*Kuwait J. Sci.* **43 (1)** pp. 150-173, 2016

# Optimal nodes selection in wireless sensor and actor networks based on prioritized mutual exclusion approach

Virender Ranga[1,*], Mayank Dave[1], Anil K. Verma[2]

[1*]*Department of Computer Engineering, National Institute of Technology, Kurukshetra, Haryana, India*

[2]*Department of Computer Science and Engineering, Thapar University, Patiala, Punjab, India*

**Corresponding Author: Email: virender.ranga@nitkkr.ac.in*

## Abstract

In this research paper we study the problem of mutual exclusionin the context of wireless sensor and actor network (WSAN) and propose two novel approaches to solve it. The major requirements for any proposed approach in such context are: (1) the proposed approach must select the minimum number of actor nodes to act on the given event region, (2) the overlaps between acting ranges should be minimum, (3) wastage of resources should be less, and finally, whole event region must be covered by one or more than one actors as per their applicability. We have proposed two algorithms, centralized prioritized *h-out-of-k* mutual exclusion algorithm (CPMEA*),* and distributed prioritized *h-out-of-k* mutual exclusion algorithm (DPMEA) in this research paper. Both proposed approaches construct an actor cover set with similar optimality. The simulation results show the performance in terms of size of actor cover set, overlapped region, non-overlapped region and maximum actor coverage degree. We have also compared our obtained results with previously proposed benchmark algorithms.

**Keywords:** h-out-of-k approach; mutual exclusion;prioritized actor cover set; wireless sensor and actor networks.

## 1. Introduction

A typical wireless sensor network (WSN) performs only one action i.e. sensing the environment. The need for smart interaction with the environment has led to the emergence of more advanced networks i.e. WSANs. The evolution from WSNs, which can be thought of to perform only read operations, to WSANs, which can perform both read and write operations, introduces unique and new challenges that need to be addressed and solved in the current scenario. Whenever an event occurs in the event area, the sensor nodes which sense the event have to send the detection information directly to actor(s) or to the sensor nodes, which may send the information to a sink or multiple sinks depend upon the type of application, which processes the information and then selects one or more actors to act

upon the event. The former model is referred to as an automated architecture, while the latter model is referred to as semi-automated architecture Akyildiz & Kasimoglu (2004). Thus, WSANs can be used in a wide area of applications, especially when some action is required on deployed site, like fire detection and control, home automation, search and emergency situations, battle field etc.

In some applications, if multiple actors are required to cover an event region, it may be necessary to ensure that the acting regions are non-overlapping or mutually exclusive in order to ensure uniform acting behaviour over the entire event region. If the acting regions are circular, the overlapping between these regions is inevitable, and hence undesirable effects may occur. For example, if there is an application where the sensors report the amount of moisture in the ground and the actors have to irrigate the area uniformly, the regions where overlap occur will overuse the actor resources (water) or will result in flooding in the field. The conventional distributed mutual exclusion methods provide atomic access to a shared critical resource among a group of processes. However, the problem of mutual exclusion is unique in the context of WSANs, and is defined as follows: "Given a set of actor nodes in an event region, what is the selected set of actor nodes and their position, such that there is a minimal or no overlap area in their acting regions and also satisfy the imposed challenges such as, minimize delay bound, wastage of actor resources and cover entire even area etc." The major requirements for any proposed approach are: (1) it must select minimum number of actors to act on given event region, (2) the overlaps between acting ranges should be minimized, (3) wastage of resources should be low, and (4) the whole event region must be covered by one or more than one actors as per their applicability in the field. For example, some applications are, the actors have to uniformly irrigate the moisture in the ground, poison gas actors, and agriculture actors for sprinkling pesticides etc.

The rest of the paper is organised into 6 sections. In section 2, we give our motivation and contribution in this research paper. Section 3 highlights the related works withits principles and limitations. In section 4, we propose our centralized solution. In section 5, we discuss another distributed solution along with the limitation of centralized approaches. The analys is of complexities and result of simulations study is presented insection 6. We have described future scope and concludedour work in section 7.

## 2. Motivation and contribution

In our proposed approaches, we consider that the whole event region is divided into unit points, which are likely identical copies of shared resource to be used or accessed by the actors. All points need to be covered by one or more than one actor nodes with the nodes have dynamically assigned priority. Each point within the event region has its actor coverage degree i.e. number of actors, which are able to cover this point in

its acting range. An actor can cover $h$ number of points within the event region (if it has some intersection with event region). Here in the context of WSAN, the following differences (in terms of requirements) with traditional *h-out-of-k* mutual exclusion (Vedantham*et al*., 2006; Derhab & Zair, 2010; Melodia *et al*., 2005; Akyildiz *et al*., 2002; Baldoni *et al*., 2002; Salarian *et al*., 2012; Derhab & Lasla, 2011; Raynal, 1991; Cheng-Zen, 2005; Bin Hu *et al*., 2014) problem are given below:

- All $k$ points must be covered i.e. it means all points should be accessed by at least one actor node. Once all points or whole event region is covered,our protocol terminates (i.e. it has computed the required actor cover set) and further actors starts acting.
- While ensuring minimum coverage degree of each point to 1, some points may be covered by more than single actor i.e. more than one actor nodes are declared to work on that point. Although we try to minimize the number of such points i.e. minimizing the overlap between acting areas of different actors, but there can be few points (if acting range is circular), which are not covered by single actor node. Therefore, concurrent access is also allowed.
- At a time maximum number of points that can be accessed is $k$.

A point in the event regionis said to be $K - actor - covered$ $(K \geq 1)$ if it is within the acting range of $K$ active actors. Given an event region and a set of active actors whose acting range intersects with event radius $ER$, denoted by $\Omega$ , then set $\xi = (R,\Omega)$ is called *actor-coverage configuration* if each point in event regionis within the acting range of at least one actor in $\Omega$. We have taken the following general assumptions to make our problem more tractable.

- We consider the case, where sensors and actors are both static and there are enough actors to cover each point in the network.
- The network contains all actors of the same type and acting range. However, there may be different types of actors (like water actor, gas actor) in a network with different acting range.
- We assume that sensors and actors can determine their location through some localization algorithms, and there is an underlying reliable routing protocol for delivering directives, gathering responses and delivering notifications to any sensor or actor node.
- The actor nodes are having synchronization, with the assumptions like network channels are reliable, and network does not get partitioned with the failure of backbone actor node.

Moreover, in our proposed algorithms, we have used different priority functions, one is for minimum number of actor selection and other is used for selecting actors in such a way that overall overlapped region can be minimized. We call these functions

*priority_min_actor* () *and priority_min_overlap* () respectively. These functions return a set of priority of intersecting actors, which is used by both centralized and distributed algorithms. To compute priority function, our proposed algorithms take inputs like event radius, event centre, and actor range with location information. In the nut shell, Table 1 shows the main differences between our proposed approaches and state-of-the-art approaches:

**Table 1.** Differences between our proposed approaches and state-of-the-art approaches

| Points | Our proposed approaches | State-of-the-art approaches |
|---|---|---|
| 1. | Our work has defined the degree of each point within event region. | They have defined degree of new overlap. |
| 2. | We compute actual points and thus actual values for selecting an actor. | The calculation is not done on actual points; therefore, they have used constants to select an actor node. |
| 3. | Our criteria is global to select the actor node i.e. we select those actor who gives maximum number of points with degree >=1 in whole event area. | Their criterion is different to select an actor node who gives maximum new area. |
| 4. | We use priority function for intersecting actor nodes. | No mechanism is used for priority assignment. |
| 5. | Termination criteria: Our approach terminates, when all selected points have non-zero degree or no actor remains in the intersecting set. | Termination criteria: The algorithms terminate, when whole event region is covered and also need each time to computeremaining area. |
| 6. | Our approaches can handle the case when actors are not able to cover the whole area. | These approaches have not included this condition in the implementations. |
| 7. | We donot involve sink node (in distributed approach). | Sink is always involved for computing of actor cover set. |
| 8. | Our approach uses one dynamically selected actor node as coordinator for a particular event. | No concept of coordinator or initiator. |
| 9. | In our proposed approach, sensors only inform to actor node based on SA or MA selection. | Sensors need to inform both, first to sink and then to actor node. No concept of SA or MA selection is taken. |
| 10. | Actor nodes use partial information of event region and then send to coordinator for further action. | Uses dependency region for coordination. |
| 11. | Actor node is informed by coordinator, (if selected) therefore, no need of messages from each actor to other actor nodes. | Each actor who wants to act needs to send a notify message to others and then wait for their response. |
| 12. | In our both approaches, coordination is perfect between actor nodes. | Coordination is not perfect, because all actors are not within dependency region. |
| 13 | Coordinator decidesthe actors for action, after waiting a common time. | It is possible that several actors can decide to act at same time; therefore wastage of resources is more. |
| 14. | Our approaches give more optimal actor set. | Give less optimal actor cover set. |

## 3. Related works

Mutual exclusion, as a fundamental problem in distributed systems has its variants, which is also found suitable for ad-hoc and sensor networks. Depending on the number of resources and nodes allowed to access the same shared resource, there are few variants of this problem. There may be the case such that only one site is allowed at a time to access shared resource or more than one sites or a group of sites (satisfying some particular conditions), for e.g., some nodes may need simultaneous access for a shared resource to perform some task. Depending upon nodes to allow for competing and access shared resources; mutual exclusion problem has been classified into different categories:

(1) Cluster-based mutual exclusion algorithms: Mellier & Myoupo (2005) proposed a clustering token based algorithm, which selects *cluster head* to act as a coordinator and forms cluster with the nodes in its communication range. It requires O($n$) broadcast rounds and a *cluster head* to act as local coordinator in the cluster. *Cluster head* is responsible for sending token to the nodes in its cluster. In this proposed paper, authors have assumed that nodes do not move while cluster is being formed and also nodes are not switched from one cluster to another cluster. The nodes are having a weight value is used to define its priority in network. It uses some flags to define the purpose of sending token; a token may be for use or needs to be forwarded to some other node. Once all nodes in the cluster are satisfied, then *cluster head* sends token to another cluster, directly or through heaviest intermediate node. This algorithm does not handle partition of the network and failure of *cluster head*. It also does not care if token holder or token destination fails. Although authors suggested to use PIF technique, for token lose. For energy saving the nodes of a cluster, which does not have token can be kept in to sleep mode. Haghighat & Mohamadi (2011) proposed a solution for *K*-mutual exclusion problem, which is based on clustering and combines the idea of Raymond (1989) and KRL algorithm proposed by Walter *et al.* (2001). The authors have replaced each cluster as a node to make a DAG (directed acyclic graph) of clusters like in KRL. Therefore, token management between clusters uses KRL algorithm and within cluster it uses Raymond's method. This algorithm defines three types of nodes, first: cluster head, manages other node within its cluster, second: nodes in range of two or more clusters called gateway nodes, third: other normal nodes i.e. members of a cluster other than gateway nodes and *cluster heads*. The channels are considered FIFO and each *cluster head* has three tuple ($h1$,$h2$,$c$), which shows height of a cluster $c$. The links are formed from higher height cluster to lower height cluster. The algorithm uses $k$ tokens, distributed in $k$ clusters, whenever any local node needs to access critical section (CS) and it sends request by Raymond tree. If any token is not present in its cluster then eventually request reaches to *cluster head,* which then makes request to other cluster heads. *Cluster head* uses priority queue to arrange these

requests, according to priority calculated from following equations:

$$S = (Total\ CS\ execution\ time + Total\ request\ in\ local\ queues + 1)$$

$$P = (total\ nodes\ in\ cluster)/S$$

Any *cluster head* having idle token sends it to requesting *cluster head*, which then forwards it to requesting local node. The authors proved that protocol insures safety, freedom from deadlock, and no starvation requirements; because system has *k* tokens at a time and only *k* nodes can be in CS, in same or different clusters. The simulation study of algorithm compares it with KRL and shows message overhead is less than KRL, on increased load it performs even better, and it also has lower response time than KRL.

Erciyes & Dagdeviren (2012) proposed a clustering permission-based algorithm, which can be called modified Ricart-Agrawala. It uses a ring of cluster coordinators for token circulation. Whenever any node needs to access the shared resource, it sends request to its cluster coordinator and coordinator is responsible to send request to next coordinator (if all waiting requests have larger timestamp than this request message) in ring, otherwise coordinator keeps the request in its *Queue*. After receiving its own message (coming after circulation) coordinator sends reply message to requesting node, indicating that now node can access CS. Myoupo *et al*. (2009) proposed a clustering group mutual exclusion protocol using weight throwing technique and adopted RL algorithm as proposed by Walter *et al*. (2001) after creating a hierarchical organisation of the network. They assigned weight and unique *Id* to each node and selects a node with maximum weight (with in cluster) as *cluster head*. Any node requires access to shared resource, it first sets its status to trying and sends request message to *cluster head*. If cluster head holds the token, it sends *Subtoken* to all its neighbours and any neighbour needs CS access can use this *Subtoken* to access CS and after use, sends it back to *cluster head*. Other Cluster heads have information of this opened session. So only after end of this session nodes of other clusters can be allowed to access CS. On receiving a *Subtoken* message the receiving node splits weight received and sends *Subtoken* to all neighbours, whose request is in *Queue*. If a node is *cluster head,* it can access (if holds token) CS and normal nodes can access CS if nodes have received *Subtoken* and they were requested for same resource. Therefore, it satisfies concurrent entering property required by a GME protocol. This protocol takes care of link failure and link farming during simulation study and message complexity of this algorithm is not presented.

(2) Non-cluster based mutual exclusion algorithms: The token based algorithm presented by Walter *et al*. (2001) is called RL (Reversing Link). It is an adaptation from wired network, which uses a unique token and maintains a DAG (directed acyclic graph). The property of a DAG is that a token holder node remains sink point, with

the help of partial reversal technique as proposed by Gafni & Bertsekas (1981). The token holder maintains its lowest height by reorienting links, which are from higher to lower height node. Each node has its height and the information of its adjacent nodes. To access shared resource a node need to send request message on one of its communication link, which eventually reaches to token holder over a tree and after that token is delivered with reverse path, of request path. In case of link creation and failure, partial rearrangement of DAG may be required. If after failure, no outgoing link is present, that means there is no path towards token holder so new route needs to be found, by partial rearrangement of DAG. The drawbacks of this protocol are: channels are assumed FIFO, token is considered reliable means it will not be lost however in real life it may lose, and starvation is also possible in this protocol. Baldoni *et al*.(2002) proposed an algorithm which combines both token asking and circulation, and uses logical dynamic ring created on-the-fly. It selects one node as a coordinator (for one cycle) and other nodes can send their requests to the selected coordinator node. It works in two states: idle state and coordinator change state. Whenever coordinator receives request from any node, it starts token circulation, otherwise token need not to be circulated. While circulating token, any node which has not yet received token can be chosen as successor (next node which will receive token), depending upon given policy such as: nearest hop node or minimum remaining energy node etc. During token circulation, when coordinator receives back the token, the protocol enters in idle state and after choosing next coordinator (the first process in pending requests), it enters in the coordinator change state. The algorithm greatly reduces message complexity on low load and on heavy load, and the optimal value for each CS access is two application messages. It also reduces nodes' power consumption by reducing number of hops used to execute critical section. This algorithm also assumed that nodes and links are reliable and token will not lose i.e. it does not deal with fault tolerance. The complexity of the algorithm is not presented although simulation study of protocol shows number of hops per CS access as a function of request load and average number of hops (in percentage) per application level message at high, medium and low mobility settings.

Wu *et al*. (2008) presented a permission based algorithm. It uses "look-ahead" technique proposed by Singhal & Manivannan (1997), was introduced for infrastructure mobile network. They have removed constraint of FIFO channel and used timeout mechanism to deal with fault tolerance. The look-ahead technique uses Lamport's time stamp algorithm proposed by Lamport (1978) and global priority for every request. Each node has *Info set*, includes the *Id* of nodes to whom this node will inform and *Status set* includes the *Id* of nodes by whom this node will get informed. The authors have also used modified algorithm as proposed by Ricart & Agrawala (1981), which involves only those sites which are currently competing. Wu *et al.* (2008) also proposed a technique to initialize *Info set* and *Status set*, introduced fault tolerance to

handle node and link failure, and introduced a *Queue (Q)* to relax FIFO constraint. When a node need access to shared resource, it request to all nodes in its *Info set,* with timeout and then it waits for reply. On receiving request message, from any node, it records that request in *Q* and sends reply message, if it is not wishing for CS or its priority is low. After sending reply, it removes *Id* of that requesting node from *Q* and moves requesting site to its *Status set*. On receiving reply message, site checks its *Q* and timestamp*,* if there is no request in *Q* then it moves requesting node to *Status set,* and then it can enters in the CS. This algorithm is able to deal with disconnections and doze mode. Whenever any node wants to go away, it sends message to other nodes, then both nodes, receiver and sender of this message modifies information in both of its sets, namely *Info set* and *Status set*. In simulation study of this algorithm, the authors used two measures, "number of messages per CS entry (MPCS)" and "number of hops per CS entry (HPCS)". The simulation shows that protocol is scalable for large system and both MPCS and HPCS increases as system scales. To study the effect of mobility, the simulation is performed in three different mobility settings, under low mobility both measures are higher than in high mobility.A permission based algorithm has been presented by Parameswaran & Hota (2010) which uses an adaptive timeout method to deal with failure of nodes and links and in case of timeout the new value for timeout is calculated as:

$$t = t + (2T - t) * g$$

Growth value *g,* used to progressively increasing or decreasing the timeout value and $2T$ sets the upper bound for timeout. They have also used "look-ahead" technique presented which was later extended by Wu *et al*. (2008). A new message type HOLD is introduced, which finds how long a wishing node should wait for a node (currently in CS). This HOLD message also helps to decide whether a node is taking long time to execute its CS or it failed in CS. Any site wants to access shared resource, sends REQUEST message to all sites in its *Info set* and on receiving request from a site, receiver sends REPLY or HOLD message to requesting site. If it is not in CS it sends REPLY message and if it also wants CS, then after checking priority of requesting site, it sends REPLY (if requesting node has higher priority) message or HOLD (if it has higher priority) message with expected time of completing CS. After sending HOLD message it adds that site to $Q_{hold}$. This protocol adaptively chooses new timeout values based on past experiences, but how to choose value of *g* (growth metric) is not discussed. Although suggestion has been given that it should not be too high or too low. They have proven both correctness and fault tolerance (both link and site failure) of this algorithm. One of the main drawbacks of the protocol that it has large message complexity and depends on number of HOLD messages sent to competing sites. To reduce number of HOLD messages they suggested "site can delay in sending HOLD message" although, it is left as future work.

An algorithm presented by Malpani *et al*. (2001), creates a dynamic ring and circulates token on that ring. Whichever node receives token, can access CS and then, token needs to be forwarded to next site, which is decided by calculating number of sites visited and factors like: recent visited, least visited. Information needed for this calculation can be obtained from local or global information of all sites in the network, and this information comes from periodic hello message; send by each node to inform its neighbours about itself. Chen &Walter (2002) presented a self-stabilizing token based algorithm which forms dynamic virtual rings by token circulation. Self-stabilization was firstly defined by Dijkstra (1974) as: "If system starts from an arbitrary configuration, it is guaranteed to converge to a legitimate configuration in finite time". This protocol allows only limited topology changes and needs network topology to be static while algorithm is converged. It is based on LRV as proposed by Malpani *et al*. (2001). In this it forwards token to a node, which is neighbour and least recently visited by the token. Each node (with restricted mobility) gets token, while it is being circulated and after getting token, site checks token status and its own local state, then it can access CS. To forward a token to next site, it checksthe *token holder list* (indicates sequence of sites who were having the token), to decide which neighbour had used the token least recently. It guarantees safety and liveness properties under limited mobility, under unrestricted mobility; it fails to assure the liveness property. The constraint, "topology should remain static during stabilization" limits this algorithm to be used in MANETs where numerous static time intervals, having length more than stabilization time are present. Later Chen & Welch (2005) presented another version of this self-stabilising protocol, which removes the constraint "static topology during stabilization" and this version can handle dynamic membership i.e. sites wanted to enter CS can be changed. However, this version is also not able to guarantee livness in arbitrary mobility. Table 2 shows the comparison table between various proposed approaches based on different parameters.

**Table 2.** Comparison table of different proposed approaches based on different parameters

| Algorithm Used | Response Time | | Number of Messages per CS Execution | | Fault Tolerant | Energy Consideration | Synchronization Delay | Non FIFO Channel Support |
|---|---|---|---|---|---|---|---|---|
| | Light Load | Heavy Load | Light Load | Heavy Load | Support | | | |
| Look-Ahead | $2T$ | $(E+1)*n/2$ | $N$ | $3N/2$ | Yes | No | $T$ | Yes |
| Permission with Clustering | $2T$ *to* $2T(m+1)$ | | $O(2)$ *to* $O(n)$ | | No | No | $2T$ *to* $2T(m+1)$ | No |
| Look-Ahead | $2T$ | $(E+1)*n/2$ | $2\|\Phi\|$ | | No | No | $T$ | No |
| Token in Clusters | $T(k+1)$ | $w((k+1)T+E)$ | $K+1$ | | No | Yes | $2T$ *to* $2T(k+1)$ | No |
| Permission with Clustering | $T(k+3d)$ | $w((k+2d-1)T+E)$ | $K+3d$ | | yes | Yes | $2dT$ *to* $T(k+2d-1)$ | No |
| Token in dynamic ring | *Depends on policy used* | | $N-2$ | 2 | No | Yes | $T$ | Yes |
| Token in Clusters | $2T$ *to* $(m+1)*T$ | | $1+m\ \frac{1}{n}+2$ | | Yes | No | $T$ | Yes |
| Look-Ahead | $2T$ | $(E+1)*n/2$ | $2(\Phi-1)+p*w$ | | Yes | No | $T$ | Yes |

## 4. Our proposed centralized approach (cpmea)

Here, we present an algorithm called centralized prioritized *h-out-of-k* mutual exclusion algorithm (CPMEA). It computes actor cover set, *M* for minimum number of actors and for minimum overlap with two different priority calculation mechanisms. Based on networks' knowledge sink node computes actor cover set or selects actors for action, which we call actor cover set. Depending upon the priority function used by sink node CPMEA can compute priority and then selects the actors based on that priority, hence there will be minimum number of actors or minimum overlapping between their coverage.

Whenever any event occurs within the range of any sensor node in the network, it detects and sends along with its location to the sink node. Based upon the reports received from sensor nodes, sink node calculates the event region i.e. its location and radius. We used very simple approach to calculate event area by taking minimum and maximum *x,y* coordinates from the observed readings of all the sensors. Then

calculate centre point, as midpoint of these x and y, and radius as distance between centre point and maximum or minimum x and y. This will cover all points where event has happened. The following steps are used to calculate actor cover set:

(1) The sink node having event centre, radius and location of each actor will identify the actors, which are intersecting the event region i.e. *intersecting_actor_set* As mentioned above, it is considered that each point of the event region is the copy of the shared resource to be used or accessed by *intersecting_actor_set.*

(2) Using priority function, sink will calculate priority of each actor from *intersecting_actor_set* till all the points of event region are having non-zero degree means every point is covered by at least one actor.

(3) Now our approach selects actors one by one according to their priority (i.e. selects higher priority actor first and then low priorityand so on) till all the points in the event region are covered or accessed. Let it is called *M* i.e. set of finally selected actors for action.

Figure 1 shows the pseudo code of our proposed CPMEA approach.

Inputs

$a_1,a_2,\ldots,a_n$: Actors in the WSAN

S: Sink

ER:Radius of event Region

EC: Centre of event region

AR: Acting range of an actor

Output

The minimal actor cover set M that covers event region OR minimum overlap cover set M

Compute the minimal actor cover or minimum overlap cover for event region

Procedure CPMEA ()

M: The set of actors selected as part of actor cover at any given stage

RM: Region covered by M actors

Ω: Set of actors whose acting range intersects with event region

PRIORITY: Set of priority of intersecting actors

1. $\omega = \Omega$
2. $P = 1$
3. PRIORITY = priority_min_actor (ER, EC, AR, ω) or priority_min overlap (ER, EC, AR, ω)

4. While (Degree of each point in ER = 0)
5. For each $a_i \in \omega$
6. if (priority of $a_i$ = P)
7. SELECTED = $a_i$
8. P = P + 1
9. end if
10. end For
11. M = M ∪ SELECTED
12. RM = RM + new area coverd by $a_i$
13. ω = ω - SELECTED
14. End while
15. Send command to actors in set M

**Fig. 1.** Pseudo code of our proposed CPMEA Algorithm

## 5. Our proposed distributed approach (DPMEA)

Here, we present our proposed distributed prioritized *h-out of k* mutual exclusion algorithm (DPMEA). It computes actor cover set in a distributed way (i.e. without using sink node or by coordination of actor nodes). In this approach, after sending the event information by sensor nodes to an actor node as decided by single actor (**SA**) or multiple actors (**MAs**) selection methods, actor node (s) can compute the partial event region. Whenever an actor receives report from sensor nodes, it will calculate partial event region and then it can share that information with other actor nodes, which may also have received some reports from same or other sensor nodes. Further, to compute actor cover set, our protocol runs on initiator actor, which has received report firstly. The following section shows single actor selection and multi actor selection procedures in this proposed approach:

a. For SA selection: In this it is assumed that all sensor nodes (after coordination), selects same actor for sending their reports. Therefore, in single actor selection, a single actor will receive whole event information from all sensors and it will also receive position of all sensor nodes or location of event as sensed by sensor nodes. There are two possible cases for the selected actor:

(1) If selected actor is inside the event area, it can announce event information to all actor nodes within the range of 2*[*Event radius + Acting range*] radius. Otherwise, it can forward this event information to an actor within event region, and then select edactor will send event information in 2*[*Event radius + Acting range*] radius.

(2) Every actor, having acting range in the event area knows whole event area. Thus, these actor nodes can calculate their intersection with event region. Actors, which are having non null intersection of their acting area with the received event area will replay back their location, acting range, residual energy etc. Now, initiator actor has all the information needed to calculate priority of actors and thus actor cover set, for both minimum number of actors and minimum overlaps is calculated.

b. For MA Selection*:* Sensor node selects an actor to send their event information based on some criteria like nearest actor or highest energy actor etc. Therefore, different sensors select different actors for sending their readings. The following steps are used in the pseudo code as shown in Figure 2 to select the multiple actor nodes in our proposed protocol:

(1) When sensors select multiple actor nodes to send their readings i.e. MA, every actor who receives event information will update its flag to initiator address or sets own address as initiator (if it is first to receive the reading). Here every actor can calculate partial information of whole event region.

(2) The actor node that first receives event information will broadcast its own *id* as initiator and all other actors will set this *id* as destination. Actors, who have received event information will replay partial estimated event region to the initiator actor, either every time they receive new report or when asked by initiator.

(3) Now initiating actor has information of whole event area (calculated by union of partial event areas received from other actors). Further, initiator will broadcast complete event area or disseminate in 2*[*Event radius + Acting range*] radius region.

(4) Each actor (which may be selected for action) is aware of whole event area, his acting range, and his location. Actors have non-zero intersection with the event region; only those actors will participate in future communication.

(5) All actors (within and nearby event region) which may not be evoked by any sensor node will also be evoked. Then the actors who have non null intersection with this event area will send their location back to initiator.

(6) After previous step, initiator actor has all the information needed to calculate priority of actors and thus actor cover set, for both minimum number of actors and minimum overlaps.

The following section shows the pseudo code of our proposed DPMEA approach.

Inputs

$a_1, a_2, \ldots, a_n$: Actors in the WSAN

AR: Acting range of actor

Output

The minimal actor cover set M that covers event region OR minimum overlap cover set M

Variables

M: The set of actors selected as part of actor cover set,

RM: Region covered by M actors,

ER: Radius of event Region,

EC: Centre of event region,

ω: actors whose acting region intersects with event region,

PRIORITY: Set of priority of intersecting actors

INITIATOR: Actor selected for coordination

Compute the minimal actor cover or minimum overlap cover for event region

Procedure DPMEA ()

1. Actor $a_i$ receives first event report
2. Set INITIATOR = a_i
3. Broadcast MSG (INITIATOR)
4. for all $a_i \neq$ INITIATOR
5. Set DESTINATION = $a_i$
6. $FRW_{reports}$ to INITIATOR //all actors will send partially computed event region to initiator.
7. End for
8. Based on partial event regions initiator will calculate event region, ER, EC.
9. $MSG_{ER,EC}$ to actors within 2* [ER + AR]
10. for all $a_i \neq$ INITIATOR
11. if ($AR_i \cup ER \neq NULL$)
12. $MSG_{Location,AR}$ to INITIATOR
13. End if
14. End for
15. ω = set of actors from which initiator had received location
16. P = 1

17. PRIORITY = priority_min_actor (ER, EC, AR, ω) or priority_min_overlap(ER, EC, AR, ω)
18. While (Degree of each point in ER = 0)
19. For each $a_i \epsilon \omega$
20. if (priority of $a_i$ = P)
21. SELECTED = $a_i$
22. P = P + 1
23. end if
24. end For
25. M = M ∪ SELECTED
26. RM = RM + new area coverd by $a_i$
27. ω =ω-SELECTED
28. End while
29. Send command to actors in set M

**Fig. 2.** Pseudo code of our proposed DPMEA Algorithm

## 6. Mathematical analysis of our proposed algorithms

In this section, we analyse the communication complexity (CC) and time complexity (TC) of our proposed protocols i.e. CPMEA and DPMEA and compare with state-of-the-art approaches i.e. Vendatham's centralized and distributed algorithm, CACI and DACI. The communication complexity measures the number of 1-hop transmissions required by the algorithm to perform the action. The time complexity, or event to action delay, measures the time difference between the occurrence of the event and the execution of its corresponding action. The worst case complexities (i.e. CC and TC) are shown in Table 3. It is assumed that *N*actor nodes are distributed in the field *A*. The node density remains constant when the number of nodes increases, and the area *A* grows with *N*. Since the expected distance of two uniformly sampled points within a square of size $l \times l$ scale with $l$, it is expected that the number of hops between two random nodes in *A* increases proportional to $\sqrt{N}$. The number of sensors with in an event region of radius $a$ is proportional to. The distance between the event and the actor in Ω is denoted by *D*. The maximum distance between two actors in Ω is $(2(a + AR))$. As *AR* is constant, the number of actors intersecting with *R*, which is |Ω|, is proportional to $a^2$. In the table, *W* denotes the actor's waiting time in NB algorithm, *Y* denotes the waiting time to ensure reception of all the events withinthe event region and *I* denotes the size of the constructed actor cover set.

In the centralized algorithms, the $a^2$ sensor nodes within the event region inform the sink node about a new event. The sink sends a command message to each actor in the actor cover set. As the average number of hops between two random nodesis proportional to $\sqrt{N}$ and $a$ is upper-bounded by $R_{max}$, this operation leads to a communication cost of $O\left((I+R^2_{max})\sqrt{N}\right)$ and a response time of $O(2\sqrt{N}+Y)$.

In Vendatham's approach, when a sensor detects a new event, it sends *REQUEST* message to all the actors in its dependency region. This notification phase incurs a communication cost of $O(a^2 2|\Omega|D_s)$. In the coordination phase, each actor sends *a NOTIFY* message to $|\Omega|-1$ actors in its dependency region in order to construct the actor cover set, which leads to a communication cost of $O(|\Omega|^2 DA)$. As *DS* and *DA* are constants, the overall communication is proportional to $a^4$ . An event needs to travel a distance proportional to $O(D)$ to reach an actor in $\Omega$. Each actor waits for $O(W)$ time to receive the rest of events, and then waits for another $O(Y)$ time before acting, which leads to an event-to-action delay proportional to $O(D+W+Y)$.

In DACI, a sensor node disseminates the event within a zone of radius equals to $(2Rmax+AR)$. This notification cost incursa communication cost of $O(a^2 \times R^2_{max})$ and a time complexity of $O(D)$. After receiving the first event, each actor needs to wait for $O(Y)$ time before executing the algorithm. The main advantage of DACI is that there is no need for signalling overhead to execute the coordination phase, which leads to low event-to-action delay. Also, its communication cost is bounded by $O(R^4_{max})$ as NB algorithm.

In our proposed CPMEA, a sensor node sends its report to sink, so upper bound for communication cost is $ER^2$, same as CACI. To send a command to the selected actor cover set it requires $I$ messages. Therefore, overall communication cost is $O\left((I+ER^2)\sqrt{N}\right)$. In our algorithm, again $a^2$ sensor nodes within the event region inform sink node about new event. The sink sends a command message to each actor in the actor cover set. As the average number of hops between two random nodes is proportional to $\sqrt{N}$ and $a$ is upper-bounded by $R_{max}$, this operation leads to a communication cost of $O\left((I+R^2_{max})\sqrt{N}\right)$ and a response time of $O(2\sqrt{N}+Y)$ which is similar to CACI.

In DPMEA, first communication is from sensors to actors, selected for reporting. Let there are T actors in radius of 2* $[ER+AR]$ and amongst them $K$ actors are selected by sensor nodes of event region to report their data. $ER^2$ are messages from sensor to actors for reporting. Then $K-1$ messages are used from *INITIATOR* to all $K$ actors to tell them about their coordinator. At end of reporting phase actors, other than *INITIATOR* will send their partial event region to *INITIATOR*. It incurs cost of $K-1$ messages. After this initiator will send messages to all actors in 2* $[ER+AR]$radius that incurs cost of $|\Omega|$ messages. Then all the actors intersecting with event region will send their location and this incurs cost of $|\Omega|$ messages. Finally,

initiator will send command to select the actors which incurs cost of $I$ messages. So, overall communication cost is $O\,(ER^2 + 2T + K - 1 + |\Omega| + I)$. An event needs to travel a distance proportional to $O\,(D)$ to reach an actor in $\Omega$. Each actor node waits for $O\,(W)$ time to receive the rest of events before acting, which leads to an event-to-action delay proportional to $O\,(D + W)$, which is less than any state-of-the-art approaches.

**Table 3.** Comparison table for worst case complexities (CC and TC) of different proposed approaches

| Algorithms' Name | Communication Complexity (CC) | Time Complexity (TC) |
|---|---|---|
| Vedantham's centralized | $O\,((I+R^2_{max})\sqrt{N})$ | $O\,(2\sqrt{N} + Y)$ |
| Vedantham's distributed | $O\,(R^4_{max})$ | $O\,(D + W + Y)$ |
| CACI | $O\,((I + R^2_{max})\sqrt{N})$ | $O\,(2\sqrt{N} + Y)$ |
| DACI | $O\,(R^4_{max})$ | $O\,(D + Y)$ |
| CPMEA | $O\,((I + ER^2)\sqrt{N})$ | $O\,(2\sqrt{N} + Y)$ |
| DPMEA | $O\,(ER^2 + 2T + K - 1 + \|\Omega\| + I)$. | $O\,(D + W)$ |

Whereas $N$: Number of actor nodes, $D$: Distance between actor node and event area, $W$: Actor's waiting time, $Y$: Waiting time for the reception of events, $I$: Size of actor cover set, $R_{max}$: Maximum range of an actor node

## 7. Simulation evaluation of our proposed approaches

We have used $C++$ programming language to implement our proposed protocols and simulate on OMNeT++ with Castalia wireless sensor network simulator. The degree of each point within the event region is implemented by checking the position of each point with respect to the actor nodes and intersection with event region. To compute actors cover set intersect with event region, we have used an algorithm to calculate the intersection between two circles, one circle is fixed i.e. event region and another circle is acting range of an actor node. Therefore, one by one we check intersection of each actor with event region, if one has then it is included in *intersecting_actor_set.* In our simulation experiments, we have taken100-600 actor nodes with acting range varies from 40m-100m are deployed in an area of $400m \times 400m$ to ensure full coverage of event region. Nodes are static actor nodes and cover the whole network region. Further, actors are deployed in a uniform grid. Our algorithm is flexible enough, so that we can change the number of actors and their deployment very easily in the configuration. We generate results for our proposed protocols with different density of actor nodes or number of actor nodes. We have used static physical process to report event information, means that the values are static for each sensor node in the experiment. Table 4 shows the simulation parameters used in our implementation and simulations.

**Table 4**. Simulation parameters

| Parameter | Value |
|---|---|
| Simulation Area | 400m x 400m |
| Actor Nodes | 100-600 |
| Topology pattern | 10 |
| Radio Model | Path loss model |
| MAC Layer | IEEE 802.15.4 |
| Routing Protocol | Multi path |
| Signal Delivery Threshold | -100dBm |
| Communication Range ($r$) | 40m-100m |
| Node Initial Energy ($E_i$) | 51 joules |
| Channel Frequency | 2.4GHz |
| Packet Size | 512 bytes |
| Antenna Model | Omni-directional |
| Data Transmission Rate | 15 packets/sec |
| Simulation Time | 100s |

In our experimental results, each plotted point represents the average of 30 executions. We have taken following different state-of-the-art metrics for measuring performance of our proposed approaches:

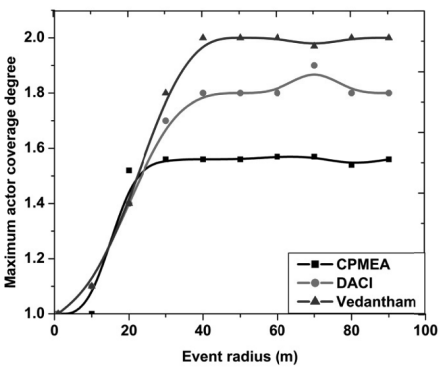


**Fig. 3.** Maximum actor coverage degree vs. event radius

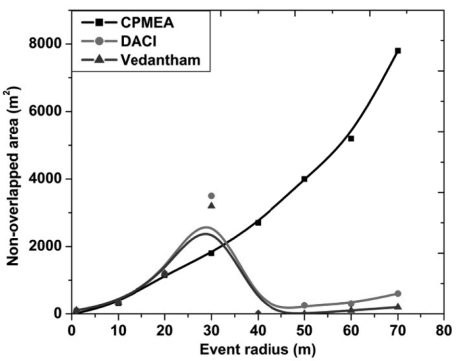


**Fig. 4.** Non-overlapped with 200 actor nodes vs. event radius

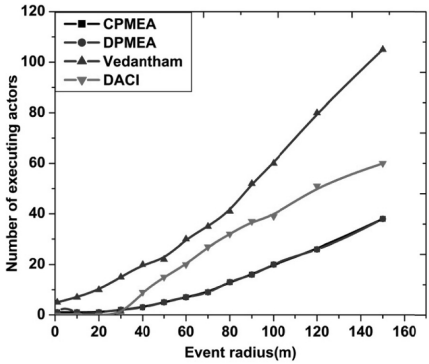


**Fig. 5.** Number of executing actors vs. event radius

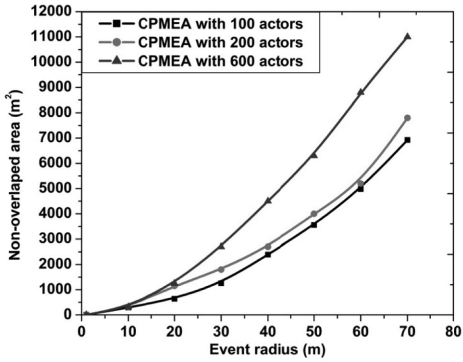


**Fig. 6.** Non-overlapped area vs. event radius

- Number of actors*:* is the number of actors selected by the algorithm to cover the entire event region. Mathematically, we can prove how our protocols minimize the number of actor nodes to cover the entire event region and verify with simulations as shown in Figure 3.

$$Res(\xi) = \sum_{ai \in \Omega} Size(Ai \cap R)$$

•

$$= \sum_{i=1}^{\Delta(\xi)} \sum_{a \in Si} i * Size(a)$$

$$Res(\xi) = \sum_{i=1}^{\Delta(\xi)} \sum_{a \in Si} i * Size(a) = \sum_{a \in Si} Size(a) + \sum_{i=2}^{\Delta(\xi)} \sum_{a \in Si} i * Size(a)$$

$$= \sum_{a \in S1} Size(a) + \sum_{a \in S2} Size(a) + \sum_{a \in S3} Size(a) + \sum_{a \in S4} Size(a) + \sum_{a \in S5} Size(a) + \cdots$$

$$= \sum_{a \in S_{\Delta(\varepsilon)}} Size(a) + \sum_{a \in S_{\Delta(\varepsilon)}} (\Delta(\epsilon) - 1) * Size(a)$$

$$= Size(R) + \sum_{i=2}^{\Delta(\xi)} \sum_{a \in Si} (i - 1) * Size(a)$$

Where $\xi = (R, \Omega)$ is called actor-coverage configuration, $\Omega$ is set of intersecting actors, $R$ is radius of event region, $EC$ is centre of event region, $AR$ is acting range of actors, $\Delta(\xi)$ is maximum actor coverage degree, $S_i$ is set of sub-regions within event region, $Size\ (a)$ is area covered by actor $a$ and $i$ is $id$ of actor node.

- Size of non-overlapping areas: is defined as the size of areas within the event region that are covered by only one actor.
- Size of overlapping areas: is defined as the size of areas within the event region that are covered by more than one actor.
- Acting cost: It is defined as the amount of resources dissipated on the event region, which is the sum of the required resources and the extra resources.
- Extra cost: is defined as the amount of extra resources dissipated on the event region.
- Maximum actor coverage degree: is the average maximum number of actors that can cover any point in the event region.

Figure 3 shows results for maximum average number of actors that can cover any point in the event region. Any point in the event region may be covered by more than one actor. Because of increasing of overlapping area with event radius, the actor coverage degree is also increased with event radius. The maximum degree is 1.56, which shows that there will be never any point where more than two actors will act. This shows that our algorithms give minimum actor coverage degree as compared with state-of-the-art approaches. It also shows that our approach will waste minimum resources as compared to other proposed algorithms. As event radius is increased, the maximum actor coverage degree is also increased and becomes constant at 2 for Vedantham'salgorithm and 1.56 for our proposed protocols.

In Figure 4, we compare all proposed approaches based on non-overlapped area with 200 actor nodes. Our algorithms performance is better, as we increase the number of actor nodes as shown in Figure 4. Figure 4 also shows non-overlapped region with large number of actor nodes deployed. It is clearly observed from the plot that the overlapped region is increased with event radius because number of actors for action is more. Therefore, it also increases the chance of overlaps. However, if we increase number of actors deployed then high chance of selecting an actor node with less overlap, more number of actors means node density is more. Thus, with increased density of actor nodes the overlaps will reduce. However, computation cost will also increase. The results clearly show that our algorithm gives best performance, because it has large non-overlapped area as compared to DACI and Vedantham's algorithm. In Figure 4, we have shown that reduction in overlap area or increase in non-overlapped area, as we have changed the number of actor nodes.

Figure 5 shows results for number of executing actors as selected by different algorithms. We called this set as minimum actor coverage set (MACS). The curves of CPMEA and DPMEA are identical because both have computed the same actor cover set on similar event radius. The results for DACI and CACI are similar, so we have taken only DACI approach in the plot. It is clear that when event radius is less than $35m$, all algorithms have selected average one actor for full coverage, because this radius can cover by a single actor node. As we have changed the acting range from $35m$ but keep less than $40m$, then one actor is enough to act. After this radius our algorithm performs better than both approaches. Although Vedantham's approach has used very large number of actors, result for our protocol remains same even with large number of actor nodes are being used. As we increase the event radius, the size of actor cover set is also proportionally increased.

Figure 6 shows our proposed approach CPMEA with different number of actor nodes. It is clear that as number of actor nodes is more, non-overlapped area has been reduced due to large number of deployment of actor nodes with small event radius.

## 8. Conclusion and future scope

We have proposed two approaches to address mutual exclusion problem in WSAN. We have also identified few new challenges to address this problem. In our proposed approaches, we consider whole event area is divided in unit points, which are like identical copies of shared resource to be used or accessed by actor nodes. Our proposed approaches focus on minimizing the number of actors selected for action and minimizing the overlapped area of action to reduce waste resources. We have proposed centralized prioritized h-out-of-k mutual exclusion algorithm (CPMEA), and its distributed version (i.e. DPMEA*)*. We have defined priority functions for deciding priority of intersecting actors; one is for selecting minimum number of actors and other for selecting actors such that overlap is low. Through simulations we compare the performance of our algorithms with state-of-the-art approaches i.e. Vedantham's algorithm, DACI, CRMEA, and CACI. Our both algorithms, CPMEA and DPMEA, performs better than previously presented algorithms in terms of, maximum actor coverage degree, size of actor cover set, and non-overlapped region. In the future we will extend our work to support multiple events. When there is more than one event happening simultaneously in the network, to solve such situation, an extra variabl *event id* canbe used to differentiate between different events and actors can be chosen based on *event id* as their *flag*, if actor intersects the event area corresponding to that *event id*. Also we will take care of the situation when event area is increasing or decreasing while action is being performed. There may be different types of actor in a single network like gas, water etc. means network is having heterogeneous actors. This type of actor selection problem can also be solved by extending our present

algorithm.We will also consider energy parameter, while deciding priority to select minimum number of actor nodes. It will not only reduce actor intermittent failure while acting,but also maintain balanced energy of the actors with in network to extend network lifetime. Fault tolerance can also be added to our proposed protocols in the future. If an actor node act as an initiator fails intermittently then how to select a new initiator or coordinator and if any other actor fails then what to do. One solution is, at the time of deciding priority, we can calculate priority for all actors in intersecting set and whenever an actor fails, we immediately select next priority actor that we already have.

## References

**Akyildiz, I.F. & Kasimoglu, I.H. (2004)** Wireless sensor and actor networks: Research challenges. Ad Hoc Networks, (Elsevier), **2**(4):351-367.

**Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002)** Wireless sensor networks: a survey. Computer Networks, **38**(4):393-422.

**Baldoni, R. Virgillito, A. & Petrassi, R. (2002)** A distributed mutual exclusion algorithm for mobile ad-hoc networks. Proceeding of the 7th IEEE Symposium on Computer and Communications (ISCC 2002), Italy, 539-545.

**Chen, Y. & Walter, J**. (**2002)** Self-stabilizing mutual exclusion using tokens in mobile ad hoc networks. Proceeding of the 6th international workshop on Discrete algorithms and methods for mobile computing and communications, 34-42.

**Chen, Y. & Welch, J.L. (2005)** Self-stabilizing dynamic mutual exclusion for mobile Ad Hoc Networks. J. Parallel and Distributed Computing, **65**(9):1072-1089.

**Cheng-Zen, Y. (2005)** A token-based h-out of-k distributed mutual exclusion algorithm for mobile ad hoc networks.Proceeding of the 3rd International Conference on Information Technology: Research and Education, (ITRE 2005)73-77.

**Derhab, A. & Zair, M. (2010)** A resource-based mutual exclusion algorithm supporting dynamic acting range and mobility for Wireless sensor and Actor Networks. Proceeding of the 6th IEEE International Conference on Distributed Computing in Sensor Systems Workshops, (DCOSSW)1-6.

**Derhab, A. & Lasla, N. (2011)** Distributed algorithm for the actor coverage problem in WSN-based precision irrigation applications. Proceeding of the 2011 International Conference on Distributed Computing in Sensor Systems and Workshops, (DCOSS)1-6.

**Dijkstra**, **E.W. (1974)** Self stabilization in spite of distributed control. Communications of the ACM, **17**: 643–644.

**Erciyes, K. & Dagdeviren, O**. (**2012)** A distributed mutual exclusion algorithm for mobile ad hoc networks. International Journal of Computer Networks and Communications, **4**(2):129-148.

**Gafni & Bertsekas, D. (1981)** Distributed algorithms for generating loop-free in networks with frequently changing topology. IEEE Transactions on Communication, 11-18.

**Haghighat, A.T. & Mohamadi, M.R. (2011)** Cluster-based *K* mutual exclusion for mobile ad hoc networks. Proceeding of the 5th International Conference on Application of Information and Communication Technologies, (AICT)1-6.

**Hu, B., Xie, Q., Bao, M. & Dong N. (2014)** Improvement of user authentication protocol with anonymity for wireless communications. Kuwait Journal of Science, **41**(1):155-169.

**Lamport, L**. (**1978)** Time, clocks and ordering of events in distributed systems. Communications of the ACM, 558-565.

**Malpani, M., Vaidya, N. & Welch, J**. (**2001)** Distributed token circulation in mobile ad hoc networks. Proceedings of the 9th ICNP, 154-165.

**Melodia, T., Pompili, D., Gungor, V.C. & Akyildiz, I.F.** (**2005)** A distributed coordination framework for wireless sensor and actor networks.proceedings of the ACM symposium on mobile Ad Hoc networking and computing (ACM MOBIHOC), Chicago, IL, USA, 99-110.

**Mellier, R. & Myoupo, J.F.** (**2005)** A clustering mutual exclusion protocol for multi-hop mobile Ad Hoc Networks.proceeding of the 13th International Conference on Networks, (ICON'05), 250-255.

**Myoupo, J.F., Naimi, M. & Thiare, O. (2009)** A clustering group mutual exclusion algorithm for mobile Ad Hoc networks.Proceeding of the IEEE Symposium on Computers and Communications, (ISCC), 693-696.

**Parameswaran, M. & Hota, C**. (**2010)** A novel permission-based reliable distributed mutual exclusion algorithm for MANETS. Proceeding of the International Conference on Wireless And Optical Communications Networks (WOCN), 1-6.

**Raynal, M.** (**1991)** A distributed solution to the k-out of-M resources allocation problem. Proceeding of the International Conference on Computing and Information: advances in computing and information, (LNCS-497), 599-609.

**Raymond, K.** (**1989)** A tree-based algorithm for distributed mutual exclusion. ACM Transactions on Computer Systems, 61-77.

**Ricart, G. & Agrawala, A.K.** (**1981)** An optimal algorithm for mutual exclusion in computer networks. Communications of ACM, **24**(1):9-11.

**Salarian, H., Chin, Kwan-Wu & Naghdy, F.** (**2012)** Coordination in wireless sensor–actuator networks. Journal of Parallel and Distributed Computing, **72**(7):856-867.

**Singhal, M. & Manivannan, D.** (**1997)** A distributed mutual exclusion for mobile environments. Proceeding of the IASTED Intl. Conf. on Intelligent Systems, 557-561.

**Vedantham, R., Zhuang, Z. & Sivakumar, R.** (**2006)** Mutual exclusion in wireless sensor and actor networks. Proceeding of the 3rd Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2006), Reston, VA, 346-355.

**Walter, J., Gaungton, C. & Mitrabhanu, M.** (**2001)** A *K*- mutual exclusion algorithm for wireless Ad Hoc networks. Proceeding of the first annual ACM workshop on Principles of Mobile Computing (POMC 2001), Newport, Rhode Island, USA, 1-11.

**Walter, J.E., Welch, J.L. & Vaidya, N.H.** (**2001)** Mutual exclusion algorithm for Ad Hoc Mobile Networks. Wireless Networks, **7**(6):585-600.

**Wu, W., Cao, J. & Yang, J.** (**2008)** A fault tolerant mutual exclusion algorithm for mobile ad hoc networks. Pervasive and Mobile Computing, **4**(1):139-160.

***Submitted*** **:** 02/08/2014

***Revised*** **:** 08/01/2015

***Accepted*** **:** 10/01/2015

# تمييز الوحدات المثالية في شبكات الاستشعار و النقل اللاسلكية باستخدام أطروحة استشعار التشارك ذي الأولوية

**فيرندررانجا، [1] مايانك ديف، [2] أنيل فيرما [1،*]**

[1] قسم هندسة الكمبيوتر – المعهد الوطني للتكنولوجيا – كوروكشيترا – هيريانا – الهند

[2] قسم علوم وهندسة الحاسوب – جامعة ثابار – باتيالا البنجاب – الهند

[*] مؤلف التواصل : virender.ranga@nikkr.ac.in

## خلاصة

ندرس في ورقة البحث هذه مسألة استبعاد التشارك في سياق شبكة الاستشعار و النقل اللاسلكية (WSAN) ونقدم أطروحتين مستحدثتين لحلها. إن المتطلبات الأساسية لأي أطروحة ضمن هذا السياق هي تمييز الحد الأدنى من الوحدات الناقلة و اللازمة للتعامل مع منطقة الحدث قبل البدء ، تقليل المساحات المشتركة بين النطاقات الفاعلة لادني حد ، تقليل هدر الموارد ، واخيرا : تغطية تامة لمنطقة الحدث بواحدة او أكثر من الوحدات الناقلة و بحسب قابليتها. لقد تقدمنا في ورقة البحث هذه بخوارزميتين هما خوارزمية استبعاد التشارك المركزية ذات الاولويات س من ص (CPMEA) وخوارزمية استبعاد التشارك الموزعة ذات الاولويات س من ص (DPMEA). كلا من الأطروحتين تقوم بإنشاء مجموعة لتغطية النواقل بمثالية متقاربة. إن نتائج المحاكاة تقيم الأداء من حيث حجم مجموعة تغطية النواقل ، ومناطق التشارك ، ومناطق عدم التشارك والدرجة القصوى لتغطية النواقل. كما وقمنا أيضا بمقارنة نتائجنا مع سابقاتها من الخوارزميات القياسية المقترحة.

**كلمات مفتاحية** : سياق س من ص؛ استبعاد التشارك ؛ مجموعة تغطية النقل ذات الأولويات ؛ استشعار لاسلكي ؛ شبكات النقل.