


An anomaly node detection method for distributed time synchronization algorithm in cognitive radio sensor networks

International Journal of Distributed
Sensor Networks
2018, Vol. 14(5)
© The Author(s) 2018
DOI: 10.1177/1550147718774467
journals.sagepub.com/home/dsn


Qi Yang¹, Xuan Zhang¹, Jingfeng Qian¹ and Qiang Ye²

Abstract

In wireless sensor networks, time synchronization is an important issue for all nodes to have a unified time. The wireless sensor network nodes should cooperatively adjust their local time according to certain distributed synchronization algorithms to achieve global time synchronization. Conventionally, it is assumed that all nodes in the network are cooperative and well-functioned in the synchronization process. However, in cognitive radio wireless sensor networks, the global time synchronization process among secondary users is prone to fail because the communication process for exchanging synchronization reference may be frequently interrupted by the primary users. The anomaly nodes that failed to synchronize will significantly affect the global convergence performance of the synchronization algorithm. This article proposes an anomaly node detection method for distributed time synchronization algorithm in cognitive radio sensor networks. The proposed method adopts the statistical linear correlation analysis approach to detect anomaly nodes through the historical time synchronization information stored in local nodes. Simulation results show that the proposed method can effectively improve the robustness of the synchronization algorithm in distributed cognitive radio sensor networks.

Keywords

Wireless sensor networks, cognitive radio, time synchronization, convergence, abnormal node detection

Date received: 22 September 2017; accepted: 26 March 2018

Handling Editor: Katsuya Suto

Introduction

Wireless sensor network (WSN) is a special distributed wireless network that can support pervasive and robust monitoring of various physical conditions such as temperature, sound, humidity, and so on. Designed for rapid and robust deployments, WSNs are widely used in both military and civilian applications.¹ Time synchronization plays a very important role in WSNs because it enables all nodes to have a unified time, on which all applications such as data collection and environment monitoring depend.^{1,2} Cognitive radio (CR) network has been widely regarded as a key technology to increase the spectrum utilization of wireless communication systems.³ CR technology enables secondary users (SUs) without spectrum license to share wireless

channels with primary users (PUs) who hold the spectrum license through dynamic spectrum access technology. The application of CR technology in WSNs can enhance the capacity of WSNs and has been widely investigated in recent years.^{4,5} However, time synchronization in cognitive radio sensor networks (CRSNs) is

¹School of Information Science and Technology, Xiamen University, Xiamen, China

²Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada

Corresponding author:

Qi Yang, School of Information Science and Technology, Xiamen University, Xiamen 361005, China.
Email: yangqi@xmu.edu.cn



prone to failure because the communication process for exchanging synchronization references among SUs may be frequently interrupted by the PUs' activity. The anomaly nodes that fail to keep synchronized will significantly reduce the convergence performance of the synchronization algorithm in CRSNs.⁴

In the literature, many-time synchronization algorithms for WSNs have been proposed. These algorithms can be broadly categorized into two types: centralized and distributed approaches.¹

In the centralized approach, there is a powerful central node servicing as a root. The root node and other nodes in the network form a tree or mesh topology. The central node periodically broadcasts the time reference information, marked as level 1. Based on the received level 1 timing information, the neighbor nodes of the root adjust their local time references and marked them as level 2. In the same way, the two-hop neighbor nodes of the root adjust their local time reference according to the root node's one-hop neighbors. The same procedure propagates through the entire network. If a node receives more than one timing references from its neighbor nodes, it will adjust its local time reference according to the node with the most superior level.^{6–10} The centralized methods are effective in achieving the convergence of time synchronization with a low signaling cost. In addition, it is easy to find an abnormal node that does not adjust its local time reference. However, these centralized algorithms are over-dependent on the root node. Moreover, the time reference bias and error will accumulate as the hops between a node and the root node increase. Therefore, distributed synchronization algorithms are proposed.

In the distributed approach, each node independently adjusts its time reference according to the time difference between its local time reference and the time reference of its neighbor nodes. In the literature, different types of the distributed algorithms have been proposed. The major types are test-based detection, majority voting detection, and node-self-detection methods.

In the test-based detection methods, nodes execute test task and then make a judgment on abnormal nodes based on the test results. In a k -connected graph, Webber let one node execute the test together with its neighbors to detect the abnormal node.¹¹ This algorithm requires that every node should have a given number of neighbors, so its application is restricted to some limited scenarios. Chessa proposed a test-based algorithm using comparison method.^{12,13} In his algorithm, the system will choose a trusted node to broadcast test information. Its neighbors, respectively, compare the information and then return a testing result to the trusted node for deciding if a node is normal or not. This algorithm is suitable to distributed networks, but how to choose a trusted node becomes a new problem. On the basis of Chessa's algorithm, Elhadeif proposed some improved

algorithms—adaptive Distributed Self-Diagnosis Protocol (DSDP) and mobile DSDP.¹⁴ Similar methods of regularly checking nodes' behavior in WSNs to detect abnormal nodes were discussed in You et al.¹⁵ and El-Koujok et al.¹⁶ Although the test-based algorithms can improve the detection accuracy, they also bring challenges in terms of the communication cost and computational complexity.

The majority voting detection method takes the advantage of the spatial similarity of the nodes in WSNs to decide abnormal nodes. Vuran et al.¹⁷ set a threshold value to test the difference between two nodes. If the difference is higher than the threshold, one of the two nodes is likely to be abnormal. If one node is voted by all its neighbors, the node will be regarded as an abnormal node. As an improvement to the simple voting, the voting methods discussed in Xiao et al.¹⁸ and Behnke et al.¹⁹ were affected by the weights between nodes. The method presented in Xiao et al.¹⁸ required that the nodes be connected only when they are similar, which make the voting credible. The Efficient Localized Detection of Erroneous Nodes (ELDEN) proposed by Behnke et al.¹⁹ computed the weight through the distance between nodes and every node chooses the median of its neighbors to compare with its own state value. Then the difference will be normalized by the weight and becomes the final difference Y . If Y is higher than the threshold, a node is regarded as abnormal. The majority voting detection method can achieve a good performance in accuracy with a relatively small signaling cost, but it shows a poor performance in scenarios with a small number of nodes.

The node-self-detection methods for distributed WSNs were discussed in Babaie et al.;^{20–22} these algorithms require extra hardware or software to complete the detection, which impose extra cost for resource limited nodes in WSNs.

Through the above literature review, we can find that detecting the abnormal nodes in distributed time synchronization is very important for cognitive WSNs. The abnormal nodes who could not adjust its local time reference according to the time synchronization algorithm will damage the convergence of the algorithm. In this article, we propose a novel method to detect the abnormal node in distributed time synchronization algorithm. In our method, the correlation coefficient is introduced to compute the local time reference correlation between nodes in the networks. As the time reference of an abnormal node is likely to be uncorrelated to the time reference of the normal node, the correlation coefficient can be used as a metric for effective detection of abnormal nodes in the network.

System model

In this article, we study the time synchronization algorithm under an assumption that the topology of the

cognitive WSN is a totally connected graph. Messages sent from a source node will successively be received by any given destination node in the network within several hops. All the nodes except the abnormal nodes in the network adjust their local time references according to the distributed synchronization algorithm.

Distributed synchronization algorithm

The algorithm proposed in this article is designed for WSNs in which nodes communicate in a distributed manner. We denote the set of all points/network nodes as V , and the neighbors of node i are denoted as $Node(i)$. The time references of node i in the current state and the next state are marked as $t_i(n)$ and $t_i(n+1)$, respectively. It is assumed that there are N nodes in the network, and the time adjustment of node i could only be computed by receiving the current state $t_j(n)$ of the time reference of its neighbors. In distributed time synchronization algorithms, each node updates its local time reference by an adjustment that is computed by the time difference between its local time reference and that of its neighbors. Therefore, every node's next state of time reference $t_i(n+1)$ will be affected by its neighbors' current state of time reference, $t_j(n)$, that is

$$t_i(n+1) = t_i(n) - \sum_{j \in Node(i)} w_{ij}(t_i(n) - t_j(n)) \quad (1)$$

Here, w_{ij} is the weight between node i and node j , which should be in the range between 0 and 1, and $Node(i)$ is the neighbor set of node i . For different algorithms, there are different manners for computing w_{ij} . Therefore, equation (1) can be simplified as

$$t_i(n+1) = \sum_{j \in V} w_{ij}t_j(n) \quad (2)$$

The weights between non-neighbor nodes which do not belong to $Node(i)$ are equal to 0, that is, $w_{ik} = 0, k \notin Node(i)$; the weight of w_{ii} is equal to $1 - \sum_{j \in Node(i)} w_{ij}$ and $\sum_{j \in V} w_{ij} = 1$.

The distributed time synchronization algorithm can be expressed as

$$\begin{aligned} T(n+1) &= \begin{bmatrix} t_1(n+1) \\ t_2(n+1) \\ \vdots \\ t_N(n+1) \end{bmatrix} = W \cdot T(n) \\ &= \begin{bmatrix} w_{11} & \cdots & w_{1N} \\ \vdots & \ddots & \vdots \\ w_{N1} & \cdots & w_{NN} \end{bmatrix} \cdot \begin{bmatrix} t_1(n) \\ t_2(n) \\ \vdots \\ t_N(n) \end{bmatrix} \end{aligned} \quad (3)$$

Here, T is a vector of the set of every node's next time state, and W is the weight matrix. As shown in equation (3), the distributed time synchronization algorithm is expressed by a distributed average formula. If the WSN is a connected network, which means that the weight matrix is an irreducible matrix, the time reference of every node in the network will converge to a same value after a few numbers of iterations. However, if more than one abnormal nodes in the network do not comply with the synchronization algorithm, it will lead to un-convergence of the algorithm.

Synchronization error model

Assume that an abnormal node appears in the network, whose state of time reference does not change according to the distributed synchronization algorithm. The existence of abnormal node will influence time reference of its neighbor nodes in the next iteration of the synchronization algorithm, such that its neighbor nodes will have incorrect time references. These neighbor nodes with error time references will further influence their nearby nodes to get incorrect time references. The error of time reference will propagate as the algorithm iterates, which is shown in Figure 1.

Supposed that the node k is an abnormal node which do not change its local time reference according to the synchronization algorithm. We use white Gaussian noise (WGN) to represent the state of k . The synchronization model with abnormal node can be expressed as

$$t_i(n) = \begin{cases} \sum_{j=1}^N w_{ij}t_j(n) & i \neq k, 1 \leq j \leq N \\ \sigma(n) & i = k, 1 \leq j \leq N, \sigma(n) \in WGN \end{cases} \quad (4)$$

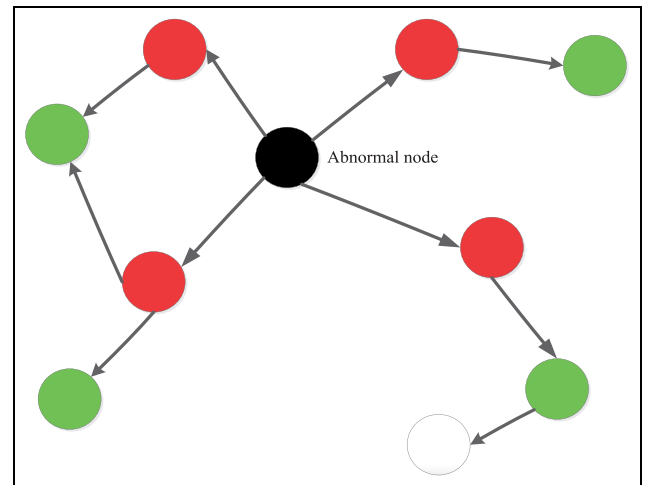


Figure 1. The spread of error information.

In equation (4), the time reference of all the normal nodes expect the abnormal node change according to equation (2). As the existence of abnormal node will affect the time reference of other nodes in the network, we should study the convergence problem of the synchronization algorithm.

If there are more than two abnormal nodes which do not change their time references in the network—all other $N - k(k \geq 2)$ nodes adjust their time references according to the synchronization algorithm—we will subsequently prove that the algorithm will not converge.

Assumed that node p and node q could not adjust their local time references, we mark the time reference of node p and node q as t_p and $t_q(t_p \neq t_q)$, respectively. If the time references of all other normal nodes are converged to t^* , $t^* \neq t_p$ ($t^* \neq t_q$ is the same similar way). According to equation (4), the time reference of node i (one of the neighbors of node p) will not be equal to t^* in the next iteration. Only when $w_{ip}t_p + w_{iq}t_q = t^*$ can we keep the time references of other normal nodes in the same value. Otherwise, the synchronization algorithm will not converge. Therefore, the detection of the abnormal nodes becomes an essential issue.

Correlation coefficient

According to the synchronization error model, all the normal nodes current state of time reference are computed by the distributed average of their neighbors' previous state of time reference. Thus, the time reference of a normal node will linearly correlate to that of their neighbor nodes. However, the time reference of the abnormal node is independent from those of its neighbors, since it does not adjust its local time reference according to the synchronization algorithm. The main idea of the proposed abnormal node detection method is that every node should compare the correlation coefficient between the local time reference and that of its neighbors.

In statistics, there are several tools to analyze the correlation between two sample groups, such as Pearson product-moment correlation coefficient (PPMCC), Spearman's correlation coefficient for ranked data, and the Kendall coefficient of concordance. These tools have different usage scenarios; we choose PPMCC in this article because PPMCC is a good metric for measuring the linear dependence.

PPMCC is a frequently used tool in statistics; it is used to measure the linear correlation between two variables. The correlation value of PPMCC ranges from -1 to 1 . If the value is larger than 0 , it means that the two variables measured are positively correlated. Otherwise, they are regarded as negatively correlated. If the value is close to 0 , it means the two variables are uncorrelated.

The PPMCC of two continuous variables is defined as the quotient of their covariance and standard deviation, that is

$$\rho_{XY} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \quad (5)$$

Here, X and Y are two values to be measured. σ_X, σ_Y are the standard deviations of X, Y . $\text{cov}(X, Y)$ is the covariance coefficient

$$\text{cov}(X, Y) = E(X - E(X))E(Y - E(Y)) \quad (6)$$

$E(X)$ and $E(Y)$ are the expectations of X and Y . For each node in the network, the time reference of state in sequence are discrete values. We use the time average, instead of ensemble average, as a statistical value to compute the mathematical expectation of the time reference of each node. Supposed that the number of historical time state used for computing the expectation is M , which is shown in equation (7)

$$\begin{cases} \hat{E}(X) = \frac{\sum_{n=1}^M x(n)}{M} \\ \hat{E}(Y) = \frac{\sum_{n=1}^M y(n)}{M} \end{cases} \quad (7)$$

The standard deviations are expressed in equation (8)

$$\begin{cases} \hat{\sigma}_X = \sqrt{\frac{\sum_{n=1}^M (x(n) - \hat{E}(X))^2}{M}} \\ \hat{\sigma}_Y = \sqrt{\frac{\sum_{n=1}^M (y(n) - \hat{E}(Y))^2}{M}} \end{cases} \quad (8)$$

Therefore, the covariance can be evaluated as

$$\hat{\text{cov}}(X, Y) = \frac{\sum_{n=1}^M (x(n) - \hat{E}(X))(y(n) - \hat{E}(Y))}{M} \quad (9)$$

The correlation coefficients of the time reference of nodes in WSNs, which is expressed in equation (9), are used as the parameter to detect the abnormal nodes.

Algorithm description

In this article, we assume that the topology is static and there is one abnormal node at most. As Figure 2 shows, the algorithm consists of two forms: non-real-time detection and real-time detection. The former is used to find out the abnormal node when the network has been un-synchronized, which needs less computation cost. The latter is working with synchronization algorithm to monitor the network and detect the abnormal node.

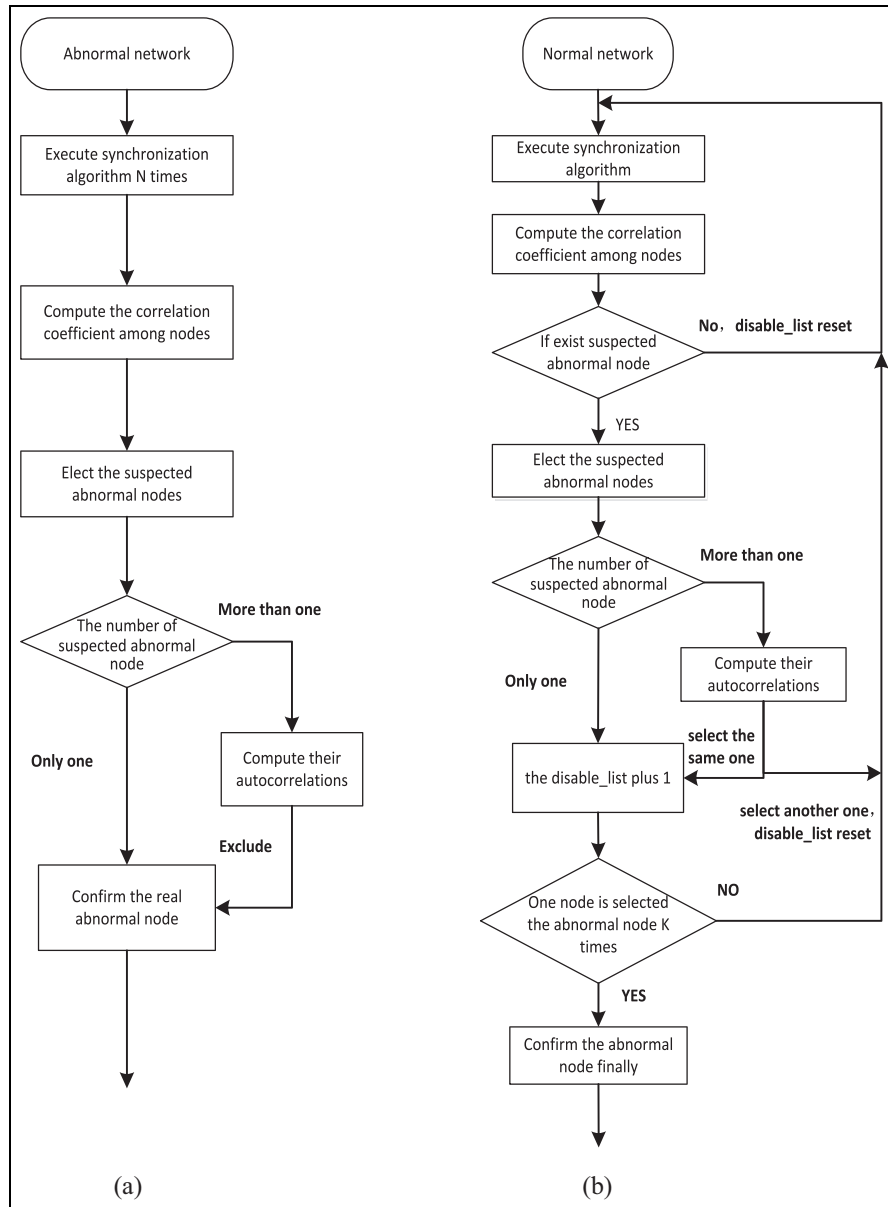


Figure 2. Algorithm flowchart: (a) non-real-time detection and (b) real-time detection.

Non-real-time detection

The non-real time detection algorithm is shown in Figure 2(a). In this form, the network is un-synchronization. Then the follow steps are used to find out the abnormal node:

1. Execute the synchronization algorithm N times according to equation (4), and each node stores the local time reference states in a sequence.
2. Each node computes the coefficient of time reference between itself and its neighbors by the PPMCC method. Because node X 's next state is affected by its neighbor Y 's current state, we

choose the samples of X from 1 to $N - 1$ and the sample of Y from 2 to N to compose an observation pair. The correlation coefficient ρ_{XY} shows the influence of X to Y .

3. After computing the PPMCC, every node chooses its neighbor node whose correlation coefficient is the smallest and notices the chosen node that it has a problem. If one node is noticed by all its neighbors, the node will be regarded as a suspected abnormal node.
4. If there is only one suspected abnormal node, the node been chosen is confirmed as the abnormal node. If there exists more than one

suspected abnormal nodes, they will compute their autocorrelations, respectively (its buffer from 1 to $N - 1$ and its buffer from 2 to N), and the one with the smallest value is confirmed to be the abnormal node.

Real-time detection

The real-time detection is shown in Figure 2(b). In this scenario, the network has been synchronized, and the algorithm is used to monitor whether one normal node becomes abnormal suddenly. Every node has a N length buffer to store the states of time reference and has a counter disable node list to record the nodes which are selected as abnormal nodes.

Before the abnormal node appears, the nodes execute the synchronization algorithm, update the buffers, and compute the PPMCC:

1. If the PPMCCs of node X are all 1, it means X is absolutely linearly correlated with its neighbors and X does not need to choose a suspected node, so the disable node list will be reset. Otherwise, X needs to select the neighbor with smallest PPMCC as non-real-time detection do and then notices it.
2. If there exists only one suspected abnormal node, its disable node list will plus 1. If there are several suspected abnormal nodes, we will do it by the same way which non-real-time detection does in step 4.
3. The node selected as the final suspected abnormal node will plus its disable node list, and the other suspected abnormal nodes' disable node list will reset.
4. If one node's disable node list comes to be k , it will be confirmed as the final abnormal node.

Feasibility analysis

As mentioned previously, the abnormal node is like a virus, it will spread error information to its neighbors, and the neighbors will also spread the error information to their nearby nodes.

In Figure 3, the strength of the influence from the abnormal node is shown. Darker color means the nodes suffer more influence from the abnormal node. Similarly, the number shown between two nodes indicates the strength of their PPMCC. The larger the number, the weaker the correlation coefficient. Since one node will have weak linear dependence with its neighbors if it is much affected by the abnormal node, the node will be selected as the problematic node with a higher probability.

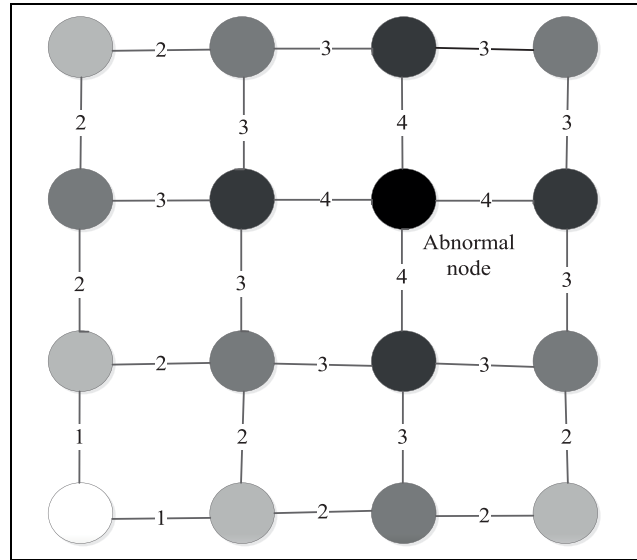


Figure 3. Influence of abnormal node.

Therefore, the abnormal node will surely be selected as the suspected abnormal node in this way. However, one of the abnormal node's neighbor also has the probability to be selected as the suspected abnormal node, because the abnormal node also chose one of its neighbor to notice as the abnormal node. If one node is noticed by the abnormal node as an abnormal node, and it is also noticed by other neighbors at the same time, the node will be another suspected abnormal node. In this way, there will be two suspected nodes at most: one of them is the abnormal node, and the other one is its neighbor.

If there exists two suspected abnormal nodes, we just need to compute the autocorrelation value. Because the next state of time reference in one normal node contains the information of the current state of time reference, the time reference of the abnormal node is uncorrelated to its previous state. The node with weaker autocorrelation can be confirmed as the abnormal node.

If the degree of abnormal node is 1, its only neighboring node will find that the correlation coefficient with the abnormal neighboring node is lower than that with another normal neighboring node. Hence, the abnormal node will be detected. If a normal node connects to the network only by an abnormal node, its time reference is adjusted by that of the abnormal node. The time reference of the normal node may be incorrect because the abnormal node is its only neighboring node. However, it does not affect other nodes in this circumstance, because the normal node with incorrect time reference does not have neighbor of any other normal nodes. Therefore, adopting the statistical linear correlation analysis approach is an effective method to detect anomaly nodes.

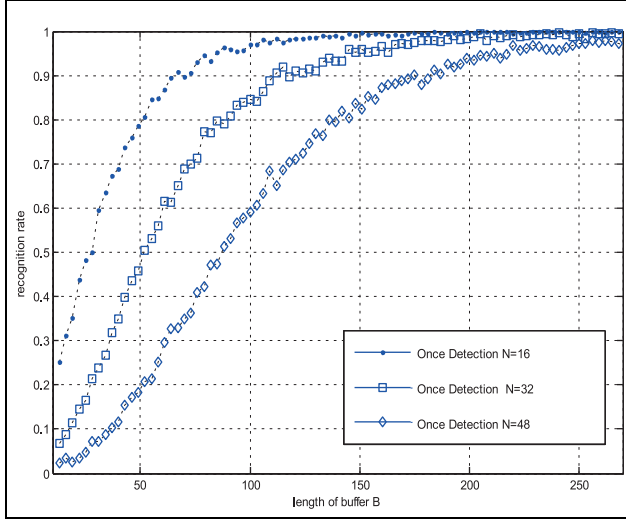


Figure 4. The influence of the length of buffer B to recognition rate in non-real-time detection ($r = 0.3, N = 16, 32, 48$).

Simulation and results

In this article, we simulate in a normalized 1×1 square area. In the simulation area, there are N nodes with uniform distribution. The communication radius is r . If the distance between two nodes is less than r , it means they are neighbor nodes. Only the neighbor nodes can communicate with each other, and each packet from a node can be sent to another node in the network with multi-hop. In the simulation, the main parameters we observed are node density N , communication radius r , and length of buffer B .

Non-real-time detection

Non-real-time detection is used to find out the abnormal node when the network has been un-synchronized.

Length of buffer B . The non-real-time scenario, the abnormal nodes detection algorithm was run one time to discover the influence of length of buffer to the detection rate, is marked as “once detection” in Figure 4.

In Figure 4, we can find that the length of buffer B has a great influence on the recognition rate. When B is small, the recognition rate will be low. Moreover, with the length of buffer B increasing, the recognition rate gets larger; thus, it will finally reach 1 and stay at this value. Our algorithm is based on the correlation of time reference between nodes, so a larger number of samples contributes to a more accurate result. Here, the length of buffer means the number of time reference samples. So large B can lead to high recognition rate.

Besides, we can find that the result with less nodes is better than the result with more nodes when the buffer is fixed.

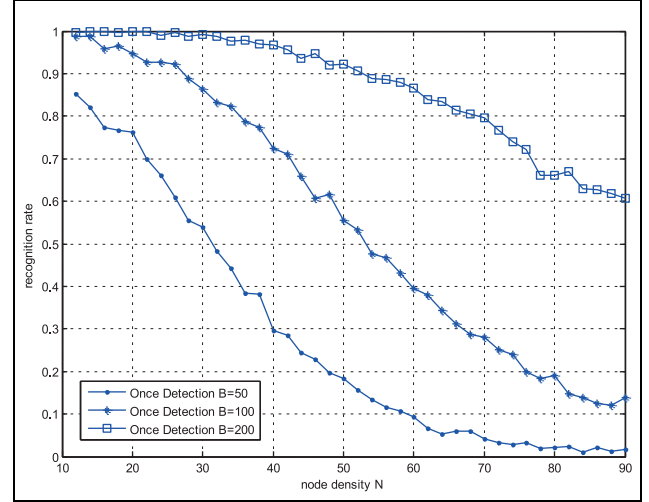


Figure 5. The influence of the node density N to recognition rate in non-real-time detection ($r = 0.3, B = 50, 100, 200$).

Node density N . The relationship between the node density and the detection rate is shown in Figure 5.

As Figure 5 shows, when the length of buffer is fixed, with the node density N increasing, the recognition rate of the system will decline. In this article, the synchronization algorithm we used is related to the weight and node degree. The bigger the node degree is, the smaller the weight will be. This means the correlation is weak and the probability of erroneous judgment is high. Therefore, if we increase the node density N , the weight will decrease, and it becomes more difficult to find the abnormal node.

On the other hand, with the whole node density N increasing, the density of abnormal node becomes small. Our algorithm is based on the unanimous voting. If one neighbor of the abnormal node does not notice the abnormal node, it cannot be detected. So, it should have more samples to compute the PPMCC accurately. From Figure 5, we can find that the performance of the system can be maximized if the length of buffer is large enough.

Real-time detection

Real-time detection is working with synchronization algorithm to monitor the network and detect the abnormal node.

Length of buffer B . In real time detection scenario, we need to observe the iteration time which began from the abnormal node appears until it has been detected. We call the iteration time as the cost. In the real-time scenario, the influence of length of buffer to the detection rate is shown in Figure 6.

Comparing Figure 6 with Figure 4, we can find that the real-time detection method is more efficient than

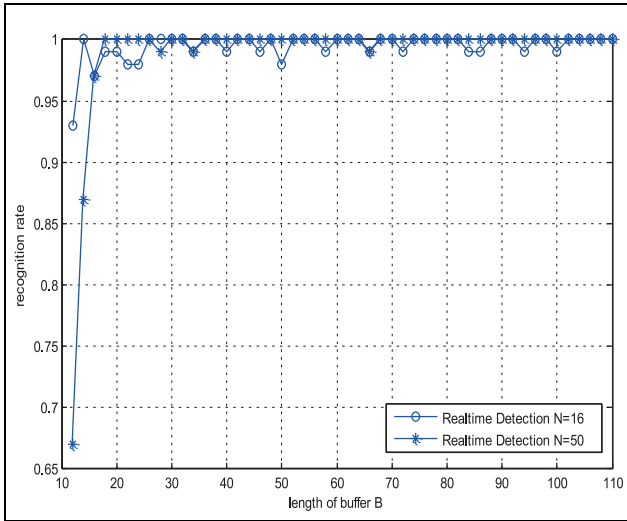


Figure 6. The influence of the length of buffer B to detection rate in real-time detection ($r = 0.3, N = 16, 50$).

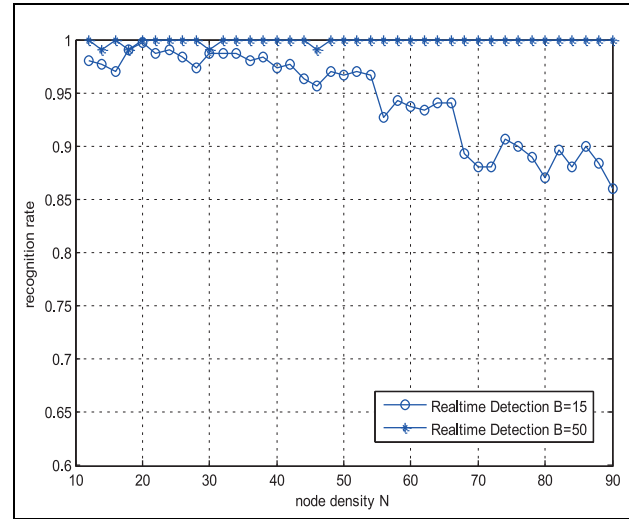


Figure 8. The influence of the node density N to detection rate in real-time detection ($r = 0.3, B = 15, 50$).

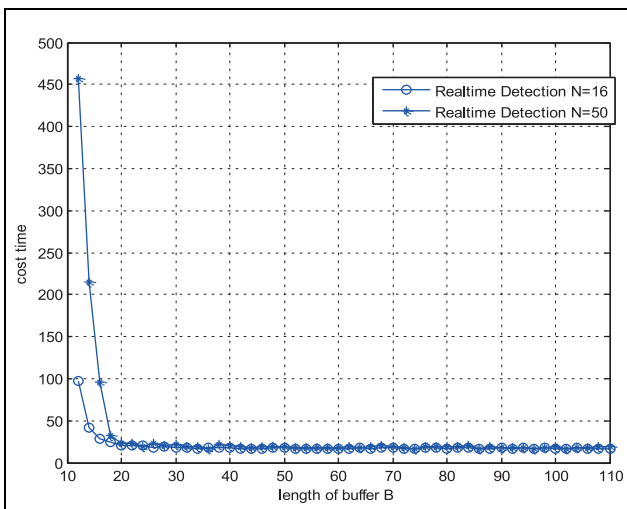


Figure 7. The influence of the length of buffer B to cost in real-time detection ($r = 0.3, N = 16, 50$).

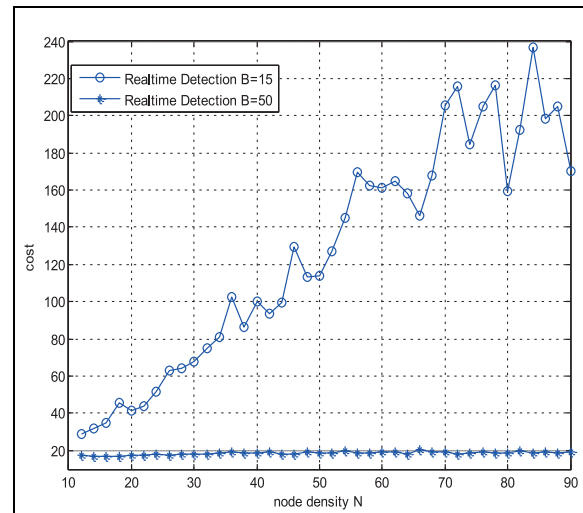


Figure 9. The influence of the node density N to cost in real-time detection ($r = 0.3, B = 15, 50$).

the non-real-time detection method. When node density and communication radius are fixed, real-time detection can reach the same performance with less buffer demand. In real-time detection, the abnormal node is immediately detected when it appears, where most of the nodes in the network, except the abnormal node and its neighbor nodes, are still in synchronization. Hence, the error information from the abnormal node has not been widely spread and it will be easier to detect the abnormal node with less time reference samples.

The influence of the length of buffer B to cost is shown in Figure 7.

It is seen from Figure 7 that the real-time detection method requires more computation cost, with the

buffer increasing, and the cost of time to detect the abnormal node would decline.

Node density N . The influence of the node density N to detection rate and the influence of the node density N to cost are shown in Figures 8 and 9, respectively.

From Figures 8 and 9, we can find that when the length of buffer is not large enough, the abnormal node's detection rate and computation cost will all decline as the node density increases. The reason is that the weight value in the distributed time synchronization algorithm declines when the node density increases; thus, the correlation value of time reference between

neighbor nodes will be smaller. Therefore, more iteration time is required to achieve the detection.

Conclusion

In this article, a novel distributed time synchronization algorithm has been proposed for CRSNs. The proposed algorithm can effectively detect the abnormal nodes that fail to comply with the normal synchronization behavior. Our algorithm relies on a key insight that the time reference trace of the abnormal node tends to be linearly uncorrelated with its neighboring nodes. This insight is used to design a voting and consensus mechanism to detect abnormal nodes using linear correlation as a metric. Simulation results show that the proposed method can effectively detect the abnormal node in CRSNs in both the non-real-time and real-time scenarios.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors would like to acknowledge the support of the China Scholarship Council (CSC) and the Natural Science Foundation of China (NSFC, nos 61201196, 61601388, 61571378, and 61371081).

References

- Ni Z and Liu W. One robust time synchronization based on time-average and maximum-likelihood estimation for WSN. In: *International conference on information science and control engineering*, Shanghai, China, 24–26 April 2015, pp.354–358. New York: IEEE.
- Dong H, Wang Z, Ding SX, et al. A survey on distributed filtering and fault detection for sensor networks. *Math Probl Eng* 2014; 2014: 858624.
- Zhang N, Cheng N, Lu N, et al. Risk-aware cooperative spectrum access for multi-channel cognitive radio networks. *IEEE J Sel Area Comm* 2014; 32(3): 516–527.
- Wang D and Song Y. A novel distributed global time synchronization protocol in cognitive radio ad hoc networks. In: *IEEE global communications conference*, San Diego, CA, 6–10 December 2015, pp.1–6. New York: IEEE.
- Zhang N, Liang H, Cheng N, et al. Dynamic spectrum access in multi-channel cognitive radio networks. *IEEE J Sel Area Comm* 2014; 32(11): 2053–2064.
- Ruiz LB, Siqueira IG, Wong HC, et al. Fault management in event-driven wireless sensor networks. In: *Proceedings of the 7th ACM international symposium on modeling, analysis and simulation of wireless and mobile systems*, Venice, 4–6 October 2004, pp.149–156. New York: ACM.
- Koushanfar F, Potkonjak M and Sangiovanni-Vincentelli A. On-line fault detection of sensor measurements. In: *2003 IEEE proceedings of sensors*, Toronto, ON, Canada, 22–24 October 2003, pp.974–979. New York: IEEE.
- Ssu KF, Chou CH, Jiau HC, et al. Detection and diagnosis of data inconsistency failures in wireless sensor networks. *Comput Netw* 2006; 50(9): 1247–1260.
- Liu Y, Liu K and Li M. Passive diagnosis for wireless sensor networks. *IEEE ACM T Network* 2010; 18(4): 1132–1144.
- Krunic V, Trumpler E, Han R, et al. *NodeMD: diagnosing node-level faults in remote wireless sensor system*. New York: ACM, 2007.
- Weber A, Kutzke AR and Chessa S. Diagnosability evaluation for a system-level diagnosis algorithm for wireless sensor networks. In: *2010 IEEE symposium on computers and communications (ISCC)*, Riccione, 22–25 June 2010, pp.241–244. New York: IEEE.
- Chessa S and Santi P. Crash faults identification in wireless sensor networks. *Comput Commun* 2002; 25(14): 1273–1282.
- Duarte EP Jr, Ziwich RP and Albin LCP. A survey of comparison-based system-level diagnosis. *ACM Comput Surv* 2011; 43(3): 22.
- Elhadeif M, Boukerche A and Elkadiki H. Diagnosing mobile ad-hoc networks: two distributed comparison-based self-diagnosis protocols. In: *Proceedings of the 4th ACM international workshop on mobility management and wireless access*, Terromolinos, 2 October 2006, pp.18–27. New York: ACM.
- You Z, Zhao X, Wan H, et al. A novel fault diagnosis mechanism for wireless sensor networks. *Math Comput Model* 2011; 54(1–2): 330–343.
- El-Koujok M, Benammar M, Meskin N, et al. Multiple sensor fault diagnosis by evolving data-driven approach. *Inform Sci* 2014; 259: 346–358.
- Vuran MC, Akan OB and Akyildiz IF. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Comput Netw* 2004; 45(3): 245–259.
- Xiao X-Y, Peng W-C, Hung C-C, et al. Using sensor ranks for in-network detection of faulty readings in wireless sensor networks. In: *Proceedings of the 6th ACM international workshop on data engineering for wireless and mobile access*, Beijing, China, 10 June 2007, pp.1–8. New York: ACM.
- Behnke R, Salzmann J, Simanowski S, et al. Efficient localized detection of erroneous nodes (ELDEN). In: *2010 IEEE 10th international conference on computer and information technology (CIT)*, Bradford, 29 June–1 July 2010, pp.2884–2890. New York: IEEE.
- Babaie S, Khosrohosseini A and Khadem-Zadeh A. A new self-diagnosing approach based on petri nets and correlation graphs for fault management in wireless sensor networks. *J Syst Architect* 2013; 59(8): 582–600.
- Mahapatro A and Khilar PM. Detection of node failure in wireless image sensor networks. *ISRN Sens Netw* 2012; 2012: 342514.
- Mahapatro A and Khilar PM. Energy-efficient distributed approach for clustering-based fault detection and diagnosis in image sensor networks. *IET Wirel Sens Syst* 2013; 3(1): 26–36.