Proceedings of the 5th WSEAS International Conference on Telecommunications and Informatics, Istanbul, Turkey, May 27-29, 2006 (pp254-259)

A Survey of Approximation Algorithms for Multicast Congestion Problems

RONGJOU YANG
Wufeng Institute of Technology
Dept. of IM
117, Chian-Kuo Rd., Sec. 2,
Ming-Hsiung , Chia-yi
Taiwan, R.O.C.

HANN-JANG HO
Wufeng Institute of Technology
Dept. of CSIE
117, Chian-Kuo Rd., Sec. 2,
Ming-Hsiung , Chia-yi
Taiwan, R.O.C.

SINGLING LEE
National Chung-Cheng University
Dept. of CSIE
160, San-Hsing,
Ming-Hsiung, Chiayi
Taiwan, R.O.C.

Abstract: Due to the recently rapid development of multimedia applications, multicast has become the critical technique in many network applications. In this paper, We investigate contemporary research concerning multicast congestion problems with the objective of minimizing the maximum sharing of a link. These problems include: multicast Steiner tree and multicast packing problem, etc. Most of these problems have already been proved as NP-complete, thus are mainly formulated as the Integer Linear Programming (ILP). Our objective is to investigate and analyze some of most recently developed approximation algorithms for the optimization of multicast congestion problems. We also discus how they are modelled and solved in the literature.

Key–Words: Approximation Algorithm, Multicast packing, Multicast Steiner trees, Randomized metarounding, Integer Linear Programming, LP-Relaxation

1 Introduction

In multicast routing, the main objective is to send data from one or more sources to multiple destinations in order to minimize the usage of resources such as bandwidth, communication time and connection costs. The multicast congestion problem is to find a set of multicast trees that minimize the maximum congestion over all its edges. The congestion of an edge is the number of multicast trees that use the edge. Given a physical network $G = (V, E)$ with a set V of n nodes, a set E of undirected network links and m multicast requests $S = S_1, S_2, \ldots, S_m$ being subsets of V, a solution to the problem is a set of m trees such that the i^{th} tree spans the nodes of the i^{th} multicast request. The objective function is to minimize the maximum congestion. The problem is formulated as an ILP (Integer Linear Programming) and its LP relaxation solution finds fractional solutions for each multicast request.

In the multicast packing problem, the network tries to accommodate simultaneously all the multicast groups (many-to-many) and avoid bottlenecks on the links to achieve higher throughput (i.e., minimize the maximum link sharing among the multicast groups). A shared tree can be considered as the backbone of a group multicast session. One way to minimize the maximum congestion is to increase the size of some multicast trees, but this also increases the delay which must be considered in the objective function of the optimal packing problem formulation. The delay is a function of the amount of dilation α from the size of the optimal tree obtained for each group multicast independently from the others (i.e., in isolation).

Priwan [1] proposed both heuristic algorithm finding approximate solution and search enumeration based algorithm finding optimal solution, and compared the approximate solution with the optimal solution in order to lower costs for the subscribers and conserves bandwidth resources for the network providers. In these algorithms, the connection approach is based on setting multicast tree routes that each participant (site) has one own multicast tree connecting to the other participants under two constraints: the delay-bounded constraint of source-destination path and the available constrained bandwidth for the service of links.

In [3], Wang formulated the problem as a tree packing problem with multiple multicast sessions under a capacity limited constraint and proposed two heuristic algorithms, Steiner-tree-based heuristic (STH) algorithm and cut-set-based heuristic (CSH) algorithm, for solving this problem. They showed that the STH algorithm can find a better approximate solution in a shorter computation time compared to CSH.

The remainder of this paper is organized as follows: Section 2 discusses the multicast Steiner Trees problems in general graphs with the objective of minimizing the maximum sharing of a link. We outline and analyze some of most recent approximation algorithms and related lower bounds for these problems. In Section 3, we discuss the Multicast Packing Problem. Discussion is given in Section 4.

2 The Multicast Steiner Trees Problem

The Steiner tree problem is solving combinatorial optimization problem when adding new vertices is permitted before finding the Minimum Spanning Tree (MST). It can be divided into three categories: Euclidean Steiner tree problem, metric Steiner tree problem, and Steiner tree in graphs which is focussed on this paper. The Steiner tree problem has applications in circuit layout or network design [12, 13, 14, 15, 16]. Most versions of the Steiner tree problem are NP-complete (computationally hard). Some restricted cases can be solved in polynomial time. In practice, heuristics are used.

In this section, we discuss the multicast Steiner Trees problems including the objective functions and approximation algorithms for minimizing the sum of the congestion over all edges.

The Steiner tree problem is to minimize the sum instead of the maximum of the congestion over all edges in which a single multicast request consists of more than two nodes. Finding edge disjoint paths (the boolean satisfiability problem) was first proved as a NP-complete problem by Karp back in 1972 [4]. Finding a minimum Steiner tree is max-SNP hard (see [6] for details) and an approximate ratio solution (1+ϵ) ($\epsilon > 0$ is a constant) for the special case of edge length equal to 1 or 2 was found in [5]. In Steiner tree problem, different graphs are formed dynamically as different multicast Steiner trees so that the maximum flows of the generated multicast Steiner trees are minimized. The congestion of an edge is the number of multicast trees that use the edge. The problem is to find a set of multicast trees that minimize the maximum congestion over all the edges.

2.1 The Iterative Randomized Rounding Algorithm

A Linear Programming (LP) for any Integer Programming (IP) can be generated by taking the same objective function and same constraints but with the requirement that variables are integer replaced by appropriate continuous constraints. The LP relaxation of the IP is the LP obtained by omitting all integer and 0-1 constraints on variables.

Randomized rounding is a probabilistic method to convert a solution of a relaxed problem into an approximate solution to the original problem. Relaxation is an optimization problem with an enlarged feasible region and extended objective function compared with an original optimization problem.

Let $G = (V, E)$ denote a physical network of n nodes, $S_1, S_2, \cdots, S_m$ denote m multicast requests, a binary variable x_{te} indicating whether edge e is chosen for the t^{th} multicast, to ensure that any solution to the ILP connects all the vertices of each multicast, the problem is formulated as follows by Vempala and Vöcking in [7] :

Minimize z

Subject to

$\sum_{e\in\delta(S)} x_{te} \geq 1, \quad \forall t, \forall S \subset V, S \cap S_t \neq \emptyset, (V \backslash S) \cap S_t \neq \emptyset$

$\sum_t x_{te} \leq z, \quad \forall e \in E$

$x_{te} \in 0, 1, \quad \forall t, \forall e \in E$

In the case for each multicast consisting of only two nodes, a LP relaxation solution was obtained by relaxing the binary variable to $0 \leq x_{te} \leq 1$ and any fractional solution of the ILP is decomposed into several paths. Each path is associated with a fractional weight so that the sum of the weights of the fractional paths for each multicast is 1 and the sum of the weights of the fractional paths crossing the edge e corresponds to the weight of that edge.

In the case for each multicast consisting of more than two nodes, the LP relaxation was described in terms of a multicommodity flow between pairs of nodes of each multicast.

Let the binary variable $f_t(i, j)$ denote the flow between the nodes i and j in the t^{th} multicast and the $x_{te}(i, j)$ denote the flow on edge e of commodity (i, j) in the t^{th} multicast, the problem is formulated as the following ILP:

Minimize z

Subject to

$x_{te}(i, j) \in f_t(i, j), \quad \forall\, t, \forall\, i < j \in S_t$

$\sum_{i\in S\cap S_t, j\in(V\backslash S)\cap S_t} f_t(i, j) \geq 1, \quad \forall t, \forall S \subset V$

$\sum_{t,i,j} x_{te}(i, j) \leq z, \quad \forall e \in E$

$0 \leq x_{te} \leq 1$

An iterative randomized rounding algorithm was proposed as follows by Vempala and Vöcking:

Step 1. Decompose the fractional solution into flow paths.

Step 2. Choose one path randomly out of each multicast node with probability equal to the value of the flow on the path, i.e., randomized rounding.

Step 3. If the multicast nodes are all connected then stop and output the solution, otherwise, contract the

vertices corresponding to the connected components and form a new multicast problem with regarding the contracted vertices as the new multicast nodes.
Step 4. The original fractional solution was derived from the solution of the new multicast problem decomposed from the fractional solution. Repeat the above steps till all multicast nodes are connected.

In this way, there are at most $log\ k$ iterations (k is the maximum number of nodes in a multicast). With high probability, the congestion of the solution found by the algorithm has an approximation bound less than $O(log\ k \cdot OPT + log\ n)$.

2.2 The General Randomized Rounding Algorithm

Randomization is a powerful technique in finding approximate solutions to difficult problems in combinatorial optimization by solving a relaxation (usually linear programming relaxations or semidefinite programming relaxations) of a problem and then using randomization to return from the relaxation to the original optimization problem.

De-randomization can be applied by using standard techniques to yield deterministic polynomial-time algorithms that yield approximations as good as those given by the randomized algorithms they are derived from, even though the process of de-randomization typically takes a relatively simple and clean randomized rounding procedure and turns it into a complex and generally slower deterministic algorithm.

In [8], Carr and Vempala proposed a general randomized rounding algorithm in polynomial time for constructing a convex combination based on the ellipsoid method.

Assume that a polynomial-time algorithm $\mathcal{A}$ is an r-approximation algorithm to a min-ILP problem with the LP relaxation $\mathcal{P}$ then $\mathcal{A}$ finds a solution with the cost being at most r (integrality gap) times the cost of the optimal solution to the LP relaxation $\mathcal{P}$ of the ILP. A min-ILP or max-ILP problem is a minimization or maximization problem, respectively, whose set of feasible solutions can be described by a positive ILP.

Let $\mathcal{P}$ denote a LP relaxation of ILP by employing the r-approximation algorithm $\mathcal{A}$, x^* denotes a feasible solution of $\mathcal{P}(I)$, Z denote an integer polytope, and P denote a LP relaxation of Z, then according to [9], $r \cdot x^*$ dominates a convex combination of extreme points of $\mathcal{Z}(I)$.

Also let I denote a min-ILP problem, $Z(I)$ denote the integer polyhedron for the ILP, $P(I)$ denotes the LP relaxation of $Z(I)$, x^j denote an extreme point of $\mathcal{Z}(I)$, x^* denote a feasible solution of $\mathcal{P}(I)$, and $ext(P)$ denote the set of extreme points for a polyhedron, the problem of the LP relaxation $\mathcal{P}$ of a positive ILP along with an r-approximation algorithm is formulated as follows:

$$r \cdot x^* \geq \sum_j \lambda_j \cdot x^j \quad (1)$$

where $\sum_j \lambda_j = 1, \quad \lambda_j \geq 0$, for $\forall\ j$.

In order to construct a set of x^j's which satisfies (1), let x^* denote a feasible solution for $P(I)$, $ext(Z) = \{x^j | j \in J\}$, E denote the index set for the variables in $P(I)$, and x^c denote the solution for each non-negative object function c returned by the r-approximation algorithm $\mathcal{A}$, the problem is formulated as the following ILP and solved in order to obtain (1):

Maximize $\sum_{j \in J} \lambda_j$
subject to
$\sum_{j \in J} \lambda_j \cdot x_e^j \leq r \cdot x_e^*, \quad \forall e \in E \quad (2)$
Maximize $\sum_{j \in J} \lambda_j \leq 1$
$\lambda_j \geq 0, \quad \forall j \in J$

The solution λ^* of (2) provides an explicit convex decomposition into points in $ext(Z)$, i.e., $r \cdot x^* \geq \sum_{j \in J'} \lambda_j^* \cdot x^j$, $J' := \{j \in J | \lambda_j^* \geq 0$. The summation in this inequality is a linear combination of $\{x^j | j \in J'\} \subset ext(Z)$ dominated by rx^* and it is a constructed convex combination if $\sum_{j \in J'} \lambda_j^* = 1$.

Obviously, (2) has an exponential number of variables and in order to obtain an approximate bound on its dual ILP as follows, it was further solved by employing the r-approximation algorithm:

Minimize $r \cdot x^* \cdot w + z$
subject to
$x^j \cdot w + z \geq 1, \quad \forall j \in J \quad (3)$
$w_e \geq 0, \quad \forall e \in E$
$z \geq 0$

(3) also has an exponential number of variables and was solved in polynomial time by using the ellipsoid method and has an optimal solution of 1.

3 The Multicast Packing Problem

In the Multicast Packing Problem, there are a number of applications which try to use the network for the purpose of establishing connection and sending information, organized in different groups. Thus, the network capacity must be shared accordingly with the requirements of each group based on known heuristics for constructing Steiner trees and the cut-set problem or using integer programming to find the minimum cost under bounded tree depth and the cost minimization under bounded degree for intermediate nodes [3, 1, 2].

In [19], Noronha and Tobagi proposed an efficient solution by employing decomposition principle to speed up LP of the problem and by enhancing value-fixing rule to prune the search space of the IP. Ofek and Yener [11] presented a window-based reliable multcast protocol with a combined sender and receiver initiation of the recovery protocol in order to combine the multicast operation with the internal flow control. They proposed two Min-Max objective functions: one for delay which is caused by the number of links needed to connect the multicast group; the other for congestion which is caused by sharing a link among multiple multicast groups. In [20], Grötschel, Martin, and Weismantel considered the Steiner tree packing problem from polyhedral point of view, called polyhedron, which can be described by means of inequalities that define the facets of the Steiner tree polyhedrons. They presented joint-inequalities for these polydrons based on cut-and-branch algorithm. Baldi, Ofek and Yener [21] proposed an approach based on the embedding of multiple virtual rings, one for each multicast group, to route messages to all the participants of multicast group while minimizing the bound on the buffer sizes and queueing delays so as to resolve two problems raised from the time-driven priority flow control scheme proposed in [22]: one is the scheduling problem in which how time intervals are reserved to each multicast group and the other is the adaptive sharing problem in which how the active (transmitting) participants can dynamically share the time intervals that have been reserved for their multicast group.

In this section, we discuss the multicast packing problem including the objective functions and approximation algorithms for minimizing the maximum link sharing among multicast groups (i.e., the congestion over all edges). In the multicast packing problem, the minimization of network congestion is defined as the total load of the most congested edge and the load (or congestion) of an edge is the total traffic demand summed over the multicast groups using that edge so as to prevent bottlenecks and thus increase utilization (or throughput).

Given a physical network $G = (V, E)$ where V is the set of nodes and E is the set of the undirected network links, a weight $w_e > 0$ (such as cost, delay, and distance) is associated with each link $e \in E$.

Let K denote the set of multicast groups, a set of nodes $M \subseteq V$ denote a multicast group, and m_i denote any arbitrary member of M, the objective of the multicast packing problem is to find a subgraph of G that spans M and has the minimum total cost. The subgraph is required to be a tree and the cost is measured as the sum of the weights of the edges in the solution. Since the objective of the multicast packing problem is to minimize the maximum sharing of a link instead of individual multicast tree cost, the solutions might produce high-cost multicast trees. In order to bound the cost of each multicast tree to guarantee service quality for some cases, the size of some trees may be increased. Assume that, for simplicity, each link has the same cost such as unit cost and the cost of a multicast tree is the total number of its links.

Given $k \in K$ (k is a multicast group and K is the set of multicast groups), Let OPT^k denote the cost of the least-cost multicast tree, if the ratio of the cost of the multicast tree in the solution to the cost of the least-cost multicast tree OPT^k exceeds a threshold $\alpha \leq 1$, then a cost based on the dilation from the size of the optimal tree obtained for each group multicast independently from the others is incurred.

Also given a link $e \in E$, let $K_e \subseteq K$ denote the set of multicast groups that use link e in the solution, z_e denotes the total congestion on link e such that $z_e = \sum_{k \in K_e} t_k$ where t_k is the amount of traffic generated by the multicast group k, $\lambda = max_e\{z_e\}$ denote the maximum congestion, a binary variables x_e^k for all $e \in E$ and $k \in K$, t^k denoting a traffic load of multicast group $k \in K$, $ST^k = \{x^k \in \{0,1\}^{|E|} : x^k$ induces a Steiner tree spanning M^k, $\sum_{e \in E} w_e \cdot x_e^k$ denote the cost of the multicast tree in the solution for multicast group $k \in K$, OPT^k denote the size of the optimal tree for multicast group k in isolation, $P^k \geq 0$ denote the cost coefficients, and $\pi^k \geq 0$ used to measure the threshold, in [10], the tree packing problem was formulated by Chen, Günlük, and Yener as the following ILP:

Minimize $\lambda(\lambda = max_e\{z_e\}) + \sum\limits_{k \in K} P^k \cdot \pi^k$

Subject to

$x_e^k \in ST^k, \quad \forall k \in K$

$\sum\limits_{k \in K} t^k \cdot x_e^k \leq \lambda, \quad \forall e \in E$

$\pi^k \geq \frac{1}{OPT^k}(\sum\limits_{e \in E} w_e \cdot x_e^k - (\alpha \cdot OPT^k)), \quad \forall k \in K$

3.1 The Multicast Packing Heuristic Algorithm

An approximation algorithm considering each multicast in isolation and taking the set of optimal multicast trees computed independently was proposed by Chen, Günlük, and Yener for packing multicast trees with minimum congestion is as follows:

Step 1. Solve the optimization problem for each multicast group independently.

Step 2. Compute the congestion for each edge and rebuild the multicast trees T' based on the results from step 1: the tree $T = \{T_x : x = 1, 2, 3, ..., m\}$ and a bound on the tree size $\alpha \cdot OPT^k$.

Step 3. Sort all edges e by z_e (the total congestion on link e) into an array in decreasing order.

Step 4. Choose an edge e' with the maximum congestion Z

Step 5. try each tree $T_x \in T$ until a new tree T_y can be found such that each edge in T_y has the congestion value no greater than $Z-1$ and tree T_y can be rebuilt without exceeding the size limit α.

Step 6. Update z_e values in the array and go to step 3. Stop if the array cannot be updated.

In [11], Ofek and Yener proposed a way to rebuild the above disconnected tree by constructing a new tree from scratch that does not use the congested link because it may cause an increase in the congestion on some links that are already used by the other multicast trees. Thus the links to be used in the new tree T_y can have a congestion value at most $Z-2$ (Z is the maximum congestion).

Chen, G$\ddot{u}$nl$\ddot{u}$k, and Yener [10] proposed a more efficient algorithm considering the cut obtained by removing the congested link e' from the new tree T_y. If there is a link e with congestion value at most $Z-2$ in this cut set, then e can be inserted into the tree to rebuild the new tree ($T_y - e' + e$). The cost of rebuilding the tree is bounded by the cardinality of cut set (i.e., $O(|E|)$) and thus the total cost is $O(m \cdot |E|^2)$ for m trees. An ILP formulation employing Steiner-cut inequalities (branch-and-cut algorithm) [17, 18] to construct such trees was as follows:

Minimize $z = \sum_{e \in E} w_e \cdot x_e$

Subject to

$$\sum_{e \in \delta(S)} x_e \geq 1, \quad \forall S \subset V, m_i \in S, M \not\subset S$$

$$x_e \in \{0, 1\}, \quad \forall e \in E$$

Let a partition $P = S_1, S_2, S_3, ..., S_k$ of V denote a Steiner partition with respect to M if each S_i contains at least one element of M and $\Delta(P)$ denote the multicut associated with P, (i.e., the collection of edges with endpoints in different members of P), they extended the above formulation as the following ILP by defining the Steiner partition inequality associated with partition P, partitioning recursively the solution spaces by branching on the variables to 0 or 1, and by solving the relaxations of the following ILP formulation in order to find the optimum multicast tree in isolation instead of generating all of the constraints at once :

$$\sum_{e} \in \Delta(P) \cdot x_e \geq k-1$$

$$1 \geq x_e \geq 0, \quad \forall e \in E$$

They further extended the above idea to partitions of V involving more than two subsets.

3.2 The Randomized Version of Partition Algorithm

A randomized algorithm contains some decision that is based on pure chance, not the inputs to the algorithm, or anything else in the environment in which the algorithm is executed. The algorithm may determine how the result is computed. The partition problem is an NP-complete problem and solving that given a set of integers, is there a way to divide the set into two independent subsets such that the sums of the numbers in each subset are equal.

Let $\Pi = \{S_1, S_2, ..., S_{|\Pi|}$ denote a partition of V, $\Delta(\Pi)) \subseteq E$ denote the associated multicut, the edge weights defined as $\bar{w}_e = \frac{1}{|\Delta(\Pi)|}$ if $e \in \Delta(S)$ or 0, otherwise, $\Lambda^k(w) \geq \frac{1}{|\Delta(\Pi)|} \cdot t^k \cdot (s^k(\Pi) - 1)$ by using these weights, in order to improve the quality of the partition Π, Chen, G$\ddot{u}$nl$\ddot{u}$k, and Yener also proposed a constructive procedure to find better partitions of V by merging some of the subsets in the partition to obtain smaller partitions and looking for a promising pair of subsets S_m and S_n with small β_{mn}. By using edge weights γ defined as $\gamma_{mn} = \beta_{mn} + u \cdot (\sum_{k \in K(S_m)} t^k + \sum_{k \in K(S_n)} t^k) + v$ with u and v being two random perturbations, they proposed a randomized version of partition algorithm that was applied repeatedly starting with the same initial partition as follows:

Step 1. Let $\Pi_0 = S_1, S_2, \cdots, S_{|V|}$ with S_i being a singleton.

Step 2. Let λ_0 be the initial best bound and $i = 0$.

Step 3. Repeat as long as $|\Pi_i| > 2$:

Step 3.1 compute γ 's for each neighboring subset pair

Step 3.2 identify a pair (S_m, S_n) with the least γ_{mn} (based on two random perturbations u and v)

Step 3.3 merge S_m and S_n to obtain a new partition Π_{i+1}

Step 3.4 compute λ_{i+1}, updated the best bound if necessary, and set $i = i + 1$

4 Discussion

The multicast congestion problems are critical to many network applications in multimedia streamlining such as multimedia distribution, software distribution, and video-conference; groupware system; game communities; and electronic design automation such as routing nets around a rectangle and moat routing. As widely known, the generalization of the problem for finding edge disjoint paths is an NP-hard combinatorial problems, i.e., minimum multicast Steiner tree problem, and also a max-SNP hard problem, i.e., there is a constant $\epsilon > 0$ such that it is NP-hard to find a $(1 + \epsilon)$ approximation.

All the approximation algorithms discussed above have been developed based on the deep analysis of the problems from different points of views, which thus have been formulated as a variety of ILPs and solved by employing different schemes. However, the main problem with the LP-relaxation is the time required to solve the LP formulation. To improve the efficiency, it is mandatory to make better use of the multicast congestion parameters in order to obtain a simplified approximation algorithm with tighter approximation bound. Finding better and fast approximation algorithms for specific classes of multicast networks is worth further investigation as well.

Acknowledgement

This work was supported in part by NSC, Taiwan under grant no. NSC 94-2213-E-274-004.

References:

[1] Priwan, V., Aida, H., and Saito, T., ”The Multicast Tree Based Routing for the Complete Broadcast Multipoint-to-multipoint Communications,” *IEICE Transactions on Communications*, E78-B(5), pp. 720-728, 1995.

[2] Chen, S., GAunlAuk, O., Yener, B., ”Optimal Packing of Group Multicastings,” *Proceedings of IEEE INFOCOM’98*, pp. 980-987, 1998.

[3] Wang, C.-F., Liang, C.-T., and Jan, R.-H., ”Heuristic Algorithms for Packing of Multiple-group Multicasting,” *Computers and Operations Research*, pp. 905-924, 2002.

[4] Karp, R. M, ”Reducibility among combinatorial problems,” *Complexity of Computer Computations*, R. E. Miller, J. W. Thatcher, Eds., Plenum Press, New York, pp. 85-103, 1972.

[5] Bern M., and Plassmann, P, ”The Steiner problem with edge lengths 1 and 2,” *Information Processing Letters*, 32, pp. 171-176, 1989.

[6] Khanna, S., Motwani, R., Sudan, M., and Vazirani, U., ”On syntactic versus computational views of approximability,” *Electronic Colloquium on Computational Complexity*, TR95-023, 1995.

[7] Vempala, Santosh and Vöcking, Berthold, ”Approximating multicast congestion,” *ISAAC Proceedings*, pp. 367-372, 1999.

[8] Carr, R. and Vempala, S., ”Randomized metarounding,” *Proceedings of the 32nd STOC*, pp. 58-62, 2000.

[9] Carr, R. and Vempala, S., ”Towards a 4/3 approximation for the asymmetric traveling salesman problem,” *proceedings of the 11th SODA* , pp. 116-125, 2000.

[10] Chen, S., Günlük, O., and Yener, B., ”The multicast packing problem,” *IEEE/ACM Transactions on Networking*, 8(3), pp 311-318, 2000.

[11] Ofek, Y. and Yener, B., ”Reliable concurrent multicast from bursty sources,” *IEEE Journal Selected Areas Communication*, 14, pp. 434-444, 1997.

[12] Ballardie, T, Francis, P., and Crowcroft, J., ”Core based trees (cbt): An architecture for scalable inter-domain multicast routing,” *Proceedings of ACM SIGCOMM*, pp. 85-95, 1993.

[13] Deering, S. E. and Cheriton, D. R., ”Multicast routing in datagram internetworks and extended LANs,” *ACM Transaction on Comp. Syst.*, 8(2), pp. 85-110, 1990.

[14] Deering, S., Estrin, D., Farinacci, D., Jacobson, V., Liu, C.-G., and Wei L., ”An architecture for wide-area multicast routing,” *Proceedings of ACM SIGCOMM*, pp. 126-135, 1994.

[15] Patridge, C. and Pink, S., ”An implementation of the revised internet stream protocol (st-2),” *Journal of Internetworking: Research and Experience*, 4, 1992.

[16] Paul, S., Sabnani, K., and Kristol, D., ”Multicast protocols for high speed networks”, *Proceedings of IEEE ICNP*, pp. 4-14, 1994.

[17] Gunluk, O., ”A branch-and-cut algorithm for capacitated network design problems,” *Math. Program*, 86(1), pp. 17-39, 1999.

[18] Padberg, M. and Rinaldi, G., ”Optimization of a 532-city traveling salesman problem by branch-and-cut,” *Oper. Res. Lett.*, 6, 1987.

[19] Noronha, C. A. and Tobagi, F. A., ”Optimum routing of multicast streams,” *IEEE INFOCOM*, pp. 865-873, 1994.

[20] Grötschel, M., Martin, A., and Weismantel, R., ”Packing Steiner trees: Polyhedral investigations,” *Math. Program*, 72, pp. 101-123, 1996.

[21] Baldi, M., Ofek, Y., and Yener, B., ”Adaptive Group Multicast with Time-Driven Priority,” *IEEE Transactions on Networking*, 8(1), pp 31-43, 2000.

[22] Li, C.S., Ofek, Y., and Yung, M.,””Time-driven priority” flow control for real-time heterogeneous internetworking,” *Proceedings of IEEE INFOCOM*, pp. 189-197, 1996.