Multicast Source Routing Algorithms

Zhansong Ma

zhansong@tct.hut.fi

ABSTRACT

Multicast services have been increasingly used by various continues media applications. The QoS requirements of these continues media applications prompt the necessity for QoS-driven, constraint-based multicast routing. This paper provides an overview of multicast routing problems, which are classified according to their optimization functions and performance constraints, of multicast source routing algorithms, which are used to solve Steiner and constrained Steiner tree problems. An example of algorithm also is introduced in order to make the understanding of solving multicast routing problem better. A summary and possible future research directions are pointed out in the end of this paper.

1 Introduction

Multicasting is the simultaneous transmission of data to multiple destinations. This is becoming a key requirement of computer networks supporting multimedia applications. In order to support large numbers of multicast sessions efficiently, the network must transport the information exchanged in those sessions using as few network resources as possible, while meeting the session's service requirements. The current approach to support a multicast session efficiently in the network consists of establishing a multicast tree for the session, along which information is transferred. Algorithms used to compute such trees in the network are called multicast routing algorithms.

Meanwhile with the increase of requesting Quality of Service on the Internet, there is a need to consider multicast routing algorithms capable of satisfying the QoS requirements. Such algorithms are generally called QoS multicast routing algorithms. It can be defined as follows: given a source node s, a set R of destination nodes, a set of constraints C and possibly an optimization goal, find the best feasible tree covering s and all nodes in R which satisfies C.

Like the traditional routing, QoS routing involves two basic tasks: (1) collecting the state information and keeping it up date; (2) searching the state information for a feasible path in unicast routing or a feasible tree in multicast routing. The search for a feasible path/tree greatly depends on how the state information is collected and where the information is stored. According to how the state information is maintained and how the search of a feasible path/tree is carried out, QoS routing can be categorized into source routing, distributed routing and hierarchical routing[1]. In source routing, each node in a network maintains a complete global state including the network topology and state information of every link, and based on the global state a feasible path/tree that satisfies a set of constraints and possibly an optimization goal is locally computed at the source node.

Source routing achieves its simplicity by transforming a distributed problem into a centralized one. By maintaining a complete global state, the source node calculates the entire path locally. It avoids dealing with distributed computing problems such as distributed state snapshot, deadlock detection, and distributed termination. It guarantees loop-free routes. Many source algorithms are conceptually simple and easy to implement, evaluate, debug, and upgrade.

Besides, source routing has some problems. First, the global state maintained at every node has to be updated frequently enough to cope with the dynamics of network parameters such as bandwidth and delay. This makes the communication overhead excessively high for large-scale networks. Second, the link state algorithm can only provide approximate global state due to the overhead concern and non-negligible propagation delay of state message. As a consequence, QoS routing may fail to find a feasible path/tree due to the imprecision in the global state used. Third, the computation overhead at the source is excessively high. This is especially true in the case of multicast routing or when multiple constraints are involved.

Many works about QoS multicast source routing algorithms have been done. The purpose of this article is to provide a survey of them as well as a detailed description of an algorithm.

The rest of this paper is constructed as follows. Section 2 presents multicast routing problems: network model and problem formulation, classification of problems. An overview of existing multicast-source routing algorithms are described in section 3. Then, one of these algorithms, BSMA, as an example, is introduced in section 4, Finally a summary and future direction are presented in the section 5.

2 Multicast Routing Problems

2.1 The Network Model

When multicast routing is concerned, a network is usually represented as a weighted graph $G=(V,E)$, where V is a set of nodes and E is a set of directed links. $|V|$ and $|E|$ denote the number of nodes and links in the network, respectively. Associated with each link are parameters that describe the current status of the link(e.g., link delay, link cost). These parameters are collectively termed link state. Similarly, associated with each node are parameters representing the current status of the node(e.g., buffer space available), which can be independently measured for each outgoing interface or aggregately measured for the node. These parameters are termed node state. The collection of the local node/link state maintained in the network is termed global network state.

Let $M \subset V$ be a set of nodes involved in a group communication. M is called multicast group with each node $v \in M$ a group member. Packets originating from a source node v_s have to be delivered to a set of receiver nodes $M\text{-}\{v_s\}$. A multicast tree T is a subgraph of G that spans all the nodes in M. The path from a source node v_s to a receiver node v_d, in the tree T is denoted by $P_T(v_s,v_d)$.

2.2 Classification of Multicast Routing Problems

Given a multicast group M, a set of constraints C, and possibly a set of optimization objective functions O, multicast routing is the process of constructing, based on network topology and network state, a multicast tree T that satisfies C and optimizes O.

2.2.1 Objective Functions and Constraints

Objective Functions:
The optimization objectives that are going to be sought are usually defined in the form of minimizing the cost of a multicast tree, where the cost may be the total bandwidth used and/or a monotonically nondecreasing function of network utilization.

Constraints:
The constraints imposed can be classified into link constraints and tree constraints.

- Link constraints are restrictions on the use of links for route selection. For example, one may request that the bandwidth or buffer available on a link be greater than or equal to a predetermined value.
- Tree constraints are one of these two: (1) bounds on the combined value of a performance metric along each individual path from the source to a receiver in a multicast tree. For example, the end-to-end delay bound on the paths from the source to all the receivers. (2) bounds on the difference of the combined value of a performance metric along the paths from the same source to any two different receivers. For example, the interreceiver delay jitter bound defined as the difference between the end-to-end delay along the paths from the same source to any two different receivers.

Depending on how a tree constraint is derived from the corresponding link metrics, tree constraints can be further classified into additive, multiplicative and concave tree constraints. Let $m(l)$ be the performance metric for link l, for any path $P_T(u,v)=(u,i,j,\ldots,k,v)$.

- If $m(u,v) = m(u,i) + m(i,j) + ... + m(k,v)$ the tree constraint is additive. For example, the end-to-end delay $d(u,v)$ from node u to node v is additive and equal to the sum of individual link metric $d(i,j)$ along the path $P_T(u,v)$.
- If $m(u,v) = m(u,i) \times m(i,j) \times ... \times m(k,v)$ the tree constraint is multiplicative. For example, the probability, $1\text{-}P_L(u,v)$, for a packet to reach node v from node u along path $P_T(u,v)$ is multiplicative and is equal to the product of individual link metric $1\text{-}P_L(i,j)$ along the path $P_T(u,v)$.
- If $m(u,v) = \min[m(u,v), m(i,j), ..., m(k,v)]$ the tree constraint is concave. For

example, the bandwidth *b(u,v)*, available along a path from node *u* to node *v*, is concave and is equal to the minimum bandwidth among the links on path $P_T(u,v)$.

2.2.2 Classification of Multicast Routing Problem

Depending on the link/tree constraints imposed and the objective functions used, a multicast routing problem can be formulated as:

1. A link-constrained problem: a link constraint is imposed to construct feasible multicast trees such as bandwidth-constrained routing.
2. A multiple-link-constrained problem: two or more link constraints are imposed to construct feasible trees such as bandwidth- and buffer-constrained routing.
3. A tree-constrained problem: a tree constraint is imposed to construct feasible trees such as delay-constrained routing.
4. A multiple-tree-constrained problem: two or more tree constraints are imposed to construct feasible trees such as delay- and interreceiver-delay-jitter-constrained routing.
5. A link- and tree-constrained problem: a link constraint and a tree constraint are imposed to construct feasible trees such as delay- and bandwidth-constrained routing.
6. A link optimization problem: a link optimization function is used to locate an optimal multicast tree such as maximization of the link bandwidth over on-tree links in a multicast tree.
7. A tree optimization problem: a tree optimization function is used to locate an optimal multicast tree such as minimization of the total cost of a multicast tree. This is also known as Steiner tree problem.
8. A link-constrained link optimization problem: a link constraint is imposed and a link optimization function is used to locate an optimal multicast tree that fulfills the constraint such as the bandwidth-constrained buffer optimization problem.
9. A link-constrained tree optimization problem: a link constraint is imposed and a tree optimization function is used to locate an optimal multicast tree that fulfills the constraint such as the bandwidth-constrained Steiner tree problem.
10. A tree-constrained link optimization problem: a tree constraint is imposed and a link optimization function is used to locate an optimal multicast tree that fulfills the constraint such as the delay-constrained bandwidth optimization problem.
11. A tree-constrained tree optimization problem: a tree constraint is imposed and a tree optimization function is used to locate an optimal multicast tree that fulfills the constraint such as the delay-constrained Steiner tree problem.
12. A link and tree constrained tree optimization problem: link and tree constraints are imposed and link optimization functions are used to locate an optimal multicast tree that fulfills the constraints such as bandwidth- and delay constrained tree optimization problem.

Problem 1 and 2 are tractable, becasue link constraints can be fulfilled by removing from the network topology links that do not meet the selection criteria. Wang and Crowcroft[2] proved that the problem of finding a path subject to two or more independent additive/multiplicative constraints in any possible combination is NP-complete. The only tractable combinations are the concave constraint and the other additive/multiplicative constraints. As a result, problem 3, 5 and 10 are polynomial time solvable, while problem 4 is NP-complete. A solution algorithm of polynomial time complexity to problem 6 has been proposed by Shacham[3]. Problem 8 reduces to problem 6 if links that do not meet the link constraints are removed from the network topology. Hence, it is also polynomial time solvable. Finally, problem 7 (Steiner tree problem) and problem 9, 11 and 12 (constrained Steiner tree problem) have been proved to be NP-complete[4]. Some source routing algorithms used to solve these problems are briefly introduced in next section. A nice survey on multicast routing problems can be found in [5].

3 Overview of Multicast Source Routing Algorithms

Many multicast routing algorithms have been proposed to solve these problems classified in section 2. This chapter only focuses on multicast source routing algorithms.

3.1 MOSPF

MOSPF[6] is a multicast extension of the unicast link-state protocol OSPF. It was based on Deering's work[7]. In addition to a global state information, at every node the protocol maintains the membership information of every multicast group in the routing domain.

Group membership changes in a subnetwork are detected by a local router, which broadcasts the information to all other nodes. Given all knowledge of network state and group membership, any node can compute the shortest-path multicast tree from a source to a group of destinations by using Dijkstra's algorithm. Such a protocol can be easily used for delay-constrained multicast routing.

3.2 Algorithms Associated with Steiner Tree Problem

The algorithm proposed by Kou, Markowski and Berman (KMB) [8] and the algorithm proposed by Takahashi and Matsuyama (TM) [9] are used to solve Steiner tree problem.

Steiner tree is the least-cost tree that spans a given subset of nodes. Strictly speaking, finding a Steiner tree is not a QoS routing problem. However, the heuristics for constructing a Steiner tree have a direct impact on how to construct a constrained Steiner tree.

The KMB Algorithm

KMB first abstracts a network to a complete graph, where the nodes represent the source and destinations, and the edges represent the represent the shortest paths among source node and destination nodes. Then KMB applies Prim's algorithms[10] to construct a minimum spanning tree in the complete graph, and the Steiner tree of the original network is obtained by expending the edges of the minimum spanning tree into the shortest paths they represent. Any loops caused by the expansion are removed.

The TM Algorithm

TM algorithm finds a Steiner tree by an incremental approach called nearest destination first(NDF). Initially, the nearest destination(in the terms of cost) to the source is found and the least-cost path between them selected. Then at each iteration the nearest unconnected destination to the partially constructed tree is found and added into the tree. This process is repeated until all destinations are included in the tree.

3.3 Algorithms Associated with Constrained Steiner Tree Problem

The Steiner tree problem has been extended to include other side constraints, for example, delay, delay jitter, or a combination thereof. These problems are NP-complete, most algorithms in this category are source routing. Four algorithms are discussed below.

The KPP Algorithm

Kompella, Pasquale and Polyzos(KPP) proposed this algorithm[11]. In KPP, the first step is to create a complete graph, where the nodes represent the source and destinations, and the edges represent the delay-constrained least-cost paths between these nodes. The link delays are assumed to be integers, and the delay constraint always bounded, so such a complete graph can be constructed in polynomial time. The second step is to construct a delay-constrained spanning tree of the complete graph. Starting with the source node, the tree is incrementally expanded by adding an edge each time until every destination node is included. The selected edge is one that:

- Connects a node in the tree and a node outside of the tree
- Does not violate the delay constraint
- Minimize a selection function

Two selection functions are considered. One is simply the cost of the edge, and the other tries to make a trade-off between minimizing the cost and minimizing the delay. The third step is to expand the edges of the constrained spanning tree into the delay-constrained least-cost paths they present. Any loops caused by this expansion is removed.

The ZPG Algorithm

Zhu, Parsa and Garcia-Luna-Aceves(ZPG)[12] proposed this problem. ZPG allows variable delay bounds to destinations. A shortest-path tree in the terms of delay is first constructed by Dijkstra's algorithm. If the delay constraint can not be satisfied for any destination, it must be renegotiated; otherwise, the algorithm proceeds by iteratively refining the tree for lower cost. The basic idea is to replace a path in the tree by another path with lower cost unless such a replacement can not be found. The algorithm always finds a delay-constrained tree(probably not least-cost), if one exists, because it starts with a shortest-path tree.

The SL Algorithm

Sun and Langendoerfer(SL)[13] proposed an algorithm which constructs an approximated constrained Steiner tree by Dijkstra's algorithm. It first computes the shortest-path tree in terms of cost. Namely, the cost of every path in the tree from the source to a destination is minimized. Then the tree is modified to satisfy the delay constraint. If the end-to-end delay to any destination in the tree violates the delay constraint, the minimum-delay path is used replace the minimum-cost path. The advantage of the algorithm is its low time complexity, *O(vlogv),* which is the same as Dijkstra's algorithm.

The Widyono Algorithm

Widyono[14] proposed several heuristic algorithms for the constrained Steiner tree problem. The one with the best performance is called the constrained adaptive ordering heuristic. At each step, a constrained Bell-man-Ford algorithm is used to find a delay-constrained least-cost path from the source to a destination that is not yet in the tree. The found path as well as the destination are then inserted into the tree. The cost of links in the tree is set to zero. The above process repeats until the tree covers all destinations.

3.4 The RB Algorithm

Rouskas and Baldine(RB)[15] proposed a heuristic for constructing a delay-delay-jitter-constrained multicast tree. The tree must have bounded delay along the paths from the source to the destinations and bounded variation among the delays along these paths. The shortest path tree in the terms of delay is first constructed by Dijkastra's algorithm. If the tree does not meet the delay jitter constraint, the algorithm finds the largest-delay path in the shortest path tree from the source to a destination, and starts from that path to incrementally construct a feasible tree. At each iteration, a "good" path from a node in the tree to a destination out of the tree is found and added into the tree. The path must be completely disjoint from the tree and must not cause the tree to violate the constraints. The above process repeats until all destinations are included in the tree. The authors showed that the heuristic demonstrates good average-case behavior in terms of the maximum interdestination delay variation.

4 An example of Multicast Source Routing Algorithms -BSMA

Multicast tree construction is becoming an integral part of multimedia application support. Hence, more attention is coming to how to construct such multicast trees. This section introduces a source-based routing algorithm named Bounded Shortest Multicast Algorithm(BSMA)[12] for the construction of delay-bounded minimum-cost multicast trees.

One purpose of this section is to serve as a guideline of solving multicast routing problem: first modeling a network and formulating the problem, then constructing a multicast tree with some constraints such as end-to-end delay, and repeatedly refining the tree to optimize it.

4.1 Network Model and Problem Formulation

In the network model described in 2.1, the nodes in set V can be of the following three types:

- Source node: the node connecting to the source that sends out the data stream.
- Destination node: the node connecting a destination that receives the data stream. The set of destination nodes in a multicast tree is denoted by $D \subseteq V\text{-}s$.
- Relay node: a node that is an intermediate hop in the path from the source to a destination.

Two positive real-valued functions are defined on E :

Link-Cost Function: $(c{:}E \rightarrow \Re^{+})$

The cost of a link can be associated with the utilization of the link; a high utilization is represented by a higher link cost.

Link-Delay Function:$(d{:}\ E \rightarrow \Re^{+})$

The delay of a link is the sum of the perceived queuing delay, transmission delay, and propagation delay over that link.

Let D be the set of destinations. For each path from s to a destination node $v \in D$, the delay of the path, or path delay is defined to be the sum of link delays along the path. The following function is defined on the set of destinations D:

Destination Delay-Bound Function or DDF: $(\delta{:}D \rightarrow \Re^{+})$

DDF assigns an upper bound to the delay along the path from the source to each destination in D. $\delta(i)$ can be different from $\delta(j)$ for destinations $i \neq j$. If DDF assigns the same upper bound delay to each destination, the upper bound is denoted by $\delta(i)=\Delta,\ \forall i \in D$.

A Delay-bounded Minimum Steiner Tree (DMST) problem is defined as:

DMST problems: *given a graph $G=(V,E)$ with a link-cost function, a link-delay function, a source s, a set of destination D, and a DDF, then construct a DMST spanning $D \cup \{s\}$, such that the cost function of the tree is minimizing while DDF is satisfied.*

4.2 Overview of BSMA

BSMA assumes that the source node has complete information regarding all network links to construct a multicast tree, hence it is a source routing algorithm. This requirement can be supported using one of many topology-broadcast algorithms, which can be based on flooding(as the case in OSPF and IS-IS) or other techniques.

BSMA is based on the feasible search optimization method. This method minimizes the objective function constrained inside a feasible region. The feasible region $\Re b$ for the BSMA problem consists of the set of all delay-bounded Steiner trees. BSMA starts with an initial tree $T_0 \in \Re_b$, and iteratively refines the tree for low cost while staying in the feasible region.

BSMA consists of two major steps, a skeleton flowchart is shown in Figure 1.

Step 1 constructs the initial tree T_0 which is a minimum-delay Steiner tree, with respect to the multicast source, using Dijkastra's shortest path algorithm. Step 2 then iteratively refine T_0 for low cost. Figure 2 is an example network and Figure 3 shows the corresponding minimum-delay Steiner tree for a multicast group in the network.

In some cases the delay bounds given by DDF may be too tight, i.e., they can not be met even in the minimum delay tree T_0. In such cases, some negotiation is required to loose the delay bounds of DDF before any feasible tree can be constructed, as shown in Figure 1. To guarantee the output tree is delay-bounded, step 1 must be invoked with reassigned delay bounds, after the negotiation has been accomplished. (in the following it is assumed that DDF gives delay bounds that have been met by the minimum-delay tree T_0).

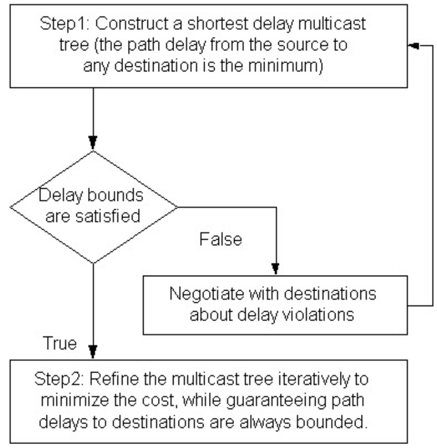


Figure 1. Flowchart of Basic approach used in BSMA

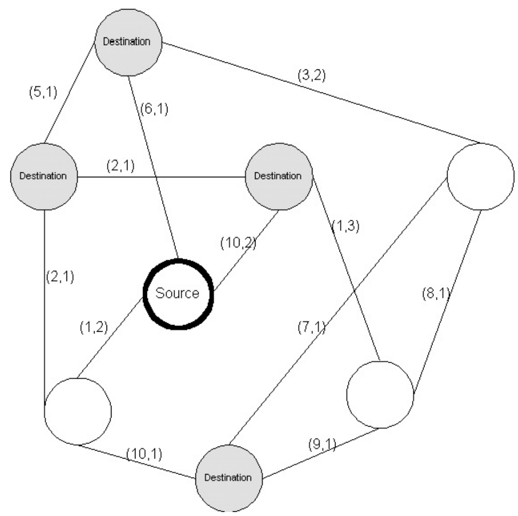


Figure 2. A network with each link(cost, delay)

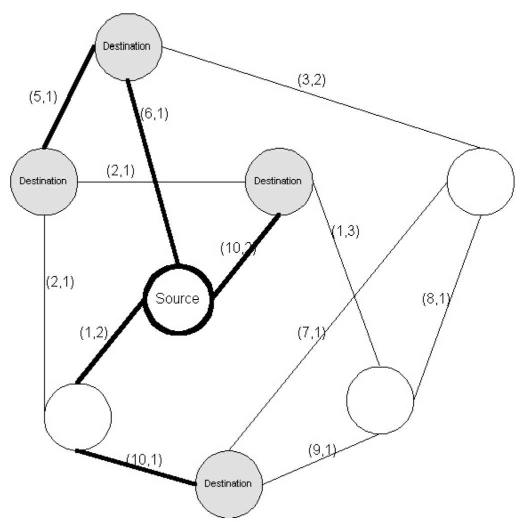


Figure 3. The shortest-delay Steiner Tree

Step 2 iteratively refines T_0 in a monotonous cost reduction. A tree obtained during each refinement is called a tree configuration, and jth tree configuration is denoted by T_j. The refinement from T_j to T_{j+1}(initially, $j=0$) is accomplished by an operation called path switching.

4.3 Delay Bounded Path Switching

Path switching means that a path in T_j is replaced by a new path that is not in T_j. This process results in a new tree configuration T_{j+1}. Delay bounded path switching guarantees that T_{j+1} is a delay bounded tree. An example is shown in Figure 4, where the goal is to obtain a DMST satisfying delay bound $\Delta=5$; accordingly, a path AB of the tree is switched to a new path BC. The cost of the tree is reduced from 31 to 23, while the maximum path delay is 4 which is still bounded by $\Delta=5$.

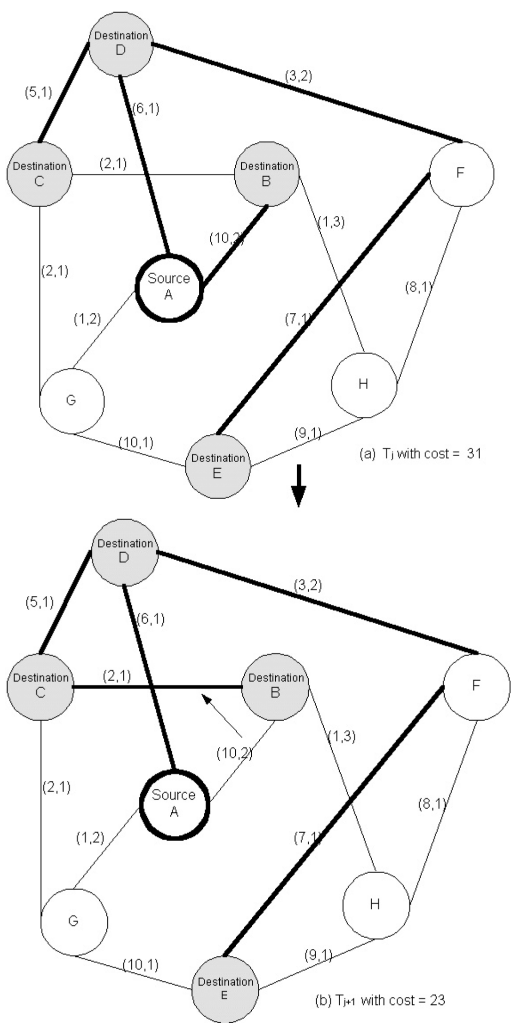


Figure 4 Delay bounded path switching: a path *AB* is replaced by a new path *BC*.

To represent the candidate paths chosen in the path switching, a new tree is defined. It is denoted by T'_j which is called collapsed tree of T_j. The collapsed tree T'_j consists of nodes and superedges.

- Nodes: The set of nodes of consists of the source node and destination nodes of T_j, and those nodes of T_j that are connected by more than two tree edges in T_j.
- Superedges: A superedge in T'_j is the longest simple path in T_j in which all internal nodes are relay nodes and each relay node connects exactly two tree edges.

To reduce the cost of T_j BSMA deletes a superedge from T'_j (which amounts to removing a path p from T_j) resulting in two subtrees T_j^1 and T_j^2, where $T_j = T_j^1 \cup T_j^2 \cup p$. BSMA then finds a delay-bounded shortest path p_s, to reconnect $T_j^1$and T_j^2

Path cost is defined as follows:

- For utilization-driven multicasting, the path cost is the sum of costs along this path.
- For congestion-driven multicasting, the path cost is the maximal link cost along the path.

The cost of a superedge in T'_j is defined to be the cost of the corresponding path in T_j.

A delay-bounded shortest path p_s between T_j^1 and T_j^2 is defined as the path with the smallest cost, subject to the constraint that the new tree $T_{j+1} = T_j^1 \cup T_j^2 \cup p_s$ is a delay bounded tree. An example is shown in Figure 5.

If there are more than one delay-bounded shortest paths, a tie-breaking choice rule is deeded: the chosen path is the one providing the maximal slack of destination delay bounds, so that there is more delay budget for iterative refinement.

To get the delay-bounded shortest path, BSMA incrementally constructs the k-shortest paths between two subtrees T_j^1 and T_j^2 by using a k-shortest path algorithm[17].

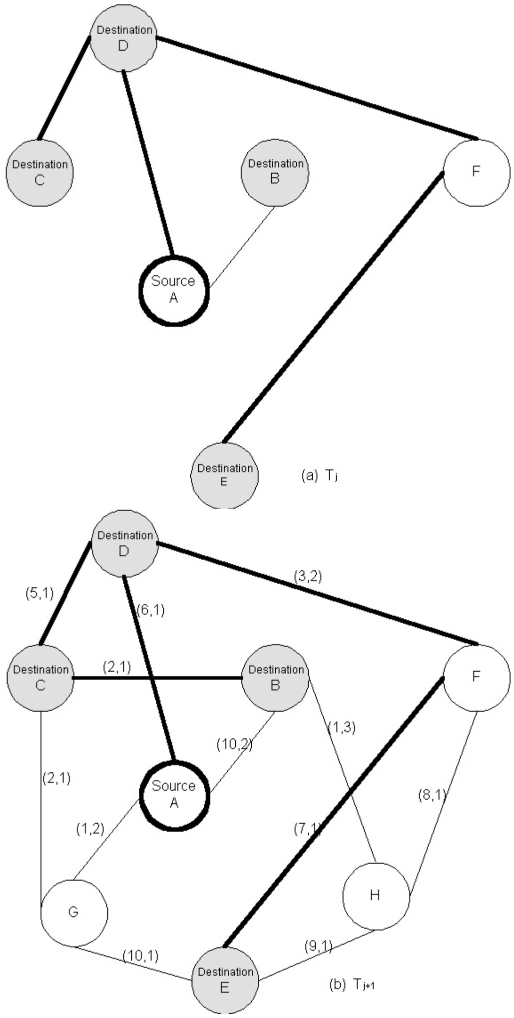


Figure 5 (a) Removing AB from the tree results in two unconnected subtree T_j^1 and T_j^2. (b) The shortest delay bounded path BC connecting T_j^1 and T_j^2.

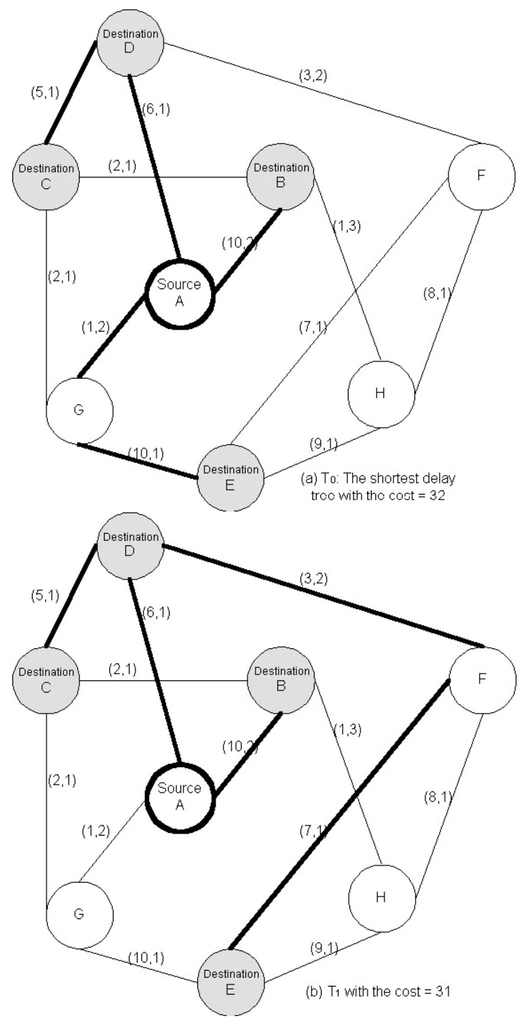
4.4 Path Switching Algorithm

The following is the pseudo code of BSMA.

INPUT:
G(V,E) = graph,
s= source node,
D=set of Destination Nodes,
DB= set of delay bounds for destination nodes
Type= cost function type for the tree, which can be utilization driven, to minimize the total link cost or congestion driven, to minimize the maximal link cost.
OUTPUT:
A delay bounded Steiner tree spanning $D \cup \{s\}$
ProcedureMulticastTree*(G(V,E), s, D, DB, Type) {*
$j \leftarrow 0$;
$T_j \leftarrow$ minimum delay tree spanning $D \cup \{s\}$ found using Dijkastra's Algtorithm;
loop *{*
if *(Type = utilization driven)*
$p_h \leftarrow$ an unmarked superedge in T_j with the highest cost among all unmarked superedges(not compared with marked superedges);
else if *(Type = congestion driven)*
$p_h \leftarrow$ an unmarked superedge in T_j withthe highest cost among all superedges (also compared with marked superedges);
if *(p_h = NULL)*
return*;*
Mark superedge p_h;
Remove p_h from tree T_j getting two subtrees T^1 and T^2;
*ps$\leftarrow$**DelayBoundedShortestPath**(G,s,DB, T^1 T^2);*
$j \leftarrow j+1$;
$T_j \leftarrow p_s \cup T^1 \cup T^2$
If *($p_s \neq p_h$)*
Unmark all marked superedges;
}
}

DelayBoundedShortestPath() corresponds to the use of a k-shortest path algorithm.

Figure 6 shows the steps of constructing a multicast tree based on utilization-driven optimization. The total edge cost of tree T is dropped from 32(the shortest delay tree) to 21, and the maximal path delay of the final tree is $4<\Delta=5$.



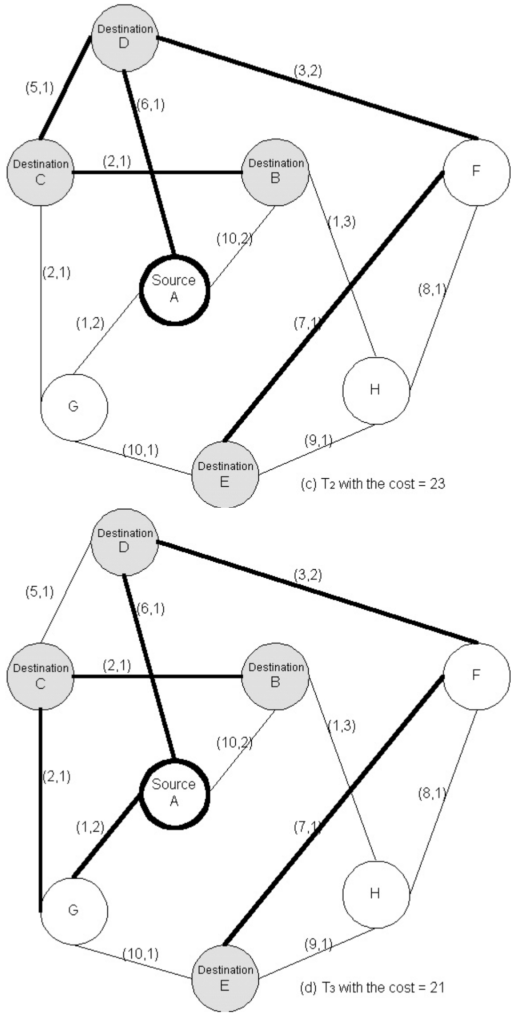


Figure 6 Example of utilization-driven iterative optimization in BSMA

4.5 Greedy Path Switching Algorithm

Another algorithm can be used in BSMA is a greedy selection of path switching at each tree refinement. The cost reduction of after a round of path switching is defined as gain. Let p be the path in tree T_j with cost c, and p^* the corresponding delay bounded shortest path to be added into tree T_{j+1} with cost c^*. The gain of this path switching is defined as $g=c-c^*$.

BSMA computes gains of all pairs of possible path switchings in T_j, and then selects the one with the maximum gain. BSMA continues the greedy path switching, and terminates when the maximum gain is zero.

4.6 Evaluation of BSMA

The proposers of BSMA proved in [12] that BSMA always constructs a delay-bounded multicast tree, if such a tree exists and monotonically decreases the cost of delay-bounded tree.

Considering a network of n nodes and denote by k the average umber of the k-shortest paths constructed to obtain the delay-bounded shortest path. It is shown that the expected time complexity of BSMA is $O(kn^3 log(n))$, and $O(kn^3)$ in a degree-bounded network where the maximal degree is upper bounded. Proof can be found in [12].

5 Summary

All multicast source routing algorithms require global state to be maintained at every node in a network. Most heuristic algorithms for the NP-complete multicast routing problem construct a constrained tree incrementally by adding one destination into the tree each time based on certain selection criteria. The KMB and KPP algorithms reduce the original problem to a spanning tree problem by constructing a logical complete graph among the source node and the destination nodes. The RB algorithm constructs a multicast tree with both bounded delay and bounded jitter, which is very useful in interactive audio-visual communication such as teleconferencing. The ZPG (or BSMA) algorithm iteratively refines the multicast tree by replacing paths in the tree for lower cost. This algorithm can handle two variants of the cost function: one minimizing the total link cost of the tree, and the another minimizing the most congested(maximal) link cost Among these algorithms for the constrained Steiner tree problem, Salama's simulation[16] showed that:

- The ZPG algorithm achieve the best average performance in terms of minimizing the cost of tree.
- The SL algorithm has the shortest execution time.

The computation is easy and fast in most multicast source routing algorithms, and policy routing can be easily integrated. However, the overhead to maintain the whole network state in the route computing node can be very large. Therefore, it is not practical for very large network, where the complete knowledge of the network is difficult to collect. In most cases, such algorithms are of theoretical interest and should be useful as tools to measure the performance of other routing algorithms. In addition, it hoped that the study of such algorithms will contribute to a better understanding of the multicast routing problem.

References

[1] Chen, S. & Nahrsted, K., "An Overview of Quality of Service Routing for Next Generation High-speed Networks: Problems and Solutions", IEEE Network, November/December 1998.

[2] Wang, Z. & Crowcroft, J., "QoS Routing for Supporting Resource Reservation", IEEE JSAC, September, 1996.

[3] Shacham, N. " Multicast Communication by Hierarchically Encoded Data", IEEE INFOCOM'92, May, 1992, pp. 2107-114.

[4] Carey, M.R. & Johnson, D.S., "Computers and Intractability", New York, Freeman, 1979.

[5]Wang, B. & Hou, J.C., "Multicast routing and its QoS extension: problems, algorithms, and protocols", IEEE Network Volume: 14 1 , Jan.-Feb. 2000 , Page(s): 22 –36.

[6] Moy, J., "Multicast Extensions to OSPF", Internet Draft, September, 1992.

[7] Deering, S. & Cheriton, D., " Multicast Routing in Datagram Internetworks and Extended LANs", ACM Trans, Comp. Sys. May, 1990, PP. 85-111.

[8] Kou, L. & Markowsky, G. & Berman, L., " A Fast Algorithm for Steiner Tree", Acta Informatica 15, 1981, pp.141-145.

[9] Takahashi, H. & Matsuyama, A., " An Approximate Solution for the Steiner Tree Problem in Graphs", Mathematica Japonica, 1980.

[10] Prim, R., "Shortest Connection Networks and Some Generalizations", Bell System Tech.J., Vol.36, 1957, pp. 1389-1401.

[11] Kompella,V.P. & Pasquale, J.C.& Polyzos, G.C., "Multicast Routing for Multimedia Communication", Networking, IEEE/ACM Transactions on Volume: 1 3 , June 1993, Page(s): 286 –292.

[12] Qing, Z & Parsa, M.& Garcia-Luna-Aceves, J.J., "A Source-based Algorithm for Delay-constrained Minimum-cost Multicasting", INFOCOM '95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. 'Bringing Information to People', Proceedings., IEEE , 1995 , Page(s): 377 -385 vol.1.

[13] Sun, Q. & Langendofer, H., " A New Distributed Routing Algorithm with End-to-End Delay Guarantees," 2th Wksp. Protocols Multimedia Sys. October , 1995, pp.452-458.

[14] Widyono, R., " The Design and Evaluation of Routing Algorithms for Real Time Channels", Technical Report, ICSI TR-94-024, Univ. CA at Berkeley Int'l. Comp. Sci. Inst., June, 1994.

[15] Touskas, G.N. & Baldine, I., " Multicast Routing with End-to-End Delay and Delay Variation Constraints", IEEE JSAC, vol. 15, April, 1997, pp. 345-356.

[16] Salama, H.F. & Reeves, D.S. & Viniotis, Y., "Evaluation of Multicast Routing Algorithms For Real-Time communication on High-Speed Networks," IEEE JSAC, vol. 15, no.3, April, 1997, pp.346-356.

[17] Lawler, E., "Combinational Optimization: Networks and Matroids", Holt, Rinehart and Winston, 1976..