The Application of $K$–Shortest Path Algorithm in Multicast Routing

MACIEJ STASIAK, MACIEJ PIECHOWIAK, PIOTR ZWIERZYKOWSKI

"Chair of Communications and Computer Networks,
Poznań University of Technology, ul. Polanka 3, Poznań 60965, Poland,
e-mail: {stasiak,mpiech,pzwierz}@et.put.poznan.pl

Received date 15 January 2009, Revised date 17 April 2009, Accepted date 1 July 2009

Abstract: Abstract. There have been many heuristic algorithms finding multicast trees proposed over the past few years. The necessity for finding a better and more suitable algorithm is still a current and important challenge. The article presents a new multicast routing algorithm for finding a multicast tree in packet networks. The objective of the $K$-Shortest Path Multicast Algorithm (KSPMA) is to minimize the cost of paths between the source and each destination node using a generalized Dijkstra’s algorithm that would comply with maximum delay bound along each path. A wide range of simulation research carried out by the authors for many network topologies and parameters has confirmed the effectiveness of the proposed algorithm.

Keywords: multicast, routing algorithms, network topology

1. Introduction

Multicasting is a way of transmission between a source node and specified destination nodes, i.e. the members of a multicast group. It requires efficient routing algorithms defining a tree with a minimum cost between the source node and the particular nodes representing the users. Such a solution prevents duplication of the same data (packets) in the links of the network.

Over the last few years, multicast algorithms have become more important due to the specificity of data transmitted in transport networks. The increase in traffic capacity of present-day networks has offered great advantages in distributed applications such as multimedia data transmission in real time, video-on-demand, teleconferencing, etc. [1]. To ensure reliable transmission, multimedia applications demand high requirements for the processing of audio and video data to be met by quality parameters (quality-of-service). Quality requirements concerning, for example, fixed guaranteed delay value
accompanying a hop of a packet between the transmitting and the receiving node down a specified route of the network still pose a challenge for designers of real time applications.

The implementation of multicasting requires solutions of many combinatorial problems accompanying the construction of optimal transmission trees. In the optimization process the following trees can be distinguished: 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 MST, which is a $\mathcal{NP}$-complete problem, results in a structure with a minimum total cost [2]. The relevant literature provides a wide range of heuristics solving this problem in polynomial time [3, 4, 5]. As far as the application in data transmission is concerned, the most commonly used algorithm is the KMB algorithm [3]. Other methods minimize 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, at first either the Dijkstra algorithm [6] or the Bellman-Ford algorithm [7] are used, then the branches of the tree that do not have destination nodes are pruned. Several routing algorithms have been proposed in the literature for this problem [8, 9, 10, 11, 12, 13].

Within the context of business relations between the user and the Internet Service Provider, or many ISPs, in real networks, a service level agreement (SLA) is negotiated to ensure the quality of transmission at the required level. From the point of view of the multicast routing algorithms, to ensure accurate transmission of real time data, a particularly defined delay along transmission path is needed. Hence, the process of optimization includes the second metrics of the network – delay ($\Delta$). With the constructions of multicast trees, the maximum delay between two end points in the network ($\Delta$) is, thus, the applicable appropriate parameter. In [3, 14], it has been proved that finding the tree poses an $\mathcal{NP}$-complete problem for one or more Quality-of-Service parameters. Due to the complexity of the problem, the presented algorithms use the appropriate techniques for approaching the solution – heuristics.

The article proposes and discusses the effectiveness of the $K$-Shortest Path Multicast Algorithm (KSPMA) and its comparison with most commonly used constrained heuristic algorithms: KPP (Kompella et all [15]), CSPT (Crawford et all [8]) and the least-delay multicast tree [16].

The article is divided into six sections. Section 2 describes the problem and the network model. Section 3 presents an overview of the KSPMA algorithm. In Section 4, the simulation methodology is described. Section 5 includes the results of the simulation of the implemented algorithms (KSPMA and other algorithms), while Section 6 sums up the presented study.
CSPT: Constrained Shortest Path Tree

designations
\( e_{ij} \in E \) is link between nodes \( i \) and \( j \)
\( s \in V \) source node for the multicast group \( M \)
\( M \subseteq V \) the multicast group: a set of nodes that are receivers of the group traffic
\( m_i \in M \) receiving (multicast) node \( m_i \)
\( T(s, M) \subseteq E \) the Steiner tree rooted in source \( s \) with leaves \( M \)
\( p(s, m_i) \subseteq T(s, M) \) path between source \( s \) and a multicast node \( m_i \)

constants
\( \Delta \) the maximum delay on the path \( p \)
\( e_{ij} \in \{0, 1\} \) a link between nodes \( i \) and \( j \)

variables
\( c_{ij} \in (10; 1000) \) the cost of a connection between nodes \( i \) and \( j \)
\( d_{ij} = |i, j| \) delay in the link (Euclidean metric) between nodes \( i \) and \( j \)
\( c_{ij}, d_{ij} \in \mathbb{R}^+ \)

objective
minimize \( c_p = \sum_{(i,j) \in p(s, m_i)} c_{ij} e_{ij} \)

subject to
\( \sum_{(i,j) \in p(s, m_i)} d_{ij} e_{ij} \leq \Delta \)

Fig. 1. Formal notation of the CSPT problem

2. Problem formulation

The constrained shortest path tree (CSPT) contains constrained shortest paths between the source and each destination node. The problem of finding CSPT is formally presented in Fig. 1. In the article we have assumed that the network is represented by an undirected connected graph \( G = (V, E) \), where \( V \) is a set of nodes, and \( E \) is a set of links. In the article we have also made the following assumptions:

- the total cost of the Steiner tree \( T(s, M) \) can be defined as \( \sum_{t \in T(s, M)} c(t) \),
- the cost of path \( p(s, m_i) \) can be expressed as: \( \sum_{p \in P(s, m_i)} c(p) \), where \( P(s, m_i) \) is a set of possible paths between \( s \) and \( m_i \in M \),
- the delay measured between the beginning and the end of the path can be expressed as: \( \sum_{p \in P(s, m_i)} d(p) \).  

The CSPT problem formally presented in Fig. 1 is \( \mathcal{NP} \)-hard [3, 14]. In the article the authors propose a new heuristic algorithm solving the CSPT problem in polynomial time.
3. KSPMA Algorithm Overview

The main goal of the algorithm proposed by the authors is to build a constrained shortest path tree calculating paths between the source and each destination node using the $K$–shortest path algorithm.

Our proposal relies on the implementation of the $K$–shortest path algorithm to construct a constrained shortest path between the source node $s$ and each destination node $m_i$. The $K$–shortest path algorithm \cite{1,2} is a classical network programming problem that has been studied particularly in practical applications \cite{19}. In the $K$–shortest paths problem, for a given network $N = (V, E)$ and two different nodes $s$ and $t$ (in our implementation $s$ – source node, $t = m_i$ – multicast node), a set of paths $P^K = \{p_1, ..., p_K\}$ should be constructed as follows:

\begin{itemize}
  \item $c(p_k) \leq c(p_{k+1})$, where $k \in \{1, ..., K - 1\}$,
  \item $c(p_K) \leq c(p)$, for any $p \in P - P^K$,
  \item $p_k$ is determined before $p_{k+1}$.
\end{itemize}

Similarly to shortest path labeling algorithms, their generalizations for the $K$–shortest paths problem are divided into two sets: the label correcting algorithms which determine the set $P^K$ after completing all operations, and the label setting algorithms when each path $p_k$ is computed during the computations. Label setting algorithms are generalizations of Dijkstra algorithm. We called them KSP ($K$–Shortest Path). The time complexity of the KSP algorithm is $O(Kn(m + n \log n))$.

The modification of the original algorithm is based on solving the delay along each of $K$ shortest paths and choosing the path with minimum cost with the delay value not exceeding the maximum delay $\Delta$ along the path.

The operation performed by the algorithm consists in determining the shortest path tree between source node $s$ and each destination node $m_i$, along which the maximum delay value ($\Delta$) cannot be exceeded. The paths, determined one by one, are added to the multicast tree $T$ (Algorithm 1, lines 3-10). If there is at least one path that does not meet the requirements, a multicast tree cannot be constructed (Algorithm 1, line 16). Since the network structure created in the process may contain cycles, in order to avoid them, Prim’s algorithm \cite{20} has been used (Algorithm 1, line 12).

In its last phase the algorithm removes leaves nodes with outdegree 1 that are not multicast nodes (Algorithm 1, line 13). On the basis of the links entering the tree and their metrics (cost and delay), the total cost of the constrained tree is calculated (Algorithm 1, line 14).
Algorithm 1 KSPMA

1: **KSPMA**\((C, D, s, M, \Delta)\)
   
   \(C\) – adjacency cost matrix, \(D\) – adjacency delay matrix,
   
   \(s\) – source node, \(m_i\) – multicast node \(m_i \in M\)

2: \(\text{tree} := \text{true}\)

3: \textbf{for each} \(m_i \in M\) \textbf{do}

4: \(p_i \leftarrow \text{KSP}(C, D, s, m_i, \Delta)\)

5: \textbf{if} \(p_i\) \textbf{exist} \textbf{then}

6: \(\text{AddPath}(p_i, T)\)

7: \textbf{else}

8: \(\text{tree} := \text{false}\)

9: \textbf{end if}

10: \textbf{end for}

11: \textbf{if} \(\text{tree} = \text{true}\) \textbf{then}

12: \(\text{Prim}(T)\)

13: \(\text{DeleteLeaves}(T)\)

14: \textbf{return} \(T\)

15: \textbf{else}

16: \textbf{return} -1 (tree cannot be constructed)

17: \textbf{end if}

4. Simulation methodology

The Internet is a set of nodes interconnected with links. This simple definition makes it possible to represent this real structure as a graph. In fact, the Internet is a set of domains – a number of grouped nodes (routers) which are under joined administration and share routing information. The Internet consists of thousands of domains and autonomous systems (AS). It is possible to generate those kinds of synthetic structures reflecting real topologies [21, 22].

In the study, a flat random graph constructed according to the Waxman method was used [2] to generate networks topologies to validate the accuracy of our algorithm. This method defines the probability of an edge between the nodes as a Euclidean distance function [23].

The second adopted method was proposed by Barabasi [24]. This model uses the power law property in the frequency of the outdegrees in network topologies: incremental growth and preferential connectivity. The network growth process is based on the incremental addition of new nodes. The preferential connectivity refers to the tendency of a new node to connect to existing nodes that are highly connected or popular.
With the construction of the network models based on the Waxman and the Barabasi-Albert methods, BRITE (Boston university Representative Internet Topology gEnera-
tor) [25] was used as a tool for generating realistic topologies. The application provides a range of network topology models and appropriate generative methods.

The research work was conducted with the application of the networks generated by the above-mentioned methods that were appropriately adopted and unified [26, 27]. The nodes were arranged on a square grid with the size of $1000 \times 1000$ (Waxman parameters: $\alpha = 0.15$, $\beta = 0.2$). Onto the existing network of connections (represented by adjacency matrix), the cost matrix $C(u, v)$ was applied (as integer values between the range of 10...1000) and the delay $D(u, v)$ resulting from the Euclidean distance between the nodes.

5. Simulation results

Due to a wide range of solutions presented in the literature of the subject, in order to compare the existing solutions with the proposed KSPMA algorithm, which is a shortest-path algorithm, the following representative algorithms were chosen: KPP [15], LDC and CSPT [8] algorithms. Their popularity in applications and the number of citations in literature were decisive in their selection. Such a set of algorithms includes solutions potentially most and least effective in terms of costs of constructed trees. This, however, makes the results of the comparison more distinct, even with comparisons with the applications of different methods of generating network topologies.

In the first phase of the experiment, the dependence between the total cost of the tree Fig. 2(a) and the average cost of the path in the tree Fig. 2(b) in relation to the number of nodes in the network $n$ was examined. The results presented in Fig. 2(a) show that the total cost of tree obtained for KSPMA, KPP, CSPT and LDC algorithms slightly increases with the increase of the number of nodes in the network, with defined maximum delay value along the path in the tree ($\Delta = 2000$). Whereas the average cost of path in the tree (Fig. 2b) increases or decreases depending on the way of operation of a given algorithm.

The results presented in Fig. 2 show that the KPP algorithm has the best performance (the authors proved in earlier works that costs of multicast trees it produces were only 5% worse than the minimum Steiner tree [28]). Despite its computational complexity and running time of the KPP algorithm, it can be successfully implemented in large networks (with thousands of nodes) and is a good benchmark in our studies. With the adopted assumptions concerning the way of generating parameters in the network (cost is a random value from the interval 10...1000, and delay is a Euclidean distance between the nodes), the influence of the method of generating the network topology on the costs of the obtained trees is not so significant as in the case when the cost was a Euclidean distance
(a) Fig. 2. Total cost of multicast tree (a) and its average path’s cost (b) versus the number of network nodes $n$ 

\[ (m = 20, D_{av} = 4, \Delta = 2000) \]

(which was the case with the algorithms without constraints). The maximum differences in costs of trees for the same algorithm with the Waxman and the Barabási-Albert models are: 6.6% for KPP, 6.1% for KSPMA and 0.8% for CSPT, respectively. Only the LDC algorithm shows a significant difference in the costs of generated trees with the mentioned generating models (11%) – the optimization process uses, however, the delay metrics that is represented as a Euclidean distance. The LDC algorithm (Least-Delay Cost) is the potentially worst solution – it constructs shortest paths trees optimizing only delay along them. Hence, the costs of constructed trees are rather high.

The dependence between the total cost of constructed trees (Fig. 3a) and the average cost of path in the tree (Fig. 3b) in relation to the number of receivers in the network ($m$) was examined in the second phase of the study. For all algorithms under scrutiny, the increase in the number of receiving nodes effects in a linear increase of the average
cost of generated trees (Fig. 3). In this particular scenario of the study, the influence of a network topology on the results of the total cost of tree (Fig. 3a) yielded by the algorithms is rather insignificant (about 4% for \( m = 180 \)). Throughout the whole of the investigation, the KPP algorithm proved to construct trees with minimum costs. Whereas the influence of the network topology on the average cost of path in the multicast tree is significant (Fig. 3b) and in the extreme point the difference is the highest for KPP (about 50% for \( m = 180 \)).

![Graph showing total cost of multicast tree](image)

![Graph showing average cost of path in multicast tree](image)

Fig. 3. Total cost of multicast tree (a) and its average cost of path(b) versus the number of multicast nodes \( m \) (\( n = 200, D_{av} = 4, \Delta = 2000 \))

An important aspect in any simulation experiment is the analysis of the operation of algorithms in those networks where the number of nodes is greater than in the previously used model of real network in relation to the number of its nodes. Studies for dense networks were carried out with the assumed average degree of graph \( D_{av} \) from the interval ranging from 4 (real Internet network) to 20. With the increase of the average network
Fig. 4. Total cost of multicast tree versus the average node degree $D_{av}$ ($n = 200, m = 20, \Delta = 2000$)

Fig. 5. Total cost of multicast tree (a) and its average path’s cost (b) versus the maximum delay $\Delta$

($n = 200, m = 50, D_{av} = 4$)
degree, the total costs of trees determined by the algorithms KPP, KSPMA and CSPT decrease (Fig. 4). The increase in the number of links in a network is followed by the increase in the number of potential paths that can be constructed between the sender $s$, and each of the receivers $m_i$, which enables us to find paths with lower costs. Hence, the lower values of the total cost of tree.

A measure of the quality of algorithms with constraints is the capability of constructing multicast trees with low costs with restrictive constraints of delay along the paths (Fig. 5). The results show that the KPP algorithm is the least sensitive to changes in the parameter $\Delta$. In extreme instances this difference was 10%, while for the KSPMA algorithm – 35%, and for CSPT as much as 48% (the Barabási-Albert method).

The implementation of the KSPMA algorithm also needs some commentary explanations. The simulation experiments were made for the parameter $K = 5$ of the generalized Dijkstra algorithm. The reliability of the analysis of the heuristics under investigation calls for additional study on the influence of the parameter $K$ on the total cost of constructed multicast tree. The relevant results are presented in Fig. 6.

![Diagram](image)

Fig. 6. The influence of the parameter $K$ of the KSPMA algorithm on the average cost of constructed multicast trees in the networks generated with the application of the Waxman model (a) and the Barabási-Albert model (b) ($n = 200$, $m = 20$, $D_{av} = 4$, $\Delta = 2000$)
It is worth emphasizing that the research study was extended to cover another feature of the multicast tree – the average cost of paths between the source and each destination node. The KSPMA algorithm has the best quality, it contracts paths in the multicast tree with minimum costs — Figs. 2–5. In this important part of the study, the network topology influence is noticeable – the paths obtained with the application of the Barabási-Albert model have much lower costs in relation to the trees obtained in Waxman networks. Considering the results mentioned above, the shortest path tree algorithms (KSPMA and CSPT) find better solutions than KPP.

The results show that the influence of the parameter $K$ is insignificant (about 2\% throughout the whole study and regardless of the method used in generating the network topology), though it is noticeable that the trees with the least costs are constructed with $K = 1$. The $K$–shortest-path algorithm (KSP) sorts them out according to the increasing value of cost, so in the case when the first path with the minimum cost does not meet the conditions concerning delay, the tree cannot be constructed and the second path is then analyzed. With such rigorous assumption ($K = 1$), time complexity of the algorithm is the lowest, however its efficiency, understood as the relation of the number of constructed trees to the number of networks in which they could be constructed, is minimal (amounts to about 15\%). Above the value $K = 4$, average costs of trees are maintained at a fixed level, independent on the $K$ parameter.

6. Conclusions

The article presents a new multicast algorithm which is more effective than popular CSPT algorithms. The algorithm was compared with representative routing algorithms for multicast connections emphasizing the quality of the network model. The effectiveness of the algorithm is confirmed by many experiments conducted by the authors for different topologies. The proposed KSPMA algorithm allows to obtain – in authors’ opinion – a reasonable compromise between the total cost of the multicast tree and the computational complexity (Figs. 2, 3 and 5).

The unique aspect of the research work consists in a reliable comparison of the new algorithm for several network parameters and methods that model network topologies. The conducted studies confirmed the effectiveness of the proposed algorithm for different network topologies. It is worth emphasizing that the proposed algorithm can be also successfully applied in large, dense networks with a high average node degree (Fig. 4).

This research study forms a basis for a development of a methodology for further studies of routing algorithms. We believe that the inclusion of the methods of topology generation as well as the basic parameters of the test network in the implemented algorithms are necessary conditions to have the existing routing algorithms compared in a reliable way.
References


Nowy heurystyczny algorytm routingu dla połączeń typu „multicast”

Streszczenie

W artykule zaproponowano nowy algorytm routingu dla połączeń rozgałęzonych. Prezentowany algorytm stosuje algorytm K–najkrótszych ścieżek do wyznaczenia ścieżek o minimalnych kosztach pomiędzy węzłem źródłowym, a każdym z węzłów
docelowych, z uwzględnieniem maksymalnej dopuszczalnej wartości opóźnienia ($\Delta$) wzdłuż każdej ścieżki.


W symulacjach zbadano również efektywność algorytmu w funkcji podstawowych parametrów sieci tj. liczba węzłów sieci, liczność grupy odbiorców, średni stopień grafu (węzła) oraz maksymalne opóźnienie wzdłuż ścieżki. W badaniach uwzględniono także średnie koszty ścieżek uzyskane po zastosowaniu algorytmów CSPT, KPP oraz LDC. Liczne doświadczenia symulacyjne przeprowadzone przez autorów potwierdziły efektywność proponowanego algorytmu KSPMA.