Performance Analysis of Multicast Heuristic Algorithms

Maciej Piechowiak, Piotr Zwierzykowski, Sławomir Hanczewski
Institute of Electronics and Telecommunication
Poznań University of Technology
ul. Piotrowo 3A, 60-965 Poznań
{mpiech,pzwierz}@et.put.poznan.pl

Abstract

The article investigates representative heuristic algorithms finding the cheapest spanning trees between a source node and a group of destination nodes (multicast connections). An algorithm that solves the Steiner problem, i.e., minimal spanning tree is also presented. The presented solutions are grouped into: i. unconstrained algorithms and ii. constrained algorithms. The key part of the article includes the efficiency analysis and a comparison of the solutions against their time complexity and the influence of the parameters of a given structure (network model generated by BRITE tool) upon the efficiency of the algorithms under scrutiny.

Keywords: *multicast routing*, *heuristic algorithms*, *network topology*, *graphs*.

1 INTRODUCTION

The increase in bandwidth of modern networks has made multimedia data transmission in real time possible (radio and television transmission, video on demand, teleconferences, etc.) [1]. To ensure accurate transmission of data of this type, a particularly defined bandwidth is needed and, primarily, a situation in which delay between the source and destination nodes can be maintained at the steady, unchanged level. The occurring jitter phenomenon is an undesirable phenomenon here.

The way transmission of data of this kind is carried out is somewhat similar to broadcasting, in practice, however, there is a given group of nodes receiving the same data that run parallel in the same time. Multicasting requires efficient routing algorithms defining a tree with minimal cost between the source node and particular nodes representing the users. Such a solution prevents duplication of the same data (packets) in the links of the network. Routing of the sent data occurs only in those nodes of the network which lead directly to destination nodes.

If the communication network is presented as a graph, the result of the implementation of the routing algorithm will be a spanning tree rooted in the source node and including all destination nodes in the multicast group. Two kinds of trees can be distinguished in the process of optimization: MST - *Minimum Steiner Tree*, and the tree with the shortest paths between the source node and each of the destination nodes - SPT (*Shortest Path Tree*). Finding the MST, which is a $\mathcal{NP}$-complete problem, effects in a structure with a minimal total cost. The relevant literature provides a wide range of heuristics solving the above problem in polynomial time [2–4]. From the point of view of the application in data transmission, the most commonly used is KMB algorithm [2]. The other method minimalizes the cost of each of the paths between the sender and each of the members of the multicast group by forming a tree from the paths having the least costs. In general, it is first either Dijkstra algorithm [5] or Bellman-Ford algorithm [6] that is used, and the branches of the tree that do not have destination nodes are then cut off.

To ensure reliable transmission, multimedia applications set up high standards for quality parameters (*Quality-of-Service* parameters). Quality requirements concerning, *inter alia*, steady and guaranteed delay values in relation to a transfer of a packet between the source and destination nodes along a defined path in the network still pose a great challenge for those designing the application in real time. Hence, the process of optimization includes the second metric of the network - delay (d). With constructions of multicast trees, maximum delay between two end points in the network (Δ) is then the applicable appropriate criterion. In [2, 7], it has been proved that finding the tree poses an $\mathcal{NP}$-complete problem for one or more QoS parameters. Due to the complexity of the problem, the presented algorithms use the techniques approaching the solution - heuristics.

The article discusses the effectiveness of most commonly used heuristic algorithms as well as their comparative usefulness. Chapter 2 presents unconstrained algorithms without considering delays in the network, i.e., KMB and SPT algorithm. Chapter 3 presents algorithms with constraints, i.e., KPP (*Kompella, Pasqualle, Polyzos*) [8] and CSPT (*Constrained Shortest Path Tree*) [3].

Chapter 4 presents the methodology of generating structures representing the topology under scrutiny, and Chapter 5 includes the results of the simulation of the implemented algorithms along with their interpretation.

2 UNCONSTRAINED ALGORITHMS

2.1 Minimum Steiner Tree (MST)

Let us assume that a network is represented by a directed, connected graph $N = (V, E)$, where V is a set of nodes, and E is a set of links. The existence of the link $e = (u, v)$ between the node u and v entails the existence of the link $e' = (v, u)$ for any $u, v \in V$ (corresponding to two-way links in communication networks). With each link $e \in E$, two parameters are coupled: cost $C(e)$ and delay $D(e)$. The cost of a connection represents the usage of the link resources. $C(e)$ is then a function of the traffic volume in a given link and the capacity of the buffer needed for the traffic. A delay in the link is in turn the sum of the delays introduced by the propagation in a link, queuing and switching in the nodes of the network. The multicast group is a set of nodes that are receivers of group traffic (identification is carried out according to a unique i address), $G = \{g_1, ..., g_n\} \subseteq V$, where $n = |G| \leq |V|$. The node $s \in V$ is the source for the multicast group G. Multicast tree $T(s, G) \subseteq E$ is a tree rooted in the source node s that includes all members of the group G. The total cost of the tree $T(s, G)$ can be defined as $\sum_{t \in T(s,G)} C(t)$. The path $P(s, G) \subseteq T(s, G)$ is a set of links between s and $g \in G$. The cost of path $P(s, G)$ can be expressed as: $\sum_{p \in P(s,G)} C(p)$, while the delay measured between the beginning and the end of the path as: $\sum_{p \in P(s,G)} D(p)$. Thus the maximum delay in the tree can be determined as: $\max_{g \in G}[\sum_{p \in P(s,G)} D(p)]$.

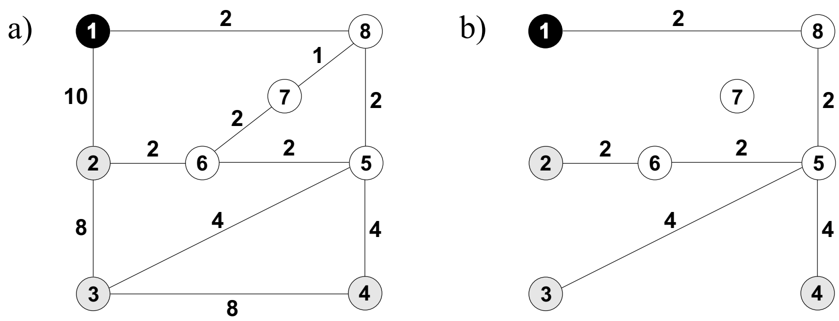

Fig. 1. Minimum Steiner tree (MST), a) original graph, b) final MST (cost = 16)

Steiner tree is a good representation for solving the routing multicast problem. This approach becomes particularly important when we have to deal with only one active multicast group and the cost of the whole group has to be minimal. However, due to the computational complexity of this algorithm ($\mathcal{NP}$-complete problem) [7], heuristic algorithms are most preferable. If the set of the nodes of the minimal Steiner tree includes all nodes of a given network, then the problem comes down to finding the minimal spanning tree (this solution can be obtained in polynomial time).

2.2 KMB algorithm

Kou, Markovsky and Bermann proposed such KMB minimum Steiner tree heuristic [2]:

- For an undirected distance graph $N = (V, E)$ and a set of destination nodes G (Fig. 2a) construct complete undirected graph $N_1 = (V_1, E_1)$, constructed from source node s and set of destination nodes G (paths in N_1 are shortest paths in N) (Fig. 2b),

- Find the minimum spanning tree T_1 for graph G_1 (if there are several minimum spanning trees, pick an arbitrary one) (Fig. 2c),
- Construct the subgraph G_S of G by replacing of each edge in T_1 corresponding shortest path in G (Fig. 2d) (if there are several shortest path, pick an arbitrary one),
- Find the minimal spanning tree T_S of G_S (Fig. 2e) if there are several minimum spanning trees, pick an arbitrary one),
- Construct a Steiner tree T_{KMB} from T_S by deleting edges in T_S, if necessary, so that all the leaves in T_{KMB} are Steiner points (Fig. 2f).

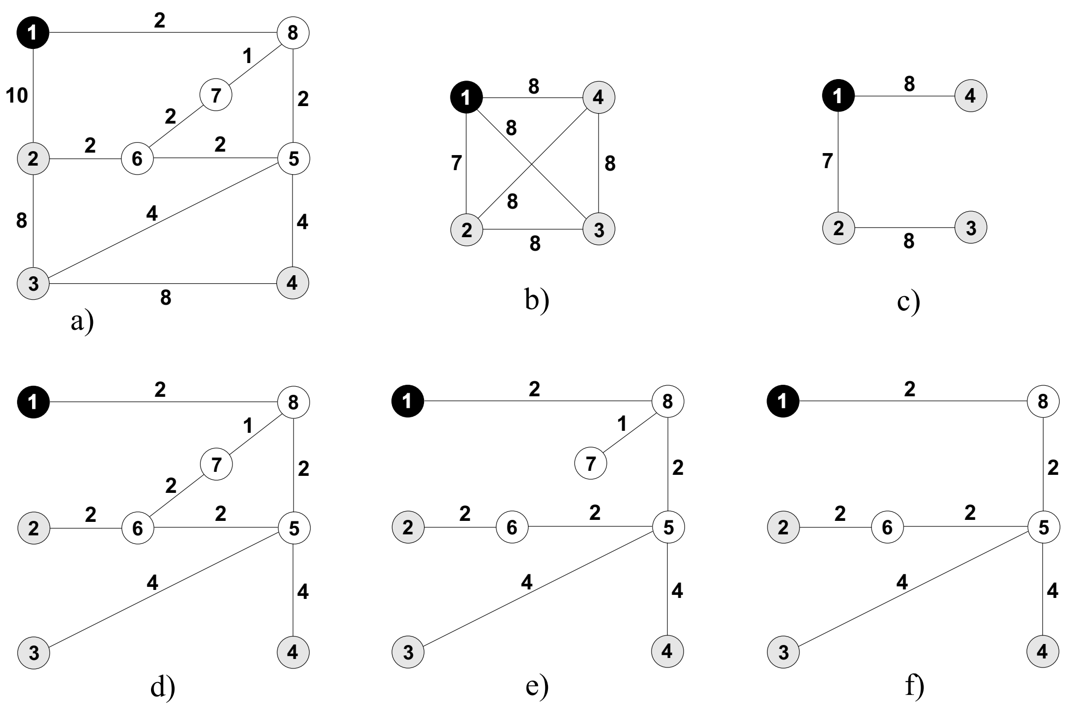

Fig. 2. Example of the KMB heuristic

The Fig. 2 shows an example of KMB heuristic. On Fig. 2 total minimum spanning tree cost is 16 and it is the minimum Steiner tree.

Doar and Leslie [9] shows, that total cost of MST find by KMB heuristic (Kou, Markowsky, Bermann) is average 5% worst in comparision with exact MST method but the time complexity of the KMB is $O(|G||V|^2)$ only.

3 CONSTRAINED ALGORITHMS

3.1 KPP algorithm

While constructing a minimal tree, KPP heuristics uses an additional parameter (delay) for each path in the network by defining the constrained shortest path as a path with minimal cost with the assumption that the value of the delay along the path does not exceed a set value Δ. This method assumes that delays in the links and maximum *end to end* delay Δ are integer values. Proposed algorithm consists of the following steps (similarly to KMB):

- For an undirected distance graph $N = (V, E)$ and a set of destination nodes G (Fig. 3a) construct complete undirected graph $N_1 = (V_1, E_1)$, constructed from source node s and set of destination nodes G only (paths in N_1 are shortest path in N) (Fig. 3b),

- Find minimum spanning tree T_1 of G_1 for each (u,v) set and cost $C(u,v)$, and delay $D(u,v)$ according to cost function f_c (Fig. 3c):

$$f_c = \begin{cases} C(u,v) & \text{if } D_s(u) + D(u,v) < \Delta \\ \infty & \text{otherwise} \end{cases}, \tag{1}$$

- Replace edges of the found tree by paths from the original graph G (if there are some loops remove them by Dijkstra algorithm - for delay constrain) (Fig. 3d).

For the example on Fig. 3 is is assumed $\Delta = 10$. In contrast with KMB, in KPP heuristic if the path 2-6-5-3 do not violate delay constrain (delay is equal to 11), then KPP finds other tree with total cost equal to 21.

The time complexity of the KPP is $O(\Delta|V|^3)$.

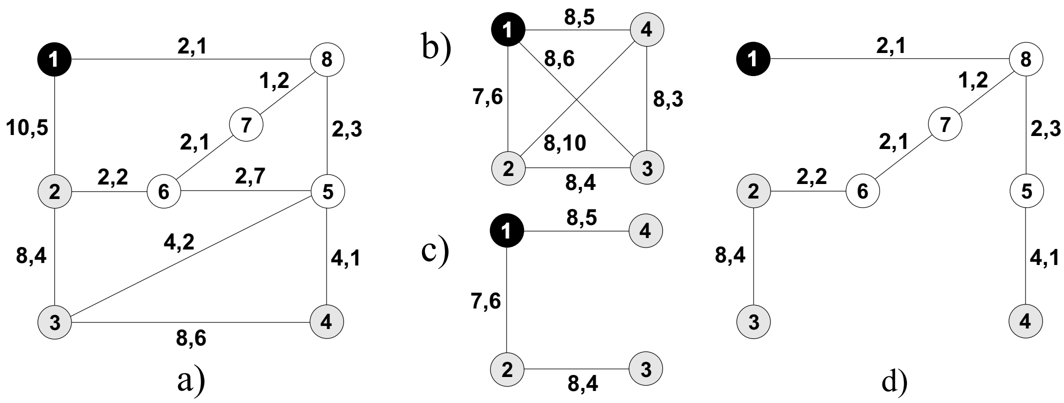

Fig. 3. Example of the KPP heuristic (for $\Delta = 10$)

3.2 CSPT algorithm

The *Constrained Shortest Path Tree* algorithm is an example of minimum cost path tree heuristic [3]. If delay on path to any destination exceed the delay bound for this path, then the path will be replaced by a minimum cost path. Thus, if the minimum cost tree exceed delay bound, we find a minimum delay tree and combine both trees. This algorithm has three steps:

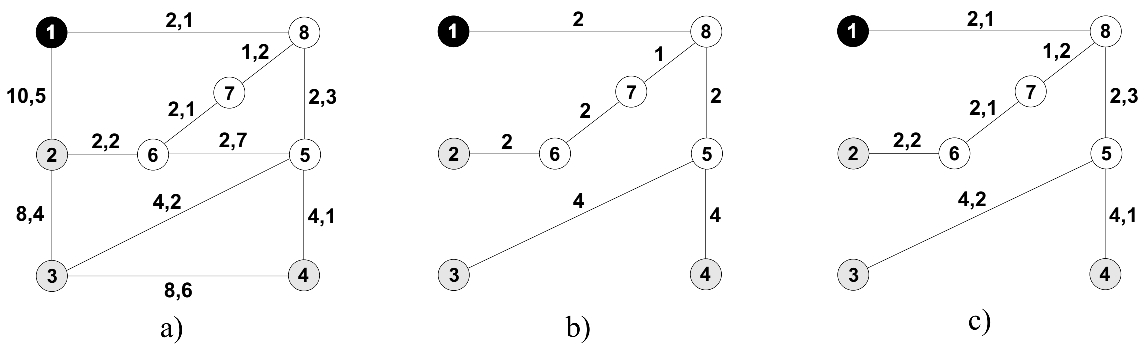

Fig. 4. Example of the CSPT heuristic (for $\Delta = 10$)

- Use Dijkstra's shortest path algorithm [5] to compute a lowest cost spanning tree to as many destination nodes in the multicast as is possible without any path breaking the arbitrary delay bound Δ (Fig. 4b).
- Use Dijkstra's algorithm to compute a shortest delay path tree to those multicast nodes not reached in the previous step.

- Combine the lowest cost spanning tree from the first step with the shortest delay path tree from the second step making sure that the delay to any destination node does not break the delay bound Δ (Fig. 4c) and that all loops are removed.

The time complexity of the CSPT is $O(|V|^2)$ [5].

4 PROBLEM FORMULATION

The main purpose of this article is to analyze the efficiency of the presented algorithms in identical network conditions - size of the network, adopted topology model, values of metrics for each of the edges, etc. To achieve that, the algorithms discussed in Chapter 2 and 3 were implemented in the C++ language. In the research, a flat random graph constructed using Waxman method was used [10]. This method defines the probability of an edge between node u and v as:

$$P(u,v) = \alpha e^{\frac{-d}{\beta L}} \tag{2}$$

where $0 < \alpha, \beta \leq 1$, d is euclidian distance between node u and v, and $L = \sqrt{2}$ is the maximal distance between two freely selected nodes. An increase in parameter α effects in the increase in the number of edges in the graph, while a decrease of the parameter β increases the ratio of the long edges against the short ones.

With the construction of the network models based on Waxman method, BRITE (*Boston university Representative Internet Topology gEnerator*) [11] was used as a tool for generation of realistic topologies. The application provides a range of network topology models and appropriate generative methods. The generative process includes the following steps:

- arrangement of network nodes in a plane,
- connection of all network nodes to form a connected structure (for instance, with the help of Waxman method),
- assigning attributes to links between the nodes (cost and delay) and to the nodes,
- export of the structure to a file of appropriate format.

A network model was adopted in which the nodes were arranged on a square grid with the size of 1000×1000 (Waxman parameters: $\alpha = 0,15$, $\beta = 0,2$). Onto the existing network of connections, the cost matrix $c(u,v)$ was applied (from the range of 500 to 1000) and the delay $d(u,v)$ resulting from euclidean distance between the nodes.

It was an important element during the simulation process to maintain a steady average node degree of the graph (for each of the generated networks) defined as: $D_{av} = \frac{2k}{n}$ (where n is the number of the nodes of the network, k is the number of edges) which, in practice, meant the necessity of maintaining a steady number of edges. It is generally accepted that for $D_{av} \geq 2$, the so-called *two-connected* network, that is a connected network, can be constructed. Toronha and Tobagi [12] proved that the efficiency of the routing algorithm implemented in a real network was identical to the efficiency of the same algorithm in a random *two-connected* network. In the implementations, the adopted degree of the graph was within the 3 to 5 bracket.

5 SIMULATION RESULTS

The research work done can be divided into the research on the total cost of the tree and the research on the time needed for constructing of an algorithm (in relation to the number of nodes). The results presented in Figs 5 and 6 show a comparison of MST, SPT and KMB algorithms in relation to a number of nodes in the network and the number of the members of the group (destination nodes). The experimental computations were carried out for a relatively small number of the members of the group and the nodes of the network due to the computational complexity of the MST algorithm. For

$n, m \leq 10$ the obtained structures were generated in acceptable time. This, however, is sufficient to asses the costs for the trees generated by the remaining heuristics, i.e., SPT and KMB, because the percentage growth of the total cost of the tree for those algorithms remains stable in relation to the MST algorithm (for the greater number of nodes).

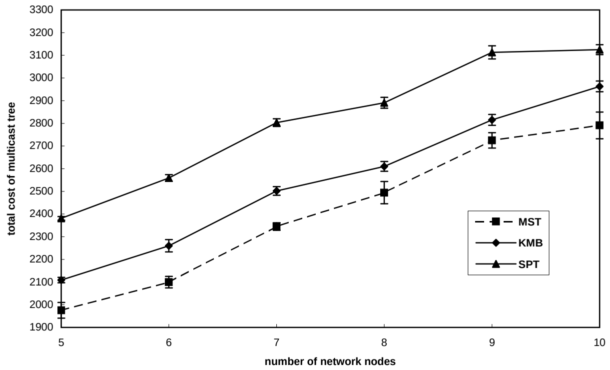

Fig. 5. Total cost of multicast tree versus number of network nodes (n) for unconstrained algorithms ($m = 3$, D_{av}=3)

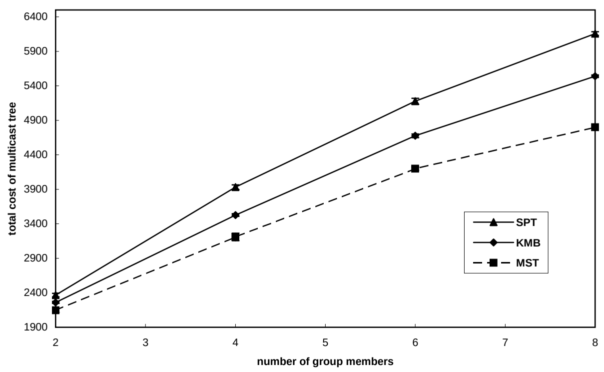

Fig. 6. Total cost of multicast tree versus number of group members (m) for unconstrained algorithms ($n = 10$, D_{av}=3)

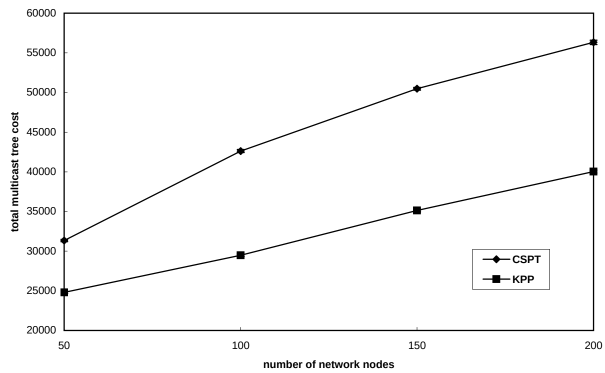

Fig. 7. Total cost of multicast tree versus number of network nodes (n) for constrained algorithms ($m = 40$, D_{av}=5, $\Delta = 10ms$)

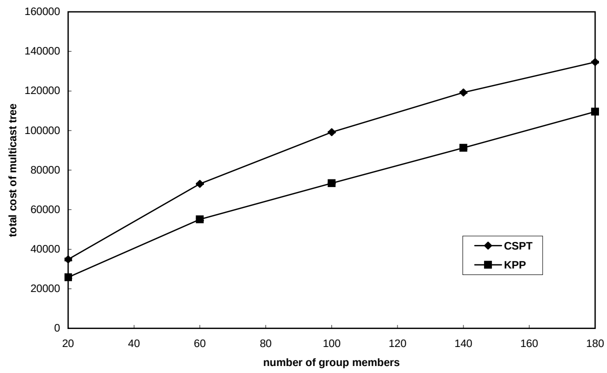

Fig. 8. Total cost of multicast tree versus number of group members (m) for constrained algorithms ($n = 200$, D_{av}=5, $\Delta = 10ms$)

The other assignment was to compare the time needed for algorithms to be carried out. The execution time was directly connected with the number of permutations in which the shortest path between the transmitting node and a member of the group was determined (Dijkstra algorithm). In SPT algorithm the number is m, while in KMB algorithm - $\frac{m(m+1)}{2}$ (Fig. 9). The time calculations were done on a PC computer with a Pentium III 550 MHz processor. Doar and Leslie [9] proved that the total cost of the trees generated with the help of KMB algorithm was, on average, only 5% worse than that obtained by the exact method (MST). KMB method is then a good approximation of the MST algorithm and can be used with the comparative analysis of other heuristics for a large number of nodes (even for $n = 1000$). Due to the above, Figs 5 and 6 show a comparison of KMB and SPT algorithms.

The next stage of the experiment included the examination of the constrained algorithms. A new parameter was introduced for this particular class, namely, maximum delay between the source node and each of the destination nodes (Δ). The introduction of this parameter entails that the cost of generated trees increases along with the increase in the number of network nodes (Figs 7 and 8).

The research work also proved that for 20% of cases CSPT algorithm failed to construct a multicast tree, while the results were always obtainable for KPP. This is directly connected with the number of paths between the source node and the destination nodes. The existence of connections *each with each* in KPP algorithm ($\frac{m(m+1)}{2}$) increases the probability of creating a spanning tree (in the third step of the algorithm - cf. Chapter 3), which will include all nodes of the group. The low complexity of the calculations translating into the time needed to construct CSPT algorithm (Fig. 10) is paid,

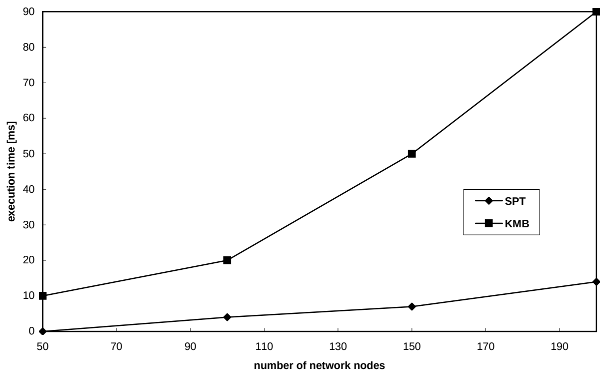

Fig. 9. Execution time versus number of network nodes (n) for unconstrained algorithms ($m=20$, D_{av}=5)

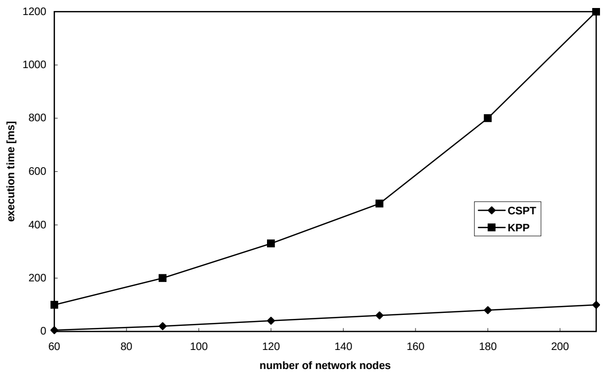

Fig. 10. Execution time versus number of network nodes (n) for constrained algorithms ($m=40$, D_{av}=5, $\Delta=10ms$)

however, with a relatively high cost of generated solutions (twice as much as compared to KPP) and low efficiency (not connected resulting structures).

Authors made a lot of experiments for different topologies to develop research methodology. Finally, they generated 1000 structures in 5 series for set parameters of the network structures (average node degree D_{av}, number of network nodes n, number of multicast nodes m and maximum *end to end* delay Δ). To evaluate the obtained simulation results (total cost of multicast tree) we used *t-Student* distribution with $n-1$ degrees of freedom:

$$\overline{c(i)} - t_{\frac{\alpha}{2}} \frac{s_n}{\sqrt{n}} \leq c_j \leq \overline{c(i)} + t_{\frac{\alpha}{2}} \frac{s_n}{\sqrt{n}} \tag{3}$$

where $\overline{c(i)}$ is the sample mean, s_n is the sample standard deviation, and $t_{\frac{\alpha}{2}}$ - critical t-value for $n-1$ degrees of freedom and $\frac{\alpha}{2}=0,025$ (for 5 series $t_{\frac{\alpha}{2}}=2,132$).

The results of simulations are shown in the charts (Fig. 5 - 8) in the form of marks with 95% confidence intervals that were calculated after the *t-Student* distribution. 95% confidence intervals of the simulation are almost included within the marks plotted in the figures (Fig. 6, 7 and 8).

The use of such simulation methodology, in the authors opinion, is important and indispencable. Many articles often lack the inclusion of this particular aspect.

6 CONCLUSIONS

Broadband packet networks transmitting data of applications in real time have to meet appropriate quality requirements for those applications. The bandwidth occupied by multimedia data streams is broad enough for the use of routing connections in the transmission to be necessary. Efficient routing protocols for multicast connections can be a sufficient and effective auxiliary tool in maintaining network resources.

The article presents and compares representative routing algorithms for multicast connections emphasizing the quality of the network model (accuracy of the illustration of a real internet topology). To this effect, topology generator BRITE, which is considered to be state-of-the-art and very good tool preferred in any research on optimization techniques of networks was used. Implementation of algorithms constructing multicast trees close to the MST method (KMB, KPP) seems to be advisable with networks of a small number of nodes (for instance LAN networks). On the other hand, algorithms determining the shortest paths (SPT, CSPT) are more appropriate for large-scale networks in which bit flow of multimedia streams can be low. Further researches should focus on implementations of fast and simple routing algorithms supporting scalability and work in distributed mode.

References

[1] Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," *IEEE Journal on Selected Area in Communications*, vol. 14, no. 7, pp. 1228–1234, 1996.

[2] L. Kou, G. Markowsky, and L. Berman, “A fast algorithm for Steiner trees,” *Acta Informatica*, no. 15, pp. 141–145, 1981.

[3] J. S. Crawford and A. G. Waters, “Heuristics for ATM Multicast Routing,” *Computer Science, University of Kent at Canterbury*, 1998.

[4] Q. Sun and H. Langendoerfer, “Efficient Multicast Routing for Delay-Sensitive Applications,” in *Proceedings of the 2-nd Workshop on Protocols for Multimedia Systems (PROMS’95)*, October 1995, pp. 452–458.

[5] E. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[6] R. Bellman, “On a routing problem,” *Quarterly of Applied Mathematics*, vol. 16, no. 1, pp. 87–90, 1958.

[7] R. Karp, “Reducibility among combinatorial problems,” *Complexity of Computer Computations*, pp. 85–104, 1972.

[8] V. P. Kompella, J. Pasquale, and G. C. Polyzos, “Multicasting for Multimedia Applications,” in *INFOCOM*, 1992, pp. 2078–2085.

[9] M. Doar and I. Leslie, “How bad is naive multicast routing?” *IEEE INFOCOM’ 1993*, pp. 82–89, 1993.

[10] B. Waxmann, “Routing of multipoint connections,” *IEEE Journal on Selected Area in Communications*, vol. 6, pp. 1617–1622, 1988.

[11] A. Medina, A. Lakhina, I. Matta, and J.Byers, “BRITE: An Approach to Universal Topology Generation,” *IEEE/ACM MASCOTS*, pp. 346–356, 2001.

[12] C. A. Toronha and F. A. Tobagi, “Evaluation of multicast routing for multimedia streams,” in *Proceedings of IEEE International Telecommunications Symposium*, 1994.