International Journal of Advanced Science and Technology
Vol.116 (2018), pp.59-74
http://dx.doi.org/10.14257/ijast.2018.116.06

A New Centralized Solution for Multi-event Wireless Sensor and Actor Networks

Utkarsh Pundir[1] and Virender Ranga[2]

[1,2]Department of Computer Engineering,
National Institute of Technology, Kurukshetra, Haryana, India
[1]pundirutkarsh@gmail.com, [2]virender.ranga@nitkkr.ac.in

Abstract

In this research, we studied the concept of mutual exclusion in Wireless Sensor and Actor Networks (WSANs). This research work deals with the solution to implement multievent concept through a centralized approach where sink nodes are responsible for the coordination and deployment of actor nodes. The area under consideration is decorated with many sensors and actors. Mutual exclusion in WSAN is a different concept. Here, it refers to using minimum resources to resolve different issues like least overlapping areas between actors and minimum actor usage. The actors must be very effective. Many undesirable conditions can be there in case mutual exclusion is not satisfied. Earlier-work was proposed only for single event, here we proposed it further for multiple events because in real environment multiple events are more common. The proposed algorithm keeps track of all the initiators and actors that act in-order to fulfill mutual exclusion in multiple event scenario. We have proposed an algorithm called CSM-WSAN (Centralized Solution for Multi-Event Wireless Sensor and Actor Network). The event sensed by the sensor nodes is called by a clustering algorithm which divide the field area into regions. Each region then separately finds the event dynamically inside its scope. Comparing the event with actor dynamicity, the favorable actors are granted to the event. The results are plotted at the end of the paper to show our proposed contribution.

Keywords: *Wireless Sensor and actor networks, Multi-event, priority*

1. Introduction

Multiple Event Mutual Exclusion in Wireless Sensor and Actor Network deals with the property of mutual exclusion in WSAN for multiple event. This is not just sensing network, it is both sensing as well as acting network where actors respond on the sensed data accordingly. To understand the concept, suppose there is a hall or a room which needs to be under surveillance, say a cold storage, where fruits are kept. We need to have sensors which detects the fruits that starts to stale, now the actors of that area get alarm from the local sen. In this way sensors and actors informs the specified authority about the fruits which must be used first. Another example is monitoring making human involvement minimal. Actors and sensors must be deployed in such a way that each point in area is covered under monitoring. With the advancement of technology, many new needs arise. The current needs are low energy consumption with high actors' capabilities. The only human involvement must be in the timely checkups and updates in the existing technology. The range of sensors and actors are set according to need and requirement with strength of actors and sensors un-compromised. Multi event sensor and actor network can be used in various fields such as battle field or in jet planes which can detect multiple attacks and offensively deals with all at once. The coordination between actors

Received (March 24, 2018), Review Result (June 5, 2018), Accepted (June 8, 2018)

Copyright ⓒ 2018 SERSC Australia

and sensors must be quick so that there is real time processing of events. The response needs to be quick and frequent, along with quick message delivery. Another important use could be in case of fire at multiple places within an area where the surrounding on fire when sensed by sensors are reported to concerned actors which in turn informs a sink which is used to manage all the actors in the concerned area to act. This is a Centralized Automated Architecture as the sink act as coordinator, various calculations are made to use the available resources as minimum as possible. In a situation when an actor which is required to perform has limited resources left, it must be saved for very crucial conditions in the future unless the resources are replenished. Actors with maximum efficiency for the particular event are chosen first and the priority of that actor is set as highest.

1.1. Motivation

Work done in the field of WSAN requires quality measures to be ungraded in the existing techniques. Various issues that are answered in this proposed work are:

(i) Answering the event in minimum time after its detection which requires minimum overhead in communication area.

(ii) Ensuring mutual exclusion.

(iii) Covering least overlapping area.

(iv) Minimizing the uncovered area which enhances the efficiency of the proposed work.

(v) Minimum computation on each node.

(vi) The most important issue which is the base of this paper is handling of multiple event in the field using centralized approach.

2. Related Work

The authors in [1] propose their work in the field of WSAN. They worked on a location studded with sensors and actors. They identify the diversity of events and proposes a Neighborhood Back-off approach for a localized and fully distributed approach. They used variables like benefit function and structures like shortest path tree to determine the usefulness of the actor w.r.t the event. This paper differentiates between WSN and WSAN networks and introduced the concept of actor and sensor network. Mutual Exclusion was first discussed in this paper with respect to WSAN. The least number of actors with maximum event-actor overlapping area are assigned the job first. This paper used event dynamics and other factors like changing intensities which are handled by the use of various function calls at the time event occurs. Flags are used to see if the required actor is already in the set cover. Different request functions are used to mark the event dynamics. It was the first time that this much of close study was done in this area, a greedy centralized approach was proposed where event dynamics and intensity is handled quite skillfully.

In [2] the authors proposed an approach for selecting some number of actor nodes, where overlapping of actors is taken as the main factor to decide minimal (yet sufficient) actor nodes in WSAN. It refers to two types of environments automated and semi-automated where a single coordinator and multiple regional coordinators are used respectively. The information that is detected by the sensors is for single event, there is no work in the field of multiple events. The main requirement of such type of networks is the coverage of entire-field by various actors (their acting range) *i.e.*, the whole field is covered by the operating actors. There is a situation where many actors are covering a single event, to find the most appropriate one is the task in hand. Checking on available resources and utilizing them efficiently is taken care in the novel approach of DOASME

(Distributed Optimal Actor node Selection based on Mutual Exclusion approach) proposed in this paper

W. Wu *et al.*, [3] proposed a Fault tolerant mutual exclusion algorithm for mobile ad-hoc networks where they discuss the first permission based MUTEX algorithm for MANETs. The authors looks toward handling the various disconnection, links and host failures in mobile network. The two basic techniques used in mutual exclusion are "permission based" and "token based". The technique used here was "Look Ahead" with a response time of "T" in light mode and $(E+1)*n/2$ under heavy load assuming the average time of an execution of Critical Section is E. The algorithm is fault tolerant with number of messages per CS execution "N" under light load and "$3N/2$" under heavy load assuming N number of hosts. The constraint of requiring FIFO channels in the look ahead techniques is relaxed here. The number of sequential messages exchanged after a host leaves the CS and before the next host enters the CS is known as Synchronization Delay which is T *i.e.*, 1 which is time of transferring single message.

A. Gupta *et al.*, [4] proposed a permission based clustering mutual exclusion algorithm for mobile ad-hoc networks. In this algorithm, the clustering based hierarchical approach which leads to reduction in message complexity, the number of messages exchanged can be reduced using the "Clustering Concept". Considering the performance parameters we have waiting time which is $2T(m+1)$ for worst case and 2T in best case. The message complexity in worst case is $O(n)$ for cluster-head and $O(2)$ for other nodes. Synchronization delay for worst case is $2T(m+1)$ and for best case it is 2T where T is propagation delay; m is the number of cluster-heads and n is total nodes.

M. Singhal *et al.*, [5] proposed a distributed mutual exclusion algorithm for mobile computing environments. The algorithm uses the technique of "look-ahead". The process of look-ahead removes the unnecessary computations by considering only the sites currently competing for critical section. This is necessary because of limited resources in the mobile network including limited battery back-up, limited bandwidth and processors which are slow. A transfer of $2|\Omega|$ messages in case the concurrency set of a site is Ω. The response time in this algorithm under heavy load is $(E+1)*n/2$ and under light load it is 2T assuming the average time of an execution of CS is E. The Synchronization delay in such case is T. This algorithm is efficient in substantial reduction of message overhead. The algorithm is free from starvation and other important aspect of deadlock.

R. Mellier *et al.*, [6] proposed a clustering mutual exclusion protocol for multi-hop mobile ad hoc network which is a token based technique used in clusters. The clusters are formed and various members are identified for each cluster along with their cluster-head based on the weighting technique. Each node in each cluster is satisfied in turn. The number of broadcasting rounds is proportional to the number n of nodes. So, it is in $O(n)$. The MUTEX is used which takes advantage of clustering, the token holding node gets critical section. The protocol has a response time of $T(k+1)$ under light load and $w((k+1)T+E)$ under heavy load. The messages exchanged per CS execution under both light and heavy load is k+1. The synchronization delay varies from 2dT to $2T(k+1)$.

R. Baldoni *et al.*, [7] proposed a distributed mutual exclusion algorithm for mobile ad-hoc network which focuses on reducing number of hops while shifting CS execution from one node to another. In this algorithm, a dynamic logical ring is built on the fly to reduce the number of hops to a very minimum value. The next process which is chosen in the ring is closest one in terms of hop number. The technique of using token in dynamic ring is made efficient. Non-FIFO channel is supported in the algorithm as ring is computed on the fly. For very small number of nodes, the number of messages exchanged per CS execution remains constant at n-2 where n is the number of nodes. Under heavy load it changes to 2.

B. Sharma *et al.*, [8] proposed a token based protocol for mutual exclusion in mobile ad hoc network. The sharing of resources in the distributed computing is a major advantage. Here algorithm uses 2 types of tokens (i) global token to ensure mutual exclusion among clusters *i.e.*, inter cluster movement of tokens and (ii) local tokens to ensure intra-cluster movement of token. The mutual exclusion is guaranteed as the local token within a cluster is activated only when the global token activates that particular cluster. Non-FIFO channel is applied which is fault tolerant. The response time is from $2T$ to $(m+1)*T$. The number of messages per CS execution is from $(1+m)$ to $(1/n+2)$ for light load and heavy load respectively where m is the total number of cluster-heads and n is the total number of nodes. The synchronization delay is T which is the propagation time of a single message.

M. Parameswaran *et al.*, [9] proposed a novel permission based reliable distributed mutual exclusion algorithm for MANETs. Here, crashing of node in CS is taken into account in this fault tolerant distributed mutual exclusion algorithm. No overhead of maintaining a logical link is there in this algorithm. It uses the technique of Look Ahead which uses HOLD message to keep the requesting sites in a waiting list if the node requested is currently using CS. Only when the priority of the requesting site is higher, it gets CS. Thus, it is a Non-FIFO channel. The response time in this algorithm is from $2T$ to $(E+1)*n/2$ under light and heavy load respectively. The synchronization delay is equal to the propagation time of a single message denoted by T. The number of messages required per CS execution is $2(\Phi\ 1)\ +\ p\ *w$.

3. Centralized Algorithm

The algorithm is designed for multiple events where sensors notify the actor about the situation which then makes suitable calculations before acting. We have considered a field of 400*400 meter where sensors are placed randomly but through localization algorithm their location can be spotted, and actors are placed with a proper order or equidistant from each other such that entire region is covered. The range (here radius) of actor is 30 meter and that of event radius is 40 meter. The actors are placed row-wise *i.e.*, keeping the y-axis constant until the single row of actor is filled then y is increased and again the entire x-axis is decorated with actors at particular (required) spots. When the entire area is covered by actors, the sensor nodes send the report packets with sensed values and sequence number of sensor nodes to the communication layer. All the report packets are send in the order of their sequence number. There may be more than one sensor which sends the report packets to communication layer at a time. All actors with their intersection area with the event region are taken into consideration. Now we have a set of actors which are having intersection with the event region. The next requirement is to have minimum number of actors with minimum overlap. Another array is taken and actors with decreasing area of intersection are put in it until all event region is covered. After each iteration the intersection area excluding the overlap by “already included actors” is calculated. This is to be considered keeping the effect of minimum overlap in mind. In case where an actor which is $(Ar\ +\ Er)$ units away from the event receives its notification. It will send the information to all actors within the radius of $2(Er\ +\ Ar)$.

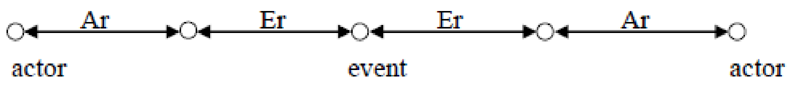


Figure 1. Actors within Range of Event Radius

Where, $Ar\ =\ Actor\ Radius$

$Er\ =\ Event\ Radius$

The distance of $(Er + Ar)$ to reach the event and another $(Er + Ar)$ to find other actors on the other/opposite side. Now all the actors within this range are aware of the event. After this, to define the intersection of all concerned actors with the event region, the radius and center of all actors must be known for calculation purposes. The calculation of minimum overlap, actor cover set and minimum number of actors is required. The center of event and the radius is calculated using the following formula.

$$Eventcenter\ = (xmin + xmax)/2, (ymin + ymax)/2 \qquad (1)$$

$$Radius = \ sqrt(pow((event_{center}.x - xmin), 2 + pow((event_{center}.y - \ ymin), 2)) \qquad (2)$$

Here *xmin*, *xmax* is the minimum and maximum x-coordinate of the event area respectively and *ymin*, *ymax* is the minimum and maximum y-coordinate of the event area respectively. In case the field area is very small *i.e.*, comparable to the event radius there is no need for calculating the number of actors and their priority, all actors must be ready to act. For example, in a field of 50*50 unit with actor radius ranging from 25-40 units. All actors must start working in the field to deal with the event. In a relatively bigger field area, the field area can be divided into regions for avoiding complex calculations. Then the actors within a region are selected using their x-y coordinates if they lie in the particular region. The number of actors in a region are calculated and the event radius and event center are measured using above equations in each region. This is the basic idea for a single event. Let us take a look on the proposed algorithm CSMWSAN. The basic idea is shown in flowchart to understand the working. The steps followed is simplified into five basic steps in the flowchart. Firstly, the flowchart begins with a start node. The second node is an input-output node which inputs the values sensed by the sensor nodes. The third node is process node which represents the process of values decoded to get the event coordinates. The next node is again used to show the clustering of the event points. After the clustering of points, the next node represents the process of finding event radius and event center in each cluster. After completing the previous step for each cluster, the related actors in each cluster are identified. These identified actors are further passed through the stage of efficiency-based sorting of actors. In the next step, we see that the events are quenched. Data of various dimensions are calculated in the next step of Output. The last node is the stop node. The flowchart is for quick overview of the algorithm that is mentioned hereafter. Figure 2 shows flow chart of our proposed solution CSM-WSAN.

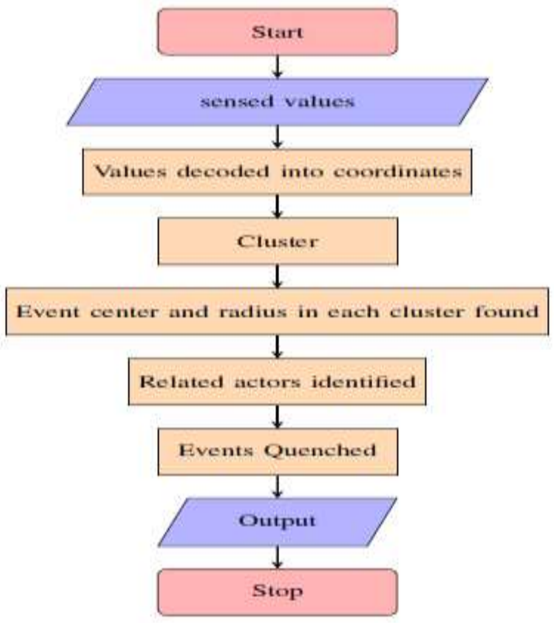

Start
sensed values
Values decoded into coordinates
Cluster
Event center and radius in each cluster found
Related actors identified
Events Quenched
Output
Stop

Figure 2. Flowchart for CSM-WSAN

The pseudo code of the above proposed solution is shown below in Figure 3:

$$SET\ ACTORS\ = Actors\ which\ receive\ first\ notification\ of\ event.$$

$$ac =\ Actors$$

$$Ec =\ Event\ Center$$

$$Er =\ Event\ Region$$

$$Ra = Range\ Of\ Actor$$

$$A\ =\ Actors\ In\ Cover\ Set$$

$$Rac =\ Region\ Covered\ By\ AC$$

```
Procedure CSM-WSAN()
1.SET_ACTORS get notified of event in the surrounding
2.SET_ACTORS will send report to COORDINATOR
3.COORDINATOR FRWmesg to aci ≠SET_ACTOR
4.Set COORDINATOR as STATION
5.Report is forwarded to STATION
6.    n=count(SET_ACTORS)
7.    for(i=1;i<=n;i++)
8.        Cluster[][] = get_actors_in_corresponding_cluster- -->
Clustering_Algo(coordinates_actors, n)
9.    end for
10.       c = get_no_of_cluster --->
Clustering_Algo(coordinates_actors, n)
11.    for (i=1;i<=c;i++)
12.        Calculate Ec and Er for ci .
13.        Broadcast MSG Er,Ec to actors within 2*(Er+ Ra)
14.            for all ac≠SET_ACTORS
15.                if(Ra U Er ≠ NULL)
16.                    convey to STATION
17.                End if
18.            End for
19.        P=1
20.        add=set of actors from which COORDINATOR received coordinates
21.        PREFERENCE=min_actor_Pref(Rai , Er,Ec, add) or min_overlap_Pref(Rai ,
Er,Ec, add)
22.        While(degree of each point in Er =0)
23.            For each aci ∈ add
24.                if (pref of aci = P)
25.                    CHOSEN = aci
26.                    P=P+1
27.                end if
28.            end for
29.            A=A U CHOSEN
30.            Rac =Rac + additional area by aci
31.            add=add-CHOSEN
32.        End While
33.        Send Command to actors in AC
34.    End for
35.End
```

Figure 3. Pseudo Code for CSM-WSAN Algorithm

Copyright ⓒ 2018 SERSC Australia

In the implementation, we used Omnett++ with Castalia. Castalia is the tool which is used to design Wireless Sensor and Actor Network configurations. Castalia uses the environment of Omnet++ where it is installed. For node value assignment, SN.physicalProcess[*].directNodeValueAssignment = ”(0) 1:50502 2:70802” statement is used, here sensor node one gives value 50502 which is decomposed to give the event coordinates. Similarly, for node 2 here, value 70802 is also decomposed into event coordinates. There can be any number of sensor node values, which are then used to cluster the sensed event coordinates. Remember, the sensors are not evenly distributed in the field area, they coordinate with each other to find their respective locations. This is the basic organization in sensor nodes. One more thing that is important is that in the “.ini” file of Castalia we can explicitly define the acting range of every actor to make the environment more dynamic and real world.

In our proposed solution, clustering is done based on the distance between event coordinates. In each cluster the number of different coordinates is saved along with their cluster number. Now in each cluster *xmin*, *xmax*, *ymin* and *ymax* are calculated and used as in equation (i) to calculate event center as shown in Figure 4.

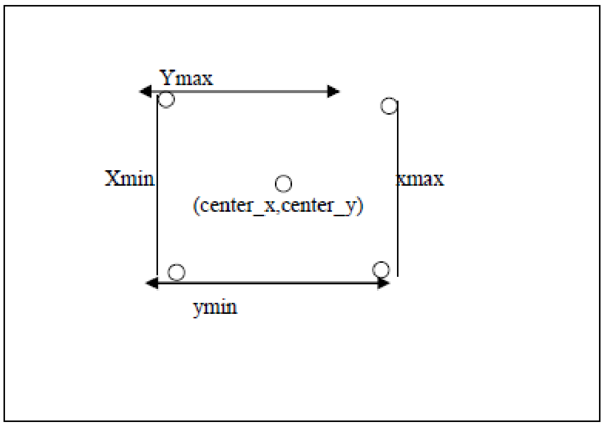


Figure 4. Finding Event Center using Sensor Coordinates

After calculating the center and radius of event using equation (ii), the event area is calculated and all the actors which have intersecting area with the event area are taken into consideration. After successfully getting all the related actors, whose center and radius are known before-hand, the algorithm goes forward with increasing the efficiency by considering the priority of actors. The actor with highest priority is assigned first.

In the proposed approach, overlapping and non-overlapping areas are very helpful at each iteration. Priority is based on maximum actor and event area intersection, the actor with maximum intersection is taken up first and next actors are taken only after taking into account: (i) remaining actors, (ii) assigned actors and (iii) event area. These three data values are used to exclude the intersecting area which is already covered by the assigned actors. Note: This procedure is repeated in each cluster.

Let us take a look at all the Figures from 5 to 9. In Figure 5, we see all the actors whose coverage area is intersected with event area. There are four actors whose area is intersecting with event. Figure 6 shows the event area. In the next Figures from 7 to 9 we see how at each iteration one actor is added to the assigned list of actors based on their priority. In the Figure 9, we notice how the actor with coverage area shown dotted in Figure 5 is excluded as it is only bearing extra cost with no extra advantage.

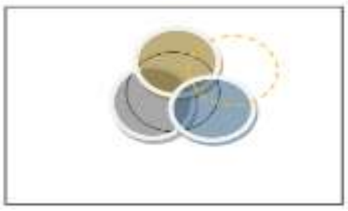


Figure 5. All Actors which Overlap the Event are Shown

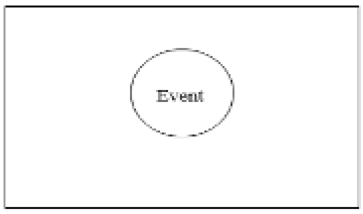


Figure 6. Event Covered

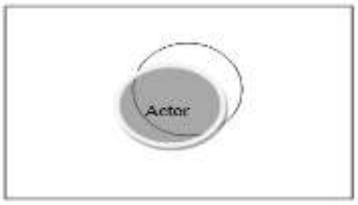


Figure 7. Actor with Maximum Event Overlap is Added

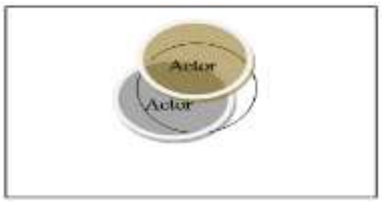


Figure 8. Second Actor is Added

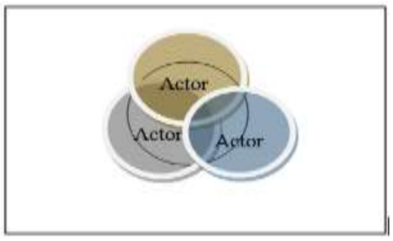


Figure 9. After Adding Third Actor, Entire Event is Covered

4. Algorithm Analysis

The proposed algorithm is a centralized solution for multiple event in wireless sensor and actor network. The sensor nodes get notified of the event and through a localization algorithm it finds the location of sensors and their covering actors. In lines 1-2, the actors get the notification of the event which forward the report to the sink known as coordinator here. In line 3, the sink forwards message to all actors who do not receive the notification and wait for some minimum time to allow the notification of any unheard actor. In lines 4-5, we see that the complete report is forwarded to the sink also referred to as STATION and COORDINATOR.

In lines 6 - 10, the algorithm calculates the number of actors in the SET ACTORS and for each actor its parent cluster is found with the total number of clusters and their members. In line 10, the number of clusters are calculated and stored in c variable. In lines 11 and 12, event center and event radius are calculated for each cluster. In lines 13 to 18, all the actors within $2(Er + Ra)$ broadcast the message and the actors which have their coverage area intersecting with the event area are conveyed to STATION. Line 19 is used to set a priority variable P to 1. In the next line, actors whose coverage area is intersecting with the event area are put in an array called *add*. In line 21, the priority based on minimum overlap among actors and minimum size of actor set is calculated. In lines 22 to 28, each actor in add variable is taken and added to another set *A* based on its priority. In line 31, the chosen actors are put in *A* and those actors are removed from *add* variable. *Rac* stores the total area covered by the actors. Finally, we get a set of actors which follows the properties of mutual exclusion.

Copyright ⓒ 2018 SERSC Australia

5. Correctness of the CSM WSAN Algorithm

The three-major correctness requirement for any algorithm are discussed here in the context of our CSM WSAN Algorithm.

1. **Safety**: Whenever a sensor in a cluster detects any event, it ensures that more than one sensor is there in any cluster before moving further. Only a single sensor can indicate an event about any faulty behavior, but its correctness is ensured after getting the same alarm from nearby sensors too and putting them combined in a single region or cluster.
2. **Liveness**: After clustering, suppose the field is divided into 4 event regions. In such case, after covering one entire event region then only the algorithm moves to the next event region. If other region also detects the event then the procedure of covering one event region is repeated otherwise after going to all detected regions, if all are satisfied, it moves towards termination.
3. **Fairness**: If a region of event is not satisfied then algorithm moves through each sensed event co-ordinate and covers every point of event area. Thus, every point is fairly covered. The actors which are activated are so placed in the region that they reach every point of the field from left to right and top to bottom. Fairness is thus guaranteed in the algorithm.

Our algorithm works fine for events when there is close proximity between event coordinates such that a single event region and only one cluster is formed.

6. Illustration with an Example

An example is used to illustrate the basic details of computations. Small patches of output are pasted which renders a great overview of steps followed. Figure 10 shows the details of a condition where the event is divided into 4 regions or clusters, the region's shape depends on the placement of actors in it.

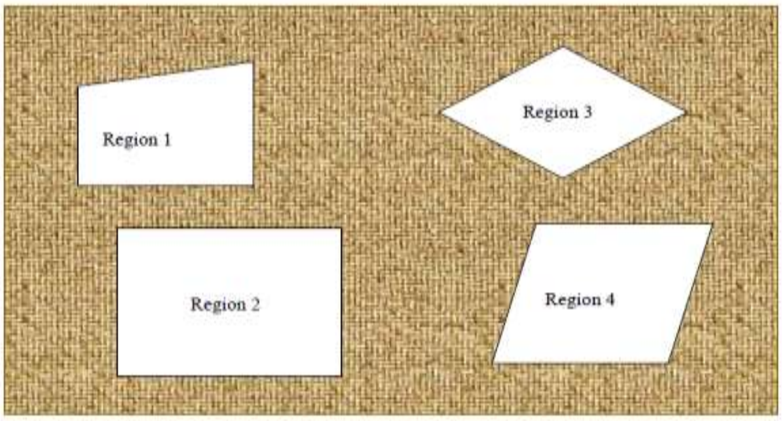
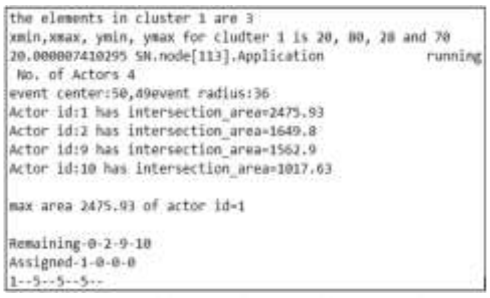


Figure 10. Showing the Field Divided into the Four Regions, each Shape based on Actor Location

Following is the output for an event in a particular cluster which has three event values 221, 351, 851 which are decoded into event coordinates (20,28), (30,70), (80,70).



```
the elements in cluster 1 are 3
xmin,xmax, ymin, ymax for cludter 1 is 20, 80, 20 and 70
20.000007410295 SN.node[113].Application                running
 No. of Actors 4
event center:50,49event radius:36
Actor id:1 has intersection_area=2475.93
Actor id:2 has intersection_area=1649.8
Actor id:9 has intersection_area=1562.9
Actor id:10 has intersection_area=1017.63

max area 2475.93 of actor id=1

Remaining-0-2-9-10
Assigned-1-0-0-0
1--5--5--5--
```

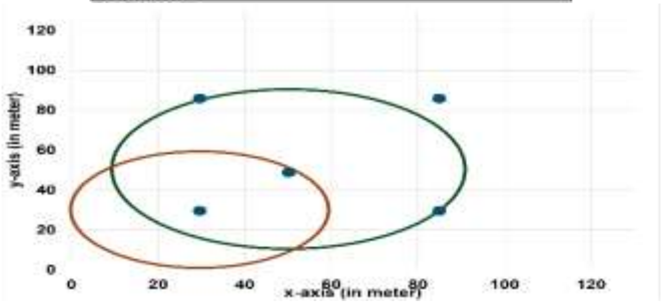
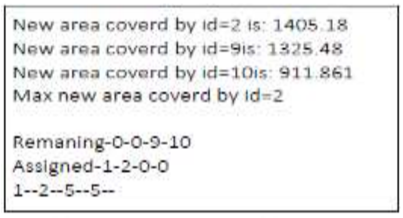


Figure 11. Actor 1 with Maximum Area of Intersection is Assigned

In Figure 11, the five dots refer to the center of event and actor centers. The circle with center (50,49) depicts event area while the other circle depicts the actor coverage area. The output shown above was the first part of execution where event center, event radius and actors with intersecting areas are identified. After selecting four actors, the area of intersection of event with each actor is calculated and actor with maximum intersection area, here actor 1 with center (30,30), is included into the assigned-actor list while the remaining in the remaining-list. The same process is repeated again.



```
New area coverd by id=2 is: 1405.18
New area coverd by id=9is: 1325.48
New area coverd by id=10is: 911.861
Max new area coverd by id=2

Remaning-0-0-9-10
Assigned-1-2-0-0
1--2--5--5--
```

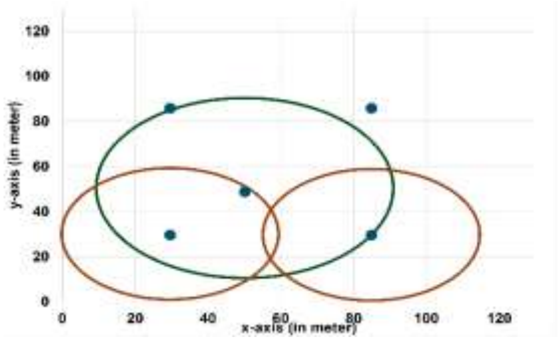
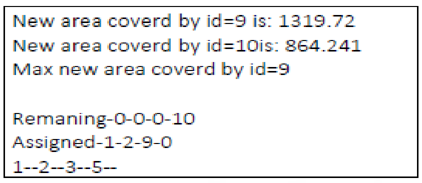


Figure 12. Actor 2 with Maximum Area of Intersection is Assigned

One thing to notice here is that after first iteration it is calculating "new area coverd" which is because it is now only considering non-covered area (after excluding the already covered area in the first iteration). In the given example "Assigned" variable collects the actors in the order of decreasing priority.

Copyright ⓒ 2018 SERSC Australia



```
New area coverd by id=9 is: 1319.72
New area coverd by id=10is: 864.241
Max new area coverd by id=9

Remaning-0-0-0-10
Assigned-1-2-9-0
1--2--3--5--
```

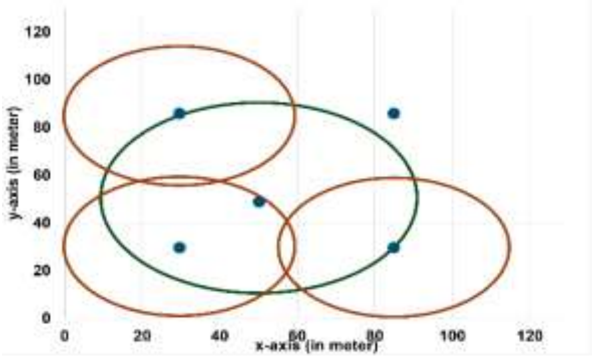
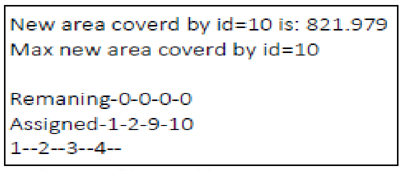


Figure 13. Actor 9 with Maximum Area of Intersection is Assigned

Similarly, in the third iteration, actor with id 9 is included in the assigned list. Now only actor with id 10 is left in the remaining list.



```
New area coverd by id=10 is: 821.979
Max new area coverd by id=10

Remaning-0-0-0-0
Assigned-1-2-9-10
1--2--3--4--
```

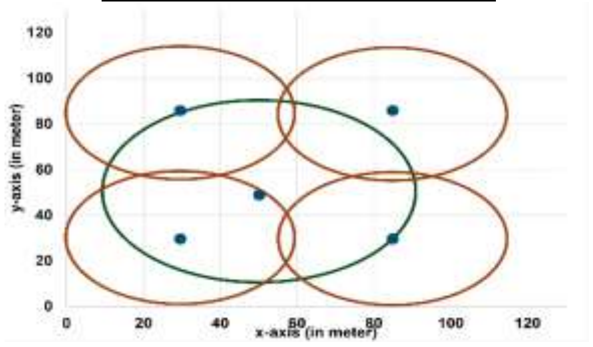
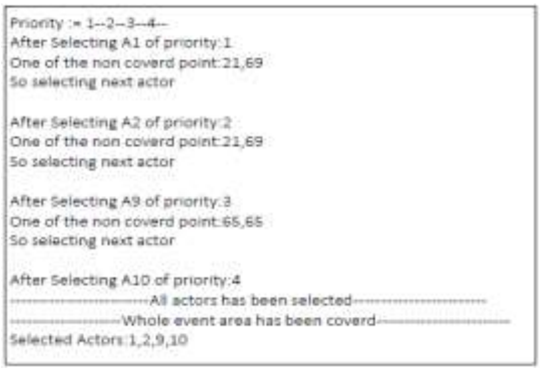


Figure 14. Actor 1 with Maximum Area of Intersection is Assigned

In this case, each actor is added in the assigned list, this is just the matter of a particular case. Usually some actors are left behind in the remaining variable whose area of coverage is already covered by one actor or the other.



```
Priority := 1--2--3--4--
After Selecting A1 of priority:1
One of the non coverd point:21,69
So selecting next actor

After Selecting A2 of priority:2
One of the non coverd point:21,69
So selecting next actor

After Selecting A9 of priority:3
One of the non coverd point:65,65
So selecting next actor

After Selecting A10 of priority:4
------------------------All actors has been selected------------------------
--------------------Whole event area has been coverd-----------------------
Selected Actors:1,2,9,10
```

This is how entire event area is covered. The output snippets shown above clearly represent the flow of events in a particular region/cluster.

7. Result Analysis and Comparison

We compare our algorithm with the basic work done in this field in past. The major comparison is done between the Vedantham's algorithm, DACI (Distributed Actor Coverage Irrigation), CACI (Centralized Actor Coverage Irrigation) and our CSM WSAN (Centralized Solution for Multievent in Wireless Sensor and Actor Network).

Figure 15 shows the comparison between the three algorithms with respect to Non-overlapped area vs Event Radius. We analyze that CSM shows the best results. As the Nonoverlapped area increases with the increase of event radius. The results with CSM are far better than those with the Vedantham, DACI and CACI. The results with DACI and CACI are similar so we consider DACI in our representations. Acting range by an actor can be varied, we have considered an average of 40m as the acting range of our actors. One thing to notice here is that at event radius less than 40m and event center very close to any actor center there is no need to consider any other actor, one single actor is enough to carry out the operation.

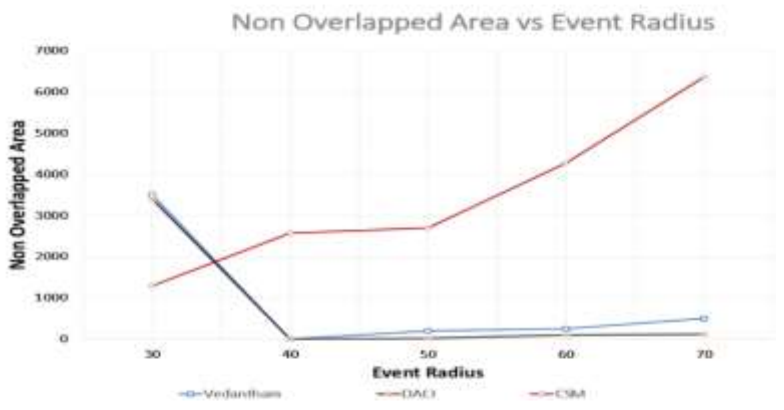


Figure 15. Graph plotting Non-Overlapped Area vs Event Radius

In Figure 16, we compare the number of actors required for execution vs the Event Radius. The results with our CSM algorithm are quite good. With increasing event-radius the number of executing actor nodes are also increasing in all the three algorithms but the best result is given by our algorithm. The number of actors requires are comparatively much lesser in CSM.

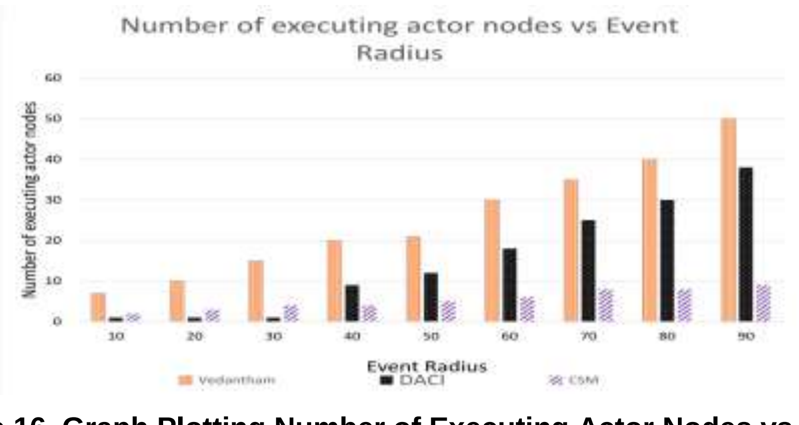


Figure 16. Graph Plotting Number of Executing Actor Nodes vs Event Radius

Copyright ⓒ 2018 SERSC Australia

In Figure 17, percentage of overlapped area is compared with Event radius for CSM. The graph follows a trend like shown for a range of event radius.

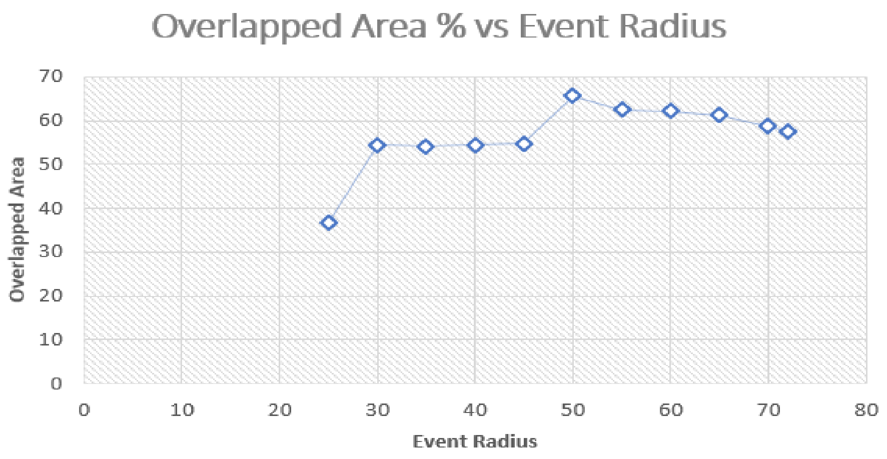


Figure 17. Graph Plotting Overlapped Area Percent vs Event Radius

In Figure 18, we can clearly see that when the number of actors are densely placed in the field, more number of actors intersects with the event area. Hence, more number of executing actor nodes are deployed. This is because when there are more number of actors studded in the field, the actors are closely placed together so more actor area intersects with the event area. Thus, more number of actors come under computation.

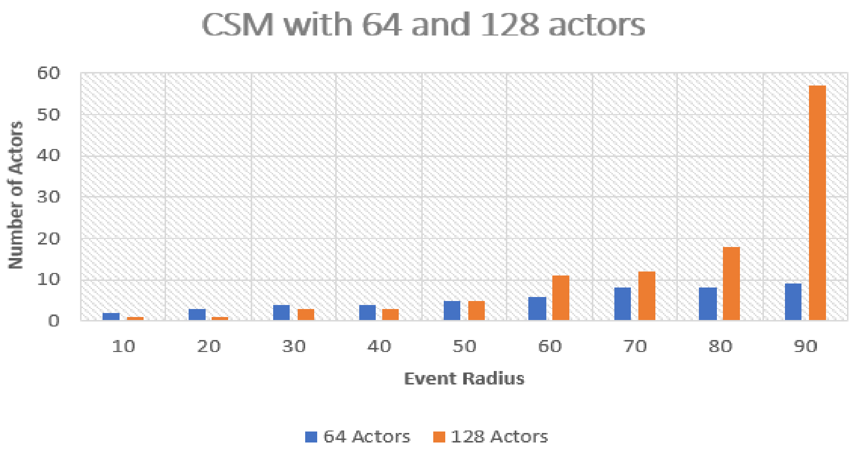


Figure 18. Graph plotting Number of Actors vs Event Radius for 64 Actor Field and 128 Actor Field

In Figure 19, the Non-Overlapped Area decreases with increasing density of actors in the field and vice-versa with the event radius increasing. Thus, we can state that the non-overlapped area is inversely proportional to the number of actors deployed in the field keeping the event radius constant. Usually we find an increasing graph in such case but there may be region of no change which may be due to event region with radius $x - a$ and $x + a$ lying within the lower and upper limits of same number of actor coverage.

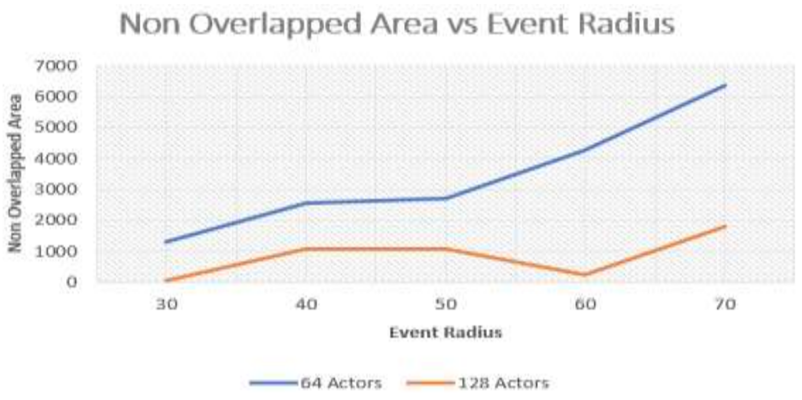


Figure 19. Graph plotting Non-Overlapped Area vs Event Radius for 64 Actor Field and 128 Actor Field

In Figure 20, time vs event radius graph is plotted which shows that more computation is required for densely populated actor field. More the number of actor nodes in the field the more is the time taken for computation. The graph plots the time vs event radius for 64 actor nodes and 128 actor nodes. We see that for very small event radius the time is equal this is because it is the time of simulation while the time of computation for very small radius is more or less constant.

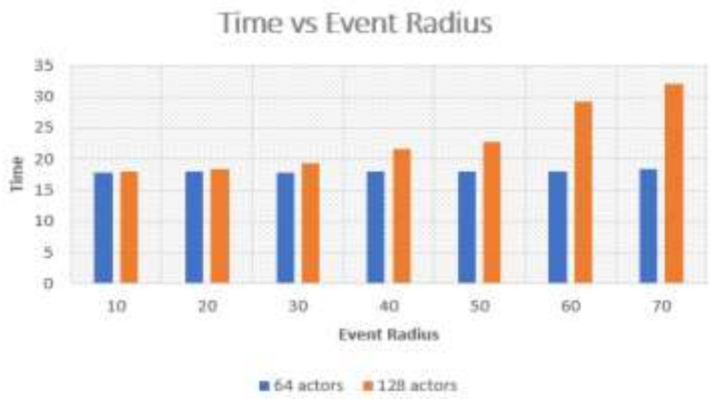


Figure 20. Graph Plotting Time vs Event Radius for 64 Actor Field and 128 Actor Field

8. Complexity Analysis

When multiple actors get event notification, let T actors get the notification, the algorithm counts the number of clusters and their respective members. Let n be the number of clusters. We have declared that Er is the event region. From among the T actors in the region of $2(Er + Ra)$ let k actors be selected by sensor nodes. $K - 1$ actors are actors who send information to the initiator. A cost of $|\Omega|$ is incurred while sending the information of actors intersecting with the event region. Finally, commands send by initiators incurs a cost of I messages. Thus, summing up, we get the communication cost of $T + n + n(Er2 + (k + 1) + |\Omega| + I)$. The time complexity of the algorithm is $O(W + nD)$ where W is the actor's waiting time and D is the distance between the actor node and event area.

Copyright ⓒ 2018 SERSC Australia

9. Conclusions and Future Work

This research work is about formation of an actor cover set for each event, in every cluster formed, in multi-event scenario. The field information is forwarded to the sink node which identifies the various active sensors in various regions. The number of active sensors in each region are counted and actors are selected based on minimum overlap priority. In future we will be looking forward to increasing the efficiency by covering the field with optimal number of actors and minimizing the region overlaps. The work could be extended for 3-D area along with heterogeneous environment where different actors have different functions.

References

[1] R. Vedantham, Z. Zhuang and R. Sivakumar, "Mutual exclusion in wireless sensor and actor networks", In Sensor and Ad Hoc Communications and Networks. SECON'06. 3rd Annual IEEE Communications Society, **(2006)**, pp. 346-355.

[2] V. Ranga, M. Dave and A. K. Verma, "A distributed approach for selection of optimal actor nodes in wireless sensor and actor networks", In Contemporary Computing and Informatics (IC3I), International Conference **(2014)** November, pp. 312-319.

[3] W. Wu, J. Cao and J. Yang, "A fault tolerant mutual exclusion algorithm for mobile ad hoc networks", Pervasive and Mobile Computing, vol. 4, no. 1, **(2008)**, pp. 139-160.

[4] A. Gupta, B. Reddy, U. Ghosh and A. Khanna, "A permission based clustering mutual exclusion algorithm for mobile ad-hoc networks", International Journal of Engineering Research and Applications, vol. 2, no. 4, **(2012)** pp. 019-026.

[5] M. Singhal and D. Manivannan, "A distributed mutual exclusion algorithm for mobile computing environments", In Intelligent Information Systems, IIS'97, Proceedings, **(1997)** December, pp. 557-561.

[6] R. Mellier and J. F. Myoupo, "A clustering mutual exclusion protocol for multi-hop mobile ad hoc networks", In Networks. Jointly held with the IEEE 7th Malaysia International Conference on Communication., 13th IEEE International Conference, vol. 1, **(2005)** November, pp. 250-255.

[7] R. Baldoni, A. Virgillito and R. Petrassi, "A distributed mutual exclusion algorithm for mobile ad-hoc networks", In Computers and Communications, Proceedings, ISCC. Seventh International Symposium, **(2002)**, pp. 539-544.

[8] B. Sharma, R. S. Bhatia and A. K. Singh, "A token based protocol for mutual exclusion in mobile ad hoc networks", Journal of information processing systems, vol. 10, no. 1, **(2014)**, pp. 36-54.

[9] M. Parameswaran and C. Hota, "A novel permission based reliable distributed mutual exclusion algorithm for manets", InWireless and Optical Communications Networks (WOCN), Seventh International Conference, **(2010)** September 1-6.

[10] M. Negahdar, M. Ardebilipour and M. Mapar, "Adaptive method for decreasing over-covered areas in wireless sensor networks", In Wireless and Mobile Communications, ICWMC'08, The Fourth International Conference, **(2008)** July 103-107.

[11] R. Baldoni, A. Virgillito and R. Petrassi, "A distributed mutual exclusion algorithm for mobile ad-hoc networks", In Computers and Communications, Proceedings, ISCC, Seventh International Symposium, **(2002)**, pp. 539-544.

[12] N. Malpani, N. Vaidya and J. Welch, "Distributed token circulation in mobile ad hoc networks", Proceedings of the 9, ICNP, **(2001)**, pp. 154-165.

[13] A. Derhab and M. Zair, "A resource-based mutual exclusion algorithm supporting dynamic acting range and mobility for Wireless sensor and Actor Networks", In Distributed Computing in Sensor Systems Workshops (DCOSSW), 6th IEEE International Conference, **(2010)** June 1-6.

[14] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges", Ad hoc networks, vol. 2, no.4, **(2004)** pp. 351-367.

[15] D. Seo, S. Kim and G. Song, "Mutual exclusion strategy in a Cloud-of-clouds", In Consumer Electronics (ICCE), IEEE International Conference, **(2017)** January, pp. 124-125.

[16] M. Negahdar, M. Ardebilipour and M. Mapar, "Adaptive method for decreasing over-covered areas in wireless sensor networks", In Wireless and Mobile Communications, ICWMC'08. The Fourth International Conference **(2008)** July 103-107.

[17] S. K. Dhurandher, D. K. Sharma, I. Woungang, R. Gupta and S. Garg, "GAER: genetic algorithm-based energy-efficient routing protocol for infrastructure-less opportunistic networks", The Journal of Supercomputing, vol. 69, no. 3, **(2014)**, pp. 1183-1214.

[18] S. J. Habib, M. Safar and N. El-Sayed, "Automatic placement of actors within wireless sensor-actor networks", In Telecommunication Networks and Applications Conference, Australasian, **(2008)** December, pp. 224-229.

[19] F. Zarafshan, A. Karimi, S. A. Al-Haddad and S. Subramaniam, "A Survey on Mutual Exclusion Algorithms for Mobile Ad Hoc Networks", Journal of Computational and Theoretical Nanoscience, vol. 13, no. 5, **(2016)**, pp. 3472-3487.
[20] A. Derhab and N. Lasla, "Distributed algorithm for the actor coverage problem in WSN-based precision irrigation applications", In Distributed Computing in Sensor Systems and Workshops (DCOSS), International Conference, **(2011)** June 1-6.
[21] J. Li, C. Chen and X. Guan, "XY-expansion: Joint search and clock synchronization for wireless sensor and actor networks", In Control Automation (ICCA), 11th IEEE International Conference, **(2014)** June, pp. 863-868.
[22] B. Sharma, R. S. Bhatia and A. K. Singh, "A Token Based Protocol for Mutual Exclusion in Mobile Ad Hoc Networks", Journal of Information Processing Systems, vol. 10, no. 1, **(2014)**, pp. 36-54.
[23] S. K. Dhurandher, D. K. Sharma, I. Woungang and A. Saini, "Efficient routing based on past information to predict the future location for message passing in infrastructure-less opportunistic networks", The Journal of Supercomputing, vol. 71, no. 5, **(2015)**, pp. 1694-1711.
[24] P. Liu, P. Yuan and C. Wang, "A Socially Aware Routing Protocol in Mobile Opportunistic Networks", Journal of Communications, vol. 12, no. 1, **(2017)**, pp. 62-71.
[25] OMNET++ with Castalia, http://www. omnetpp.org, downloaded on 10 October 2017.

Authors

Utkarsh Pundir received his B.Tech. in Computer Science and Engineering from AKGEC Affiliated to Abdul Kalam Technical University, India in 2016. Currently, he is pursuing his M.Tech. from National Institute of Technology, Kurukshetra, India. His interest area is Wireless Sensor and Actor Networks.

Dr. Virender Ranga received his M.Tech. in Computer Science and Engineering from Guru Jambheshwar University of Science & Technology, India in 2004. Currently, he is Assistant Professor in the Computer Engineering Department, NIT Kurukshetra, India. His interest areas are partition recovery in Wireless Sensor Networks, Fault Tolerance in Sensor Networks, Coverage and Connectivity in Sensor and Actor Networks in WSAN.

Copyright ⓒ 2018 SERSC Australia