If consecutive heartbeat messages are not received from the critical node, backup actor node handles the recovery process. Let's assume the critical actor is designated as (AC) and its backup node is designated (AB), the following instructions illustrate the pre-failure process:

```
{  
  If (AB.HeartbeatMonitor(AC) == false);  
  Ab.Recover(AC);  
}
```

While post failure process is handled by the backup node (AB). Thus, AB moves towards the failed actor (AC) location in order to recover the network partition. Also, the neighbors of the critical node will use the stored information to communicate with the backup node (AB) in order to restore connectivity.

```

PostFailure(AC, AB)
{
  Move(AB, c);
  Connect(AB, Neighbors(AC))
}

```

Big-O notation is used to illustrate the complexity description of the algorithms.. As EAR and other competing algorithms are recursive by nature. Thus, divide-and-conquer method using Master method is used to determine the complexity of those algorithms. Therefore, time complexity can be calculated using recursive properties given as

$$T(n) = \begin{cases} \theta(1) & \text{If } n = 1 \\ at\left(\frac{n}{b}\right) + \theta(n) & \text{If } n > 1 \end{cases} \quad (4.2)$$

To calculate the time complexity for the previous operation, we assume that each process takes a time  $T(n)$  which is illustrated in Table 4.4. As shown in Table 4.4, each statement takes  $O(1)$ .

All these algorithms are solving the big problems so that we can apply divide-and-conquer method using Master method to determine the complexity of EAR algorithms. The proof is shown in Table 4.4 .

Table 4.4. Mathematical proof of time complexity for EAR

Algorithm	Time complexity
EAR	$T(n) = at \left(\frac{n}{b}\right) + \theta(n)$
	Where problem is divided into two parts with similar size.
	$T(n) = 2t \left(\frac{n}{2}\right) + \theta(1)$
	It is finite because it takes constant values, therefore it can be written as
	$T(n) = 2t \left(\frac{n}{n}\right) + \theta(1)$
	$T(n) = 2t + \theta(1)$
	Where ignore $2t$ ; thus
OR	$T(n) = \theta(1)$
	$T(n) = O(1)$

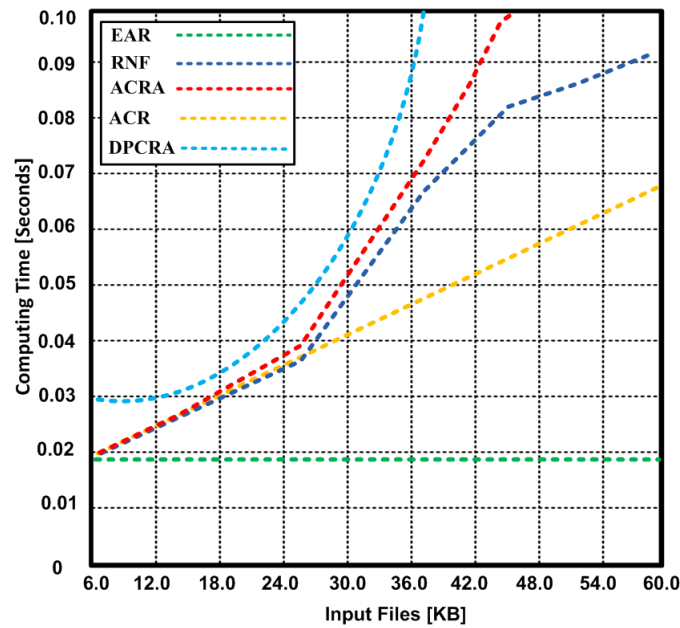


Figure 4.11. Big-O Complexity Chart of EAR, RNF, DPCRA, ACR, ACRA

Figure 4.11 is used to represent the time complexity analysis of EAR compared to RNF, DPCRA, ACR algorithms using Big-O notation. The time complexity of EAR and other competing algorithms is obtained using Big O notation is given in Table 4.5. Therefore, Based on experimental results, EAR shows the significance and improvement of Over competing approaches ,depicted in Table 4.5.

Table 4.5. Time Complexity of EAR, RNF, DPCRA, ACR, and ACRA using O Big operation

Name of Approaches	Excellent ()
EAR	$O(1)$
RNF [10]	$O(\log n)$
ACRA [13]	$O(n \log (n))$
ACR [7]	$O(n)$
DPCRA [11]	$O(n^2)$

#### 4.5. Reliability

Reliability is very essential for networks systems in general. It's one of the most important factors especially in WSN where most of the MAC protocols used lack to offer reliability measurement. An actor failure can degrade network reliability in such uneven environment. Thus, when the actor fails, then it is important to initiate the prompt recovery process to avoid the reduction in the network performance.

If the network works efficiently, and all of the components should operate properly, then the reliability of the network ' $R_{ws}$ ' is obtained as

$$R_{ws} = P_r\{\varepsilon(k(\tau) = 1\} \prod_{k=0}^n P_r\{k(\tau) = 1\} = \prod_{k=0}^n R_k * \omega(t) \quad (4.3)$$

$k(\tau)$ : Functioning probability of either actor/sensor node,  $R_k * \omega(t)$ : Reliability of total components used in the network.

EAR manage to handle actor failure while maintaining QoS in compared to existing actor failure/recovery algorithms. Thus, this total improvement reflects positively over the EAR reliability. Figures 4.12 and 4.13 show the network reliability of the proposed EAR algorithm along with other competing approaches: RNF, DPCRA, ACR, and ACRA. In these experiments, different network topologies are used:  $600 \times 600$  m<sup>2</sup>,  $800 \times 800$  m<sup>2</sup>,  $1200 \times 1200$  m<sup>2</sup> and  $1400 \times 1400$  m<sup>2</sup>. The network consists of 54 actor nodes and 346 sensor nodes. In Figure 4.12, we used  $600 \times 600$  m<sup>2</sup>. EAR shows slight improvement over competing techniques with 0.07-0.2% improvement. While when the size of the deployment area reaches  $1400 \times 1400$  m<sup>2</sup>, EAR shows improvement of 1.1-2.1%. Thus, EAR shows significant improvement over competing algorithms while network deployment area increases, Figure 4.14.

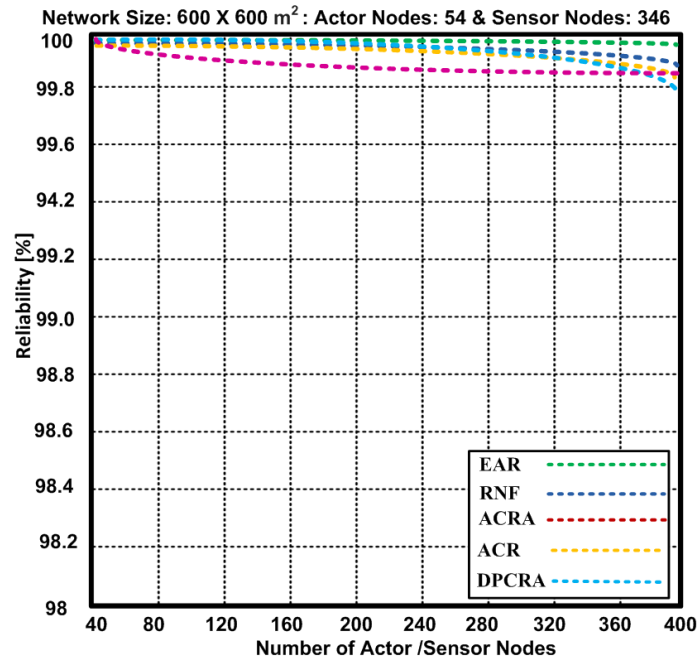


Figure 4.12. Reliability of EAR, RNF, DPCRA, ACR, and ACRA approaches with  $600 \times 600$  m<sup>2</sup>

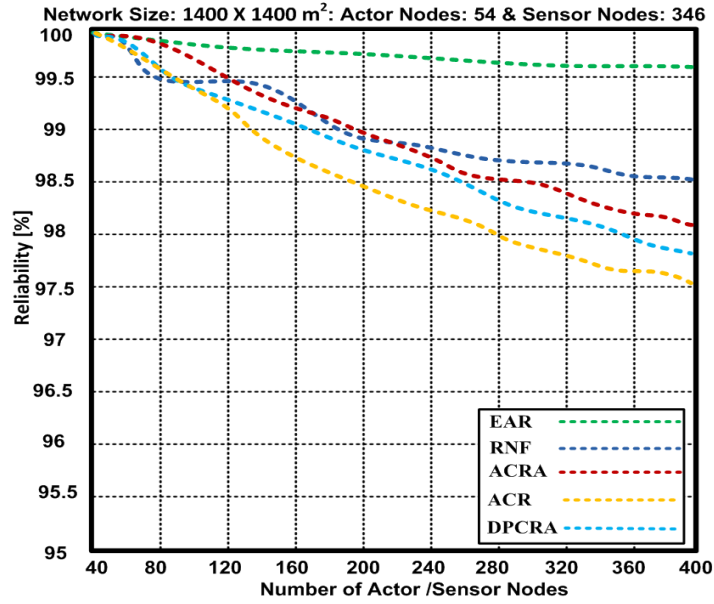


Figure 4.13. Reliability for EAR, RNF, DPCRA, ACR, and ACRA approaches with 1400 X1400 m<sup>2</sup>

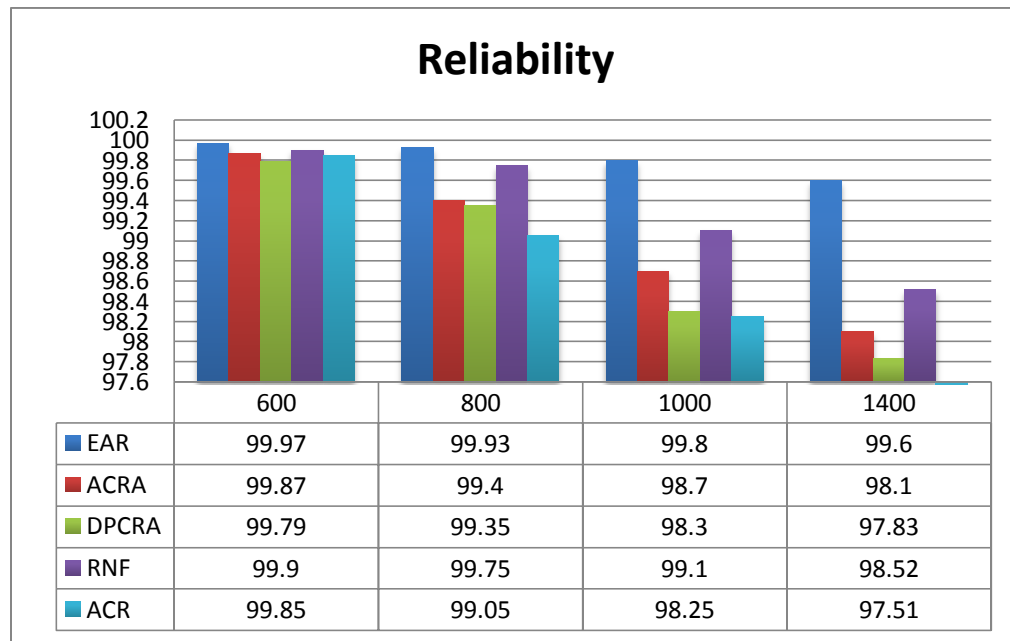


Figure 4.14. Reliability for EAR, RNF, DPCRA, ACR, and ACRA approaches in regards to the size of deployment area

## 4.6. Overall Performance of EAR

EAR model maintains to improve the overall QoS for variety of parameters in compared to competing algorithms in WSN.

In the overall network life time, EAR had an overall improvement of 0.5- 27.71% compared to RNF, DPCRA, ACR, and ACRA. The reason of the better stability of EAR is the usage of the RSSI model that helps to determine the proper distance between sensor-to-sensor, sensor-to-actor and actor-to-actor nodes. Furthermore, the network integration message process connects the entire network. As a result, bottlenecks are avoided. Moreover, in case of the node failure, the recovery process is initiated by the backup node, that does not only improve the throughput, but also reduce total recovery time and extends the nodes' lifetime.

Due to the harsh environment and limited sensors components in WSN, energy is one of the important parameters. Thus, nodes residual energy gain important consideration while implementing EAR. EAR proposed model determines the forwarding capacity of each sensor/actor node prior to transmission which helps to avoid the node failure. Furthermore, the implementation of PRNFA algorithm has effectively assist to monitors and manages nodes power resources. Hence, EAR had a residual energy improvement of 2- 33% compared to RNF, DPCRA, ACR, and ACRA. Indeed, EAR has successfully handle to manage nodes power resources in WSN.

On the other hand, node failure can occur even if the node has sufficient energy. Failure can occur in respond to the node's mobility, environmental change, or topology

change. Thus, WSN should provide robust mechanism for failure detection and recovery. As critical actor failure has a high impact, EAR took into consideration of critical actor failure. Not only by providing an effective recovery mechanism for such actor, but also by preventing the critical actor failure using a variety of algorithms. Even though if failure occurs, failure is handled with minimal impact. Consequently, the results confirm the soundness of EAR in terms of an actor recovery time due to contention-free forwarding capacity of the nodes. In addition, particular RSSI value is selected for traffic forwarding process that makes the process of actor recovery much easier. In contrast to the existing approaches which either attempt to recover the failure actor or try to reduce the overhead, EAR approach handles to manage both criteria in addition to reduce the power consumption and to deliver the data without contention. Furthermore, it improves the backup node selection process in case of node failure or disjoint. These characteristics of EAR help reduce the actor recovery time as compared with other approaches. Moreover, results also validate that EAR has 2.41%–20.66% data recovery improvement over other competing approaches.

This total improvement in EAR reflects positively over the overall network reliability. Results confirm the improvement of EAR with a variety of network sizes, with total improvement of 0.2-2.7%, Figure 2.14.

In conclusion, based on the experimental results, EAR performance shows significant improvement compared to existing approaches. Table 4.6 summarizes EAR improvements in regards to QoS parameters network lifetime, residual energy, actor/sensor recovery time, data recovery, complexity and reliability.



Table 4.6. Improvement of EAR in percentile as compared to competing approaches: RNF, DPCRA, ACR, and ACRA.

Parameters	EAR			RNF			ACRA			ACR			DPCRA			Improvement in EAR		
Number of alive Days (Scenario-1)	367			323			362			361			353			0.5-11%		
Number of alive Days (Scenario-2)	643			588			572			512			572			7.93-20.37%		
Number of alive Days (Scenario-3)	671			531			485			531			571			14.9-27.71%		
Residual Energy in Joules at 9, 18, & 27 (Events)	9	18	27	9	18	27	9	18	27	9	18	27	9	18	27	9	18	27
	8.4	7.4	6.7	6.9	4.5	4.0	8.2	5.2	5.1	7.95	5.8	3.4	7.82	4.6	4.3	2-15%	16-29%	16-33%
Actor Recovery time (Seconds) with 1200 X 1200m <sup>2</sup> network	3.25			4.19			4.7			3.4			3.6			3.19-20%		

Parameters	EAR	RNF	ACRA	ACR	DPCRA	Improvement in EAR
Actor Recovery time (Seconds) with 1400 X 1400m <sup>2</sup> network	3.25	4.31	4.75	3.61	3.76	3.21-20.8%
Data Recovery of 15KB with 10 Events	15	11.1	12.6	13.5	13.1	10-26%
Data Recovery of 30 KB with 20 Events	29.82	26.1	29.1	27.2	23.8	2.41-20.66%
Time Complexity (Seconds)	0.065	0.067	0.082	0.074	0.094	0.2-2.9%
Reliability (600 m <sup>2</sup> )	99.97	99.9	99.87	99.85	99.79	0.07-0.2%
Reliability (800 m <sup>2</sup> )	99.93	99.75	99.40	99.05	99.35	0.2-0.9%
Reliability (1000m)	99.8	99.1	98.7	98.25	98.3	0.7-1.55%
Reliability (1400 m <sup>2</sup> )	99.6	98.52	98.1	97.51	97.83	1.1- 2.1%
Reliability (1400 m <sup>2</sup> )Including malicious nodes	99.3	96.75	97.05	96.6	97.11	2.19-2.7%

## CHAPTER 5 CONCLUSION

The features WSNs are difficult to deploy even though WSNs are known to improve the overall network performance. One short coming of WSNs is that these networks adversely affected by inadequate positioning, power restraints, and routing limitations. To avoid these issues, the sensor and actor nodes should be deployed randomly or at fixed position based on the application requirements. Actor nodes can either be mobile or static. Hence, the node mobility improves the network performance metrics such as coverage, connectivity and lifetime [27]. A number of localization techniques were studied and introduced for WSNs. Some techniques focused on nodes positioning are provided in [147], while a few studies focused on failure node recovery process. Even though, those techniques failed to address QoS features.

The actor node is a major component of WSN. The failure of an actor node can degrade the overall network performance. Furthermore, the failure of an actor node may result in a partitioning of the WSN and may limit event detection and handling. Thus, maintaining the inter-actor connectivity is indispensable in WSNs. The failure of an actor may cause loss of communication or a network disconnect. Thus, actors must communicate with each other to guarantee the entire network coverage and to harmonize their actions for the best response. In case of actor failure, adjacent actors should restore the process or they may be replaced by a backup actor. This solution, however, could be costly and infeasible.

In this dissertation, Efficient Actor Recovery paradigm (EAR) for Wireless Sensor and actor networks is introduced. The main contribution of this work is to provide an efficient actor failure detection and recovery for WSN while maintaining the network QoS. The approach is based on the received signal strength. Unlike previous studies, EAR aims to provide efficient failure detection and recovery mechanism while maintaining the Quality of service. EAR differentiates between critical and non-critical nodes and allocates a suitable backup node from its neighboring node, which is also chosen based on the signal strength and regulates the nodes in its surrounding locality. EAR consists of novel RSSI model that helps apply probability density function for finding the correct location of the actor and sensor nodes. In addition, it shows the relationship between received energy of the wireless signals and transmitted energy including required distance among the actor-sensor nodes. Furthermore, EAR is supported with algorithms for performing the network monitoring process, backup node selection, network integration and message forwarding process, and routing process for actor node to avoid the failure node:

- Node Monitoring and Critical Node Detection (NMCND) algorithm that monitors the activities of the nodes to determine the nodes types and distinguish critical nodes. The NMCND algorithm checks the entire network to determine the critical node during the network life time and pre-assign a backup node for each critical node; so in case the failure of critical node, this node takes place in order to improve and balances the network performance.

- Optimized RSSI model is introduced that selects the different power strengths for each beacon in order to ensure the proper delivery of the beacon to each node. This aims to reduce the latency and estimating the prediction of the node energy-level. As a result, QoS provisioning is maintained and extended the network lifetime.
- Network Integration and Message Forwarding (NIMF) improves the QoS by handling the packet forwarding process. NIMF works to reduce packet forwarding through critical nodes and enhances network lifetime. Moreover, NIMF has the capability to decide the source of the forwarded packet which enhances the packet forwarding flow. Thus, accurate packet forwarding process reduces the latency and bandwidth consumption.
- Priority-Based Routing for Node Failure Avoidance algorithm (PRNFA) handles the routing process. PRNFA analyzes and evaluates the information of the packet in order to route it to the next node. It determines the priority of the forwarded packets. In addition, PRNFA eliminates redundant data prior to routing the packets.

EAR approach has been validated using simulation of OMNET++ and compared with other known approaches: RNF, DPCRA, ACR, and ACRA. The experimental results confirm and given in Table-4.6 that EAR outperforms to other competing approaches in terms of data recovery, number of alive days of the nodes, residual energy, and data loss. In Future, we will focus on extending the efficiency of EAR by implementing security mechanisms while maintaining QoS.

## REFERENCES

- [1] I. Akyildiz and M. Vuran, *Wireless Sensor Networks*: John Wiley & Sons, 2010.
- [2] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz, "Communication and Coordination in Wireless Sensor and Actor Networks," *IEEE Transactions on Mobile Computing*, vol. 6, 2007, pp. 1116-1129.
- [3] J. Kakarla, B. Majhi, and R. Babu Battula, "Comparative Analysis of Routing Protocols in Wireless Sensor–Actor Networks: A Review," *International Journal of Wireless Information Networks*, vol. 22, 2015, pp. 220-239.
- [4] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, 2004, pp. 351-367.
- [5] K. Morita, K. Ozaki, K. Watanabe, N. Hayashibara, T. Enokido, and M. Takizawa, "Sensor-Actuator Communication Protocols in Wireless Networks," in *Network-Based Information Systems: First International Conference, NBIS 2007, Regensburg, Germany, September 3-7, 2007. Proceedings*, T. Enokido, L. Barolli, and M. Takizawa, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 11-19.
- [6] D. J. A. Bijwaard, P. J. M. Havinga, and H. Eertink, "Analysis of Mobility and Sharing of WSNs By IP Applications," *International Journal of Distributed Sensor Networks*, vol. 8, 2011, p. 923594.

- [7] R. Mahjoub and K. Elleithy, "Low Latency Actor Recovery Mechanism for Wireless Sensor and Actor Networks," presented at the Wireless Telecommunications Symposium (WTS), Chicago, IL, 2017, pp. 1-8.
- [8] T. Melodia, D. Pompili, and I. F. Akyildiz, "A Communication Architecture for Mobile Wireless Sensor and Actor Networks " in *3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*, Reston, VA, 2006, pp. 109-118.
- [9] A. Liu, J. Ren, X. Li, Z. Chen, and X. Shen, "Design principles and improvement of cost function based energy aware routing algorithms for wireless sensor networks," *Computer Networks*, vol. 56, pp. 1951-1967, 3 May 2012 2012.
- [10] N.A Vasanthi and S. Annadurai, "Sleep Schedule for Fast and Efficient Control of Parameters in Wireless Sensor-Actor Networks," presented at the International Conference on Communication Systems Software & Middleware, New Delhi, 2006.
- [11] P. Lameski, E. Zdravevski, A. Kulakov, and D. Davcev, "Architecture for Wireless Sensor and Actor Networks Control and Data Acquisition," in *International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, Barcelona, 2011, pp. 1-3.
- [12] M. A. Khan, G. A. Shah, M. Ahsan, and M. Sher, "An efficient and reliable clustering algorithm for wireless sensor actor networks (WSANs)," in *Circuits and Systems (MWSCAS), 2010 53rd IEEE International Midwest Symposium on*, 2010, pp. 332-338.

- [13] S. Khan, F. Khan, and S. A. Khan, "Delay and throughput performance improvement in wireless sensor and actor networks," in *Information Technology: Towards New Smart World (NSITNSW), 5th National Symposium on*, 2015, pp. 1-5.
- [14] R. Kudale and R. Satao, "Green Routing for Wireless Sensor and Actor Networks," *International Journal of Scientific and Research Publications*, vol. 2, 2012, pp. 1-4.
- [15] A. Boukerche and A. Martirosyan, "An Efficient Algorithm for Preserving Events' Temporal Relationships in Wireless Sensor Actor Networks," in *32nd IEEE Conference on Local Computer Networks (LCN 2007)*, 2007, pp. 771-780.
- [16] S. J. Habib, M. Safar, and N. El-Sayed, "Automatic Placement of Actors within Wireless Sensor-Actor Networks," in *Australasian Telecommunication Networks and Applications Conference*, 2008, pp. 224-229.
- [17] A. Martirosyan and A. Boukerche, "Preserving Temporal Relationships of Events for Wireless Sensor Actor Networks," *IEEE Transactions on Computers*, vol. 61, 2012, pp. 1203-1216.
- [18] V. Rafe, H. Momeni, and M. Sharifi, "Energy-Aware Task Allocation in Wireless Sensor Actor Networks," in *Second International Conference on Computer and Electrical Engineering*, 2009, pp. 145-148.
- [19] E. Cayirci, T. Coplu, and O. Emiroglu, "Power aware many to many routing in wireless sensor and actuator networks," in *Proceedings of the Second European Workshop on Wireless Sensor Networks*, 2005, pp. 236-245.



- [20] V. Ranga, M. Dave, and A. K. Verma, "Recovery of Lost Connectivity in Wireless Sensor and Actor Networks using Static Sensors as Bridge Routers," *Procedia Computer Science*, vol. 70, 2015, pp. 491-498.
- [21] V. Ranga, M. Dave, and A. K. Verma, "A Hybrid Timer Based Single Node Failure Recovery Approach for WSAWs," *Wireless Personal Communications*, vol. 77, 2014, pp. 2155-2182.
- [22] W. Li, D. Liu, B. Zhu, X. Wei, W. Xiao, and L. Yang, "SDN Control Model for Intelligent Task Execution in Wireless Sensor and Actor Networks," in *IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016, pp. 1-5.
- [23] H. M. Almasaeid and A. E. Kamal, "Data delivery in fragmented wireless sensor networks using mobile agents," presented at the Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems, Chania, Crete Island, Greece, 2007.
- [24] F. Senel, K. Akkaya, and M. Younis, "An Efficient Mechanism for Establishing Connectivity in Wireless Sensor and Actor Networks," in *IEEE GLOBECOM Global Telecommunications Conference*, 2007, pp. 1129-1133.
- [25] A. Alfadhly, U. Baroudi, and M. Younis, "Least Distance Movement Recovery approach for large scale Wireless sensor and actor networks," *7th International Wireless Communications and Mobile Computing Conference*, 2011, pp. 2058-2063.
- [26] A. A. Abbasi, M. Younis, and K. Akkaya, "Movement-Assisted Connectivity Restoration in Wireless Sensor and Actor Networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, 2009, pp. 1366-1379.

- [27] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 6, pp. 621-655, 2008.
- [28] A. A. Abbasi, M. F. Younis, and U. A. Baroudi, "A least-movement topology repair algorithm for partitioned wireless sensor-actor networks," *Int. J. Sen. Netw.*, vol. 11, 2012, pp. 250-262.
- [29] A. Abbasi, U. Baroudi, M. Younis, and K. Akkaya, "C2AM: an algorithm for application-aware movement-assisted recovery in wireless sensor and actor networks," presented at the Proceedings of the International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, Leipzig, Germany, 2009.
- [30] D.-I. Curiac, "Towards wireless sensor, actuator and robot networks," *J. Netw. Comput. Appl.*, vol. 63, 2016, pp. 14-23.
- [31] T. Melodia, D. Pompili, V. C. Gungo, and I. F. Akyildiz, "Communication and coordination in wireless sensor and actor networks," *IEEE Trans. Mobile Comput.*, vol. 6, 2007, pp. 1116-1129.
- [32] S. Thessler, L. Kooistra, F. Teye, H. Huitu, and A. Bregt, "Geosensors to support crop production: Current applications and user requirements," *Sensors*, vol. 11, 2011, p. 6656.
- [33] N. Sabri, S. A. Aljunid, R. B. Ahmad, M. F. Malik, A. Yahya, R. Kamaruddin, and M. S. Salim, "Cognitive wireless sensor actor networks: An agricultural perspective," in *IEEE Symp. Wireless Technology and Applications (ISWTA)*, Langkawi, 2011, pp. 90-95.

- [34] N. Sabri, S. Aljunid, M. Salim, R. Kamaruddin, B. Ahmad, and M. Malek, "Cognitive wireless sensor actor network: An agricultural perspective," *Int. J. Innov. Comput. Inf. Control*, vol. 10, 2014, pp. 631-658.
- [35] N. Sabri, S. A. Aljunida, R. B. Ahmad, M. F. Malek, A. Yahya, and R. Salim, "Smart prolong fuzzy wireless sensor-actor network for agricultural application," *J. Inf. Sci. Eng.*, 2011, pp. 1-21.
- [36] A. D. Elia, L. Perilli, F. Viola, L. Roffia, F. Antoniazzi, R. Canegallo, and T. S. Cinotti, "A self-powered WSN for energy efficient heat distribution," in *IEEE Sensors Applications Symp. (SAS)*, 2016, pp. 1-6.
- [37] A. Pawlowski, J. Guzman, F. Rodríguez, M. Berenguel, J. Sánchez, and S. Dormido, "Simulation of greenhouse climate monitoring and control with wireless sensor network and event-based control," *Sensors*, vol. 9, 2009, pp. 232-252.
- [38] P. Andrade-Sanchez, F. Pierce, and T. V. Elliott, "Performance assessment of wireless sensor networks in agricultural settings," in *The American Society of Agricultural and Biological Annual Meeting ASAE* 2007.
- [39] L. Ruiz-Garcia, L. Lunadei, P. Barreiro, and J. I. Robla, "A review of wireless sensor technologies and applications in agriculture and food industry: State of the art and current trends," *Sensors (Basel, Switzerland)*, vol. 9, 2009, pp. 4728-4750.
- [40] D. Truong-Xuan and K. Younghan, "Information-centric wireless sensor and actor network in the industrial network," in *Int. Conf. ICT Convergence (ICTC)*, 2013, pp. 1095-1096.

- [41] N.-T. Dinh and Y. Kim, "Potential of information-centric wireless sensor and actor networking," in *Int. Conf. Computing, Management and Telecommunications (ComManTel)*, , 2013, pp. 163-168.
- [42] A. Potsch, "A scalable testbed infrastructure for embedded industrial wireless sensor and actuator networks: Ph.D. forum abstract," presented at the Proc. 15th Int. Conf. Information Processing in Sensor Networks, Vienna, Austria, 2016.
- [43] A. Information Resources Management, Ed., *E-Health and Telemedicine: Concepts, Methodologies, Tools, and Applications (3 Volumes)*. Hershey, PA, USA: IGI Global, 2016.
- [44] L. Suet-Fei, "Wireless sensor actuator network for light monitoring and control application," in *3rd IEEE Consumer Communications and Networking Conference, 2006.*, 2006, pp. 974-978.
- [45] Binodkumar, M. Balaji, R. Dbalaji, and R. Lakshminarayanan, " Wireless Sensor and Actor Network - Security Analysis," *Journal of Wireless Networking and Communications*, vol. 4, 2014, pp. 1-6.
- [46] G. Reese, *Cloud Application Architectures: Building Applications and Infrastructure in the Cloud*: O'Reilly Media, Inc., 2009.
- [47] R. Liu and I. J. Wassell, "Opportunities and challenges of wireless sensor networks using cloud services," presented at the Proceedings of the workshop on Internet of Things and Service Platforms, Tokyo, Japan, 2011.
- [48] R. Mahjoub and K. Elleithy, "Wireless Sensor and Actor Network Applications, Advantages, Direction, and Challenges," in *Wireless Telecommunications Symposium (WTS)*, 2017, pp. 1-6.

- [49] N. Sabri, S. A. Aljunid, R. B. Ahmad, M. F. Malik, A. Yahya, R. Kamaruddin, and M. S. Salim, "Cognitive Wireless Sensor Actor Networks: An Agricultural Perspective," in *IEEE Symposium on Wireless Technology and Applications (ISWTA)*, Langkawi, 2011, pp. 90-95.
- [50] N. Sabri, S. A. Aljunida, R. B. Ahmad, M. F. Malek, A. Yahya, and R. Salim, "Smart Prolong Fuzzy Wireless Sensor-Actor Network for Agricultural Application," *Journal of Information Science and Engineering*, 2011, pp. 1-21.
- [51] A. Cuevas, M. Uruena, R. Cuevas, and R. Romeral, "Modelling data-aggregation in multi-replication data centric storage systems for wireless sensor and actor networks," *IET Communications*, vol. 5, 2011, pp. 1669-1681.
- [52] H. S. Alotaibi, R. Mahjoub, and M. Faezipour, "An intelligent system to control the operation of tunnels and subways: Application in Makkah and Madina traffic and pedestrian tunnels," in *2016 Annual Connecticut Conference on Industrial Electronics, Technology & Automation (CT-IETA)*, 2016, pp. 1-5.
- [53] S. P. Peinado, A. J. V. Rico, J. V. Cañón, J. J. d. I. Heras, and J. M. Mieres, "RFID-based quality and safety building site improvement during execution phase," in *2009 35th Annual Conference of IEEE Industrial Electronics*, 2009, pp. 2659-2664.
- [54] A. Potsch, "A scalable testbed infrastructure for embedded industrial wireless sensor and actuator networks: Ph.D. forum abstract," presented at the Proceedings of the 15th International Conference on Information Processing in Sensor Networks, Vienna, Austria, 2016.

- [55] C. Chiu, Z. Chaczko, and X. Kong, "Design of an Intelligent Health System Using Evolutionary Middleware for Sensor Actor Networks," in *International Conference on Information Science and Applications*, 2012, pp. 1-6.
- [56] A. Lombardi, M. Ferri, G. Rescio, M. Grassi, and P. Malcovati, "Wearable wireless accelerometer with embedded fall-detection logic for multi-sensor ambient assisted living applications," in *IEEE Sensors*, 2009, pp. 1967-1970.
- [57] J. A. Stankovic, Q. Cao, T. Doan, L. Fang, Z. He, R. Kiran, S. Lin, S. Son, R. Stoleru, and A. Wood, "Wireless sensor networks for in-home healthcare: Potential and challenges," Department of Computer Science, University of Virginia 2005.
- [58] A. Milenkovi, #263, C. Otto, and E. Jovanov, "Wireless sensor networks for personal health monitoring: Issues and an implementation," *Comput. Commun.*, vol. 29, 2006, pp. 2521-2533.
- [59] D. Martinez, F. Blanes, J. Simo, and A. Crespo. (2008, Wireless sensor and actuator networks: Charecterization and case study for confined spaces healthcare applications *Int. Multiconference on Computer Science and Information Technology*, 687-693.
- [60] Abhiman Hande, Todd Polk, W. Walker, and D. Bhatia, "Self-Powered Wireless Sensor Networks for Remote Patient Monitoring in Hospitals," *Sensors*, vol. 6, 2006, pp. 1102-1117.
- [61] Y.-J. Chang, C.-H. Chen, L.-F. Lin, R.-P. Han, W.-T. Huang, and G.-C. Lee, "Wireless Sensor Networks for Vital Signs Monitoring: Application in a Nursing

- Home," *International Journal of Distributed Sensor Networks*, vol. 2012, 2012, p. 12.
- [62] S. Haque, M. Rahman, and S. Aziz, "Sensor Anomaly Detection in Wireless Sensor Networks for Healthcare," *Sensors*, vol. 15, 2015, p. 8764.
  - [63] C. J. Rawstorn, N. Gant, I. Warren, N. R. Doughty, N. Lever, K. K. Poppe, and R. Maddison, "Measurement and Data Transmission Validity of a Multi-Biosensor System for Real-Time Remote Exercise Monitoring Among Cardiac Patients," *JMIR Rehabil Assist Technol*, vol. 2, 2015, p. e2.
  - [64] F. González, O. Villegas, D. Ramírez, V. Sánchez, and H. Domínguez, "Smart Multi-Level Tool for Remote Patient Monitoring Based on a Wireless Sensor Network and Mobile Augmented Reality," *Sensors*, vol. 14, 2014, p. 17212.
  - [65] O. Ergul, A. O. Bicen, and O. B. Akan, "Opportunistic reliability for cognitive radio sensor actor networks in smart grid," *Ad Hoc Netw.*, vol. 41, pp. 5-14, 5/1/2016.
  - [66] A. Grilo, A. Casaca, P. Pereira, L. Buttyan, J. Goncalves, and C. Fortunato, "A Wireless Sensor and Actuator Network for improving the electrical power grid dependability," in *8th EURO-NGI Conference on Next Generation Internet (NGI)*, 2012, pp. 71-78.
  - [67] A. M. Grilo, J. Chen, D. M. x00Ed, az, D. Garrido, and A. Casaca, "An integrated WSAN and SCADA system for monitoring a critical infrastructure," *IEEE Trans. Ind. Informat*, vol. 10, 2014, pp. 1755-1764.

- [68] N. Matta, R. Rahim-Amoud, L. Merghem-Boulahia, and A. Jrad, "Enhancing smart grid operation by using a WSN for substation monitoring and control," in *2012 IFIP Wireless Days (WD)*, 2012, pp. 1-6.
- [69] I. L. Santos, L. Pirmez, F. C. Delicato, and L. F. R. d. C. Carmo, "Ensuring energy efficiency of power quality applications in smart grids through a framework based on Wireless Sensor and Actuator Networks," in *International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2015, pp. 763-768.
- [70] C. Vespi and J. Hazen, "Big data bigger opportunities plans and preparedness for the data deluge," *Oracle*, 2012.
- [71] I. Al-Anbagi, M. Erol-Kantarci, and H. T. Mouftah, "Priority- and Delay-Aware Medium Access for Wireless Sensor Networks in the Smart Grid," *IEEE Systems Journal*, vol. 8, 2014, pp. 608-618.
- [72] A. E.-D. Mady, G. Provan, and N. Wei, "Designing cost-efficient wireless sensor/actuator networks for building control systems," in *Proc. 4th ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, Toronto, Ontario, Canada, 2012, pp. 138-144.
- [73] E. de Freitas, T. Heimfarth, A. Vinel, F. Wagner, C. Pereira, and T. Larsson, "Cooperation among Wirelessly Connected Static and Mobile Sensor Nodes for Surveillance Applications," *Sensors*, vol. 13, 2013, p. 12903.
- [74] B. Pannetier, J. Dezert, and G. Sella, "Multiple target tracking with wireless sensor network for ground battlefield surveillance," presented at the 17th International Conference on Information Fusion (FUSION), 2014.



- [75] R. Kudale and R. Satao, "Green routing for wireless sensor and actor networks," *International Journal of Scientific and Research Publications*, vol. 2, 2012, pp. 1-4.
- [76] B. Płaczek and M. Bernas, "Data Suppression Algorithms for Surveillance Applications of Wireless Sensor and Actor Networks," in *Computer Networks: Proceedings 22nd International Conference.*, P. Gaj, A. Kwiecień, and P. Stera, Eds., ed Cham: Springer International Publishing, 2015, pp. 23-32.
- [77] S. Hashish and M. Ahmed, "Efficient wireless sensor network rings overlay for crowd management in Arafat area of Makkah," in *IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, 2015, pp. 1-6.
- [78] D. S. Linthicum, *Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-Step Guide*: Addison-Wesley Professional, 2009.
- [79] M. Yuriyama and T. Kushida, "Sensor-Cloud Infrastructure - Physical Sensor Management with Virtualized Sensors on Cloud Computing," in *13th International Conference on Network-Based Information Systems (NBIS)*, 2010, pp. 1-8.
- [80] R. Liu and I. J. Wassell, "Opportunities and challenges of wireless sensor networks using cloud services," presented at the Proc. Workshop on Internet of Things and Service Platforms, Tokyo, Japan, 2011.
- [81] M. S. Aslam, S. Rea, and D. Pesch, "Provisioning within a WSAN cloud concept," *SIGBED Rev.*, vol. 10, 2013, pp. 48-53.

- [82] R. Porkodi and V. Bhuvaneswari, "The Internet of Things (IoT) Applications and Communication Enabling Technology Standards: An Overview," in *International Conference on Intelligent Computing Applications*, 2014, pp. 324-329.
- [83] S. Deshpande and R. Shankar, "A discrete addressing scheme for wireless sensor networks based internet of things," in *22nd Nat. Conf. Communication (NCC)*, 2016, pp. 1-6.
- [84] F. Janu, x00E, rio, A. Santos, L. Palma, A. Cardoso, and P. Gil, "A distributed multi-agent approach for resilient supervision over a IPv6 WSN infrastructure," in *IEEE International Conference on Industrial Technology (ICIT)*, 2015, pp. 1802-1807.
- [85] P. Zhibo, C. Yuxin, M. E. Johansson, and G. Bag, "Work-in-progress: Industry-friendly and native-IP wireless communications for building automation," in *1st Int. Conf. Industrial Networks and Intelligent Systems (INISCom)*, 2015, pp. 163-167.
- [86] C. Zhu, V. C. M. Leung, L. T. Yang, L. Shu, J. J. P. C. Rodrigues, and X. Li, "Trust assistance in Sensor-Cloud," in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2015, pp. 342-347.
- [87] H. Shibo, L. Xu, C. Jiming, C. Peng, S. Youxian, and D. Simplot-Ryl, "EMD: Energy-efficient P2P message dissemination in delay-tolerant wireless sensor and actor networks," *IEEE J. Sel. Areas Commun.*, vol. 31, 2013, pp. 75-84.
- [88] P. Y. Chen and K. C. Chen, "Optimal control of epidemic information dissemination in mobile ad hoc networks," in *IEEE Global Telecommunications Conference (GLOBECOM 2011)*, 2011, pp. 1-5.

- [89] C. Anagnostopoulos, S. Hadjiefthymiades, and E. Zervas, "Information dissemination between mobile nodes for collaborative context awareness," *IEEE Trans. Mobile Comput.*, vol. 10, 2011, pp. 1710-1725.
- [90] C. Anagnostopoulos, O. Sekkas, and S. Hadjiefthymiades, "An adaptive epidemic information dissemination model for wireless sensor networks," *Pervasive Mob. Comput.*, vol. 8, 2012, pp. 751-763.
- [91] H. Byun and J. So, "Node scheduling control inspired by epidemic theory for data dissemination in wireless sensor-actuator networks with delay constraints," *IEEE Trans. Wireless Commun.*, vol. 15, 2016, pp. 1794-1807.
- [92] A. Klemm, C. Lindemann, and O. P. Waldhorst, "A special-purpose peer-to-peer file sharing system for mobile ad hoc networks," in *IEEE 58th Vehicular Technology Conf.*, vol.4, 2003, pp. 2758-2763.
- [93] X. Wang, J. Wang, and Y. Xu, "Data Dissemination in Wireless Sensor Networks with Network Coding," *EURASIP Journal on Wireless Communications and Networking*, 2010, p. 465915.
- [94] Z. Yang and Y. Liu, "Understanding node localizability of wireless ad hoc and sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, 2012, pp. 1249-1260.
- [95] E. Ruiz-Ibarra, L. Villasenor-Gonzalez, and R. A. Santos, "Design issues and considerations for coordination mechanisms in wireless sensor and actuator networks," in *Electronics, Robotics and Automotive Mechanics Conf. (CERMA)*, 2007, pp. 80-88.
- [96] Beg, m. Turgut, and R. P. Martin, "Using a-priori information to improve the accuracy of indoor dynamic localization," in *Proc. 12th ACM Int. Conf. on*

*Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Tenerife, Canary Islands, Spain, 2009, pp. 396-404.

- [97] S. Schmid and R. Wattenhofer, "Algorithmic models for sensor networks," in *20th International Parallel and Distributed Processing Symposium*, 2006, p. 11 pp.
- [98] N. Tamboli and M. Younis, "Coverage-aware connectivity restoration in mobile sensor networks," presented at the IEEE Int. Conf. on Communications, 2009.
- [99] A. A. Abbasi, K. Akkaya, and M. Younis, "A Distributed Connectivity Restoration Algorithm in Wireless Sensor and Actor Networks," presented at the 32nd IEEE Conference on Local Computer Networks 2007.
- [100] N. Haider, M. Imran, N. M. Saad, and M. A. Zakariya, "Performance analysis of reactive connectivity restoration algorithms for wireless sensor and actor networks," in *IEEE Malaysia Int. Conf. Communications (MICC)*, 2013, pp. 490-495.
- [101] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, 2002, pp. 660-670.
- [102] T. Melodia, D. Pompili, V. C. Gungor, and I. F. Akyildiz, "A distributed coordination framework for wireless sensor and actor networks," in *Proc. 6th ACM Int. Symp. Mobile ad hoc networking and computing*, Urbana-Champaign, IL, 2005, pp. 99-110.

- [103] E. C. H. Ngai, M. R. Lyu, and L. Jiangchuan, "A real-time communication framework for wireless sensor-actuator networks," presented at the IEEE Aerospace Conf., 2006.
- [104] D. Vassiss, G. Kormentzas, and C. Skianis, "Performance evaluation of single and multi-channel actor to actor communication for wireless sensor actor networks," *Ad Hoc Netw.*, vol. 4, 2006, pp. 487-498.
- [105] G. A. Shah, M. Bozyi, O. B. Akan, and B. Baykal, "Real-Time Coordination and Routing in Wireless Sensor and Actor Networks," in *Proceedings of the 6th international conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking*, St. Petersburg, Russia, 2006, pp. 365-383.
- [106] Y. Yibin, X. Xiaoning, and W. Xiaozhen, "Location discovery for three-dimensional sensor-actor networks using alternating combination quadrilateration," in *6th Int. Conf. ITS Telecommunications Proceedings*, 2006, pp. 1021-1024.
- [107] G. Ghidini, C. M. Pinotti, and S. K. Das, "A semi-distributed localization protocol for wireless sensor and actor networks," in *8th IEEE Int. Conf. Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2010, pp. 438-443.
- [108] L. Mo and B. Xu, "Dynamic collaborative processing based on event-driven task allocation in wireless sensor and actuator networks," in *31st Chinese Control Conference (CCC)*, 2012, pp. 6567-6573.

- [109] M. Akbas, M. Erol-Kantarci, and D. Turgut, "Localization for wireless sensor and actor networks with meandering mobility," *IEEE Trans. Comput.*, vol. PP, 2014, pp. 1-1.
- [110] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, 2004, pp. 6-28.
- [111] N.-T. Dinh, M. Vuong, H. Nguyen, and H. Nguyen, "Wireless sensor actor networks and routing performance analysis," in *Int. Workshop on Wireless Ad-hoc Network*, 2005, pp. 23-26.
- [112] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: Research challenges," *Ad Hoc Netw.*, vol. 2, 2004, pp. 351-367.
- [113] C. Chee-Yee and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," in *Proc. of the IEEE*, 2003, pp. 1247-1256.
- [114] J. Bhola and S. Soni, "A study on research issues and challenges in WSAN," in *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2016, pp. 1667-1671.
- [115] B. P. Gerkey and M. J. Mataric, "A market-based formulation of sensor-actuator network coordination," in *Proc. AAAI Spring Symposium on Intelligent Embedded and Distributed Systems*, 2002, pp. 21-26.
- [116] F. Xia, "QoS challenges and opportunities in wireless sensor/actuator networks," *Computing Research Repository*, 2008, p. abs/0806.0.
- [117] K. Hyunbum and J. A. Cobb, "Optimization trade-offs in the design of wireless sensor and actor networks," in *IEEE 37th Conf. Local Computer Networks (LCN)*, 2012, pp. 559-567.

- [118] N. Sabri, S. A. Aljunid, R. B. Ahmad, M. F. Malik, A. Yahya, R. Kamaruddin, and M. S. Salim, "Towards smart wireless sensor actor networks: Design factors and applications," in *IEEE Symp. Industrial Electronics and Applications (ISIEA)*, 2011, pp. 704-708.
- [119] M. Imran, M. Younis, A. M. Said, and H. Hasbullah, "Volunteer-instigated connectivity restoration algorithm for Wireless Sensor and Actor Networks," in *IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS)*, 2010, pp. 679-683.
- [120] A. alfadhly, U. Baroudi, and M. Younis, "Optimal node repositioning for tolerating node failure in wireless sensor actor network," in *25th Biennial Symposium on Communications (QBSC)*, 2010, pp. 67-71.
- [121] N. Haider, M. Imran, N. M. Saad, and M. A. Zakariya, "Performance analysis of reactive connectivity restoration algorithms for wireless sensor and actor networks," in *IEEE Malaysia International Conference on Communications (MICC)*, 2013, pp. 490-495.
- [122] A. Abbasi, M. Younis, and U. Baroudi, "Restoring Connectivity in Wireless Sensor-Actor Networks with Minimal Topology Changes," in *IEEE International Conference on Communications (ICC)*, 2010, pp. 1-5.
- [123] M. Imran, A. M. Said, M. Younis, and H. Hasbullah, "Application-centric connectivity restoration algorithm for wireless sensor and actor networks," presented at the Proceedings of the 6th international conference on Advances in grid and pervasive computing, Oulu, Finland, 2011.

- [124] A. Abbasi, M. Younis, and U. Baroudi, "Restoring Connectivity in Wireless Sensor-Actor Networks with Minimal Topology Changes," presented at the IEEE International Conference on Communications (ICC), 2010.
- [125] A. A. Abbasi, M. F. Younis, and U. A. Baroudi, "Recovering From a Node Failure in Wireless Sensor-Actor Networks With Minimal Topology Changes," *IEEE Transactions on Vehicular Technology*, vol. 62, 2013, pp. 256-271.
- [126] K. Akkaya, F. Senel, A. Thimmapuram, and S. Uludag, "Distributed Recovery from Network Partitioning in Movable Sensor/Actor Networks via Controlled Mobility," *IEEE Transactions on Computers*, vol. 59, 2010, pp. 258-271.
- [127] A. Alfadhly, U. Baroudi, and M. Younis, "Least Distance Movement Recovery approach for large scale Wireless sensor and actor networks," in *7th International Wireless Communications and Mobile Computing Conference*, 2011, pp. 2058-2063.
- [128] J. Du, L. Xie, X. Sun, and R. Zheng, "Application-Oriented Fault Detection and Recovery Algorithm for Wireless Sensor and Actor Networks," *International Journal of Distributed Sensor Networks*, vol. 8, 2012.
- [129] Y. K. Joshi and M. Younis, "Autonomous recovery from multi-node failure in wireless sensor network," in *IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 652-657.
- [130] H. Afzaal and N. A. Zafar, "Formal modeling and algorithm of subnet-based backup assigning in WSAN," in *International Conference on Information and Communication Technologies (ICICT)*, 2015, pp. 1-6.



- [131] A. Alfadhly, U. Baroudi, and M. Younis, "An effective approach for tolerating simultaneous failures in wireless sensor and actor networks," in *Proceedings of the first ACM international workshop on Mission-oriented wireless sensor networking*, Istanbul, Turkey, 2012, pp. 21-26.
- [132] N. Haider, M. Imran, M. Younis, N. Saad, and M. Guizani, "A novel mechanism for restoring actor connected coverage in wireless sensor and actor networks," in *IEEE International Conference on Communications (ICC)*, 2015, pp. 6383-6388.
- [133] R. Hema and M. Amutha, "An Effective Approach for Recovering From Simultaneous Node Failures in Wireless Sensor Networks," *IJRITCC*, vol. 2, Mar-14 2014, pp. 471 - 478.
- [134] Y. K. Joshi and M. Younis, "Autonomous recovery from multi-node failure in Wireless Sensor Network," in *IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 652-657.
- [135] S. Lee, M. Younis, and M. Lee, "Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance," *Ad Hoc Networks*, vol. 24, Part A, 2015, pp. 1-19.
- [136] J. L. V. M. Stanislaus and M. Younis, "Mobile relays based federation of multiple wireless sensor network segments with reduced-latency," in *IEEE International Conference on Communications (ICC)*, 2013, pp. 6407-6411.
- [137] G. Sumalatha , N. Zareena , and C. Gopi Raju, "A Review on Failure Node Recovery Algorithms in Wireless Sensor Actor Networks," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 12(2), June 2014 2014, pp. 94-98.

- [138] Z. Xinyuan and W. Neng, "Optimal restoration approach to handle multiple actors failure in wireless sensor and actor networks," *Wireless Sensor Systems IET*, vol. 4, 2014, pp. 138-145.
- [139] M. Younis, S. Lee, S. Gupta, and K. Fisher, "A Localized Self-Healing Algorithm for Networks of Moveable Sensor Nodes," in *IEEE Global Telecommunications Conference GLOBECOM*, 2008, pp. 1-5.
- [140] S. Zhang, X. Wu, and C. Huang, "Adaptive Topology Reconfiguration from an Actor Failure in Wireless Sensor-Actor Networks," *International Journal of Distributed Sensor Networks*, 2015, p. 11.
- [141] S. Zhang, X. Wu, and H. Wang, "Length-aware topology reconfiguration in wireless sensor-actor networks to recover from an actor failure," in *33rd Chinese Control Conference (CCC)*, 2014, pp. 304-309.
- [142] K. Akkaya and S. Janapala, "Maximizing connected coverage via controlled actor relocation in wireless sensor and actor networks," *Comput. Netw.*, vol. 52, 2008, pp. 2779-2796.
- [143] A. Alfadhly, U. Baroudi, and M. Younis, "An effective approach for tolerating simultaneous failures in wireless sensor and actor networks," in *Proc. 1st ACM Int. Workshop on Mission-oriented wireless sensor networking*, Istanbul, Turkey, 2012, pp. 21-26.
- [144] K. Ozaki, N. Hayashibara, and M. Takizawa, "Coordination protocols of multiple actuator nodes in a multi-actuator/multi-sensor model," presented at the 21st Int. Conf. Advanced Information Networking and Applications Workshops, 2007.

- [145] K. Ozaki, K. Watanabe, S. Itaya, N. Hayashibara, T. Enokido, and M. Takizawa, "A fault-tolerant model for wireless sensor-actor system," in *20th Int. Conf. Advanced Information Networking and Applications*, 2006, p. 5 pp.
- [146] M. Imran, M. Younis, A. M. Said, and H. Hasbullah, "Localized motion-based connectivity restoration algorithms for wireless sensor and actor networks," *Journal of Network and Computer Applications*, vol. 35, 2012, pp. 844-856.
- [147] Y. Kim and C. Jeon, "Actor Positioning with Minimal Movement in Wireless Sensor and Actor Networks," *International Journal of Distributed Sensor Networks*, vol. 2015, 2015, p. 9.
- [148] H. M. Almasaeid and A. E. Kamal, "Data delivery in fragmented wireless sensor networks using mobile agents," presented at the Proc. 10th ACM Symp. on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, Chania, Crete Island, Greece, 2007.
- [149] M. A. Khan, G. A. Shah, M. Ahsan, and M. Sher, "An efficient and reliable clustering algorithm for wireless sensor actor networks (WSANs)," in *53rd IEEE Int. Midwest Symp. on Circuits and Systems (MWSCAS)*, 2010, pp. 332-338.
- [150] F. Senel and M. Younis, "Optimized relay node placement for establishing connectivity in sensor networks," in *2012 IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 512-517.
- [151] B. M. Acharya and S. V. Rao, "Location-based coordinated routing protocol for wireless sensor and actor networks," in *Proceedings of ICCD Intelligent Computing, Communication and Devices*, vol. 2, L. C. Jain, S. Patnaik, and N. Ichalkaranje, Eds., ed New Delhi: Springer India, 2015, pp. 71-77.

- [152] K. Akkaya, F. Senel, and B. McLaughlan, "Clustering of wireless sensor and actor networks based on sensor distribution and connectivity," *J. Parallel Distrib. Comput.*, vol. 69, 2009, pp. 573-587.
- [153] S. Khan, F. Khan, and S. A. Khan, "Delay and throughput performance improvement in wireless sensor and actor networks," in *2015 5th Nat. Symp. Information Technology: Towards New Smart World (NSITNSW)*, 2015, pp. 1-5.
- [154] A. W. Khan, A. H. Abdullah, M. A. Razzaque, and J. I. Bangash, "VGDR: A virtual grid-based dynamic routes adjustment scheme for mobile sink-based wireless sensor networks," *IEEE Sensors J.*, vol. 15, 2015, pp. 526-534.
- [155] A. A. Abbasi, M. F. Younis, and U. A. Baroudi, "Recovering from a node failure in wireless sensor-actor networks with minimal topology changes," *IEEE Trans. Veh. Technol.*, vol. 62, 2013, pp. 256-271.
- [156] A. Abbasi, M. Younis, and U. Baroudi, "Restoring connectivity in wireless sensor-actor networks with minimal topology changes," presented at the IEEE Int. Conf. Communications (ICC), 2010.
- [157] A. Kumar and V. Ranga, "A Cluster-Based Coordination and Communication Framework Using GA for WSANs," in *Proceedings of Ninth International Conference on Wireless Communication and Sensor Networks: WCSN 2013*, R. Maringanti, M. Tiwari, and A. Arora, Eds., ed New Delhi: Springer India, 2014, pp. 111-124.
- [158] M. Y. Sir, I. F. Senturk, E. Sisikoglu, and K. Akkaya, "An optimization-based approach for connecting partitioned mobile sensor/Actuator Networks," in *IEEE*

*Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2011, pp. 525-530.

- [159] I. F. Senturk, K. Akkaya, and F. Senel, "An effective and scalable connectivity restoration heuristic for Mobile Sensor/Actor Networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2012, pp. 518-523.
- [160] I. F. Senturk, S. Yilmaz, and K. Akkaya, "A Game-Theoretic approach to connectivity restoration in Wireless Sensor and Actor Networks," in *IEEE International Conference on Communications (ICC)*, 2012, pp. 7110-7114.
- [161] J. Chen, M. Díaz, L. Llopis, B. Rubio, and J. M. Troya, "A survey on quality of service support in wireless sensor and actor networks: Requirements and challenges in the context of critical infrastructure protection," *Journal of Network and Computer Applications*, vol. 34, 2011, pp. 1225-1239.
- [162] D. Martínez, A. González, F. Blanes, R. Aquino, J. Simo, and A. Crespo, "Formal Specification and Design Techniques for Wireless Sensor and Actuator Networks," *Sensors*, 2011.
- [163] R. Mahjoub and K. Elleithy, "Efficient Actor Recovery Paradigm for Wireless Sensor and Actor Networks," *Sensors*, 2017, p. 858.