Steiner Tree Based Optimal Resource Caching Scheme in Fog Computing

SU Jingtao[1], LIN Fuhong[1*], ZHOU Xianwei[1], LÜ Xing[2]

[1]School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China

[2]School of Science, Beijing Jiaotong University, Beijing, 100044, China

Abstract: Fog Computing is a new platform that can serve mobile devices in the local area. In Fog Computing, the resources need to be shared or cached in the widely deployed Fog clusters. In this paper, we propose a Steiner tree based caching scheme, in which the Fog servers, when caching resources, first produce a Steiner tree to minimize the total path weight (or cost) such that the cost of resource caching using this tree could be minimized. Then we give a running illustration to show how the Fog Computing works and we compare the traditional shortest path scheme with the proposed one. The outcome shows that the Steiner tree based scheme could work more efficiently.

Keywords: steiner Tree; resource caching; fog computing; architecture

I. Introduction

The Fog Computing, first proposed by Cisco in 2012 [1], is a platform providing computing, storage and network serving to mobile devices, which is still in its infancy. In [2], Luis M. Vaquero and Luis Rodero-Merino in 2014 defined it as "Fog computing is a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralized devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third-parties. These tasks can be for supporting basic network functions or new services and applications that run in a sandboxed environment. Users leasing part of their devices to host these services get incentives for doing so". Based on this definition, Fog Computing mainly serves wireless terminals, while the Cloud Computing mainly serves the general Internet users. Wireless terminals or mobile devices have some special characteristics, such as limited storage, computing power, wireless interface, and one-hop wireless connection during physical and communication [3]. These natural features require the Fog Computing designing a new service architecture.

A common architecture of Fog Computing is shown in Figure 1. There are three layers, namely Cloud Computing layer, Fog cluster layer and wireless terminal layer. The Cloud Computing is in the center of the network and mainly in charge of the resources storing. The Fog Clusters are in the edge of the network and responsible for serving the mobile devices. The mobile devices in a certain region communicate with the Fog Cluster who is in charge of them for services. Under this mechanism, the mobile devices can have the following advantages: (1) They need not store many resources, since they can retrieve required resources from the Fog cluster as long as they need; (2) They need not communicate with other mobile devices for resources sharing, and one retrieval action from the Fog cluster can get the proper resource. This can reduce the energy consumption; (3) They have a low latency when retrieving resources. This is because the Fog cluster is at the edge of the whole network. They can communicate and transmit data using a fewer hops.

In the Fog Computing, all the resources need to be shared or cached in the Fog cluster in order to provide the local service to the mobile devices. One key issue is how to efficiently accomplish this task, i.e., how to share or cache resources among certain Fog server with minimum costs. To this end, we leverage graph theory in this study, and find that the Steiner tree can be a good tool to deal with this problem [12]. The main idea is to find a connected tree which has the following characteristics: (1) The selected weighted edges are in the original graph; (2) The sum of the edge weights is the smallest; (3) The node set should contain some pre-determined nodes. Specifically, we propose the Steiner tree based scheme to analyze the resources sharing or caching in the Fog cluster. Then, we run an illustration to show how the Steiner tree is produced in the Fog Computing. And we compare the Steiner tree scheme and the traditional shortest path scheme to show the efficiency of our proposed one. The outcome shows that the Steiner tree scheme can bring down the cost of data sharing or caching.

The rest of this paper is organized as follows. Section II briefly summarizes the literatures on Fog Computing and the Steiner tree. Section III formally presents the problem formulation in the Fog Computing. We will introduce the Steiner tree building algorithm in Section IV, and Section V presents an illustration to show how the Steiner tree is produced in the Fog Computing. We will conclude this paper in Section VI.

II. RELATED WORK

To our knowledge, no work has been done for on the resources sharing or caching in Fog Computing so far. So in this Section, we give some related works on Fog Computing researching and Steiner tree researching which are the two important parts of this paper. The references [1-11] are almost all the works on Fog Computing, from the definition, architecture to the applications. We will give some representative ones to show what have done on this research area.

The Fog Computing was firstly proposed by Cisco in 2012 [1]. The authors stated that it should have the following characteristics: firstly, supports local process which brings down the time delay; Secondly, supports resources geographical distribution; Thirdly, supports end devices' Mobility; Fourthly, has the capacity of processing large number of nodes; Fifthly, supports wireless access; Sixthly, can provide streaming and real time applications; And lastly, supporting heterogeneity.

Authors in [4] added the webpage optimizing methods in the Fog Computing architecture. In this way, these methods can have the information of the Fog (edge) nodes

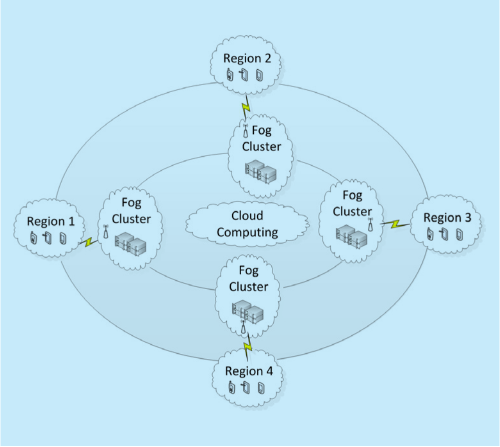


Fig.1 *The fog computing architecture*

In this paper, the authors propose a Steiner tree based caching scheme, in which the Fog servers, when caching resources, first produce a Steiner tree to minimize the total path weight (or cost) such that the cost of resource caching using this tree could be minimized.

to optimize the network performance. Using the network edge specific knowledge, user's conditions can be dynamically adapted. The user's webpage rendering performance is higher than just using these methods only.

Authors in [5] used the Fog Computing in augmented brain computer. They stated that the users' brain states in real-life situations can be detected using the augmented brain computer interface and those wearable interfaces would greatly impact the future application. So they used the Fog Computing to build ubiquitous system architecture. From the simulation, they verified the scheme characteristics of responsiveness and easiness-of-use.

Authors in [6] stated that cloud computing can meet the requirements of elastic provisioning of resources, and ubiquitous and on-demand access to data. Internet of Things connected the devices physically to the Internet. So it has the ability of sharing data among surrounding devices. The cooperation of these two techniques needs a new computing platform which can provide networking services, storage and computation between the traditional end-users and cloud computing data centers. They proposed a policy-based resources management in Fog Computing which can support secure collaboration and interoperability between the two communicated entities.

The above mentioned works give us an insight of what have done in Fog Computing. Some authors research on the definitions and architecture, and some authors work on Fog Computing applications. In this paper, we will focus on improving the performance of the Fog Computing.

The second related part is Steiner tree researching. The Steiner tree problem was named after Jakob Steiner and belongs to in combinatorial optimization problem. The aim is finding the shortest interconnect for a given set of nodes. The fundamental difference between the minimum spanning tree problem and the Steiner tree problem is that extra intermediate nodes and edges may be added to reduce the cost in Steiner tree scheme [12].

Authors in [13] proposed approximation algorithms for the Steiner tree problems. They added a new pre-processing phase so as to choose Steiner points in dependence on the possible deviation of the best solutions. According to the performance ratio test, their scheme could achieve higher performance comparing with others.

Authors in [14] researched on the Steiner tree problem in the condition of the given graph is acyclic directed. In order to get Steiner tree in this graph, a minimum cost sub-network should be obtained. In this sub-network, there should contain the paths from root to each node. The analysis showed that the new proposed scheme could improve the network performance in comparing with k – approximation.

Authors in [15] aimed at finding a faster algorithm for solving the Steiner tree problem. They used dynamic programming paradigm and relied on exhaustive search. They defined a new entity called region which is a maximal sub-tree and in this tree every leaf is a terminal and vice versa. Then, they proved that the optimal Steiner tree can be constructed by cooperating regions

So far, a lots of research work on Steiner tree can be found in the literature, such as BST [16], EBFSTP [17, 18], NWST or NWGST [19]. However, we just need the basic one to solve our problem, for it perfectly fits the scenario that we will build in Fog Computing.

III. Problem Formulation

In this paper, we focus on the improving of the Fog cluster layer's performance. This layer provides services to mobile devices. Before this action, the Fog cluster needs to share or cache resources with other Fog cluster in order to provide local service. We give a working illustration shown in Figure 2, which has 8 Fog server clusters. Each Fog cluster is composed by movie server, web server, file server and game server. All the servers are connected randomly to form a topology, and we call it the connecting layer. If a web service provider wants to publish some new webpages to certain web servers, it needs to share or cache these pages among them as shown in Figure 2 with yellow colour, and we call this layer as serving layer.

We know that the data transmission between any two connected servers will introduce some cost. How to efficiently cache resources among these web servers becomes an important issue. In other words, we need to complete the resources caching among certain servers in the condition of given the cost of the connection between any two servers.

This optimizing problem can be interpreted in a mathematical way as follows. There is a graph $G(G{=}N,E)$. N is the set of nodes and E is the set of edges. The weight of each edge i is e_i. There is another node set P ($P \subset N$). We needs to find a graph $G^*(G^*{=}P,E^*)$ in the condition of minimizing the total weight of E^*, where E^* is a set of edges connecting two end points in P. So the optimizing problem can written as:

$$OPT = \min\left(\sum_{i \in E^*} e_i\right) \tag{1}$$

$$\text{s. t.,} \begin{cases} P \subset N \\ e_i \subset E \\ G^* \quad being\_connected \end{cases} \tag{2}$$

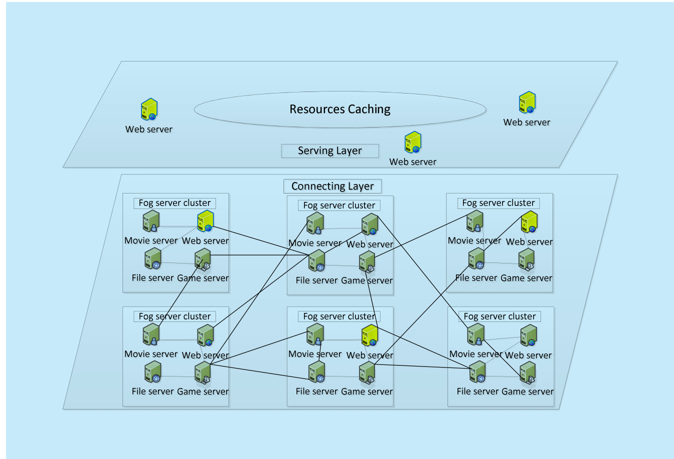


Fig. 2 *The Fog cluster resources caching model*

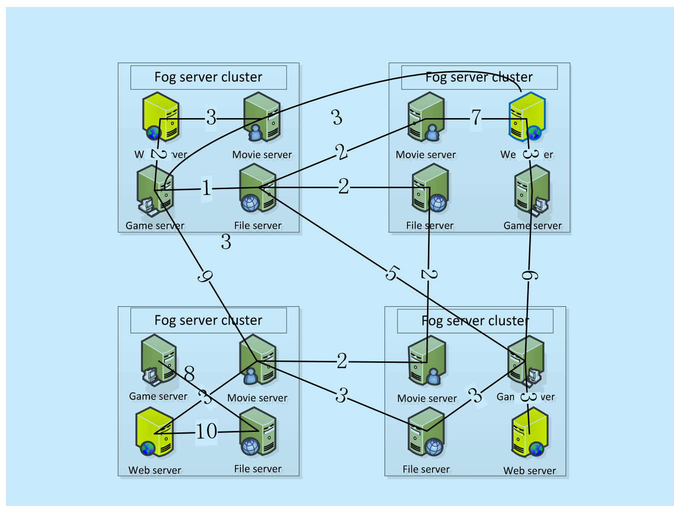


Fig. 3 *The connections and the costs of fog servers*

IV. Algorithm Design

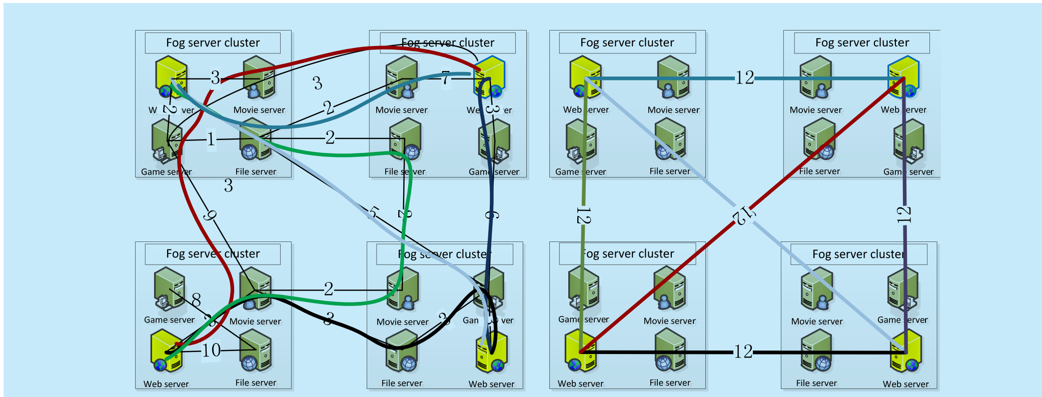
In order to solve the introduced problem, we turn to Steiner tree. The Steiner tree deals with the following problem. Given a set V of nodes which are interconnected by graph of shortest weighted length, where the length is the sum of the weighted lengths of all edges, finds a spanning tree using extra intermediate vertices and edges [12]. This is the same problem as we mentioned in Section 3. Lawrence T. Kou et. al. in [20] proposed a fast algorithm for Steiner trees which is suitable for our problem. In the following, we will first give some definitions on the Fog Computing and Steiner tree. And then introduce the fast Steiner tree algorithm.

Definition 1: In the Fog Computing topology, all the servers compose a node set N and all the edges compose a edge set E. The certain kinds of servers compose another node set P and $P \subset N$. The cost weight of the i th edge is e_i.

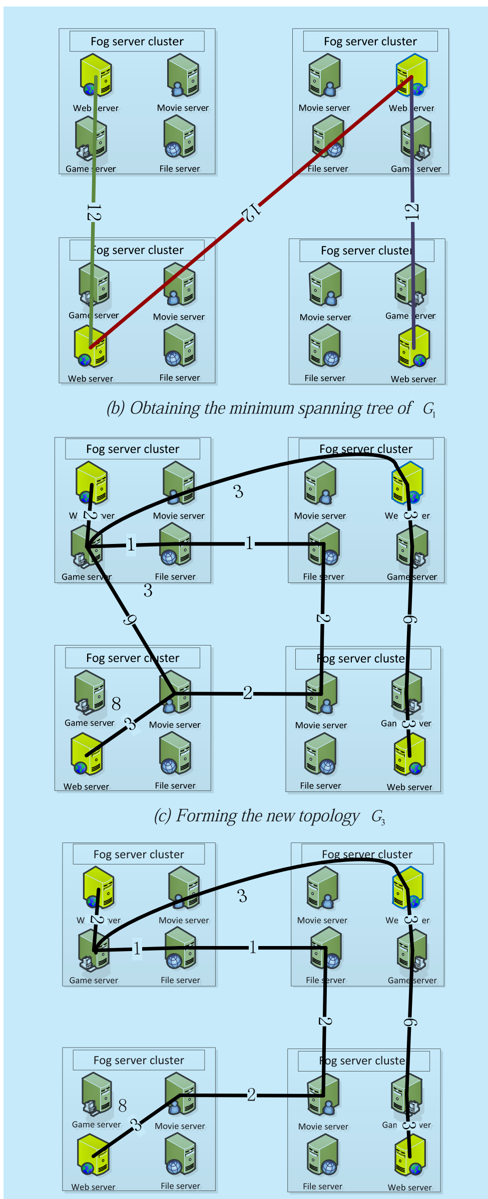
Definition 2: In the Steiner tree, the set of terminal vertices is also P. The set of Steiner Vertices is Q and $Q{=}N{-}P$. The set of initial edges is also E. The weight of each edge is the same as in Fog Computing topology.

We introduce the procedures of building a Steiner tree which was proposed by Lawrence T. Kou *et. al.* in [20]. The steps are as follows. We just use it solving our problem.

1) Calculate the shortest weight paths between each pair of nodes in P and



(a) Calculating the shortest weight paths between each pair of nodes



(b) Obtaining the minimum spanning tree of G_1

(c) Forming the new topology G_3

(d) Obtaining the built Steiner tree

Fig. 4 *The procedures of building a Steiner tree in fog computing*

form a new topology G_1 with node set P and new edge set E_1. The weight of i th edge $e_i^{'}$ is the sum of the short weight paths between nodes P_i and P_{i+1};

2) Obtain the minimum spanning tree of G_1 and form a new topology G_2. If there is more than one minimum spanning tree, just pick anyone of them;
3) Replace the edges in G_2 using the shortest weight paths obtained in 1) and form a new topology G_3. Also if there is more than one shortest weight paths, just pick one;
4) Obtain the minimum spanning tree of G_3 and form a new topology G_4;
5) Delete the nodes who is in the set of G_4 and have one connected neighbor;
6) Delete the edges that have one connected node.

V. Running Illustration

In this Section, we will give an example to show how this scheme is used to minimize the total cost of resources caching in Fog Computing, and then compare the proposed one with the traditional shortest path scheme. There are 16 Fog servers in the Fog cluster layer. The connections and the costs of each pair of servers are shown in Figure 3. The yellow Web servers want to cache resources among them. The problem can be written as finding a minimum spanning tree in the given condition.

Firstly, we calculate the shortest weight paths between each pair of yellow servers and get the results shown in Figure 4 (a). The shortest path between either two web servers is the same which is 12. In the next steps, we calculate the minimum spanning tree of the newly built topology (Shown in Figure 4 (b)). The four web servers are connected with each other with the minimum total path weight. Replace the edges in this graph using the corresponding shortest weight paths which is shown in Figure 4 (c). Then, we calculate the minimum spanning tree of the new graph and the result is shown in Figure 4 (d). In this graph, we can see there are no additional nodes that have one connected neighbour in the terminal set and edges that have one connected node. So this topology is the Steiner tree that we require.

VI. Numerical Simulation

In this Section, we give a numerical simulation to show the caching scheme performance. The topology used here is the same with that shown in Figure 3. We assume that there are 2, 3, 4 yellow Web servers wanting to share or cache resources among them respectively. Comparing with the traditional shortest path scheme, the Steiner tree based scheme can achieve better performance.

From Figure 5, we can see that when we use Steiner tree scheme to caches resources, the minimum path weight is 24 in the condition of there are 4 web servers needing data caching. In other words, the Fog Computing can complete the caching task with the cost of 24. If we use the shortest path scheme, the total cost will be 36. Comparing these two costs, we can see that the Steiner tree can greatly reduce the cost. Further, we can get that if there are 3 Fog servers who want to communicate, the costs are 12 and 24 respectively in the condition of Steiner tree scheme and shortest path scheme. Lastly, we can get that if there are only 2 Fog nodes who want to communicate, the shortest path and the Steiner tree scheme has the same performance, for the topologies that they build are the same. For more than three communicating Fog nodes, we believe the Steiner tree scheme can work better. The reason is that this scheme is design for minimizing the total edge weight of a connected graph.

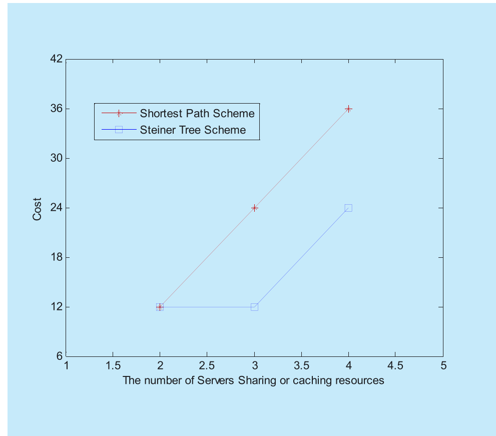


Fig. 5 *Costs comparing between steiner tree scheme and shortest path scheme*

VII. Conclusion

Fog Computing is a new research area which meets the requirements of wireless end users who has the characteristics of mobility, storage limited, bandwidth limited. In order to provide the local area services, the Fog Computing widely distributes Fog clusters geographical, and all the resources will be shared or cached in the Fog servers of the Fog Cluster. An important research point is that how we efficiently share or cache resources among certain Fog servers. In this paper, we turn to Steiner tree theory as our tools to analyse this problem. Steiner tree minimizing the total path weight in condition of a given node set. We introduced a Steiner tree building scheme which was proposed by Lawrence T. Kou et. al. Then, using a running illustration, we explained how this scheme is used in Fog Computing. Lastly, we compared the traditional shortest path scheme and Steiner tree scheme to show the efficiency of our proposed one.

ACKNOWLEDGEMENT

We gratefully acknowledge anonymous reviewers who read drafts and made many helpful suggestions. This work is supported by the National High-Tech R&D Program (863 Program) No. 2015AA01A705, the National Natural Science Foundation of China under Grant No. 61202079, the China Postdoctoral Science Foundation under Grant No. 2014T70031, and the Fundamental Research Funds for the Central Universities of China No. 2015JBM111.

Reference

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things" in Proceedings of ACM MCC, pp. 13–16, 2012

[2] Luis Miguel Vaquero Gonzalez, Luis Rodero-Merino: Finding your Way in the Fog: Towards a Comprehensive Definition of Fog Computing. Computer Communication Review (CCR) 44(5):27-32, 2014

[3] Tom H. Luan, Longxiang Gao, Zhi Li, Yang Xiang, Limin Sun: Fog Computing: Focusing on Mobile Users at the Edge.CoRR abs/1502.01815, 2015

[4] Jiang Zhu, Douglas S. Chan, Mythili Suryanarayana Prabhu, Preethi Natarajan, Hao Hu, Flavio Bonomi: Improving Web Sites Performance Using Edge Servers in Fog Computing Architecture. SOSE 2013:320-323, 2013

[5] John K. Zao, Tchin Tze Gan, Chun Kai You, Sergio Jose Rodriguez Mendez, Cheng En Chung, Yu Te Wang, Tim Mullen,Tzyy Ping Jung: Augmented Brain Computer Interaction Based on Fog Computing and Linked Data. Intelligent Environments 2014:374-377, 2014

[6] Clinton D'Souza, Gail-Joon Ahn, Marthony Taguinod: Policy-driven security management for fog computing: Preliminary framework and a case study. IRI 2014:16-23, 2014

[7] Mohammad Aazam, Eui-Nam Huh: Fog Computing and Smart Gateway Based Communication for Cloud of Things. FiCloud 2014:464-470, 2014

[8] Flavio Bonomi, Rodolfo A. Milito, Preethi Natarajan, Jiang Zhu: Fog Computing: A Platform for Internet of Things and Analytics. Big Data and Internet of Things 2014:169-186, 2014

[9] Ivan Stojmenovic, Sheng Wen: The Fog Computing Paradigm: Scenarios and Security Issues. FedCSIS 2014:1-8, 2014

[10] Marcelo Yannuzzi, Rodolfo A. Milito, René Serral-Gracià, D. Montero, Mario Nemirovsky: Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing. CAMAD 2014:325-329, 2014

[11] Tom H. Luan, Longxiang Gao, Zhi Li, Yang Xiang, Limin Sun: Fog Computing: Focusing on Mobile Users at the Edge.CoRR abs/1502.01815, 2015

[12] Steiner tree problem, http://en.wikipedia.org/wiki/Steiner_tree_problem

[13] Marek Karpinski, Alexander Zelikovsky: New Approximation Algorithms for the Steiner Tree Problems. J. Comb. Optim. (JCO) 1(1):47-65 , 1997

[14] Alexander Zelikovsky: A Series of Approximation Algorithms for the Acyclic Directed Steiner Tree Problem. Algorithmica 18(1):99-110, 1997

[15] Daniel Mölle, Stefan Richter, Peter Rossmanith: A Faster Algorithm for the Steiner Tree Problem. Symposium on Theoretical Aspects of Computer Science. Lecture Notes in Computer Science 2006:561-570, 2006

[16] A. Karim Abu-Affash, Paz Carmi, Matthew J. Katz: Bottleneck Steiner tree with bounded number of Steiner vertices. J. Discrete Algorithms (JDA) 30:96-100, 2015

[17] Karim Abu-Affash: The Euclidean Bottleneck Full Steiner Tree Problem. Algorithmica 71(1):139-151 , 2015

[18] Ahmad Biniaz, Anil Maheshwari, Michiel H. M. Smid: An optimal algorithm for the Euclidean bottleneck full Steiner tree problem. Comput. Geom. (COMGEO) 47(3):377-380 , 2014

[19] Erik D. Demaine, Mohammad Taghi Hajiaghayi, Philip N. Klein: Node-Weighted Steiner Tree

and Group Steiner Tree in Planar Graphs. ACM Transactions on Algorithms (TALG) 10(3):13 , 2014

[20] Lawrence T. Kou, George Markowsky, Leonard Berman: A Fast Algorithm for Steiner Trees. Acta Inf. (ACTA) 15:141-145, 1981

Biographies

SU Jingtao, is a PHD student of university of Science and Technology Beijing. His research interests include Fog Computing, Satellite Network.

LIN Fuhong, received his M.S. and Ph.D. degrees from Beijing Jiaotong University, China in 2006 and 2010, respectively, both in electronics engineering. He is currently a Lecturer in the Department of Computer and Communication Engineering, University of Science and Technology Beijing, China. His research interests include wisdom network, social network and P2P network. *The corresponding author, Email: FHLin_ustb@yeah.net.

ZHOU Xianwei, received his B.S. degree in Department of Mathematics from Southwest Normal University in 1986 and his M.S. degree from Zhengzhou University in 1992, and in 1999, he obtained Ph.D. degree in Department of Transportation Engineering from Southwest Jiaotong University, China. He was engaged in postdoctor study at Beijing Jiaotong University, China, from 1999 to 2000. Now, as a professor in Department of Communication Engineering, School of Computer and Communication Engineering, University of Science and Technology Beijing, his research interests include the security of communication networks, next generation networks, scheduling and game theory.

LÜ Xing, received his Ph. D degree from Beijing University of Posts and Telecommunications, Beijing, P. R. China, in 2012, in computer science and technology. Now he is a lecturer in school of science, Beijing Jiaotong University, P. R. China. His research interests include soliton theory, symbolic computation and optical soliton communication.