New Strategies and Extensions in Kruskal’s Algorithm in Multicast Routing

Mohamed Aissa, University of Nizwa, Oman
Adel Ben Mnaouer, Dar Al Uloom University, Saudi Arabia
Rian Murray, University of Trinidad and Tobago, Trinidad and Tobago
Abdelfettah Belghith, HANA Research Group University of Manouba, Tunisia

ABSTRACT

Multimedia applications are expected to guarantee end-to-end quality of service (QoS) and are characterized by stringent constraints on delay, delay-jitter, bandwidth, cost, and so forth. The authors observe that Kruskal’s algorithm is limited to minimal (maximal) spanning unconstrained tree. As such, the authors extend Kruskal’s algorithm to incorporate the delay bound constraint. Consequently, a novel algorithm is proposed, called EKRUSt (Extended Kruskal), for constructing multicast trees. The EKRUSt’s distinguishing features consists of a better management of Kruskal’s priority queues, and in the provision of edge priority aggregation. Preliminary results show that the proposed EKRUS algorithm performs as well as the best-known algorithms (such as the DDMC, DMCTC algorithms) while exhibiting reduced complexity. The authors conducted an extensive analysis and evaluations of different strategies of assigning edges into the classes of the queue as well as edge selection. As a result, the EKRUS algorithm was further extended with different edge assignment and selection strategies. Through extensive simulations, the authors have evaluated various versions of the EKRUS and analyzed their performance under different load conditions.

Keywords: Delay-Constrained Routing, End-to-End Quality of Service, Kruskal’s Algorithm, Multicast Routing Algorithms, Multimedia Applications

INTRODUCTION

Minimum spanning tree algorithms find applications in diverse areas such as: least electrical wiring, minimum cost connecting communication and transportation network, network reliability problems, minimum stress networks, clustering and numerical taxonomy, etc.

The problem of finding a spanning subtree of a given connected network which has minimum total length was first solved by Kruskal (1956). Shortly thereafter, Prim (1959) and Dijkstra (1959) suggested another algorithm
which appeared to be more efficient. So far, all the problems relating to spanning trees are solved using either Prim or Dijkstra’s algorithm.

However, recent works suggest that a suitable implementation of Kruskal’s algorithm is computationally more efficient in a number of interesting cases, in particular when the network under consideration is sparse.

Navneet and Jianer (2002) showed that maximum bandwidth paths can be constructed by a modified Kruskal’s algorithm. They demonstrate that their approach is simpler, easier in implementation, more flexible, and faster than using a modified Dijkstra’s algorithm.

For complex communication network architecture, communication network architecture has a heavy workload in the process. If we constructed it in the traditional way, it will result in a great waste of resource. Consequently, we need to ensure good efficiency and increase economic efficiency. Only by using rational design, we can get the best results, otherwise, the workload will be large, and the efficiency will be low (Chenghui & ChuanJun, 2010).

Kershenbaum and Van (1972) observed that for sparse networks a dramatic reduction in execution time can be obtained by the use of Kruskal’s algorithm.

The motivation for the present work is three-fold. First, we have identified a weakness in the classical Kruskal’s algorithm. It consists in the fact that the priority queue contains all the edges in only one class. In this class, all the edges in the priority queue are sorted either in increasing order to form a maximal spanning tree or in decreasing order to find the minimal spanning tree. Second, in most cases found in the literature, the extraction of edges from the priority queue, in the classical Kruskal’s algorithm, is done according to one single criterion, which is the priority of that edge in the queue. Third, in most cases, Kruskal’s algorithm is used without considering constraints.

The contributions of our work are as follows. At first, we propose to organize the priority queue of the original Kruskal’s algorithm into multiple classes. These classes are formed by the edge containing the source node, the edges containing destination nodes and the edges containing relay nodes respectively. Then, we introduce a new strategy in edge selection, giving priority to edges containing one or two destination nodes to be selected. Finally, based on these two strategies, we address the problem of constructing the delay constrained multicast tree using a fast and simple heuristic algorithm named the Extended Kruskal’s (EKRUS) algorithm. Furthermore, the assignment strategy of edges into the classes of the queue, as well as the edge selection process are investigated. As a result, of this investigation, the EKRUS algorithm was divided into different variants depending on different edge assignment and selection strategies. These strategies are subjected to extensive simulation in order to assess their performances and their suitability under different load conditions.

In the remainder of this paper, the next section provides an overview of some related work. The network model is then presented. The formal definition of the EKRUS algorithm and its analysis are given. A comparative study of the various variants of the EKRUS algorithm against some well-known algorithms from the literature is also provided. We prove the correctness and time complexity analysis of the EKRUS algorithm and discuss simulation results. Concluding remarks are provided in the last section.

RELATED WORK

Prim’s algorithm is easy to implement and is very fast for large nearly complete graphs (Kershenbaum & Van, 1972). Unfortunately, for sparse graphs, Prim’s algorithm becomes considerably more complex and some speed is lost in analyzing such graphs (Kershenbaum & Van, 1972). Dijkstra algorithm assures a rapid and correct dissemination of topological information (McDonald, 1997). Unfortunately, the algorithm suffers from scalability problem (Alpert et al., 1993), and the written programs are long and complicated (Navneet & Chen, 2002). For constructing a maximum bandwidth
Co-Modeling of Embedded Networks Using SystemC and SDL
[www.igi-global.com/article/modeling-embedded-networks-using-systemc/51648?camid=4v1a](www.igi-global.com/article/modeling-embedded-networks-using-systemc/51648?camid=4v1a)

Telecommunication Customer Demand Management
[www.igi-global.com/chapter/telecommunication-customer-demand-management/21676?camid=4v1a](www.igi-global.com/chapter/telecommunication-customer-demand-management/21676?camid=4v1a)