EAI Endorsed Transactions
on Internet of Things

Research Article EAI.EU

Algorithm and Formal Model of Recovering Network Connectivity in Battlefield Surveillance

Hamra Afzaal, and Nazir Ahmad Zafar [1,*]

COMSATS Institute of Information Technology, Department of Computer Science,
Sahiwal, Pakistan. hamraafzaal@hotmail.com, nazafar@gmail.com

Abstract

Battlefield surveillance requires mission-critical operations and tasks which can effectively be performed using Wireless Sensor and Actor Networks (WSANs). We have used clustering approach for deployment of WSAN to minimize energy consumption and to limit the processing cost. The adverse environment conditions in battlefield may cause a loss of connectivity but there is a need of continuous flow of information in this application. Therefore an algorithm for network recovering is proposed. Firstly, graphical model of the system is presented using graph theory which is then transformed into a formal model by developing formal specification using Vienna Development Method-Specification Language (VDM-SL). Invariants and pre/post-conditions are defined for its validation. The correctness of the formal specification is assured by an analysis through VDM-SL toolbox.

Keywords: Battlefield Surveillance, Wireless Sensor and Actor Networks, Recover Connectivity, Formal Verification, VDM-SL

Received on 28 March 2017, accepted on 05 May 2017, published on 26 July 2017

doi: 10.4108/eai.26-3-2018.154377

*Corresponding author. Email:nazafar@gmail.com

1. Introduction

In recent years, army has started to take interest to use advance technologies for handling tactical tasks in a battlefield [1-2]. In a battlefield, the environment may change rapidly but the soldiers need to be connected continuously to share information with each other and to take action collectively. Wireless Sensor and Actor Networks (WSANs) are used in various military applications as these are autonomous and are able to handle tactical tasks according to the environment [3]. Therefore WSANs are used in this work for battlefield surveillance. In our previous work, an abstract model for battlefield surveillance was presented [4] but this work is mainly focused on recovering connectivity of the network in a battlefield.

WSANs consist of sensors and actors where sensors have limited capabilities as compared to actors in terms of power, communication and computation. The topology for cluster-based WSAN in a battlefield is shown in Fig. 1. Graphical way is used to describe the topology as it provides better and easier understanding. Graphical models are effective to represent networks as nodes, e.g., sensors or actors, are represented through vertices and communication between them can be modeled through edges.

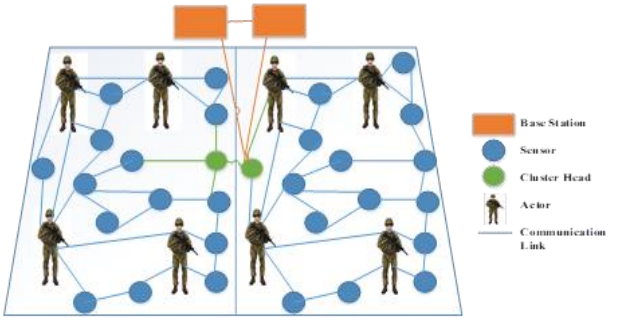

Fig. 1. Representation of a Battlefield Surveillance Scenario.

Most of the existing work on WSANs is based on simulations which have certain limitations. For example, simulations cannot be performed for the whole system and do not have an ability to prove correctness of a system completely. That is why formal methods are used here to minimize the limitations of simulations techniques. Formal methods are mathematical techniques used for developing specification of a system and verifying its properties. In this work, we have used Vienna Development Method-Specification Language (VDM-SL) [5] for specifying proposed algorithm for battlefield termed as Recovering Network Connectivity (RNC). VDM-SL Toolbox [6] is used for proving correctness of the algorithm. Rest of the paper is organized as follow: Section 2 discusses the relevant work in this area. The system model and proposed algorithm are presented in Section 3. Formal specification of the algorithm is described in Section 4. Finally, Section 5 concludes the paper with discussion.

2. Related Work

A group mobility model for battlefield to simulate behaviors of realistic soldiers and leader in the battlefield is presented in [1]. The demand of autonomous sensors in battlefield is increased therefore this leads to the demand of localized and independ-ent energy harvesting capabilities for sensor nodes which are summarized in [7]. An application of battlefield based on MANET is reviewed in [8] and compared with the two emerging commercial MANET scenarios, i.e., campus network and urban vehi-cle grid. Robots can provide better surveillance in battlefield in case of war. That is why different projects have been presented in using robots systems [9-10].

Many methods are proposed to protect link failure in ad hoc networks as the movement of nodes is unpredictable due to dynamic topology. For example, to pre-dict partition and replication of services on the server nodes, a model is described in [11]. Replication of data with dynamic deployment and network partitioning are discussed in [12], which increase overhead of memory and communication band-width. To overcome the limitations of simulations formal methods are used in this work. For various safety and mission critical systems, formal methods have been used [13]. The algorithms are proposed for WSANs using Z notation in [14-16] and VDM-SL [17].

3. System Model and Algorithm

WSANs are deployed in pre-planned way in the form clusters. A cluster head exists in every cluster which collects information from the cluster and reports to the base station. If any node is lost from the cluster and network connectivity is lost, it is recovered from the highest degree neighbor. Then base station decides where and when to deploy a new node to replace the lost node. It is assumed that sensors, actors and a cluster head are mobile nodes while base station is assumed as static. It is assumed that every node has at least two and at most four neighbors.

```
RNC(Sensors, Actors, Cluster Heads).
  ∀ Sensors SRi, for i = 1,2,…,m
  ∀ Actors ARj, for j = 1,2,…,p
   ∃ ClusterHead CLk, for k = 1,2,…,n
  ClusterDeployment(SRi, ARj, CLk)
  Neighbors(SRi) >= 2 and  Neighbors(SRi) <= 4
  Neighbors(ARj) >=2 and  Neighbors(ARj) <= 4
  Neighbors(CLk) >=2 and  Neighbors(CLk) <= 4
  ∀ Base stations BASl, for l = 1,2,…,q
  Neighbors(BASl) >=2 and  Neighbors(BASl) <=4
  ∀ Clusters CTk, for k = 1,2,…,n
  ∀ Sensors SRi, for i = 1,2,…,m
  ∀ Actors ARj, for j = 1,2,…,p
  ∃ ClusterHead CLk, for k = 1,2,…,n
  DetectMines(SRi) and DetectEnemy(SRi)
  DetectAttack(SRi) and TransmitBInfo(ARj, SRi)
  FireGun(ARj) and FBomb(ARj) and FTank(ARj)
   ∃ Base station BASl, for l = 1,2,…,q
  TranBInfo(CLk,ARj)andTranBInfo(BASl, CLk)
  ∃ Sensor SRi, SRj, for i = 1,2,…,m, j = 1,2,…,m
  ConnectivityLoss ← PathLoss(SRi, SRj)
  ∃ Actor ARi, ARj, for i,j = 1,2,…,p
  ConnectivityLoss ← PathLoss(ARi, ARj)
  ∃ Sensor SRi, for i = 1,2,…,m
  ∃ Actor ARj, for j = 1,2,…,p
   ConnectivityLoss ← PathLoss(SRi, ARj)
  ∃ Sensor SRi, for i = 1,2,…,m
  ∃ ClusterHead CLk, for k = 1,2,…,n
  ConnectivityLoss ← PathLoss(SRi, CLk)
  ∃ Actor ARi, for i = 1,2,…,p
  ∃ ClusterHead CLk, for k = 1,2,…,n
  ConnectivityLoss ← PathLoss(ARi, CLk)
  ∃ ClusterHead CLi, CLk, for i, k = 1,2,…,n
  ConnectivityLoss ← PathLoss(CLi, CLk)
  ∃ ClusterHead CLk, for k = 1,2,…,n
  ∃ Base station BASl, for l = 1,2,…,q
  ConnectivityLoss ← PathLoss(CLk, BASl)
  ∃ Base station BASi, BASl, for i, l = 1,2,…,q
  ConnectivityLoss←PathLoss(BASi, BASl)
  RecoverConnectivi-
ty←HighestDegreeNeighbor
  DeployNNode
```

Fig. 2. Pseudo of the Algorithm.

between sensor-cluster head, actor-cluster head, any two cluster heads, cluster head and base station or between any two base stations (lines 26-38). Initially, the connectivity is recovered by the neighbor of highest degree and then a new node is deployed to replace the failed node (lines 39-40).

. (1)

4. Formal Specification

The proposed algorithm is transformed into formal specification using VDM-SL which is used as it is effective to specify the specification at detailed level. In the specification, various notations of VDM-SL are used, for example, sets, composite objects, numeric and quote types. The specification consists of static and dynamic models. The static model includes definitions of composite objects. Invariants are defined on composite objects to define the correct behavior. The dynamic model includes the definitions of state and operations. Invariants are defined on the state and pre/post conditions are defined on the operations for the correct execution.

In the model, four types of nodes are assumed, i.e., sensors, actors, gateways and base stations. The common fields among them are described in the Node object. The field *nidt* describes that every node has a unique identifier. The *npwr* represents the power set of the node. The field *mobility* is used to define mobility status of a node whether static or

mobile. The *npos* is used because every node is deployed on a certain position. The field *c_status* shows the connectivity status of a node in the network. The variable *battleinfo* records whether the information is transmitted or received. The field *cngbrs* represents the connected neighbor nodes.

types

```
Idt = token;
NPwr = <<HIGH>> | <<LOW>>; Mobility =
<<STATIC>> | <<MOBILE>>;
NPos :: xcor : int
        ycor : int;
CStatus = <<CONNECTED>> |
<<DISCONNECTED>>;
BattleInfo = <<BITRANSMITTED>> |
<<BIRECEIVED>>;

Node :: nidt : Idt
        npwr : NPwr
        mobility : Mobility
        npos : NPos
        c_status : CStatus
        battleinfo : BattleInfo
        cngbrs : set of Node;
```

Connectivity between the nodes is defined through communication links. Any two nodes connected by a link can communicate with each other. A node does not have connectivity with itself which represents that the network does not have a loop.

```
ComLinks = set of ComLink
inv comlinks == forall mk_(nidt1, nidt2)
in set comlinks &
 mk_(nidt2, nidt1) in set comlinks;
ComLink = Idt * Idt
inv comlink == let mk_(idt1, idt2) =
comlink in idt1 <> idt2;
```

A sensor node has three fields. The first one *srnode* describes that a sensor is assumed as a node. The field *sdetect* represents that sensors are deployed to detect mines, enemy and any other attacks. The variable *sc_ngbrs* is required to keep record of connected neighbor nodes.

```
SDetect = <<MINES>> | <<ENEMY>> |
<<ATTACK>>;
Sensor :: srnode : Node
          sdetect : SDetect
          sc_ngbr : set of Node
inv mk_Sensor(srnode, sdetect, sc_ngbr)
==
srnode.npwr = <<LOW>> and
srnode.mobility = <<MOBILE>> and
sdetect = <<MINES>> and sdetect =
<<ENEMY>> and
sdetect = <<ATTACK>> and
srnode.c_status = <<CONNECTED>> <=>
sc_ngbr <> {} and
srnode.c_status = <<DISCONNECTED>> <=>
sc_ngbr = {};
```

Invariants: (1) A low power and mobile sensor is deployed in the battlefield. (2) A sensor node detects mines, enemy or any other attack. (3) The connectivity status of a sensor node is connected if it has neighbors otherwise it is disconnected.
In formal specification of actor, *arnode* describes that an actor is assumed as a node. The field *paction* defines actions performed by actors. The field *ac_ngbr* is used to keep record of connected neighbor nodes.

```
Tank = token; PAction = <<FIREGUN>> |
<<FIREBOMB>> | <<FTANK>>;
Actor :: arnode : Node
         paction : PAction
         ac_ngbr : set of Node
inv mk_Actor(arnode, paction, ac_ngbr)==
  arnode.npwr = <<HIGH>> and
arnode.mobility = <<MOBILE>> and
  paction = <<FIREGUN>>  and paction =
<<FIREBOMB>> and
  paction = <<FTANK>> and
  arnode.c_status = <<CONNECTED>> <=>
ac_ngbr <> {} and
  arnode.c_status = <<DISCONNECTED>> <=>
ac_ngbr = {};
```

Invariants: (1) A high power mobile actor is deployed in the battlefield. (2) Actor performs actions like firing gun, bomb and functioning tank. (3) The connectivity status of an actor is connected if and only if it has neighbors otherwise disconnected.
In the definition of cluster head, *crnode* shows that a cluster head is assumed as a node. The field *cmonitor* represents that a cluster head monitors sensors and actors. The *cc_ngbr* shows the set of connected neighbor nodes.

```
CMonitor :: sensors : set of Sensor
            actors : set of Actor;
CLTHead :: crnode : Node
           cmonitor : CMonitor
           cc_ngbr : set of Node
inv mk_CLTHead(crnode, -, cc_ngbr)==
  crnode.npwr = <<HIGH>> and
crnode.mobility = <<MOBILE>> and
  crnode.c_status = <<CONNECTED>> <=>
cc_ngbr <> {} and
  crnode.c_status = <<DISCONNECTED>> <=>
cc_ngbr = {};
```

Invariants: (1) A high power and mobile cluster head is deployed in every cluster. (2) The connectivity status of a cluster head is connected if and only if it has neighbors otherwise it is disconnected.
A cluster consists of cluster head, sensors and actors which communicate through wireless links. The formal specification is given below including invariants.

```
Cluster :: clthead : CLTHead
```

```
  sensors : set of Sensor
  actors : set of Actor
  comlinks : ComLinks
inv mk_Cluster(clthead, sensors, actors,
comlinks)==
forall sr1, sr2 in set sensors & exists
col in set comlinks &
 col = mk_(sr1.srnode.nidt,
sr2.srnode.nidt) and
forall acr1, acr2 in set actors & exists
col in set comlinks &
 col = mk_(acr1.arnode.nidt,
acr2.arnode.nidt) and
exists sr in set sensors & exists acr in
set actors &
 exists col in set comlinks &
 col = mk_(sr.srnode.nidt,
acr.arnode.nidt) and
 col = mk_(clthead.crnode.nidt,
sr.srnode.nidt) and
 col =
mk_(clthead.crnode.nidt,acr.arnode.nidt)
and
forall sr in set sensors &
 card sr.sc_ngbr >= 2 and card
sr.sc_ngbr <= 4 and
forall acr in set actors &
 card acr.ac_ngbr >= 2 and card
acr.ac_ngbr <= 4 and
card clthead.cc_ngbr >= 2 and card
clthead.cc_ngbr <= 4 ;
```

Invariants: (1) A communication link exists between any two sensors and actors in a cluster. (2) Sensor and actor are connected through a communication link. (3) A cluster head is connected with a sensor and an actor. (4) Every node in a cluster has minimum two and maximum four neighbors which show that there is no leaf node.
A cluster is connected with a base station to disseminate the information. A base station is assumed as a node which is a set of nodes, and used to perform actions.

```
BAction = <<NNDEPLOY>>;
BAStation :: basnode : Node   ba_ngbr :
set of Node
             baction : BAction
inv mk_BAStation(basnode, ba_ngbr,
baction) ==
basnode.npwr = <<HIGH>> and
basnode.mobility = <<STATIC>> and
basnode.c_status = <<CONNECTED>> <=>
ba_ngbr <> {} and
basnode.c_status = <<DISCONNECTED>> <=>
ba_ngbr = {} and
baction = <<NNDEPLOY>>;
```

Invariants: (1) A base station is of high power. (2) The connectivity status of a base station is connected if and only if it has neighbors otherwise disconnected.

The network topology consists of set of clusters and base station having communication links between them as defined below.

```
Topology :: clusters : set of Cluster
            bastation : BAStation
            comlinks : ComLinks
inv mk_Topology(clusters, bastation,
comlinks)==
forall cls1,cls2 in set clusters &
exists col in set comlinks &
col =
mk_(cls1.clthead.crnode.nidt,cls2.clthea
d.crnode.nidt)and
exists cls in set clusters & exists col
in set comlinks &
 col = mk_(cls.clthead.crnode.nidt,
bastation.basnode.nidt) and
card bastation.ba_ngbr >= 2 and card
bastation.ba_ngbr <= 4 ;
```

Invariants: (1) Any two clusters are connected through cluster heads having communication link. (2) A cluster is connected with a base station through a cluster head. (3) A base station has minimum of two and maximum of four connected nodes.
The state of battlefield consists of the attributes which are specified above.

```
state BField of
   sensors : set of Sensor
   actors : set of Actor
   cltheads : set of CLTHead
   bastations : set of BAStation
   clusters : set of Cluster
   cpath : seq of Idt
   comlinks : ComLinks
   topology : [Topology]
inv mk_BField(-, -, -, bastations,
clusters, -, -, -)==
forall cls in set clusters & forall sr
in set cls.sensors &
 sr.sdetect = <<MINES>> or sr.sdetect =
<<ENEMY>> or
 sr.sdetect = <<ATTACK>> =>
  sr.srnode.battleinfo =
<<BITRANSMITTED>> and
 exists acr in set cls.actors &
 acr.arnode.battleinfo = <<BIRECEIVED>>
and
 acr.paction = <<FIREGUN>> or
acr.paction = <<FIREBOMB>> or
 acr.paction = <<FTANK>> and
 acr.arnode.battleinfo =
<<BITRANSMITTED>> and
 cls.clthead.crnode.battleinfo =
<<BIRECEIVED>> and
 cls.clthead.crnode.battleinfo =
<<BITRANSMITTED>> and
exists bas in set bastations &
bas.basnode.battleinfo =
```

```
 <<BIRECEIVED>> and bas.baction =
<<NNDEPLOY>>
init bf == bf = mk_BField({}, {}, {},
{}, {}, [], {}, nil) end
```

Invariants: (1) In a cluster, a sensor detects mines, enemy and other attack. (2) The sensor transmits detected information to an actor. (3) The actor performs actions, for example, firing gun, bomb and tanks. (4) The actor transmits the information to cluster head and then to base station which decides to deploy the new nodes.
Nodes may fail during the operation and network does not remain connected. The operation *ConnectivityLoss* defined below is for this purpose. It takes a lost node as input and returns true if the connectivity is lost. The set of sensors, actors, cluster heads, base stations, communication path and links are modified below.

```
operations
ConnectivityLoss(LostNode : Node)cl :
bool
ext wr sensors : set of Sensor
  wr actors : set of Actor
  wr cltheads : set of CLTHead
  wr bastations : set of BAStation
  rd cpath : seq of Idt
  wr comlinks : ComLinks
pre true
post cl <=> exists sns1, sns2 in set
sensors &
 exists acr1, acr2 in set actors &
 exists clt1, clt2 in set cltheads &
 exists bas1, bas2 in set bastations &
 forall k in set inds cpath & cpath(1)
<> sns1.srnode.nidt or
 cpath(len cpath) <> sns2.srnode.nidt
and
 LostNode = sns1.srnode or LostNode =
sns2.srnode or
 cpath(1) <> sns1.srnode.nidt or
 cpath(len cpath) <> acr1.arnode.nidt
and
 LostNode = sns1.srnode or LostNode =
acr1.arnode or
 cpath(1) <> acr1.arnode.nidt or
 cpath(len cpath) <> acr2.arnode.nidt
and
 LostNode = acr1.arnode or LostNode =
acr2.arnode or
 cpath(1) <> acr1.arnode.nidt or
 cpath(len cpath) <> clt1.crnode.nidt
and
 LostNode = acr1.arnode or LostNode =
clt1.crnode or
 cpath(1) <> clt1.crnode.nidt or
 cpath(len cpath) <> clt2.crnode.nidt
and
 LostNode = clt1.crnode or LostNode =
clt2.crnode or
 cpath(1) <> clt1.crnode.nidt or
 cpath(len cpath) <> bas1.basnode.nidt
and
 LostNode = clt1.crnode or LostNode =
bas1.basnode or
 cpath(1) <> bas1.basnode.nidt or
 cpath(len cpath) <> bas2.basnode.nidt
and
 LostNode = bas1.basnode or LostNode =
bas2.basnode and
nodes = nodes~ \ {LostNode} and
exists sns in set sensors & exists acr
in set actors &
 exists clt in set cltheads & exists bas
in set bastations &
 card sns.sc_ngbr < 2 or card
acr.ac_ngbr < 2 or
 card clt.cc_ngbr < 2 or card
bas.ba_ngbr < 2 and
 k < len cpath and exists col in set
comlinks &
 col = mk_(cpath(k), cpath(k+1)) and col
not in set comlinks and
 comlinks = comlinks~ \ {col};
```

Pre/Post-Conditions: (1) There may exist two sensors such that the path for communication may loss between them. (2) The path may loss between a sensor and an actor which exhibits that either sensor or the actor is lost. (3) There may be two actors such that the path may loss between them. (4) The communication between an actor and a cluster head may loss due to loss of connectivity. (5) The path to communicate a cluster head with a base station may loss which shows that either the cluster head or the base station failed. (6) There may be two base stations which may loss from the network. (7) The network nodes are updated by removing the lost node. (8) The connectivity of sensor, actor, cluster head or base station may loss if they have no neighbor. (9) Communication links are updated by removing the lost link.
The lost of connectivity is recovered as there is a need for continuous operation which is defined as *RecoverNConnectivity*. In this operation, recovered node and new node are returned while the lost connectivity and lost node are taken as input.

```
RecoverNConnectivity(cl:bool,LostNode:No
de)RNode:Node,NNode:Node
ext wr sensors : set of Sensor
  wr actors : set of Actor
  wr cltheads : set of CLTHead
  wr bastations : set of BAStation
  rd cpath : seq of Idt
  wr comlinks : ComLinks
pre cl = true
post forall sr in set sensors & exists sr1 in
set sensors &
  LostNode = sr1.srnode => forall sngbr in
set sr1.sc_ngbr &
  exists sngbr1 in set sr1.sc_ngbr &
  card sngbr1.cngbrs > card sngbr.cngbrs and
  RNode = sngbr1 and sngbr1 = sr.srnode and
```

```
  card sr.sc_ngbr >= 2 and card sr.sc_ngbr <=
4 and
  sensors = sensors~ union {NNode} or
forall acr in set actors & exists acr1 in set
actors &
 LostNode = acr1.arnode => forall angbr in
set acr1.ac_ngbr &
  exists angbr1 in set acr1.ac_ngbr &
  card angbr1.cngbrs > card angbr.cngbrs and
  RNode = angbr1 and angbr1 = acr.arnode and
  card acr.ac_ngbr >= 2 and card acr.ac_ngbr
<= 4 and
  actors = actors~ union {NNode} or
forall clt in set cltheads & exists clt1 in
set cltheads &
 LostNode = clt1.crnode => forall cngbr in
set clt1.cc_ngbr &
  exists cngbr1 in set clt1.cc_ngbr &
  card cngbr1.cngbrs > card cngbr.cngbrs and
  RNode = cngbr1 and cngbr1 = clt.crnode and
  card clt.cc_ngbr >= 2 and card clt.cc_ngbr
<= 4 and
  cltheads = cltheads~ union {NNode} or
forall bas in set bastations & exists bas1 in
set bastations &
 LostNode = bas.basnode => forall bngbr in
set bas1.ba_ngbr &
  exists bngbr1 in set bas1.ba_ngbr &
  card bngbr1.cngbrs > card bngbr.cngbrs and
  RNode = bngbr1 and bngbr1 = bas.basnode and
  card bas.ba_ngbr >= 2 and card bas.ba_ngbr
<= 4 and
  bastations = bastations~ union {NNode};
```

Conclusion

For battlefield surveillance, WSAN is used to provide better handling of complex tasks. This requires continuous connectivity of the network but harsh environment conditions in battlefield may cause a loss of connectivity. Therefore, a recovery algorithm using clustering approach is presented in this work. The clustering approach is used for increasing energy efficiency and minimizing processing cost. Graph theory is used for topological representation for its effectiveness of modeling networks. Formal methods are observed for correctness addressing limitations of simulations techniques. VDM-SL Toolbox is used for analyzing, validating and verifying the formal specification.

The syntactic and semantic correctness of the specification is ensured through syntax and type checkers of the toolbox. For providing assistance at the implementation level the equivalent C++ code is produced through C++ code generator. The pretty printer generated the specification which is useful for documentation. To

Pre/Post-Conditions: (1) If the sensor node is lost then it is recovered by the highest degree neighbor. (2) The recovered sensor node must have neighbors greater than the minimum limit. (3) A new sensor is deployed with the passage of time.

is the body text with indent. This is the body text with indent. This is the body text with indent. This is the body text with indent. This is the body text with indent. This is the body text with indent. This is the body text with indent. This is the body text with indent. This is the body text with indent. This is the body text with indent.

(i) This is a list, note the hanging indent. This is a list, note the hanging indent. This is a list, note the hanging indent. This is a list, note the hanging indent. This is a list, note the hanging indent. This is a list, note the hanging indent.

(ii) This is a list, note the hanging indent. This is a list, note the hanging indent. This is a list, note the hanging indent. This is a list, note the hanging indent. This is a list, note the hanging indent. This is a list, note the hanging indent.

(iii) This is a list, note the hanging indent. This is a list, note the hanging indent. This is a list, note the hanging indent. This is a list, note the hanging indent. This is a list, note the hanging indent. This is a list, note the hanging indent.

identify run-time errors dynamic checking is used. The integrity examiner is used to visualize the formal specification in terms of predicates which are evaluated as true.

References

1. Regragui, Y., Moussa, N.: Agent-based system simulation of wireless battlefield networks, Computers & Electrical Engineering, vol. 56, pp. 313-333 (2016)
2. Cotton, S. L., Scanlon, W. G., Skafidas, E., Madahar, B. K.: Millimeter-wave stealth radio for special operations forces, Microwave Journal, (2010)
3. National Research Council: Energy-efficient technologies for the dismounted soldier, National Academies Press. (1998)
4. Afzaal, H., Iqbal, Z., Saeed, T., Zafar, N. A.: Battlefield surveillance formalism using WSANs, IEEE International Conference on Electrical Engineering (ICEE), pp. 1-6 (2017)
5. Khan, F., Rehman, A, U., Usman, M., Tan, Z., & Puthal, D. (2017). Performance of Cognitive Radio Sensor Networks using Hybrid Automatic Repeat reQuest: Stop-and-Wait. Mobile and Network Application. Springer.

6. Fitzgerald, J., Larsen, P. G., Sahara, S.: VDM Tools: Advances in support for formal modeling in VDM, ACM Sigplan Notices, vol. 43(2) (2008)
7. Qian, H., Sun, P., Rong, Y.: Design proposal of self-powered WSN node for battle field surveillance, Energy Procedia, vol. 16, pp. 753-757 (2012)
8. Gerla, M.: From battlefields to urban grids: New research challenges in ad hoc wireless networks, Pervasive and Mobile Computing, vol. 1(1), pp. 77-93 (2005)
9. Everett, H. R., Gage, D. W.: Third-generation security robot, International Society for Optics and Photonics, pp. 118-126 (1997)
10. Krotkov, E., Blitch, J.: The defense advanced research projects agency tactical mobile robotics program, The International Journal of Robotics Research, 18(7), pp. 769-76 (1999)
11. Wang, K. H., Li, B.: Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks, IEEE Twenty-First Annual Joint Conference of Computer and Communications Societies (INFOCOM), vol. 2, pp. 1089-1098 (2002)
12. Karumanchi, G., Muralidharan, S., Prakash, R.: Information dissemination in partitionable mobile ad hoc networks, 18th Symposium on Reliable Distributed Systems, pp. 4-13 (1999)
13. Zafar, N. A., Khan, S. A. Araki, K.: Towards the safety properties of moving block railway interlocking system, International Journal of Innovative Computing Information and Control, vol. 8(7), pp. 5677-5690 (2012)
14. Imran, M., Zafar, N. A., Alnuem, M. A., Aksoy, M. S., Vasilakos, A. V.: Formal verification and validation of a movement control actor relocation algorithm for safety–critical applications, Wireless Networks, pp. 1-19 (2015)
15. Imran, M., Zafar, N.A.: Formal specification and validation of a hybrid connectivity restoration algorithm for wireless sensor and actor networks, Sensors, vol. 12(9), pp. 11754-81 (2012)
16. Alnuem, M., Zafar, N. A., Imran, M., Ullah, S., Fayed, M.: Formal specification and validation of a localized algorithm for segregation of critical/noncritical nodes in MAHSNs, International Journal of Distributed Sensor Networks, (2014)
17. Riaz, S., Afzaal, H., Imran, M., Zafar, N. A., Aksoy, M. S.: Formalizing mobile ad hoc and sensor networks Using VDM-SL, Procedia Computer Science, vol. 63, pp. 148-153 (2015)

[1]