Research on key network node identification based on depth - first search algorithm

Hao Sun[1, a], Xiaoxia Cai[2, b] and Zufeng Zhang[3, c]

[1] No.460 Huangshan Road, Shushan District, Hefei, China

[2] No.460 Huangshan Road, Shushan District, Hefei, China

[3] No.460 Huangshan Road, Shushan District, Hefei, China

[a]1582892650@qq.com, [b]caixiaoxia@163.com, [c]191893549@qq.com

Key words: complex networks; topological properties; key nodes

Abstract: The recognition of critical nodes has important application value in the real world. On the one hand, it can improve the survivability of the network by improving their performance, on the other hand, they can also destroy some networks by deliberately attacking them. In this paper, DSF algorithm is proposed to identify key nodes in the network. Experimental results show that the proposed method can identify key nodes efficiently and effectively.

Introduction

At present, the protection and protection of key nodes in complex networks are mainly focused on the protection of nodes in random networks[1]. However, it is found that complex networks are not scale-based networks, but are not stochastic networks[2,3]. Therefore, the previous experience in identifying and protecting critical nodes is not suitable for complex networks with robustness, fragility and scale-free characteristics[4,5] , There is an urgent need for new methods and techniques to carry out research.

In this paper, the key nodes in wireless sensor networks are probed, so it is assumed that the graphs formed by all the nodes and each node and its neighbor nodes are connected in advance, so we need to give the concept of graph traversal[6,7]. A traversal of a graph means that each node in the network accesses each node along the path from one node to another[8].

DFS Algorithm Flow

The specific process of the algorithm is as follows: Assume that each node of graph *G* has not been detected before, then select one node *i* as the initial starting node in graph *G*, first detect node *i* and mark the node, then If the node *j* has not been detected, then the node *j* is detected and the node *j* has been visited. If the node *j* has been visited, the node *i* is reselected. A neighbor node, and then repeat the previous step with the node *k* as the new starting node *k* until all the nodes that the node *j* wants to detect are detected. If at this time there is still a node in the graph *G* that has not been probed, Node as the starting point, repeat the above process until all the nodes are detected so far.

From the above can be seen, *DFS* algorithm is as far as possible the first direction of the depth of search, which is the specific origin of its name.

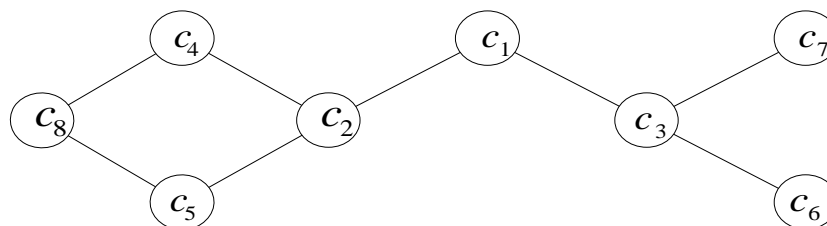The following is an example of the *DFS* algorithm, assuming that the network topology is shown in Fig. 1.

Fig. 1 undirected graph

As shown in Fig. 1, it searches from its node c_1. When it detects the node c_1 and marks it, it should probe its neighbor node and select its neighbor node c_2 because it is not detected. c_2 searches and marks, and then searches the neighbor of c_2 from c_2. Assuming that neighbor c_4 of c_4 first searches c_4 and detects that c_4 has not been detected, c_4 is detected and labeled, and then c_4 is a new starting point, If the neighbor node c_8 is not detected, it detects and marks c_8, then detects the neighbor node c_8 with c_8 as the new starting point, detects the neighbor node c_5 that is not detected, detects and marks c_5, For the new starting point, search the neighbors of B, found that all have been marked detection, so this path is detected completed.

Since the neighbor node of node c_2 has already probed the path of neighbor node c_4, it also has neighbor node c_5, so it detects the node c_5 has been detected, so c_2 and its child node Be considered a real search is completed.

As in the detection of c_2, we found that its neighbors more than c_2, c_2 has just been on the path of the search, so the following c_1 of the other neighbors c_3 where the path detection, detection of c_1's neighbor c_3 is not detected, so the c_3 is detected and marked, and then c_3 as a new starting point to detect the neighbors of c_3, c_3 detected neighbor c_6 is not detected, so the detection of c_6 and mark, and then to c_6 as a new starting point to detect c_6's neighbors, Since c_6 has no other neighbors, path $c_1 - c_3 - c_6$ has been searched.

As in the detection of c_3, found its neighbors more than c_6, there are c_7, and has not been detected, so the detection of c_7 and mark, and then to c_7 as a new starting point, the neighbors of c_7 detection, detection has no other Neighbors, so this path $c_1 - c_3 - c_7$ is considered to be completed.

So far all the nodes in Fig. 1 have been detected.

DFS for the Detection of Key Nodes

The detection of key nodes in the graph using the DFS algorithm will be further analyzed below.

In the above discussion, a possible order in which the nodes in the entire network are searched is pointed out, that is, their order of precedence is as follows: $c_1 - c_2 - c_4 - c_8 - c_5 - c_3 - c_6 - c_7$。 Then the DFS algorithm is how to determine that the node is the key node that the cut point of the map it?

First, the following definition is given.

$DFS(i)$ is defined as the sequence number of the node i is searched, because the order in which the nodes are accessed is $c_1 - c_2 - c_4 - c_8 - c_5 - c_3 - c_6 - c_7$, so $DFS(c_1) = 1$, $DFS(c_2) = 2$, $DFS(c_4) = 3$, $DFS(c_8) = 4$, $DFS(c_5) = 5$, $DFS(c_3) = 6$, $DFS(c_6) = 7$, $DFS(c_7) = 8$

Let c_2 be a child of c_1, c_4 be a child of c_2, c_8 be a child of c_4, c_5 be a child of c_8, and for a child of c_5, let j be the child of node i, as in the above search order, Because the search sequence according to the above discussion, it has a neighbor node logic is no subsequent child nodes, so that it does not have children, and then c_3 is a child of c_1, is a child node, the same c_7 is a child of c_3, c_6 and c_7 has no child nodes.

If j is a child of node i, then A defines the search number of the earliest node in the subtree of j that can be traced back to node i by non-parent-side edges (i and j sides). The following is the key node to determine the method, that is:

$$DFS(i) \le A(j) \qquad (1)$$

Then node i is the key node.

Take the example of Fig. 2 as an example. According to the definition of the key nodes, we can see that the nodes c_1 and c_2 and c_3 are the key nodes in the graph, because the failure of the node c_1 will cause the network to form two regions.

Failure of Node c_2 results in two areas as shown in Fig. 3. If node c_3 fails, it will cause two areas as shown in Fig. 4. In addition, if any other nodes fail, they can not divide the network into two or two or more.
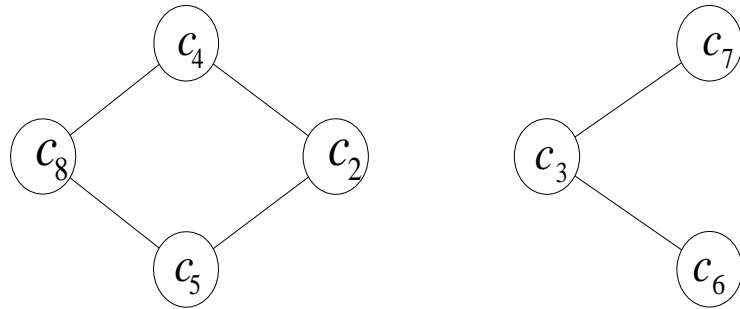

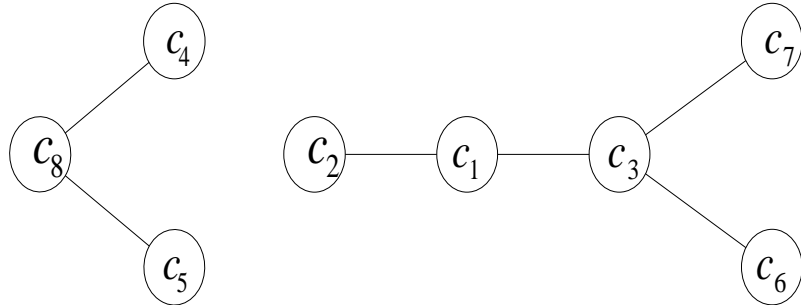Fig. 2 Topology after node c_1 failure

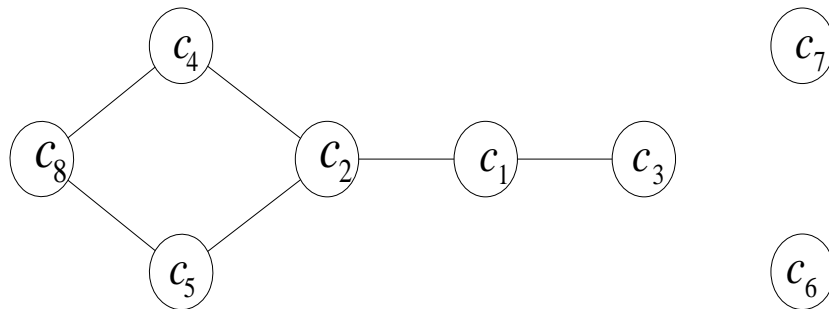Fig. 3 Topology after node c_2 failure

Fig. 4 Topology after node c_3 failure

The following formula to determine whether each node is a key node, so that the results of the above analysis and comparison.

For node c_1, $DFS(c_1) = 1$, it must be the smallest, and it must satisfy the formula, so it is judged as the key node.

For node c_2, $DFS(c_2) = 2$ c_4 is its child node, $A(c_4) = DFS(c_5) = 5$ and the formula is satisfied, so node c_2 is the key node.

For node c_4, $DFS(c_4) = 3$, c_8 is its child node, $A(c_8) = DFS(c_2) = 2$, and the formula is satisfied, so node c_4 is not the key node.

For node c_8, $DFS(c_8) = 4$, c_5 is its child node and does not satisfy the formula, so node c_8 is not the key node.

For node c_5, $DFS(c_5) = 5$, since it has no child nodes, it is considered to be at the edge, so it is not the key node. In fact, even if c_5 has child node c_2, $A(c_2) = DFS(c_8) = 4$, it is not satisfied, so it can still be judged Node c_5 is not a key node. As for the key nodes, it is a bridge connecting two networks, no matter from the start of the search, it will not be considered to be at the edge.

For node c_3, $DFS(c_3) = 6$, where c_6 is its child node, $A(c_6) = \infty$, the formula is satisfied, so node C is the key node. It should be noted that, $A(c_6) = \infty$ since, according to the definition of A, the sub-edges of c_6 must pass through some nodes and edges to c_3, but here c_3 has no sub-edges and therefore can not be reached by its sub-edges.

Conclusions

In this paper, the *DSF* algorithm is introduced and the key nodes are identified by using DSF algorithm. The experimental results show that the algorithm is simple and effective for network key nodes identification. At the same time, *DSF* algorithm is relatively simple and easy to program simulation, we can see that the algorithm used in the identification of key nodes is very simple and effective.

Acknowledgements

References

[1]Wasserman S, Faust K. Social Network Analysis. Cambridge University Press, 1994.
[2]Ying-Dar L, Yu-Ching H. Multihop cellular: a new architecture for wireless communications [A]. In: INFOCOM 2000 Proceedings [C]. US: IEEE Press, 2000, 1273-1282.
[3] Jeong H, Tombor B, Albert R,et al. The large-scale organization of metabolic networks. Nature, 2000, 407: 651-654.
[4] Williams R J, Martinez N D. Simple rules yield complex food webs. Nature 2000,
[5] Huang L, Wang G, Wang Y, Enrico Blanzieri, Chao Su. Link Clustering with Extended. Link Similarity and EQ Evaluation Division[J]. Plo S one, 2013, 8(6): e66005.
[6] Watts DJ, Strogatz SH. Collective dynamics of Small-World networks. Nature, 1998, 393(6638):440 442.
[7] Barabási AL, Albert R. Emergence of scaling in random networks. Science, 1999, 286(5439):509 512.
[8] Barabási AL, Albert R, Jeong H, Bianconi G. Power-Law distribution of the World Wide Web. Science, 2000, 287(5461): 2115a.