Minimum Spanning Tree for Variant Weighting

Wen-Pinn Fang

Yuanpei Universiyy, Department of Computer Science and Information Engineer, Taiwan , R.O.C

Summary
This paper proposed a method to find out the number of minimum spanning tree with time variant weighting network. Using the proposed method, it is possible to find out how many devices are required to keep the quality of transmission good in a ad hoc network.
Key words:
Minimum spanning tree, Linear programming, network topology, optimum

1. Introduction

Recently, the progress of mobile device technology is very fast. More and more people use mobile network every day. To make the quality of transmission better and make the price cost down is very important. This paper focuses on the improvement to the transmission quality of a mobile network. The aim of this paper is find the time to change the network topology in order to keep the quality of transmission and keep lowest power requirement. Because the location of nodes in a network is dynamic and the power need is inversely proportional to the distance of two connect nodes. That is the best topologies of an ad hoc network will change. The traditional algorithm is not very suitable for this situation. The minimum spanning tree will change after weighting are different.

The rest of this paper is organized as follows: the problem definition introduced in Section 2; Relative knowledge is described in the section 3; the proposed method is demonstrated in Section 4; and the results are presented in Section 5; the conclusions are given in Section 6.

2. Problem definition

In this section, the problem will be defined. In the original idea, the paper will design an algorithm to find out how many device are necessary to keep the cost is minimum. The problem can be transform to find out how many minimum spanning tree when the weighting is time variant. It is very difficult to determine the number of device if the weighting functions alternate. To simply it, assume the weighting function is a polynomial function.

Manuscript received July 5, 2012
Manuscript revised July 20, 2012

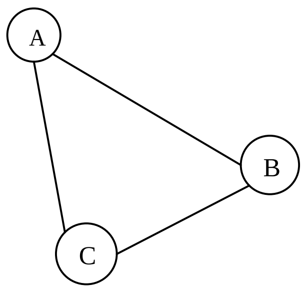
To describe the problem clear, here show the symbolic table and an example.

The question is shown as below: There are three nodes in the network. To connect the network, just two edges are necessary.

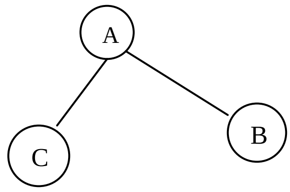
Table 1: Symbolic tables

Symbol	*meaning*
G	A Graphy
V	Vertex
E	Edge

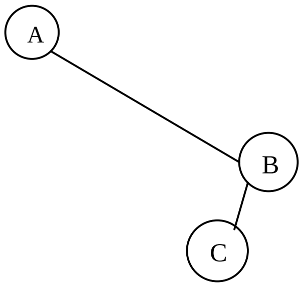
As shown in Fig.1. if there are three nodes(vertices) :A,B and C. The physical meaning of A,B and C are mobile phone, tablet or notebook. Because A, B and C will change their location continually. The minimum spanning tree will change, just like Fig.1(b)-(d)



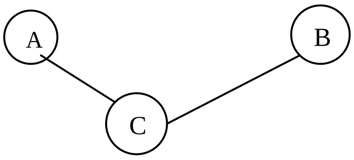
(a)



(b)



(c)



(d)

Fig. 1 an example of the problem, .(a) is the initial state, (b)-(d) are the minimum spanning tree in different weight

Assume there are *n* nodes in an ad hoc network system. It can be described a

input: a complete graph G(V,E)

output: the number of minimum spanning tree

3. Relative knowledge

In order to describe the proposed method, some background knowledge is shown as below:

3.1 Kruskal's algorithm[1] is a greedy algorithm in graph theory that finds a minimum spanning tree for a connected weighted graph. The procedure is shown as below:

1. create a forest F (a set of trees), where each vertex in the graph is a separate tree
2. create a set *S* containing all the edges in the graph
3. while *S* is nonempty and F is not yet spanning

    3.1 remove an edge with minimum weight from *S*

    3.2 if that edge connects two different trees, then add it to the forest, combining two trees into a single tree

    3.3 otherwise discard that edge.

The time complexity of Kruskal's algorithm is O(E log E) = O(E log V).

3.2 Depth-first search [2] is an algorithm for traversing or searching a tree, tree structure, or graph. One starts at the root (selecting some node as the root in the graph case) and explores as far as possible along each branch before backtracking.

The time complexity of Kruskal's algorithm is O(|V|+|E|)

4. Proposed method

Input : G(V,E,)

Output: number of minimum spanning tree

Step 1: find out all tree in G

Step 2: determine all

$$y_i(N) = \sum_{n=0}^{m-1} w_n(N) b_n(N)$$
$$= \sum_{n=0}^{m-1} \overline{b_n^*(N) r_i(N)} \cdot b_n(N)$$

5. An example

Assume that there are three node A, B and C. The locations are P_{A0}, P_{B0},and P_{C0}. The weighting functions are linear equation. $V_A(t)$, $V_B(t)$ and $V_C(t)$.

$$P_A(t)=P_{A0}+V_A(t) \quad (1)$$
$$P_B(t)=P_{B0}+V_B(t) \quad (2)$$
$$P_C(t)=P_{C0}+V_C(t) \quad (3)$$

The weightings are

$$W_{AB}(t)=|P_A(t)-P_B(t)| \quad (4)$$
$$W_{AC}(t)=|P_A(t)-P_C(t)| \quad (4)$$
$$W_{BC}(t)=|P_B(t)-P_C(t)| \quad (4)$$

6. Conclusions

If you would like to itemize some parts of your manuscript, please make use of the specified style “itemize” from the drop-down menu of style categories

In the case that you would like to paragraph your manuscript, please make use of the specified style “paragraph” from the drop-down menu of style categories

References

[1] Joseph. B. Kruskal: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. In: Proceedings of the American Mathematical Society, Vol 7, No. 1, 1956), pp. 48–50

[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. MIT Press and McGraw-Hill, 2001, pp.540–549.

Wen-pinn Fang received his BS degree in Mechanical Engineering in 1994 from National Sun Yat-sen University and his MS degree in Mechanical Engineering in 1998 from National Chiao Tung University, where he also got his PhD degree in Computer Science in 2006. His recent research interests include image sharing, pattern recognition, image processing and e-learning.