A Distributed Coverage-based Connectivity Restoration in Wireless Sensor Networks

Shiow-Fen Hwang, Jing-Huang Yang, Wen-Lin Chao, Chyi-Ren Dow

Feng Chia University, Taiwan

0306

The Asian Conference on the Social Sciences 2013

Official Conference Proceedings 2013

Abstract

In recent years, wireless sensor networks have been widely used in many real-life scenarios, and have become more and more important. However, network connectivity is necessary for sensor network applications. When some of sensor nodes fail (due to energy depletion or destruction), it will cause the network disconnected. How to restore network connectivity is very important.

Hence, in this paper we propose a Distributed Coverage-based Connectivity Restoration (DCCR) which just needs a few nodes to execute critical nodes detection; coverage rate, residual energy, distance and degree are also considered to choose a backup node. Then, to promote the coverage rate, a better location is searched for a backup node to move to. Finally, the simulation results demonstrate that our proposed scheme has good performance in terms of coverage rate and residual energy than DARA and NORAS.

Keyword： WSN, connectivity, coverage rate, critical nodes, restoration, backup node

I. INTRODUCTION

Wireless Sensor Networks consist of a large number of sensor nodes which execute the task in a special environment. These sensor nodes are capable of computing data, monitoring environment, collecting information and transferring data etc. In recent years, wireless sensor networks have been widely used in many applications such as monitoring the action in the battlefield, tracking wild animals in the natural environment and sensing temperature/humidity in the forest. Additionally, volcano monitoring, debris flow detection and disaster notification are also used widely.

How to use limited energy to implement tasks of sensor nodes in the specific area is an important issue. In general, the nodes communicate and coordinate to send the data in the connected network. When some of sensor nodes failed, it would cause the network disconnected and tasks failed possibly. Therefore, the network connectivity is very important in the application of wireless sensor networks. The environment is usually harsh or difficult for people to reach in wireless sensor network. Hence, the nodes must have the ability to recover network connectivity by itself when network is disconnected.

There are many studies in how to recover network connectivity when nodes fail in wireless sensor networks (e.g.[2-7][10][13-16]). However, most of the researches have to execute the critical node detection in advance. For example, K.Akkaya et al.[2] modified CDS[8] to get the algorithm of critical node detection; some of researchers directly utilized DFS[9] algorithm. However, they are not suitable for wireless sensor networks, which consist of the large number of nodes, because they have to keep topology information of entire networks. In addition, some critical node detection algorithms [11] are not efficient in a special topology although it only needs the local information.

The network connectivity restoration mechanisms will be triggered when the failure node is the critical node. N. Tamboli et al. [14] used neighbors of a failure node to recover the temporary connectivity. Nevertheless, the neighbor nodes moving excessively will cause energy depletion. In addition, it was not suitable for emergency events because the latency of data delivery would be long. M. Younis et al. [16] proposed that all nodes moved inward to the location of a failure node to recover connectivity. However, the drawbacks are that more sensor nodes moved and the coverage rate became worse. A. Alfadhly et al. [7] improved the problem of node's movement. But the coverage rate will be low. Most of researchers looked for a backup node to move to the location of the failure node. The selection of a backup node will affect the network performance. A.A. Abbasi et al. [5] proposed that when the network was divided into number of areas and a critical node failed, the backup node, which is close to the failure node was selected from the least number of nodes area. The premise of this situation is that every node knows all information of network topology. A.A. Abbasi et al. [3] proposed the node should be one hop neighbor with minimum degree of failure nodes to be backed up. If there are two or more backup nodes, it considers the closest one. But if the backup node is the critical node, it will initiate a series connectivity restoring processes. K. Vaidya et al. [15] avoided choosing the critical node and took coverage rate, degree of node into account. Unfortunately, maybe it will not find any backup node in some special topology. However, the sensing energy of nodes is limited and are not considered by the above researchers. In addition, the coverage rate will affect the performance of the sensor nodes. Thus, we propose the DCCR to achieve the high coverage rate and restore connectivity in this paper.

The rest of this paper is organized as follows. Section II presents the design of our connectivity restoration. We evaluate the proposed scheme by simulations in Section III. Section IV concludes the paper.

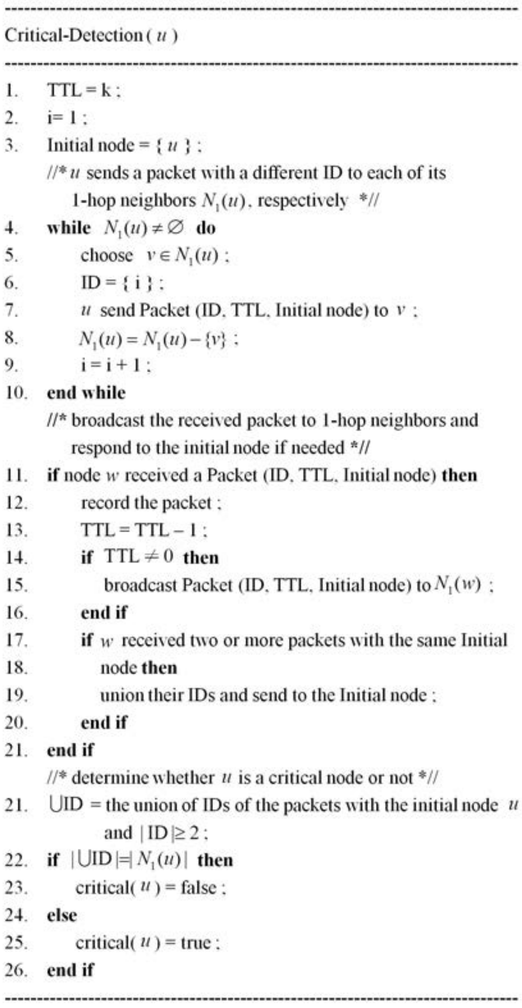
II. NETWOK CONNECTIVITY RESTORATION

We describe our proposed scheme as follows: i. Critical node detection. ii. Choose the backup node. iii. The location a backup node moves to. iv. Maintenance mechanism.

i. Critical node detection

In this section, we propose the simple and effective critical node detection which just needs location information and executes when it needs to reduce energy consumption and control overhead.

First, the nodes execute the critical node algorithm and send the packet with different ID of its neighbors (ex: Packet(ID, TTL, Initial node)). The value of TTL will be decreased by one when a node receives the packet. The packet will be broadcasted to the one-hop neighbors if the value of TTL is not equal to zero. The node returns the packets to the initial node when it receives more than or equal to two different numbers of packets from some initial nodes. Finally, the initial node unites the packets when it receives all returned packets. If the union equals to the all initial ID, this node will be regarded as the non-critical node (the detail of algorithm is shown in Fig. 1). As shown in Fig. 2, node A sends the packets (ID={1, 2 and 3}) of its neighbors ({B, C and D}) and sets the value of TTL (TTL=2). The nodes (B, C and D) broadcast the packet to their one hop neighbors and set the new value of TTL (TTL = TTL -1) until the value of TTL equals zero. The node G receives two different ID of packets (2 and 3), then it returns the packets to initial node A. Finally, the node A determines that it is the critical node because the received packet ({2 and 3}) is not equal to the all initial ID ({1, 2, and 3}). In another example as shown in Fig. 3, nodes (E and F) receive two different packets ({1 and 2}, {2 and 3}), then each returns the packets to initial node A. Because the union of received packet ({1, 2, and 3}) equals to the all initial ID, hence, node A is the non-critical node. However, this method may have errors in the detection of critical node when there are large loops in the network. However, the critical node detection has more than 96% accurate rate when the value of TTL equals three ([10]). Therefore, we can modify the value of TTL to increase the accurate rate in different network size and applications.



```
Critical-Detection ( $u$ )
TTL = k ;
i= 1 ;
Initial node = { $u$ } ;
//* $u$ sends a packet with a different ID to each of its
    1-hop neighbors $N_1(u)$, respectively  *//
while $N_1(u) \neq \varnothing$ do
    choose $v \in N_1(u)$ ;
    ID = { i } ;
    $u$ send Packet (ID, TTL, Initial node) to $v$ ;
    $N_1(u) = N_1(u) - \{v\}$ ;
    i = i + 1 ;
end while
//* broadcast the received packet to 1-hop neighbors and
    respond to the initial node if needed *//
if node $w$ received a Packet (ID, TTL, Initial node) then
    record the packet ;
    TTL = TTL – 1 ;
    if TTL $\neq 0$ then
        broadcast Packet (ID, TTL, Initial node) to $N_1(w)$ ;
    end if
    if $w$ received two or more packets with the same Initial
        node then
        union their IDs and send to the Initial node ;
    end if
end if
//* determine whether $u$ is a critical node or not *//
$\bigcup$ID = the union of IDs of the packets with the initial node $u$
        and $|$ ID $| \geq 2$ ;
if $| \bigcup \text{ID} | = | N_1(u) |$ then
    critical( $u$ ) = false ;
else
    critical( $u$ ) = true ;
end if
```

Fig. 1. Critical node detection algorithm.

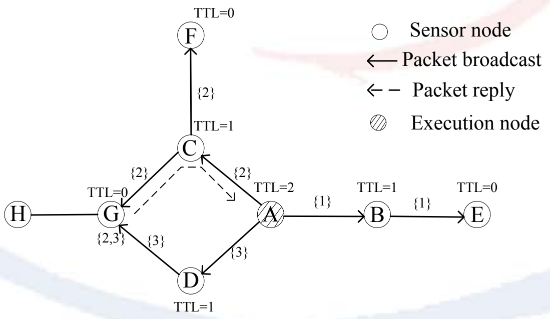


Fig. 2. The node A is the critical node.

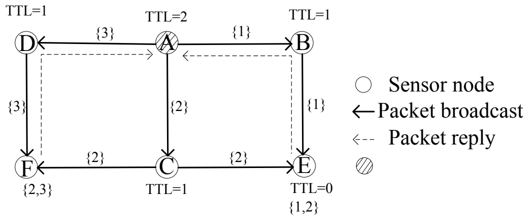


Fig. 3. The node A is the non-critical node.

ii. Choose the backup node

To avoid disconnection of network when nodes depleted energy, we set the threshold of energy. Therefore, we must choose a backup node which moves to some location (detail in next chapter) to maintain network connectivity when the sensor node is a critical node and its energy is below the threshold.

The purpose of choosing a backup node is to restore network connectivity. So, the critical node is not suitable to be the backup node, because it will cause a series connectivity restoring processes. Hence, in this paper the critical node is not regarded as the backup node. In addition, the coverage rate will affect the performance of the network. Therefore, we hope to select the backup node which influences the minimal coverage rate after moving. Moreover, the node's residual energy, distance from this node to failure node and degree are factors to select this node as a backup node or not. Based on the above observation and analysis, we set the weight value relative to a failure node. The high weight value stands for more probability of being the backup node.

Assume that the node u will be failed (call failure node). The weight function $w_u(v)$ that the node v to u is defined as follows:

$$w_u(v) = \begin{cases} -1 & , v \in \text{critical node} \\ \alpha \mathrm{A} + \beta \mathrm{E} + \gamma \mathrm{D} + \delta \mathrm{K}, \in v & \text{non - critical node} \end{cases}$$

$$0 \leq \alpha, \beta, \gamma, \delta \leq 1, \quad \alpha+\beta+\gamma+\delta=1$$

$$\mathrm{A} = \frac{\text{the overlapped coverage of } v \text{ and } \mathrm{N}_1(v)}{\text{the sensor area of } v}$$

$$\mathrm{E} = \frac{\text{the residual energy of } v}{\text{the initial energy of } v}$$

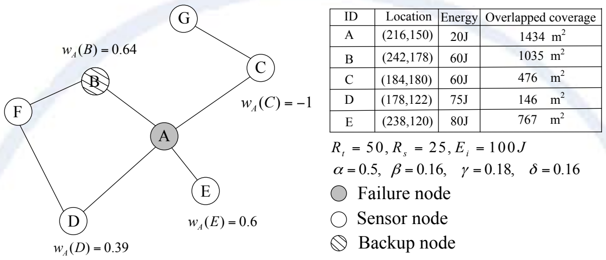
$$\mathrm{D} = \frac{\text{the transmission range of } v}{\text{the distance between u and } v}$$

$$\mathrm{K} = \frac{1}{\text{averge degree of } v \text{ neighbor nodes}}$$

$\alpha, \beta, \gamma, \delta$ are parameters which are adjusted for different applications. By the above definitions, all the parameters are greater than or equal to zero, so the weight value of the node is greater or equal than zero when it is a non-critical node. Furthermore, the sensor node needs to execute critical node detection to determine whether it is the critical node before computing the weight values.

The sensor node u executes critical node detection when it will fail. The selection of the backup node mechanism will be triggered when the node u is a critical node. The detailed mechanism is as follows:

First, the sensor node u informs one hop neighbors of calculating the weight value, and chooses the sensor node p_1 with the highest weight value. If $w_u(p_1) \neq -1$ (p_1 is non-critical node), then the sensor node p_1 is the backup node for node u. If there are the same weight values in more than two nodes, the node with maximum ID will be the backup node. Otherwise, the sensor node u informs two hop neighbors to calculate the weight value and get their one hop neighbor information in the meanwhile. Then, the sensor node u chooses the backup node with the highest weight value. Similarly, the nodes with the maximum ID will be the backup node when they have the same weight value. If all of $w_u(p) = -1, \quad p \in N_2(u)$, the node u uses the received information to execute DFS algorithm in the local to check if there is a circular loop or not. If the circular loop exists, the node u will correct the node misjudged as a critical node for the non-critical node. Then, it has to calculate the weight value again to find the new backup node. If there is no circular loop, the node u repeats the above steps to find the backup node in three hop neighbors. Generally, the node usually can find the backup node, because it must exist the non-critical node in any graphics (the detail of algorithm is shown in Fig. 4). For example, in Fig. 5, node A notifies one hop neighbors to calculate the weight value and then selects the highest weight value node B as the backup node when node A will fail. In another example in Fig. 6, the node A notifies node (B, C, D and E) of its one hop neighbors to execute the critical node detection algorithm (TTL = 2) and calculate the weight value. The weight values of nodes (B, C, D and E) are -1, so they are the critical nodes. Then, node A informs the nodes (F, G, H and I) to calculate the weight value and their weight values are also -1. However, the node A has the nodes' (F, G, H and I) one hop neighbor information so that it can find there are two cycles (J-G-C-A-B-F-J and K-I-E-A-D-H-K) in the network as shown in Fig. 6(a). After updating the weight value, the node D is selected as the backup node because it has the highest weight value, as shown in Fig. 6(b).

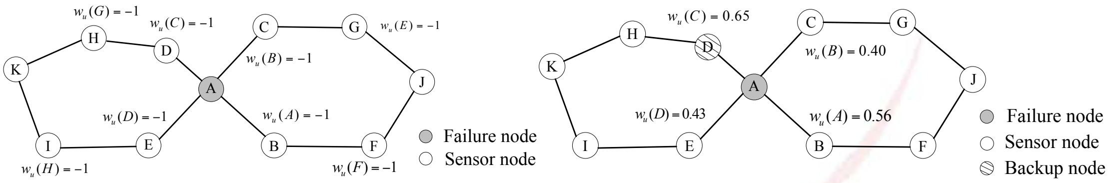


ID	Location	Energy	Overlapped coverage
A	(216,150)	20J	1434 m^2
B	(242,178)	60J	1035 m^2
C	(184,180)	60J	476 m^2
D	(178,122)	75J	146 m^2
E	(238,120)	80J	767 m^2

$R_t = 50, R_s = 25, E_i = 100J$

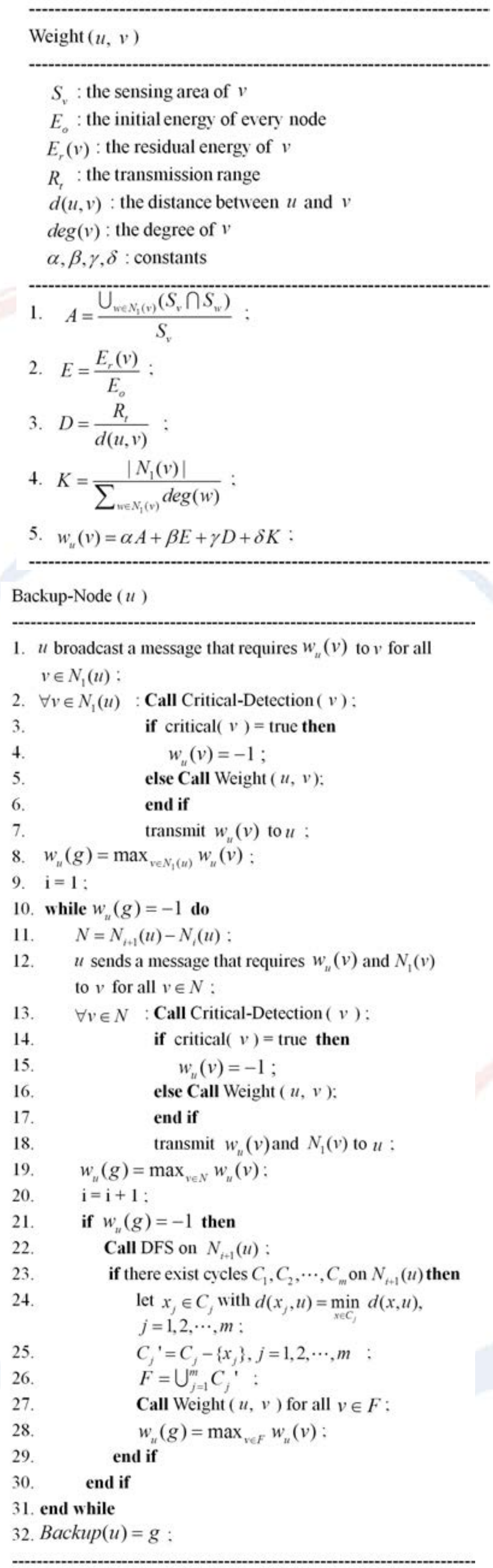
$\alpha = 0.5,\ \beta = 0.16,\ \gamma = 0.18,\ \delta = 0.16$

Fig. 5. The selection of backup node (there is the non-critical node in two hop neighbors).



(a) (b)

Fig. 6. The selection of backup node (the nodes is misstated to be critical node).



```
Weight ($u$, $v$)

    $S_v$ : the sensing area of $v$
    $E_o$ : the initial energy of every node
    $E_r(v)$ : the residual energy of $v$
    $R_t$ : the transmission range
    $d(u,v)$ : the distance between $u$ and $v$
    $deg(v)$ : the degree of $v$
    $\alpha, \beta, \gamma, \delta$ : constants

$A = \frac{\bigcup_{w \in N_1(v)} (S_v \cap S_w)}{S_v}$ ;
$E = \frac{E_r(v)}{E_o}$ ;
$D = \frac{R_t}{d(u,v)}$ ;
$K = \frac{|N_1(v)|}{\sum_{w \in N_1(v)} deg(w)}$ ;
$w_u(v) = \alpha A + \beta E + \gamma D + \delta K$ ;

Backup-Node ( $u$ )

$u$ broadcast a message that requires $w_u(v)$ to $v$ for all
  $v \in N_1(u)$ ;
$\forall v \in N_1(u)$ : Call Critical-Detection ( $v$ ) ;
                if critical( $v$ ) = true then
                    $w_u(v) = -1$ ;
                else Call Weight ( $u$, $v$);
                end if
                transmit $w_u(v)$ to $u$ ;
$w_u(g) = \max_{v \in N_1(u)} w_u(v)$ ;
i = 1 ;
while $w_u(g) = -1$ do
    $N = N_{i+1}(u) - N_i(u)$ ;
    $u$ sends a message that requires $w_u(v)$ and $N_1(v)$
    to $v$ for all $v \in N$ ;
    $\forall v \in N$ : Call Critical-Detection ( $v$ ) ;
                if critical( $v$ ) = true then
                    $w_u(v) = -1$ ;
                else Call Weight ( $u$, $v$ );
                end if
                transmit $w_u(v)$ and $N_1(v)$ to $u$ ;
    $w_u(g) = \max_{v \in N} w_u(v)$ ;
    i = i + 1 ;
    if $w_u(g) = -1$ then
        Call DFS on $N_{i+1}(u)$ ;
        if there exist cycles $C_1, C_2, \cdots, C_m$ on $N_{i+1}(u)$ then
            let $x_j \in C_j$ with $d(x_j,u) = \min_{x \in C_j} d(x,u)$,
            $j = 1,2,\cdots,m$ ;
            $C_j' = C_j - \{x_j\}, j = 1,2,\cdots,m$ ;
            $F = \bigcup_{j=1}^{m} C_j'$ ;
            Call Weight ( $u$, $v$ ) for all $v \in F$ ;
            $w_u(g) = \max_{v \in F} w_u(v)$ ;
        end if
    end if
end while
$Backup(u) = g$ ;
```

Fig. 4. Backup node algorithm

iii. The location a backup node moves to

The backup node usually moves to the location of a failure node to maintain the network connectivity. But we think that there is a better location to improve the coverage rate than the location of a failure node.

- **Candidate locations**

For example, in Fig. 7, the degree of failure node u is two and the backup node p is node $u's$ one hop neighbor (assume $N_1(u)=\{p,q\}$). Then, the location of w is the intersection of line $\overline{pq}$ and circle q (the center is q and the radius is R_t). The location of u and w will be the candidate locations where the backup node may move to. Otherwise, the center of the circle is node $u's$ one hop neighbors, R_t is the radius, the location of intersection on the circle which are less or equal to R_t and u will be the candidate location where the backup node may move to, as shown in Fig. 8.

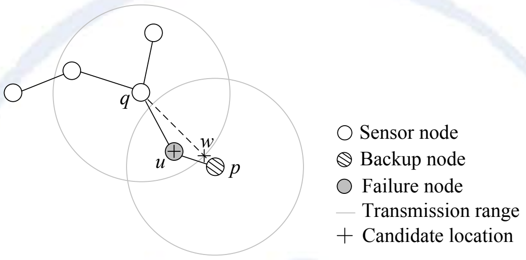


Fig. 7. The backup node p will move to the candidate locations (when $deg(u)=2$ and $p\in N_1(u)$).

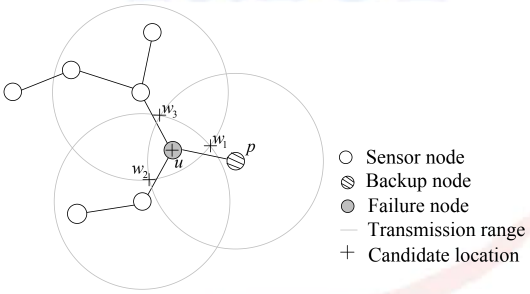
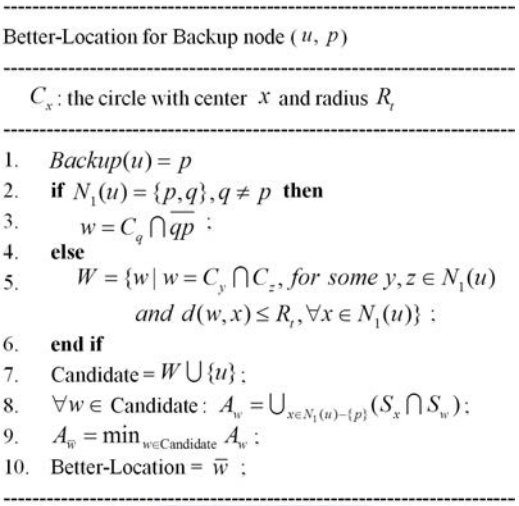


Fig. 8. The backup node p will move to the candidate locations (when $d(u)\neq 2$ or $p\in N_i(u), i\geq 2$).

- **The better candidate locations**

After calculating the candidate location, we calculate the lowest overlapped coverage with $u's$ one-hop neighbor to be the better candidate location. However, the coverage of node p and node u will be ignored when calculating the overlapped coverage (the detail of algorithm is shown in Fig. 9). For example, in Fig. 10, the candidate locations are w_1, w_2, w_3 and u. Then, w_1 is the better candidate location, because it has lowest overlapped coverage.



```
Better-Location for Backup node ( $u$, $p$ )
    $C_x$ : the circle with center $x$ and radius $R_t$
Backup(u) = p
if $N_1(u) = \{p, q\}, q \neq p$ then
    $w = C_q \cap \overline{qp}$ ;
else
    $W = \{w \mid w = C_y \cap C_z, for\ some\ y, z \in N_1(u)$
        $and\ d(w, x) \leq R_t, \forall x \in N_1(u)\}$ ;
end if
Candidate = $W \cup \{u\}$ ;
$\forall w \in$ Candidate : $A_w = \bigcup_{x \in N_1(u)-\{p\}} (S_x \cap S_w)$ ;
$A_{\bar{w}} = \min_{w \in \text{Candidate}} A_w$ ;
Better-Location = $\bar{w}$ ;
```

Fig. 9. Better-Location for Backup node algorithm

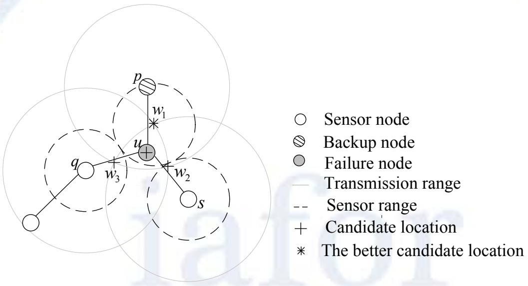


Fig. 10. The better candidate location.

- **The backup node moves to the better candidate locations**

When determining the better candidate location, the backup node p directly moves to the better candidate locations to restore network connectivity when the node p is one-hop neighbor of node u. Otherwise ($p \in N_k(u)$, $k \geq 2$), assume that the shortest path between the node u and the node p is $u - p_1 - p_2 - \cdots - p_{k-2} - p_{k-1} - p$, $p_i \in N_1(u), i = 1,2,\cdots,k-1$. Then, when p_1 moves to the better candidate location, p_2 moves to p_1 until p moves to p_{k-1}. We can balance the energy consumption and restore the network connectivity by these steps. As shown in Fig. 11, the backup node p directly moves to the better candidate locations w_1. As shown in Fig. 12, first, the node p_1 moves to better candidate locations and the node p moves to location of p_1.

Nevertheless, the node u only uses one-hop neighbor to calculate the better candidate location. if the location where the backup node may move to is not the location of node u. it have to recalculate the overlapped coverage again. The backup node will move to the location of node u to get the better coverage when the overlapped coverage of a better candidate location is higher than the value $c(c > 0)$ of node u.

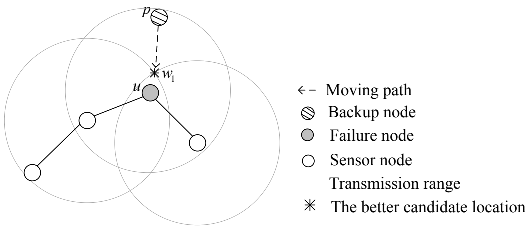


Fig. 11. The backup node is the failure node's one hop neighbor.

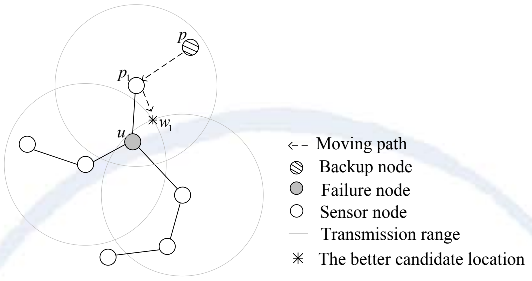


Fig. 12. The backup node is no failure node's one hop neighbor.

iv. Maintenance mechanism

The sensor node cannot find the backup node to restore networks connectivity on time when the critical node is damaged suddenly. We also propose a mechanism to solve this case.

The node u will detect whether it is the critical node or not when it finds out that its next forwarding node v is damaged (call damaged node). When node u is a non-critical node, it directly moves to the location of damaged node v, as shown in Fig. 13. In another case, the node u first triggers the network connectivity recovery mechanism to find the backup node p and the better candidate location w. Then, the node u moves to the location of the damaged node v and the backup node p moves to the locations of w, as shown in Fig. 14.

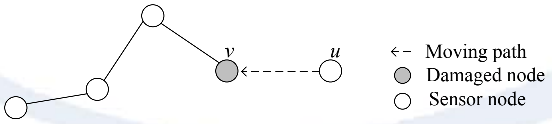


Fig. 13. The non-critical node moves to the location of the damaged node.

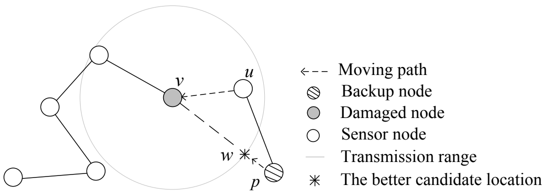


Fig. 14. The critical node finds the backup node and then moves to the location of the damaged node.

IV. EXPERIMENTAL EVALUATIONS

In this chapter we will describe the experimental simulation environment, as well as the basic parameter setting. We compare the proposed scheme DCCR with DARA[3] and NORAS[15] in terms of coverage rate, total travelled distance, control overhead, average residual energy and success rate.

We use the C# language to simulate the performance. The parameters are shown in Table 1 .

Table. 1. Simulation parameters.

Parameter	Value
Network size	500×500 m^2, 750×750 m^2, 1000×1000 m^2, 1250×1250 m^2
Number of sensors	20, 40, 60, 80, 100
Transmission range	50m, 75m, 100m, 125m
Sensor range	50m
Energy of sensors	$50mJ$, $125mJ$, $200mJ$ (random)
α, β, γ, δ, c	$\alpha:0.5$, $\beta:0.16$, $\gamma:0.17$, $\delta:0.16$, $c:0.1$

1. **Total travelled distance:** the distance that all the nodes moved when restoring network connectivity.
2. **Reduction in field coverage:** the initial field coverage minus the recovery field coverage.
3. **Control overhead:** the message produced when restoring network connectivity.
4. **Average residual energy:** the average residual energy of moved node.
5. **Success rate:** the success rate of recovering network connectivity.

Fig. 15 and 16 compare the total travelled distance with different number of nodes and transmission range. Our proposed method DCCR has better performance than others because DCCR does not choose the critical node to be the backup node and it considers the distance between a backup node and a failure node.

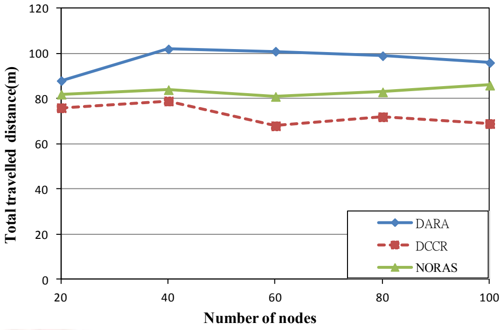


Fig. 15. The total travelled distance with different number of sensor nodes(size: 500*500, R_t: 100 m).

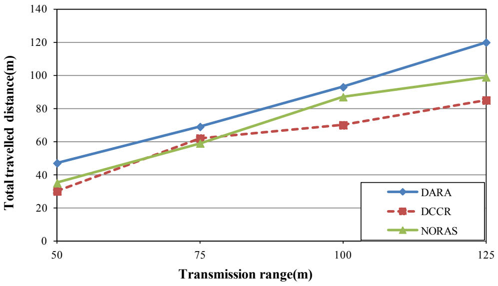


Fig. 16. The total travelled distance with different transmission range(size: 500*500, sensors: 60).

Fig. 17 and 18 compare the reduction in field coverage with different number of nodes and transmission range. The DARA does not have good performance, because it does not consider the node coverage when it chooses the backup node. Although both DCCR and NORAS consider the node coverage, DCCR has better performance than NORAS, because it selects the better location for backup node to move to.

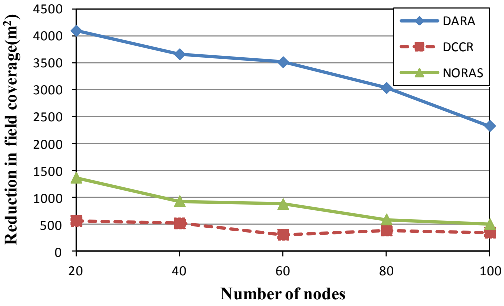


Fig. 17. The coverage rate with different number of sensor nodes(size: 500*500, R_t: 100 m)

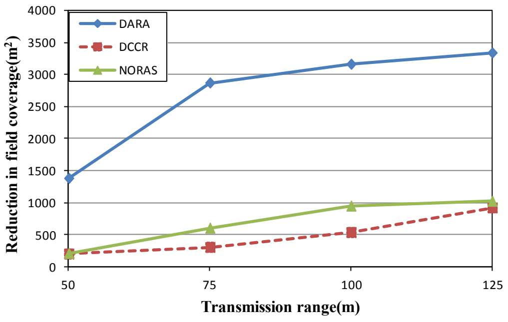


Fig. 18. The coverage rate with different transmission range(size: 500*500, sensors: 60).

Fig. 19 and 20 compare the control overhead with different number of nodes and transmission range. The NORAS has great number of control overhead because all nodes have to execute the critical node detection. However, the control overhead of DCCR is not much, because it only needs some parts of nodes to execute the critical node detection. The DARA does not execute the critical node detection, so the control overhead is less than DCCR.

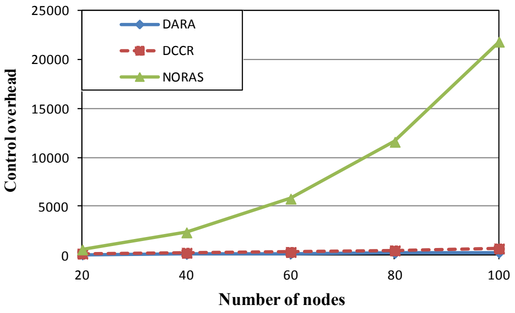


Fig. 19. The control overhead with different number of sensor nodes(size: 500*500, R_t: 100 m).

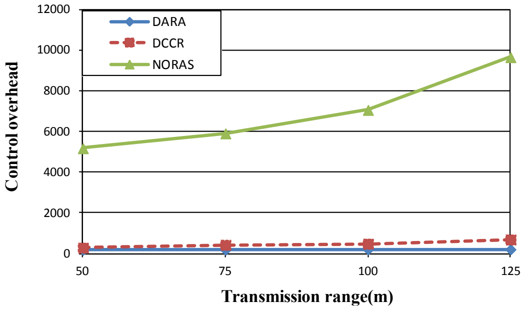


Fig. 20. The control overhead with different transmission range(size: 500*500, sensors: 60).

Fig. 21 and 22 compare the average residual energy with different number of nodes and transmission range. The DCCR has better performance than DARA and NORAS, because it considers the node residual energy when it selects the backup node.

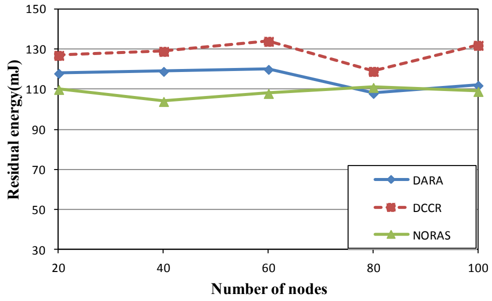


Fig. 21. The average residual energy with different number of sensor nodes(size: 500*500, R_t: 100 m).

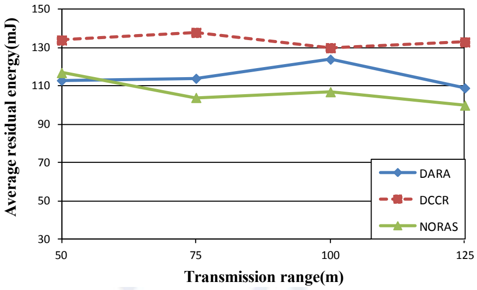


Fig. 22. The average residual energy with different transmission range(size: 500*500, sensors: 60).

Fig. 23 and 24 compare the success rate with different number of nodes and transmission range. The DCCR has better performance than DARA and NORAS, because it considers the node energy to avoid the node with less energy to be the backup node.

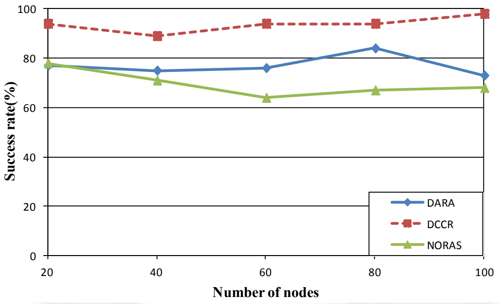


Fig. 23. The success rate with different number of sensor nodes(size: 500*500, R_t: 100 m).

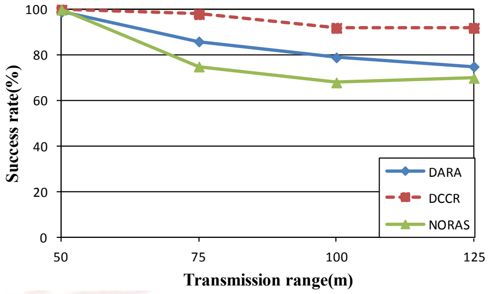


Fig. 24. The success rate with different transmission range(size: 500*500, sensors: 60).

Fig. 25-29 compare the different methods with 100m transmission range and different network size.

Fig. 25 compares the total travelled distance. The DCCR chooses a non-critical node to be the backup node which is near the failed node, so the network sizes does not affect total travelled distance. On the other hand, the DARA may choose the critical node to be the backup node, so the total travelled distance has increased with growing network size.

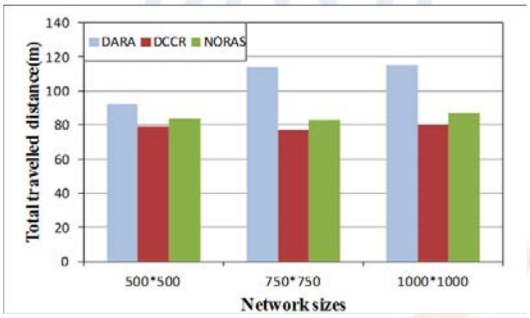


Fig. 25. The total travelled distance with different network sizes.

Fig. 26 and 27 compare the reduction field coverage and average residual energy. The values are not significantly changed in different network sizes, because the two elements are less influenced by network sizes.

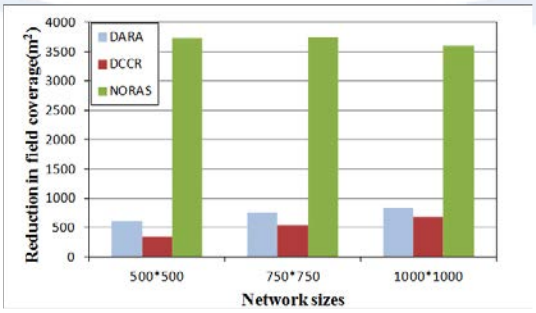


Fig. 26. The reduction field coverage with different network sizes.

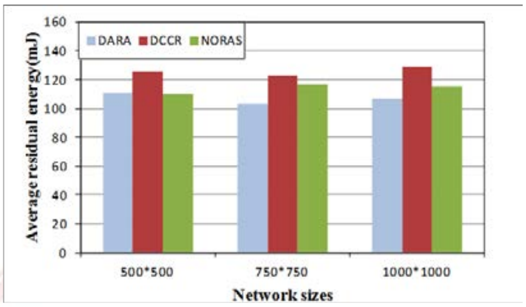


Fig. 27. The average reduction energy with different network sizes.

Fig. 28 compares the control overhead. The control overhead of NORAS increases a lot with growing network sizes. Therefore, we show DCCR and NORAS in Fig. 29. The control overhead of DARA increases with growing network sizes. However, the increase of DCCR is not obvious.

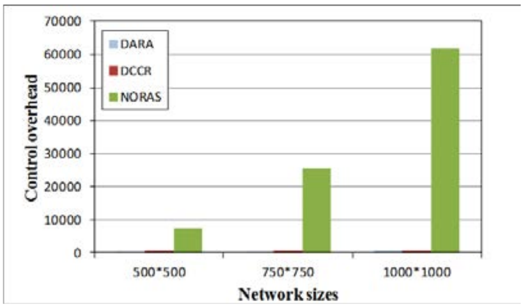


Fig. 28. The control overhead with different network sizes(DARA, DCCR and NORAS).

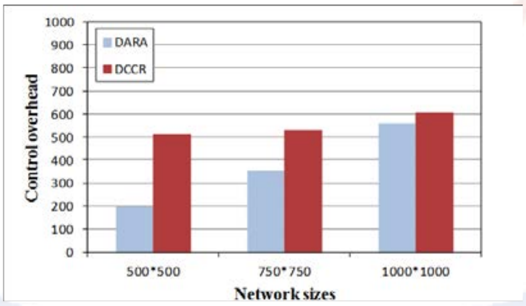


Fig. 29. The control overhead with different network sizes(DARA and DCCR).

Fig. 30-32 compare with DCCR in 100m transmission range and different parameters (α,β,γ). The simulation result shows that the coverage rate has improved when increasing the value α. Similarly, the average residual energy and total travelled distance are better when increasing β and γ. So, the users can adjust the parameters by themself to meet the different demands.

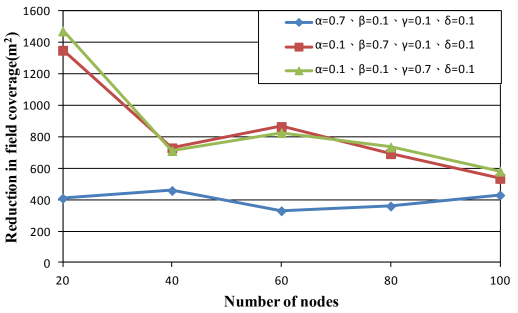


Fig. 30. The reduction in field coverage with different number of sensor nodes.

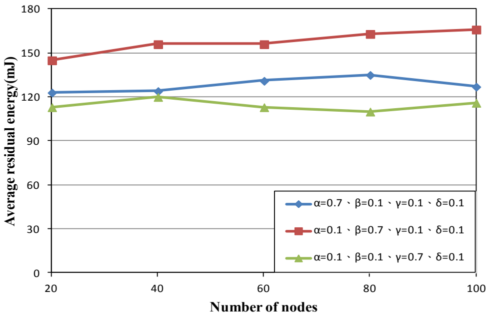


Fig. 31. The average reduction energy with different number of sensor nodes.

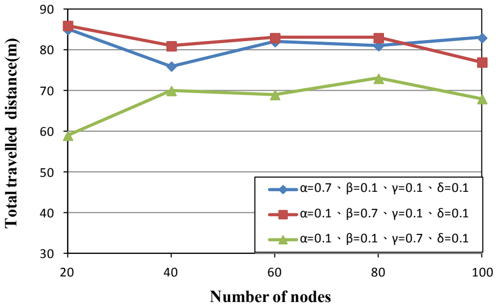


Fig. 32. The total travelled distance with different number of sensor nodes.

V. CONCLUSION

In this paper, we propose a Distributed Coverage-based Connectivity Restoration (DCCR). Firstly, we design the simple algorithm to detect the failure node which cause network partitioned. Second, the backup node takes many important factors into account, such as coverage rate, residual energy, travelled distance and the degree of node. Finally, the backup node will move to the better candidate location to restore network connectivity. Besides, we also propose the effective maintenance mechanism when a node fails suddenly. The simulation results demonstrate that our proposed scheme has good performance in terms of coverage rate、residual energy than DARA and NORAS. Simultaneously, the total travelled distance has a good performance.

ACKNOWLEDGMENT

This research is supported by the National Science Council of the Republic of China, Taiwan, under grant No. NSC-100-2221-E-035-056.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A Survey on Sensor Networks,” *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-114, 2002

[2] K. Akkaya, F. Senel, A. Thimmapuram, and S. Uludag, “Distributed Recovery from Network Partitioning in Movable Sensor/Actor Networks via Controlled Mobility, ” *IEEE Transactions on Computers*, Vol. 59, No. 2, pp. 258-271, 2010

[3] A.A. Abbasi, M. Younis, and K. Akkaya, “Movement-Assisted Connectivity Restoration in Wireless Sensor and Actor Networks,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 20, pp.1366-1379, 2009

[4] A.A. Abbasi, M. Younis, and U. Baroudi, “Restoring Connectivity in Wireless Sensor-Actor Networks with Minimal Node Movement,” *Proceedings of International Wireless Communications and Mobile Computing Conference*, pp. 2046-2051, 2011

[5] A.A. Abbasi, M. Younis, and U. Baroudi, “Restoring Connectivity in Wireless Sensor-Actor Networks with Minimal Topology Changes,” *Proceedings of IEEE International Conference on Communications*, pp. 1-5, 2010

[6] A.A. Abbasi, U. Baroudi, M. Younis, and Akkaya. K, “C2AM: An Algorithm for Application-Aware Movement-Assisted Recovery in Wireless Sensor and Actor Networks,” *Proceedings of International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, pp. 655-659, 2009

[7] A. Alfadhly, U. Baroudi, and M. Younis, “Least Distance Movement Recovery Approach for Large Scale Wireless Sensor and Actor Networks,” *Proceedings of International Wireless Communications and Mobile Computing Conference*, pp. 2058-2063, 2011

[8] K. Akkaya, A. Thimmapuram, F. Senel , and S. Uludag, “Distributed Recovery of Actor Failures in Wireless Sensor and Actor Networks,” *Proceedings of Wireless Communications and Networking Conference*, pp. 2480-2485, 2008

[9] F. Dai and J. Wu, “An Extended Localized Algorithm for Connected Dominating Set Formation in Ad Hoc Wireless Networks,” *IEEE Transactions on Parallel and Distributed Systems*, Vol. 15, pp.908-920, 2004

[10]K.I. Gandhi, P. Narayanasamy, and P. Ushapreethi, “Coverage Holes Reduction in Movable Sensor Networks Using Minimum Movement Technique,” *Proceedings of International Conference on Recent Trends in Information Technology*, pp. 275-280, 2011

[11]M. Imran, M. Younis, A.M. Said, and H. Hasbullah, “Partitioning Detection and Connectivity Restoration Algorithm for Wireless Sensor and Actor Networks,” *Proceedings of International Conference on Embedded and Ubiquitous*

Computing, pp. 200-207, 2011

[12]M. Imran, A.M. Said, M. Younis, and H. Hasbullah, “Application-Centric Connectivity Restoration Algorithm for Wireless Sensor and Actor Networks,” *Proceedings of international conference on Advances in grid and pervasive computing, Springer-Verlag*, pp. 243-253, 2011

[13]M. Imran, M. Younis, A. M. Said and H. Hasbullah, “Localized Motion-based Connectivity Restoration Algorithms for Wireless Sensor and Actor Networks,” *Journal of Network and Computer Applications*, Vol 35, No 2 , pp. 844-856, 2012

[14]T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, “Introduction to Algorithms,” Second Edition. MIT Press and McGraw-Hill, 2001.

[15]X. Liu, L. Xiao, A. Kreling, and Y. Liu, “Optimizing Overlay Topology by Reducing Cut Vertices,” *Proceedings of international workshop on Network and operating systems support for digital audio and video*, No. 17, 2006

[16]J. Milenko, I. Stojmenovic, M. Hauspie, and D. Simplot-ryl, “Localized Algorithms for Detection of Critical Nodes and Links for Connectivity in Ad Hoc Networks,” *Proceedings of the Third Annual IFIP Mediterranean Ad Hoc Networking Workshop* , pp. 360-371 , 2004

[17]N. Tamboli and M. Younis, “Coverage-Aware Connectivity Restoration in Mobile sensor networks,” *Journal of Network and Computer Applications*, Vol. 33, No 4, pp. 363-374, 2010

[18]N. Tamboil and M. Younis, “Coverage-Aware Connectivity Restoration in Mobile Sensor Networks,” *Proceedings of International Conference on Communications*, pp. 1-5 , 2009

[19]K. Vaidya and M. Younis, “Efficient Failure Recovery in Wireless Sensor Networks Through Active, Spare Designation,” *Proceedings of IEEE International Conference on Distributed Computing in Sensor Systems Workshops*, pp. 1-6 , 2010

[20]S. Vemulapalli and K. Akkaya, “Mobility-Based Self Route Recovery from Multiple Node Failures in Mobile Sensor Networks,” *Proceedings of IEEE Conference on Local Computer Networks*, pp. 699-706, 2010

[21]M. Younis, L. Sookyoung, S. Gupta, and K. Fisher, “A Localized Self-Healing Algorithm for Networks of Moveable Sensor Nodes,” *Proceedings of IEEE Global Telecommunications Conference*, pp. 1-5, 2008

[22]M. Younis and R. Wasknis, “Connectivity Restoration in Wireless Sensor Networks Using Steiner Tree Approximations,” *Proceedings of IEEE Global Telecommunications Conference*, pp.1-5 , 2010

[23]X. Zhao and N. Wang, “ Coordination-Assisted Connectivity Recovery Approach in Wireless Sensor and Actor Networks, ” *Proceedings of International Conference on Computer Research and Development*, pp. 82-86, 2011


iafor