Algorithm and Formal Specification of Subnet-Based Communication in WSANs

Hamra Afzaal, Nazir Ahmad Zafar

Abstract — **Recently Wireless Sensor and Actor Networks (WSANs) have drawn increased interest of researchers due to a large number of applications in large scale mission and safety-critical systems. Although significant improvements have been made in WSANs but still there is a need to overcome many challenges. Energy management is a major challenge in WSANs, that's why we present an energy efficient model of Subnet-Based Communication (SBC) algorithm in WSANs. The proposed model pursues WSAN and partitions into subnets. Firstly, in each subnet it proactively distinguishes between critical and non-critical nodes. A critical node selects suitable neighbor as a backup within the subnet for its monitoring and maintaining the inter-actor connectivity within the subnet. Then it chooses one gateway node from each subnet for the continuous communication among subnets and gateway nodes continuously communicate intelligently using Open Shortest Path First (OSPF) routing protocol. Backups are assigned to the gateway nodes to maintain the inter-gateway connectivity among subnets. VDM-SL is introduced as a formal technique for the implementation of SBC algorithm. Validation and verification of the algorithm are done through VDM-SL toolbox which proves its correctness.**

Keywords — **Formal Verification and Validation, Open Shortest Path First, Subnet-Based Communication, WSAN, VDM-SL.**

I. INTRODUCTION

Wireless sensor and actor networks (WSANs) have gained popularity in the world of communications because of smart sensor and actor nodes. Smart sensor nodes cooperate to sense the environment and inform smart actor nodes using wireless links. Smart actors are mobile, autonomous and on the basis of information gathered from the environment able to take decision. Smart actors work together to achieve the planned mission. WSANs have wide range of applications, e.g., fire detection, search and rescue, border protection, battlefield reconnaissance and disaster management etc. The smart actors are more powerful because of long radio used for long range communication, battery, processing than smart sensors. Fig. 1 presents connectivity of subnets in WSAN.

Developing and modelling algorithms for WSANs have raised several research issues which have captured attention of the researchers' community. Deployment, coordination, real-time requirement, security, localization, synchronization, data aggregation and dissemination are some of the critical issues in WSANs. There are several optimization issues in WSAN, for example, scheduling, routing, mobility, latency, energy efficiency, failure recovery, coverage and topology control, etc. This work focuses on communication through energy efficient way. This paper is an extension of the work originally presented in [1].

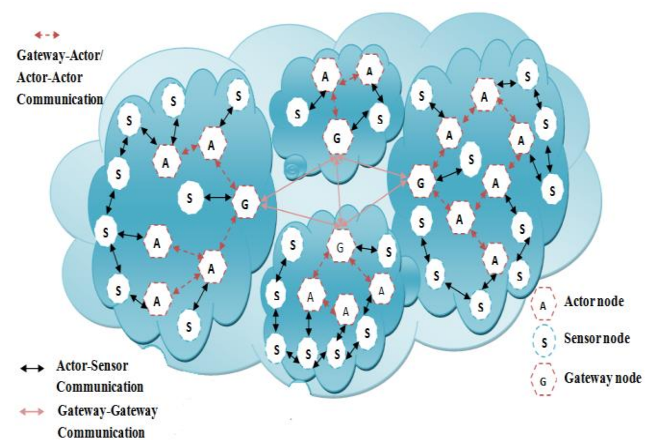

Fig. 1 Connectivity of subnets in WSAN.

In our previous works [2-3] we proposed models of Subnet-Based Backup Assigning and failure recovery for WSANs. This paper describes a model of energy efficient Subnet-Based Communication (SBC) in WSANs which assumes partitioning of WSAN into subnets to save energy and selects one gateway node from each subnet for communication among subnets. In this approach the communication takes place through energy efficient way which increases the network lifetime. In this model each subnet employs smart sensor, actor and gateway nodes. The smart gateway nodes are basically actor nodes. The nodes within the subnet communicate using their radios. The gateway nodes have maximum information of the subnet and share intelligently with other subnets gateway nodes using Open Shortest Path First (OSPF) routing protocol. OSPF is a link state routing protocol used by the routers to share topology information in an intelligent way with the nearest neighbors. As 1-hop neighbors are the most nearest so Internet Protocol packets never travel more than 1-hop. It computes the route using the method based on Dijkstra's algorithm. In the proposed model each gateway node multicasts information of the subnet to one hop gateway nodes of the other subnets and all the subnets remain continuously connected through gateway nodes using OSPF routing protocol. The critical node or gateway node failure due to functioning in unprotected environment affects the connectivity of the network. For this reason critical nodes and gateway nodes are assigned backups to recover the failed node and for the continuous communication in the network.

Hamra Afzaal, Nazir Ahmed Zafar, COMSATS Institute of Information Technology, Sahiwal, Pakistan. Emails: hamraafzaal@hotmail.com.
Manuscript received June 11, 2016; revised on July 9, 2016; accepted on Sep 02, 2016

This model is different from the previous proposed models [2-3] as nodes are smart and communicate in a different way.

Most of the existing work on modeling WSANs is based on simulation techniques which have some disadvantages, for example, these techniques cannot guarantee about the correctness of the model because simulations are performed for limited number of inputs but the formal approaches are effective for proving the correctness of the model. Therefore formal methods based technique is used for the SBC algorithm proposed in this paper for the validation and verification. Formal methods are basically mathematics-based techniques, which have roots in discrete mathematics and are employed for the specification of properties of software and hardware systems [4]. Even though, formal methods are tremendously used for the verification and validation of safety-critical [5-6] and mission-critical systems such as air traffic control, missile control systems and railway interlocking system etc., but still there is a need to pay attention on the use formal techniques for verification and validation of algorithms in WSANs. The objective of our work is to present energy efficient model.

The main contribution in this work is that we have used graph theory for effective modeling of WSANs as graph theory is very useful for modeling of networks. The subnet-based approach is used in this work which is energy efficient as the whole computation and processing takes place at subnet level. We have proposed the Subnet-Based Communication (SBC) algorithm for the intelligent communication among subnets. The graph based model and algorithm is transformed into an equivalent formal specification using formal specification language, i.e., VDM-SL to prove its correctness. Moreover formal methods are used to overcome the limitations of simulations. For example, simulations lacks in proving correctness of an algorithm but are useful only for the performance analysis of non-functional properties of a system. The proposed algorithm is verified, validated and analyzed through existing techniques in VDM-SL Toolbox.

The rest of the paper is organized as follows. In Section II system model and problem statement is described. In Section III the related work is discussed. Section IV describes the proposed algorithm in detail. Section V presents the formal specification of the proposed algorithm. Finally conclusion of the paper is described in Section VI.

II. SYSTEM MODEL AND PROBLEM STATEMENT

SBC is applicative to WSANs as partitioning in WSAN into subnets is assumed in this approach. This approach is energy efficient as it selects one gateway node from each subnet for communication among subnets which requires less computation and less processing. The WSANs employ smart sensor and actor nodes. The smart nodes have a processing unit with a RAM, a program memory, power supply and a wireless transceiver. The smart actor nodes are more powerful as they have more processing ability, memory and wireless communication. The subnets of WSAN also employ smart sensor and actor nodes. Each subnet consists of smart gateway nodes. The smart gateway nodes are actually smart actor nodes, most powerful in terms of battery and are centered for communication among subnets. In our work, we assume smart sensor nodes and actor nodes are randomly placed in an area of interest. The smart actors discover each other after deployment and form inter-actor communication links within the subnets and the gateway nodes form inter-gateway communication links among subnets as shown in Fig. 2. Communication takes place within the subnet through radios.

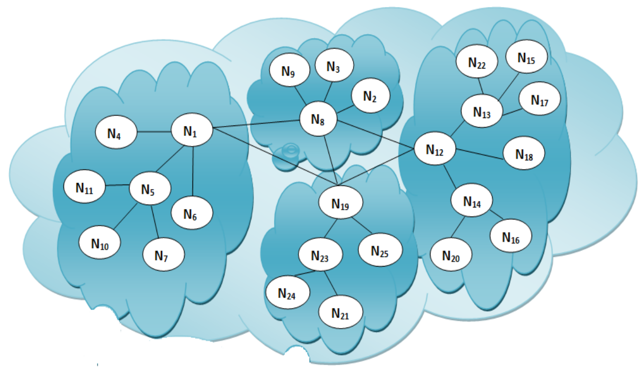

Fig. 2 Representation of Scenario.

The communication range that the smart actor has is the maximum Euclidean distance that its radio can reach. The communication range of a smart actor is assumed to be longer than that of a smart sensor. The gateway node intelligently discovers other subnets gateway nodes using OSPF routing protocol. One gateway node communicate with all the 1-hop neighbor gateway nodes and the 1-hop neighbor gateway nodes communicate with the next 1-hop neighbor gateway nodes and in this way the process is continued. Node failure effect on connectivity of the network depends on the location of a node in the network. Leaf node failure or non-critical node failure within the subnet, e.g., N4, has no effect on the inter-actor connectivity whereas failure of critical node in the subnet, e.g., N5, divides the subnet into disjoint segments. The gateway node failure, e.g., N19, effects inter-gateway connectivity and disconnects the subnet with other subnets. For this reason we propose Subnet-Based Communication (SBC) algorithm which uses localized information to distinguish between critical and non-critical nodes in each subnet. A critical node selects a suitable neighbor as a backup to recover the failed node. A gateway node also selects an appropriate neighbor for backup from the same subnet to recover the failed gateway and for continuous communication among subnets. The communication among subnets takes place through gateway nodes using OSPF routing protocol. The smart sensor, actor and gateway are used interchangeably with sensor, actor and gateway afterward respectively.

III. RELATED WORK

In recent years the interest of researchers have increased in WSANs and because of their increased interest a lot of open research challenges for coordination and communication between sensors and actors are introduced [7]. Maintaining connectivity and energy efficiency are major problems in WSANs. The focus of existing connectivity restoration

 ISSN – 1999-4974

schemes is to maintain inter-actor connectivity [8-9]. These techniques recover inter-actor connectivity in case of critical node failure with minimum overhead. Unlike these techniques our proposed technique focuses on the recovery of both inter-actor and inter-gateway connectivity which requires minimum overhead. For tolerance of fault various models are presented for WSANs [9-10], e.g., the model for fault tolerance is described in [10], which is achieved by redundancy. Routing protocols are basically of three types [11], i.e., Distance-vector routing protocols (e.g., RIP), Link-state routing protocols (e.g., OSPF), hybrid routing protocols (e.g., EIGRP). OSPF was used for the wired networks. Previously it was also used for the wireless sensor networks [11] but we have employed it for the wireless sensor and actor networks. Clustering is an energy efficient approach. The model for intelligent mobility of actors that uses clusters information is described in [12]. Clustering algorithms survey for wireless sensor networks is provided in [13]. Cluster-based routing schemes are discussed in [14]. Unlike these we have presented a subnet-based communication model in WSANs which employs gateway nodes for the intelligent communication among subnets.

Formal methods are basically mathematics-based techniques, have roots in discrete mathematics, and are employed for the specification of properties of software and hardware systems [4] and tremendously used for the safety critical systems [5-6]. Some researchers recently focus on the application of formal methods in WSANs [1-3, 9, 15-17]. In [9, 15], the authors used Z notation for formal verification. Formal validation is done through simulations and analysis is done using Z Eves tool. Z notation is used for analyzing complex systems at abstract level. In [1-3, 16-17], VDM-SL is used for formal specification of the algorithms in WSANs. VDM-SL is also employed for MAHSNs in [18]. Unlike [9, 15], but like [1-3, 16-17], VDM-SL is used in this paper as a formal specification language for the detailed level analysis of WSANs. VDM-SL specification uses data types for static modeling and operations for dynamic modeling. Invariants are defined for static modeling and pre and post conditions for dynamic modeling. VDM-SL uses several constructs e.g., sets, sequences, composite objects and maplets. We have done static modeling in the proposed algorithm using VDM-SL. Formal validation and verification is done using VDM-SL toolbox. Formal verification of algorithm proposed for MAHSNs is described using Z notation in [19] and the algorithm is validated through simulation technique.

IV. SUBNET-BASED COMMUNICATION IN W

Subnet-Based Communication (SBC) algorithm in WSANs which is energy efficient is proposed in this work. It is energy efficient as it assumes partitioning of network into subnets and selects one gateway node from each subnet for the connectivity of subnets which increases the network lifetime. Each gateway actor chooses an appropriate neighbor from the same subnet as a backup to recover the failed gateway node and for the continuous communication among subnets. For intelligent communication among subnets gateway actors are selected from each subnet and gateway nodes use their radios for communication and communicate intelligently using OSPF routing protocol. In all the subnets the nodes use radios of different ranges for communication. In each subnet the SBC proactively distinguishes between critical and non-critical actors using localized information. Each critical actor selects an appropriate neighbor within the subnet as a backup for its monitoring and to recover the failed critical actor. In Fig. 3 pseudo code of the SBC algorithm is presented. The backup for the critical actor is selected in each subnet on the basis of power, neighbor actor status, actor degree and distance. The gateway actor or backup of gateway actor is selected in each subnet on the basis of power, degree and distance. The gateway and its backup selection procedure and backup selection procedure for critical actor node is explained below.

A. Gateway Selection

An actor node is selected as a gateway in a subnet having highest power and highest degree. If more than one candidates appear then least distance actor node from other gateway nodes is selected as a gateway.

B. Backup Selection for Gateway

The gateway node selects the neighbor actor node having highest power, preferably non-critical otherwise critical, and having highest degree as a backup. If more than one candidates appear on the basis of these properties then least distance actor node is selected as a backup.

C. Backup Selection for Actor

The identified critical actors in a subnet are assigned backup. The backup is selected on the basis of highest power, preferably non-critical otherwise critical, and having highest degree as a backup. If more than one candidates appear on the basis of these properties then least distance actor node is selected as a backup.

D. Notations

- S = set of sensors
- A = set of actors
- G = set of gateways
- SN = (S, A, {Gi})
- NetworkPartitionedIntoSubnets(SN, N) returns network in the form of subnets
- GatewaySelection(G, SN) returns gateway from a subnet
- AssignBackup(B, A) returns backup B of A

SBC algorithm assumes partitioning the WSAN into subnets (line 1). For connectivity among subnets the gateway node is selected from each subnet. The gateway node selection procedure is discussed already. Then to monitor the gateway node in a subnet backup is assigned to each gateway node (line 2-5). Backup assigning to gateway nodes is discussed already. In all the subnets a gateway node in one subnet communicate with 1-hop gateway nodes of other subnets using OSPF routing protocol and multicasts the message to all the 1-hop neighbor subnet gateway nodes (line 6-9). In all the subnets all the nodes communicate using radios of different communication ranges (line 10-12). In all the

 ISSN – 1999-4974

subnets critical actors' identification and backup assigning is according to the following procedure. All the actors are described as non-critical initially. Through cut vertex detection procedure, critical and non-critical actors are distinguished. If an actor is critical then it selects a backup (line 13-20). Backup assigning procedure is described already.

```
SBC (N)
1.  NetworkPartitionedIntoSubnets(SN, N)
2.  forall subnets SN_j, for j = 1, 2,…m
3.  GatewaySelection(G_j, SN_j)
4.  AssignBackup(B, G_j)
5.  end for
6.  forall subnets SN_j, for j = 1, 2,…m
7.  Gateway-GatewayCommunication(G_j, SN_j, N)
8.  Every Gateway communicate with other 1-hop Gateways in
    other Subnets and multicast the message to 1-hop Gateways
    using OSPF Routing Protocol
9.  end for
10. forall subnets SN_j, for j = 1, 2,…m
11. Sensors S, actors A and gateways G communicate using radios
    of different ranges in subnets
12. end for
13. forall subnets SN_j, for j = 1, 2,…m
14. CriticalIdentification(A, SN_j)
15. Critical-Node (A) <= FALSE
16. if Neighbors become disconnected without A then
17. Critical-Node (A) <= TRUE
18. AssignBackup (B, A)
19. end if
20. end for
```

Fig. 3 Pseudo code of SBC algorithm.

V. FORMAL SPECIFICATION OF SBC USING VDM-SL

SBC algorithm is implemented using formal specification language, i.e., VDM-SL. The specification is verified, validated and analyzed using VDM-SL toolbox. The WSAN is described as a dynamic graph and is defined as a composite object type *Topology*. It has two fields, i.e., *tnodes* and *tedges*.

```
types

node = token;

TEdge :: en1 : node
         en2 : node
inv mk_TEdge(en1, en2) == en1 <> en2;

Topology :: tnodes : set of node
            tedges : set of TEdge
inv mk_Topology(tnodes, tedges) ==
forall eg in set tedges &
 eg.en1 in set tnodes and
 eg.en2 in set tnodes and
forall nd in set tnodes &
 (exists eg in set tedges & (nd = eg.en1 or nd = eg.en2));
```

Invariants: (1) For every edge there must exists two nodes. (2) For all the nodes that are part of the network there must exists an edge between any two nodes. The isolated node is not the part of the network.

In WSAN, sensor, actor and gateway nodes use radios for communication. Sensors have short range radios than actors. Gateways also have long range radios. Sensor-Sensor Communication is defined by the composite object type *SensorSensorComm* which consists of four fields. The first field *sensor* has type sensor which is the composite object as sensors are used for Sensor-Sensor communication. The second field *tedges* has type set of edge as there should exists an edge between any two sensors that participate in communication. The third field *radio* is used, as sensors communicate using their radios. The fourth field *range* is used as the sensors communicate within the range of radios.

```
Radios = <R1> | <R2> | <R3> | <R4> | <R5> | <R6>;
Range = <SHORT> | <LONG>;
SensorSensorComm :: sensor : Sensor
                    tedges : set of TEdge
                    radio : Radios
                    range : Range
inv mk_SensorSensorComm(sensor, tedges, radio, range) ==
forall s1, s2 in set sensor.tnodes &
 (exists eg in set tedges &
 (s1 = eg.en1 and s2 = eg.en2)) and
  radio = <R1> => range = <SHORT>;
```

Invariants: (1) In Sensor-Sensor communication, for all the sensor nodes that are involved in the communication there exists an edge between them. One sensor node is on one side of the edge and another sensor node is on the other side of the edge. (2) The radio for the Sensor-Sensor communication is R1 that has short range.

Sensor-Actor Communication is defined by the composite object type *SensorActorComm* which consists of five fields. The first field *sensor* has type sensor which is the composite object as sensor is required for Sensor-Actor communication. The second field *actor* has type actor which is the composite object as actor is also required for Sensor-Actor communication. The third field *tedges* has type set of edge as there should exists an edge between sensors and actors that participate in communication. The fourth field *radio* is used as sensors and actors communicate using their radios. The fifth field *range* is used as the sensors and actors communicate within the range of radios.

```
SensorActorComm :: sensor : Sensor
                   actor : Actor
                   tedges : set of TEdge
                   radio : Radios
                   range : Range
inv mk_SensorActorComm(sensor, actor, tedges, radio,
range) ==
forall s in set sensor.tnodes &
 (forall a in set actor.tnodes &
 (exists eg in set tedges &
 (s = eg.en1 and a = eg.en2))) and
 radio = <R2> => range = <SHORT>;
```

Invariants: (1) In Sensor-Actor Communication, for all the sensor and actor nodes that are involved in the

 ISSN – 1999-4974

communication there exists an edge between them. Sensor node is on one side of the edge and actor node is on the other side of the edge. (2) The radio for the Sensor-Actor communication is R2 that has short range.

Sensor-Gateway Communication is defined by the composite object type *SensorGatewayComm* which consists of five fields. The first field *sensor* has type sensor which is the composite object as sensor is required for Sensor-Gateway communication. The second field *gateway* has type gateway which is also the composite object as gateway is also required for Sensor-Gateway communication. The third field *tedges* has type set of edge as there should exists an edge between sensors and gateways that participate in communication. The fourth field *radio* is used as sensors and gateways communicate using their radios. The fifth field *range* is used as the sensors and gateways communicate within the range of radios.

```
SensorGatewayComm ::
 sensor : Sensor
 gateway : Gateway
 tedges : set of TEdge
 radio : Radios
 range : Range
inv mk_SensorGatewayComm(sensor, gateway, tedges,
radio, range) ==
forall s in set sensor.tnodes &
 (forall g in set gateway.tnodes &
 (exists eg in set tedges &
 (s = eg.en1 and g = eg.en2))) and
  radio = <R3> => range = <SHORT>;
```

Invariants: (1) In Sensor-Gateway Communication, for all the sensor and gateway nodes that are involved in the communication there exists an edge between them. Sensor node is on one side of the edge and gateway node is on the other side of the edge. (2) The radio for the Sensor-Gateway communication is R3 that has short range.

Actor-Actor Communication is defined by the composite object type *ActorActorComm* which consists of four fields. The first field *actor* has type actor which is the composite object as actors are required for Actor-Actor communication. The second field *tedges* has type set of edge as there should exists an edge between all actors that participate in communication. The third field *radio* is used as actors communicate using their radios. The forth field *range* is used as actors communicate within the range of radios.

```
ActorActorComm :: actor : Actor
                  tedges : set of TEdge
                  radio : Radios
                  range : Range
inv mk_ActorActorComm(actor, tedges, radio, range) ==
forall a1, a2 in set actor.tnodes &
 (exists eg in set tedges &
 (a1 = eg.en1 and a2 = eg.en2)) and
  radio = <R4> => range = <LONG>;
```

Invariants: (1) In Actor-Actor communication, for all the actor nodes that are involved in the communication there exists an edge between them. One actor node is on one side of the edge and another actor node is on the other side of the edge. (2) The radio for the Actor-Actor communication is R4 that has long range.

Actor-Gateway Communication is defined by the composite object type *ActorGatewayComm* which consists of five fields. The first field *actor* has type actor which is the composite object as actors are required for Actor-Gateway communication. The second field *gateway* has type gateway which is also the composite object as gateways are also required for Actor-Gateway communication. The third field *tedges* has type set of edge as there should exists an edge between actors and gateways that participate in communication. The fourth field *radio* is used as actors and gateways communicate using their radios. The fifth field *range* is used as the actors and gateways communicate within the range of radios.

```
ActorGatewayComm :: actor : Actor
                    gateway : Gateway
                    tedges : set of TEdge
                    radio : Radios
                    range : Range
inv mk_ActorGatewayComm(actor, gateway, tedges, radio,
range) ==
forall a in set actor.tnodes &
 (forall g in set gateway.tnodes &
 (exists eg in set tedges &
 (a = eg.en1 and g = eg.en2))) and
 radio = <R5> => range = <LONG>;
```

Invariants: (1) In Actor-Gateway communication, for all the actor and gateway nodes that are involved in the communication there exists an edge between them. An actor node is on one side of the edge and a gateway node is on the other side of the edge. (2) The radio for the Actor-Gateway communication is R5 that has long range.

Gateway-Gateway Communication is defined by the composite object type *GatewayGatewayComm* which consists of six fields. The first field *gateway* has type gateway which is the composite object as gateways are required for Gateway-Gateway communication. The second field *tedges* has type set of edge as there should be an edge between all gateways that participate in communication. The third field *radio* is used as gateways communicate using their radios. The fourth field *range* is used as gateways communicate within the range of radios. The fifth field *routingprotocol* that is Open Shortest Path First (OSPF) is used for continuous communication among all gateway nodes. The sixth field *data* records the message that is used to communicate.

```
GatewayGatewayComm ::
gateway : Gateway
tedges : set of TEdge
radio : Radios
range : Range
routingprotocol : OpenShortestPathFirst
data : Message
```


ISSN – 1999-4974

```
inv mk_GatewayGatewayComm(gateway, tedges, radio,
range, routingprotocol, data) ==
forall g1,g2 in set gateway.tnodes &
(exists eg in set tedges &
(g1 = eg.en1 and g2 = eg.en2) and
radio = <R6> => range = <LONG> and
{g1} = routingprotocol.onehop and
{g2} = routingprotocol.onehop and
data = routingprotocol.multicast);
```

Invariants: (1) In Gateway-Gateway communication, for all the gateway nodes that are involved in the communication there exists an edge between them. One gateway node is on one side of the edge and another gateway node is on the other side of the edge. (2) The radio for the Gateway-Gateway communication is R6 that has long range. (3) All the gateway nodes communicate using Open Shortest Path First (OSPF) routing protocol with their 1-hop neighbors. (4) The gateway nodes multicast the message among all the 1-hop neighbors.

The Open Shortest Path First (OSPF) routing protocol is used for communication among gateway nodes that are in different subnets. The gateway nodes use OSPF routing protocol for communication with 1-hop neighbors' gateway nodes. It is defined by the composite object type *OpenShortestPathFirst* which consists of two fields, i.e., 1-hop having type set of node as there may be more than one nodes in 1-hop neighbors and multicast has type message as message is multicast among all 1-hop neighbors.

```
Message = token;
OpenShortestPathFirst ::
onehop : set of node
multicast : Message;
```

The proposed algorithm assumes partitioning the WSAN into subnets. Subnet is described as composite object *Subnet.* It consists of twelve fields. The first field is *subnet* having type set of topology. The second field *tnodes* having type set of node as subnet is the collection of nodes. The third field *tedges* having type set of edge as there must exist an edge between any two nodes. The fourth, fifth, sixth and seventh fields, i.e., sensor, actors, gateway, neighbor are defined by the composite object types as the subnet must consists of all these nodes. The remaining fields are *Sensor-Sensor communication*, *Sensor-Actor communication*, *Sensor-Gateway communication*, *Actor-Actor communication*, *Actor-Gateway communication* and *Gateway-Gateway communication*, are also the composite objects which are required for communication in the subnet.

```
Subnet ::
subnet : set of Topology
tnodes : set of node
tedges : set of TEdge
sensor : Sensor
actor : Actor
gateway : Gateway
neighbor : Neighbour
sensorsensorComm : SensorSensorComm
sensoractorComm : SensorActorComm
sensorgatewayComm : SensorGatewayComm
actoractorComm : ActorActorComm
actorgatewayComm : ActorGatewayComm
inv mk_Subnet(subnet, tnodes, tedges, sensor, actor,
gateway, -, sensorsensorComm, sensoractorComm,
sensorgatewayComm, actoractorComm,
actorgatewayComm) ==
forall s in set subnet &
(card s.tnodes >= 2) and
forall s1, s2 in set subnet &
(exists eg in set tedges &
(s1.tnodes = {eg.en1} and
s2.tnodes = {eg.en2})) and
forall s in set sensor.tnodes &
{s} = tnodes and
forall a in set actor.tnodes &
{a} = tnodes and
forall g1, g2 in set gateway.subnets &
(g1.subnet <> g2.subnet) and
forall g1, g2 in set gateway.tnodes &
(g1 <> g2) and
forall s1,s2 in set
sensorsensorComm.sensor.tnodes &
(s1 <> s2) and
forall s in set
sensoractorComm.sensor.tnodes &
(forall a in set
sensoractorComm.actor.tnodes &
(s <> a)) and
forall s in set
sensorgatewayComm.sensor.tnodes &
(forall g in set
sensorgatewayComm.gateway.tnodes &
(s <> g)) and
forall a1 in set
actoractorComm.actor.tnodes &
(forall a2 in set
actoractorComm.actor.tnodes &
(a1 <> a2)) and
forall a in set
actorgatewayComm.actor.tnodes &
(forall g in set
actorgatewayComm.gateway.tnodes &
(a <> g));
```

Invariants: (1) Each subnet must consists of at least two nodes. (2) The subnets must be connected, i.e., there must exists an edge between any two subnets. (3) Every subnet must consists of sensor nodes and actor nodes (4) All gateways must exists in different subnets (5) The gateways should not be the same they must be unique. (6) Sensor-Sensor communication must take place among different sensors. (7) Sensor-Actor Communication must take place among sensor and actor nodes. (8) Sensor-Gateway Communication must take place among sensor and gateway nodes. (9) Actor-Actor Communication must take place among different actor nodes. (9) Actor-Gateway

 ISSN – 1999-4974

Communication must take place among actor and gateway nodes.

The composite object Sensor is defined by thirteen fields. The first field *id* is the sensor identifier. The second field *subnets* is defined, as sensors must exists in a subnet. The third field *tnodes* describes all sensors are the nodes. The fourth field *pwr* describes the status of power of the sensor. The fifth field *sstate* describes the sensor state which may be ok or not ok. The sixth field *information* is used for storing information about events. The seventh field *sneighbours* is required for the communication of sensors with their neighbors. The eighth field *sconnectivity* is required to analyze the connectivity status of sensors. The ninth field *radio* is required as sensors use radios for communication. The tenth field *range* describes that the radio of sensors has certain range in which sensors can communicate. The remaining three fields, i.e., *sensor-sensor communication*, *sensor-actor communication*, *sensor-gateway communication* are defined by the composite objects as sensors communicate with each other as well as with actors and gateway nodes.

```
Pwr = <HIGH> | <LOW>;
SState = <SOK> | <SNOTOK>;
Data=token;
NConnectivity = <NCONNECTED> |
<NDISCONNECTED>;
Sensor ::
id : node
subnets : set of Subnet
tnodes : set of node
pwr : Pwr
sstate : SState
information : set of Data
sneighbours : set of node
sconnectivity : NConnectivity
radio : Radios
range : Range
sensorsensorComm : SensorSensorComm
sensoractorComm : SensorActorComm
sensorgatewayComm : SensorGatewayComm
inv mk_Sensor(-, subnets, tnodes, pwr, sstate, information,
sneighbours, sconnectivity, radio, range,
sensorsensorComm, sensoractorComm,
sensorgatewayComm) ==
forall s in set subnets &
(s.tnodes = tnodes) and
pwr = <LOW> and
sstate = <SOK> <=> information = {} and
sstate = <SNOTOK> <=> information <> {} and
sconnectivity = <NCONNECTED> <=>
sneighbours <>{} and
sconnectivity = <NDISCONNECTED> <=>
sneighbours = {} and
radio = <R1> =>
sensorsensorComm.radio = <R1> and
range = <SHORT> and
radio = <R2> =>
sensoractorComm.radio = <R2> and
range = <SHORT> and
radio = <R3> =>
sensorgatewayComm.radio = <R3> and
range = <SHORT>;
```

Invariants: (1) All subnets must employ sensor nodes. (2) Sensors have low power with respect to actors. (3) Sensors detect events from the environment. If the sensor state is ok then it contains no information. (4) But if the sensor state is not ok then the sensor has information about the detected event. (5) For sensor-sensor communication radio R1 is used which has short range. (6) For sensor-actor communication radio R2 is used which also has short range. (7) For sensor-gateway communication radio R3 is used which also has short range.

The composite object *Actor* combines the types for actor identifier, criticality type, power, backup, nodes, neighbors, connectivity, subnets, radio, range, sensor-actor communication, actor-actor communication and actor-gateway communication. An actor is identified by its identifier. The criticality type is required to know whether the actor is critical or non-critical. An actor must have some power. The backup is required for critical actor. Actors are the set of nodes. As actors communicate with neighbor nodes that's why neighbors information is required. The connectivity status of the actors is stored in connectivity field. The field subnet describes actors must exist in a subnet. The field radio is required as actors use radios for communication. The radio has certain range in which actors can communicate. The remaining three fields i.e., sensor-actor communication, actor-actor communication and actor-gateway communication are the composite object types as actors communicate with each other as well as with sensor and gateway nodes.

```
Criticality = <CRITICAL> | <NONCRITICAL>;
Actor ::
id : node
actype : Criticality
pwr : Pwr
backup : Neighbour
tnodes : set of node
aneighbours : set of Neighbour
aconnectivity : NConnectivity
subnets : set of Subnet
radio : Radios
range : Range
sensoractorComm : SensorActorComm
actoractorComm : ActorActorComm
actorgatewayComm : ActorGatewayComm
inv mk_Actor(id, actype, pwr , backup, tnodes, aneighbours,
nconnectivity, subnets, radio, range, sensoractorComm,
actoractorComm, actorgatewayComm) ==
actype = <CRITICAL> =>
card aneighbours >= 2 and
actype = <NONCRITICAL> =>
card aneighbours < 2 and
pwr = <HIGH> and
backup.type = <ACTOR> and
exists nr in set aneighbours &
```

 ISSN – 1999-4974

```
nr.neighbour = backup.neighbour and
backup.pwr = <HIGH> and
backup.criticality = <NONCRITICAL> and
backup.criticality <> <NONCRITICAL>  =>
backup.criticality = <CRITICAL>  and
forall nr in set aneighbours &
(nr.neighbour in set tnodes and
backup.neighbour in set tnodes and
card backup.neighbours >=
card nr.neighbours) and
backup.distance < nr.distance and
nconnectivity = <NCONNECTED> <=>
aneighbours <> {} and
nconnectivity = <NDISCONNECTED> <=>
aneighbours = {}and
forall s in set subnets &
(s.tnodes = tnodes and
s.gateway.id = id) and
radio = <R2> =>
sensoractorComm.radio = <R2> and
range = <SHORT>  and
radio = <R4> =>
actoractorComm.radio = <R2> and
range = <LONG>  and
radio = <R5> =>
 actorgatewayComm.radio = <R5> and
 range = <LONG> ;
```

Invariants: (1) An actor must have high power. (2) If the actor has two or more than two neighbors then it is critical, non-critical otherwise. (3) The critical actor requires backup which must be an actor. The backup for the critical actor is selected according to the following procedure. (4) The backup actor must have high power. (5) The non-critical actor is preferred for the backup of the critical actor. (6) The critical actor is selected for backup if non-critical actor is not found among neighbors. (7) The selection of the backup must be from its neighbors. (8) Strongly connected and least distance actor is preferred for backup. (9) The connectivity status of the actor node is checked using neighbors' information if it is a non-empty set then the node is connected otherwise disconnected. (10) Every subnet must consists of actor nodes. (11) In every subnet the gateway must be an actor. (12) For sensor-actor communication radio R2 is used which also has short range. (13) For actor-actor communication radio R4 is used which has long range. (14) For actor-gateway communication radio R5 is used which also has long range.

Neighbor is described as a composite object *Neighbour* having six fields. First field is the neighbor identifier. Second field *neighbours* is the set of nodes. Third field type describes the neighbor of the node can be sensor, actor or gateway node. Fourth field *criticality* is needed for assigning backups. The remaining two fields, i.e., power and distance are also required to assign backups.

```
Type = <ACTOR> | <SENSOR> | <GATEWAY>;
Neighbour :: neighbour : node
             neighbours : set of node
             type : Type
             criticality : Criticality
             distance : nat
             pwr : Pwr;
```

The composite object Gateway combines the types for actor, id, power, type, backup, subnets, nodes, neighbors, connectivity, distance, radio, range, sensor-gateway communication, actor-gateway communication and gateway-gateway communication. Gateway node must be actor node. It must have unique identifier. The gateway node must have power. The type field records criticality status of the gateway node. Backup is required for the gateway nodes which is selected from neighbors within the subnet. The field subnets is used as every subnet employs gateways. The field nodes is used as all gateways are the nodes. The field neighbors records the set of neighbors that are connected with the gateway node. The field connectivity is required to record the connectivity status of the gateway node. The field distance has type natural number which records the distance of gateway node from its neighbors. The field radio is required as gateways use radios for communication. The radio has certain range in which gateways can communicate. The next three fields i.e., sensor-gateway communication, actor-gateway communication and gateway-gateway communication are the composite object types as gateways communicate with each other as well as with sensor and actor nodes. The last field routing protocol is the Open Shortest Path First (OSPF) routing protocol which is used for communication among gateway nodes that are in different subnets.

```
Gateway ::
actor : Actor
id : node
pwr : Pwr
gtype : Criticality
backup : Neighbour
subnets : set of Subnet
tnodes : set of node
gneighbours : set of Neighbour
gconnectivity : NConnectivity
distance : nat
radio : Radios
range : Range
sensorgatewayComm : SensorGatewayComm
actorgatewayComm : ActorGatewayComm
gatewaygatewayComm : GatewayGatewayComm
routingprotocol : OpenShortestPathFirst
inv mk_Gateway(-, -, pwr, gtype, backup, -, tnodes,
gneighbours, nconnectivity, distance, radio,
range,sensorgatewayComm, actorgatewayComm,
gatewaygatewayComm, -)
 ==
pwr = <HIGH> and
gtype = <CRITICAL> =>
card gneighbours >= 2 and
gtype = <NONCRITICAL> =>
card gneighbours < 2 and
backup.type = <ACTOR> and
```

 ISSN – 1999-4974

backup.pwr = <HIGH> and
exists nr in set gneighbours &
nr.neighbour = backup.neighbour and
card gneighbours >= card nr.neighbours
and distance <= nr.distance and
forall nr in set gneighbours &
(nr.neighbour in set tnodes and
backup.neighbour in set tnodes and
card backup.neighbours >=
card nr.neighbours) and
backup.distance <= nr.distance and
nconnectivity = <NCONNECTED> <=>
gneighbours <> {} and
nconnectivity = <NDISCONNECTED> <=>
gneighbours = {} and
radio = <R3> =>
sensorgatewayComm.radio = <R3> and
range = <SHORT> and
radio = <R5> =>
actorgatewayComm.radio = <R5> and
range = <LONG> and
radio = <R6> =>
gatewaygatewayComm.radio = <R6> and range = <LONG>;

Invariants: (1) The gateway node must have high power. (2) Gateway node must have high degree. (3) Gateway node must communication first with the other subnet gateway node that has least distance. (4) If the gateway node has two or more than two neighbors then it is critical otherwise non-critical. (5) Backup that is required for the gateway node must be actor node. (6) The backup of gateway node must have high power. (7) The selection of backup of gateway node must be from its neighbors. (8) Backup of gateway must have high degree. (9) The backup node must have least distance from the gateway node as compared to other neighbors. (10) If neighbors of gateway node are non-empty then the gateway node is connected otherwise disconnected. (11) For sensor-gateway communication radio R3 is used which has short range. (12) For actor-gateway communication radio R5 is used which has long range. (13) For gateway-gateway communication radio R5 is used which also has long range.

VI. MODEL ANALYSIS

The model analysis of the proposed algorithm is done through VDM-SL Toolbox. The formal specification of the proposed algorithm has been analyzed by syntax/type checker, C++ Code generator and pretty printer which reported no errors in the specification as shown in Fig. 4. Syntax checker checks that the developed specification is according to the syntax of VDM-SL. The type checker checks the correct use of data types. The C++ code generator generates the equivalent code in C++. The pretty printer facilitates the analysis of inconsistencies. The dynamic checking is also enabled for checking run time errors. The integrity properties of the specification are generated through integrity examiner which are evaluated to true which shows the correctness of the specification. The results are shown in Table I.

TABLE 1. RESULTS

Composite objects, State, Functions and Operations	**Syntax Check**	**Type Check**	**Pretty Printer**
Topology	Y	Y	Y
SensorSensorComm	Y	Y	Y
SensorActorComm	Y	Y	Y
SensorGatewayComm	Y	Y	Y
ActorActorComm	Y	Y	Y
ActorGatewayComm	Y	Y	Y
GatewayGatewayComm	Y	Y	Y
OpenShortestPathFirst	Y	Y	Y
Subnet	Y	Y	Y
Sensor	Y	Y	Y
Actor	Y	Y	Y
Neighbour	Y	Y	Y
Gateway	Y	Y	Y

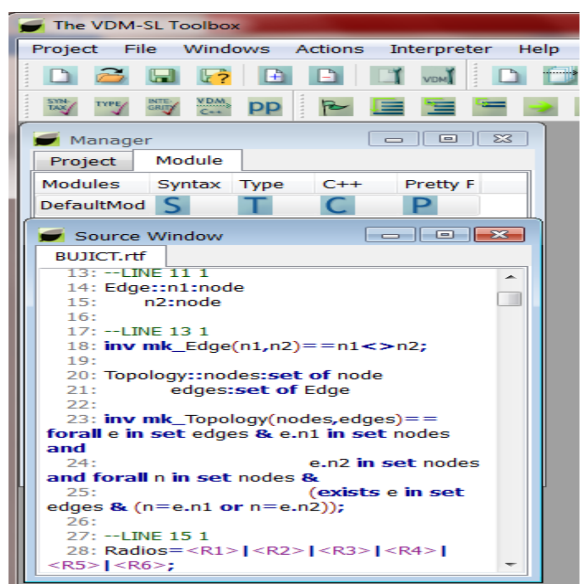

Fig. 4 Model Analysis through VDM-SL Toolbox

VII. CONCLUSION

This paper presents an energy efficient model of Subnet-Based Communication algorithm in WSANs for critical systems. Subnet-Based Communication (SBC) algorithm pursued WSAN and partitions into subnets. It proactively

 ISSN – 1999-4974

distinguishes between critical and non-critical nodes within the subnet. Each critical node designates an appropriate backup selected from the neighbors for maintaining connectivity within the subnet. The nodes communicate using radios which have certain range. From each subnet a gateway node is selected which uses radios for communication and communicate intelligently using OSPF routing protocol. Backup is assigned to each gateway node for the continuous communication among the subnets. This scheme is energy efficient because it reduces the computation overhead as the recovery process takes place in the particular subnet in which the failure is detected. The proposed algorithm is implemented through formal specification language VDM-SL. Validation and verification of the proposed algorithm have been done through VDM-SL toolbox which proves its correctness.

REFERENCES

[1] H. Afzaal, and N.A. Zafar, "Algorithm and Formal Specification of Subnet-Based Communication in WSANs," 1st International Conference on Dependable Embedded Wireless and Sensing Networks (DEWSNET), 2015.

[2] H. Afzaal, and N.A. Zafar, "Formal Modeling and Algorithm of Subnet-Based Backup Assigning in WSAN," 6th IEEE International Conference on Information and Communication Technologies, (ICICT), 2015.

[3] H. Afzaal, and N.A. Zafar, "Formal Localized Reactive Subnet-based Failure Recovery Model for Sparsely Connected Wireless Sensor and Actor Networks, 9th IEEE International Conference on Open Source Systems and Technologies, (ICOSST), pp.64-71, 2015.

[4] M. Conrad, and D. Hötzer, "Selective integration of formal methods in the development of electronic control units," In IEEE Proceedings of Second International Conference on Formal Engineering Methods, 1998.

[5] N.A. Zafar, "Formal Model for Moving Block Railway Interlocking System Based on Un-directed Topology," In IEEE International Conference on Emerging Technologies (ICET'06), 2006.

[6] N.A. Zafar, "Formal Specification and Validation of Railway Network Components using Z Notation," IET software, 3(4): pp.312-320, 2009.

[7] E. Cayirci, "Wireless Sensor and Actuator Network Applications and Challenges," In Autonomous Sensor Networks, Springer, pp.1-15, 2013.

[8] K. Akkaya, A. Thimmapuram, F. Senel, and S. Uludag, "Distributed recovery of actor failures in wireless sensor and actor networks," In IEEE Wireless Communications and Networking Conference (WCNC), 2008.

[9] M. Imran, and N.A. Zafar, "Formal Specification and Validation of a Hybrid Connectivity Restoration Algorithm for Wireless Sensor and Actor networks," Sensors, 12(9): pp.11754-11781, 2012.

[10] K. Ozaki, K. Watanabe, S. Itaya, N. Hayashibara, T. Enokido, and M. Takizawa, "A Fault-tolerant Model of Wireless Sensor-Actor Network," In Ninth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), 2006.

[11] K. Shobha, V. Mamatha Jadhav, "Simulation of a Secure Efficient Dynamic Routing In Wireless Sensor Network," International Journal of Engineering and Advanced Technology, ISSN: 2249 – 8958, 2(5), 2013.

[12] S. S. Krishnakumar, and R. T. Abler, "Intelligent Actor Mobility in Wireless Sensor and Actor Networks," In Wireless Sensor and Actor Networks, Springer US, pp.13-22, 2007.

[13] A. A., Abbasi, and M. Younis, "A Survey on Clustering Algorithms for Wireless Sensor Networks," Computer Communications, 30(14), pp.2826-2841, 2007.

[14] G. Sikander, M. H. Zafar, A. Raza, M. I. Babar, S. A. Mahmud, and G. M. Khan, "A Survey of Cluster-based Routing Schemes for Wireless Sensor Networks," SmartCR, 2013, 3(4), pp.261-275.

[15] M. Imran, N. A. Zafar, M. A. Alnuem, M. S. Aksoy, and A. V. Vasilakos, "Formal Verification and Validation of a Movement Control Actor Relocation Algorithm for Safety–Critical Applications," Wireless Networks, pp.1-19, 2015.

[16] H. Afzaal, M. Imran, and N. A. Zafar, "Implementing Partitioning Detection and Connectivity Restoration in WSAN using VDM-SL," 13th IEEE International Conference on Frontiers of Information Technology, pp.71-76, 2015.

[17] H. Afzaal, and N. A. Zafar, "Formal analysis of subnet-based failure recovery algorithm in WSAN," Complex Adaptive Systems Modeling, 4(27), pp. 1-27, 2016.

[18] S. Riaz, H. Afzaal, M. Imran, N. A. Zafar, and M.S. Aksoy,"Formalizing Mobile Ad Hoc and Sensor Networks using VDM-SL," Procedia Computer Science, 63, pp.148-53, 2015.

[19] M. Alnuem, N.A. Zafar, M. Imran, S. Ullah, and Fayed, "Formal specification and validation of a localized algorithm for segregation of critical/non-critical nodes in MAHSNs," International Journal of Distributed Sensor Networks, 2014.

 ISSN – 1999-4974