1998 International Computer Symposium
Workshop on Computer Networks, Internet, and Multimedia
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

Multicast Routing with Multiple QoS Constraints Based on Genetic Algorithms

Jang-Jiin Wu and Ren-Hung Hwang

Department of Computer Science & Information Engineering
National Chung-Cheng University, Chia-Yi, Taiwan, R.O.C.
Email: {wjj85,rhhwang}@cs.ccu.edu.tw

ABSTRACT

Recently, more and more applications provide multiparty communication services, e.g., video conferencing, distance learning, etc. Therefore, the routing problem of multicast with multiple QoS constraints becomes more important. For example, in order to ensure smooth play back of audio and video data, a video conference requires guarantee on both end-to-end delay and loss probability. In this paper, we propose a novel multiple-constraint multicast routing algorithm based on genetic algorithms, called MCMGA. Our numerical experiments show that MCMGA yields very good performance.

1. Introduction

As World Wide Web gradually becomes world wide wait, the demand for high bandwidth and Quality of Service (QoS) guarantee received significant attention. Asynchronous Transfer Mode (ATM) has been recognized as the most promising solution. Many features of ATM have made it the unique solution for transmission multimedia traffic. For example, the cell switching technique, the ability of providing Quality of Service guarantee, and flexible bandwidth allocation all make it suitable for integrating multiple classes of multimedia traffic with different bandwidth requirement and QoS guarantee.

Many multimedia applications require multicast support. For example, multiparty video conferences, distance learning, and selected video broadcast all require multicasting. In the past, the multicast routing problem has been formulated as the minimum cost multicast tree problem, i.e., the Steiner tree problem[1], which is well-known to be NP-complete [2].

While total tree cost as a efficiency measure is an important parameter, networks supporting multimedia applications are also required to provide certain quality of service guarantees, e.g., the end-to-end delay along the individual paths from the source to each of the destination nodes. The problem of routing multicast traffic with one QoS constraint, referred to as the constrained multicast problem, has been extensively studied in the literatures [3-5]. Heuristics are developed to compute low-cost multicast trees which guarantee an upper bound on the end-to-end delay.

However, multimedia applications involved in real-time applications require multiple classes of quality of service guarantee. For example, video and audio conferences require QoS guarantees on both end-to-end delay and loss probability to provide a smooth play-out at the receiver. If the delay is guaranteed, but the loss probability constraint is not, important video data may be lost, resulting a degradation of play-out quality. If the network only guarantees the loss probability, but not the delay, video cannot be played smoothly. In the past research, the multicast problem with multiple QoS constraints does not received much attention. For considering two QoS constraints, a heuristic for constructing a multicast tree with end-to-end delay and delay variation constraints has been developed in [6]. However, [6] did not attempt to optimize the multicast tree in terms of cost (bandwidth consumption). In this paper, we study the heuristics for finding low-cost multicast trees that satisfy multiple QoS constraints.

In this paper, we solve the multiple-constraint multicast problem based on genetic algorithms. For performance comparison, we also propose a simple heuristic algorithm, referred to as the constrained shortest path tree (CSPT) algorithm. Our numerical experiments show that MCMGA yields much better performance than the CSPT algorithm.

The remainder of this paper is organized as follows. Section 2 presents our network model and defines the multicast routing problem. Section 3 describes the proposed MCMGA algorithm. The CSPT algorithm is described in Section 4. The two multicast routing algorithms are evaluated in Section 5. Section 6 concludes this paper and discusses future work.

2. Network Model

An ATM network is represented as a directed graph $G=(V, E)$ consisting of a set of switches, V, and a set of directed links, E. Let the link from node i to nod j be denoted by $e(i, j)$. Each link $e \in E$ is associated with a cost $C(e)$ and several QoS parameters, such as delay, loss probability, and jitter. The cost function, $C(e)$ is a positive real function, i.e., $C: E \rightarrow R^+$. The QoS supported on a link is described by QoS functions. Each QoS function, $Q_i(e)$, is a positive real function which gives the quality of the parameter that can be guaranteed on the link e.

For a multicast connection, packets originating at the source node $s \in V$, have to be delivered to a set of destination nodes $M \subseteq V - \{s\}$. We refer to M as the *destination set* or *destination group*, and $\{s\} \cup M$ the *multicast group*. Multicast packets are routed from the source to the destinations via the links of a multicast tree $T = (V_T, E_T)$. A multicast tree is a subgraph of G spanning s and the nodes in M. In addition, V_T may contain relay nodes, that is, nodes in the multicast tree but not in the multicast group.

The QoS guarantee for a multicast connection is defined as follows. Let Q_1, ..., Q_n be the n QoS functions and Δ_1, ..., Δ_n be the corresponding QoS constraints that need to be satisfied. A multicast tree is said to be able to provide the required QoS guarantee if the end-to-end QoS of each source-destination pair of the multicast connection is satisfied. In this paper, we only consider QoS parameters that are additive, i.e., the end-to-end QoS of a path is the sum of individual QoS of each link on the path. QoS parameters such as delay and jitter are additive in nature. The end-to-end loss probability of a path can be approximated by the sum of loss probabilities of all links on the path if the link loss probability is very small. Formally, for each $v \in M$, the end-to-end QoS is guaranteed by

$$\sum_{e \in P(s,v)} Q_i(e) \le \Delta_i, \ \forall v \in M, \ i=1, \ldots, n,$$

where $P(s, v)$ is the path in T from s to v.

The multiple-constraint multicast routing problem is defined as follows:

$$\min_T \sum_{e \in T} C(e),$$

$$\text{s.t.} \quad \sum_{e \in P(s,v)} Q_i(e) \le \Delta_i, \ \forall v \in M, \ i=1, \ldots, n.$$

3. Multiple-Constraint Multicast Routing Algorithm based on Genetic Algorithms (MCMGA)

In this section, we present a novel multicast routing algorithm with multiple QoS constraints based on genetic algorithms (GAs). We will refer to this algorithm as the Multiple-Constraint Multicast routing algorithm based on Genetic Algorithms (MCMGA).

3.1 Genetic Algorithms

Genetic algorithms are used to solve optimization problems based on the principle of evolution. A population of candidate solutions, called chromosomes, is maintained at each iteration of the evolution. Each chromosome consists of linearly arranged genes which are represented by binary strings. Three basic operations, reproduction, crossover, and mutation, are adopted in the evolution to generate new offspring. Reproduction is based on the Darwinian survival of the fittest among strings generated. Chromosomes with larger fitness function values are selected to generate new offspring bit strings by crossover operations and convert the offspring to new parameter solutions. Crossover is used to cut individually two parent bit string into two or more segments and then combine the segments undergoing crossing over to generate two offspring bit strings. Crossover can produce offspring that are radically different from their parents. Mutation is to perform random alternation on bit strings by some operations, such as bit shifting, inversion, rotation, etc. The mutation operation will create new offspring bit strings different from those generated by reproduction and crossover operations. Mutation can extend the scope of the solution space and reduce the possibility of falling into local extremes.

3.2 The MCMGA Algorithm

3.2.1 Routing table

In the network graph, $G=(V, E)$, there are $|V| \times (|V|-1)$ possible source-destination pairs. There are usually many possible routes between any source-destination pair. Our MCMGA algorithm assumes that a routing table, consist of R possible routes, has been constructed for each source-destination pair using the K-shortest-path algorithm proposed in [7]. The size of the routing table, R, is a parameter of our algorithm.

3.2.2 Representation of chromosomes

For a given source node s and a destination set $M = \{m_1, m_2, \ldots, m_k\}$, a chromosome can be presented by a string of integers with length k. A gene, g_i, $1 \le i \le k$, of the chromosome is an integer in $\{1, 2, \ldots, R\}$ which represents a possible route between s and m_i, where $m_i \in M$.

Obviously, a chromosome represents a candidate solution for the multicast routing problem since it guarantees a path between the source node to any of the destination nodes. However a chromosome does not necessarily represent a tree. Therefore, we trim the extra edges using a minimum directed spanning tree algorithm, modified from the Optimum Branchings Algorithm proposed in [8].

3.2.3 Description of the algorithm

The MCMGA algorithm maintains a population of chromosomes, each of which has a fitness value. The fitness value defines the quality of the chromosome. Beginning with a set of random chromosomes, a process of evolution is simulated. The main components of this process are reproduction, crossover, and mutation. After a number of generations, highly fit chromosomes will emerge corresponding to good solutions. The MCMGA algorithm stops when a pre-defined number of iterations is encountered.

The outline of our MCMGA algorithm, schematically illustrated in Figure 1, is given as follows.

3.2.3.1 Initialization of chromosomes

The initial procedure generates P different chromosomes at random which form the first generation. The set of chromosomes is called the chromosome pool (or population), and P is the size of the gene pool.

3.2.3.2 Evaluation of chromosomes

The fitness value of a chromosome is the value of the objective (fitness) function for the solution (e.g. a multicast tree) represented by the chromosome. Given a chromosome pool $H = \{h_0, h_1, \ldots, h_{P-1}\}$, the fitness value of each chromosome is computed as follows. Let $C(h_i)$ be the overall link cost of the graph represented by the chromosome h_i. Let $C(L)$ be the sum of the costs of all links of the network. The fitness value of the chromosome h_i, $F(h_i)$, is given by

$$F(h_i) = \begin{cases} 1 - \dfrac{C(h_i)}{C(E)} & \text{if } \sum_{e \in P(s,v)} Q_i(e) \le \Delta_i \quad \forall v \in M, i = 1,2,\ldots,n, \\ 0 & \text{otherwise} \end{cases}$$

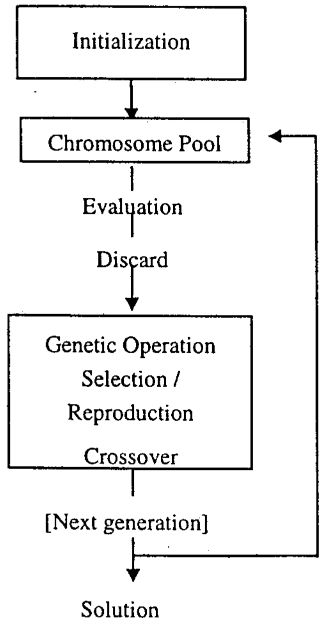

Figure 1: The MCMGA algorithm.

After evaluating the fitness values of all chromosomes, chromosomes are then sorted according to their fitness values such that $F(h_0) \ge F(h_1) \ge \ldots \ge F(h_{P-1})$. (That is, the first chromosome in the pool is the best solution found so far.)

3.2.3.3 Discard duplicate chromosomes

The might be duplicated chromosomes in the pool. Apply some of the genetic operations, e.g. crossover, on two duplicate chromosomes will yield the same offspring. Therefore, too many redundant chromosomes will reduce the ability of searching. Once this situation occurs, the redundant chromosomes must be discarded. In our algorithm, they are replaced by new randomly generated chromosomes.

4. The CSPT Algorithm

In this section, a simple multiple-constraint multicast algorithm, referred to as the constrained shortest path tree (CSPT) algorithm, is introduced. The CSPT algorithm first finds an unconstrained minimum cost tree by the TMR method which is an extension of the algorithm proposed by Takahashi and Matsuyama [9]. If the end-to-end QoS of a path violates one of the constraints, the path is replaced by a "*guaranteed path*". A guarantee path is a path which satisfies all of the QoS constraints. Any shortest-path algorithm, such as Dijkstra's algorithm [10], can be used to find a guaranteed path under one QoS constraint by using the QoS function as the link cost function. However, it becomes an NP-complete problem if there are more than one QoS constraint [11]. We solve this problem by choosing one of the QoS parameters as the major QoS constraint, denoted by Q_1. Using Q_1 as the link cost function, a loop-free *K*-shortest path algorithm (modified from DOUBLE-SWEEP algorithm [7]) is used to find *K* paths which have the best end-to-end QoS in terms of Q_1. A guaranteed path is then chosen from these *K* paths such that all QoS constraints are satisfied while the end-to-end Q_1 is minimized. Figure 2 shows the CSPT algorithm.

5. Numerical Results

In this section, performance of the two multicast routing algorithms is evaluated via simulations under various random graphs. The random graphs used in the experiments are constructed to resemble real-world networks using the method proposed by Waxman [12]. Nodes of a graph are randomly placed on a coordinate grid. For a given pair of nodes, (u, v), the probability that there exists a direct link between them is given by

$$P(u,v) = \lambda \exp\left(\frac{-d(u,v)}{\rho L}\right),$$

```
/*
 * Let G=(V, E) describe the network topology.
 * s = source node
 * M = destination set
 * Δ = QoS constraints
 * n = number of QoS parameters
 */
CSPT_algorithm (G, s, M, Δ)
begin
  /* begin with the TMR method */
  Let T_0=(V_0, φ) where V_0 = s.
  for i=1 to |M| do
  begin
    Find a node v∈ M\V_{i-1} such that the cost from T_{i-1}
      to v is minimal. The cost from T_{i-1} to v is the
      minimal cost of any path of T_{i-1} to v. Construct the
      tree T_i = (V_i, E_i) by adding to T_{i-1} the nodes
      and edges of the shortest path that connects T_{i-1} to
      v.
  end
  T_tmr = T_{|M|}
  Q_1 ← choose one of the QoS parameters
  for each v ∈ M do
  begin
    if the path from s to v in T_tmr violates one of the
constraints then
    begin
      Find K shortest paths from source s to v using Q_1
    as the link cost function.
      /* Find a guaranteed path in K shortest paths from
         s to v to replace the un-guaranteed path */
      for i = 1 to K do
      begin
        if ( the i-th path of the K shortest paths from s to
             v satisfies all QoS constraints ) do
        begin
          Replace the path from s to v in T_tmr by the
             i-th shortest path
          break /* for */
        end /* if */
      end /* for */
    end /* if */
  end /* for */
end /* CSPT algorithm */
```

Figure 2: The CSPT algorithm

where $d(u, v)$ is the distance between u and v, and L is the maximum distance between two nodes in this graph. The parameters λ and ρ are chosen from the range of (0, 1). They are set in such a way that the average degree of the resulting network is around five. A 20-node random graph with an average degree of four is shown in Figure 3. The average degrees of the graphs used for simulations are listed in Table 1. The cost of each edge was uniformly distributed between [1, 10]. In our simulations, we consider two kinds of QoS constraints: end-to-end delay and loss probability. Both delay, Q_1, and loss probability, Q_2, of an edge are set to a discrete uniform random variable with range [1..10] (due to the limit of the KPP algorithm).

Number of nodes	Average degree
20	3.61
40	5.36
60	5.74
80	5.9
100	5.8

Table 1: Average degrees for the networks in the simulations.

5.1 MCMGA Parameter Setup

Parameters of the MCMGA algorithm, such as the size of chromosome pool, mutation rate, and the number of generations, must be properly selected to yield the best performance.

We first examine the effect of the size of chromosome pool on the performance of the MCMGA algorithm. Figure 4 shows the minimum cost obtained after 200 generations with different chromosome pool sizes under 60-node random graphs where the size of the destination group is set to be 20% of the graph nodes. The mutation rate is set to 0.4. From Figure 4, we can observe that after 200 generations, the chromosome pool size of 80 yields the best performance.

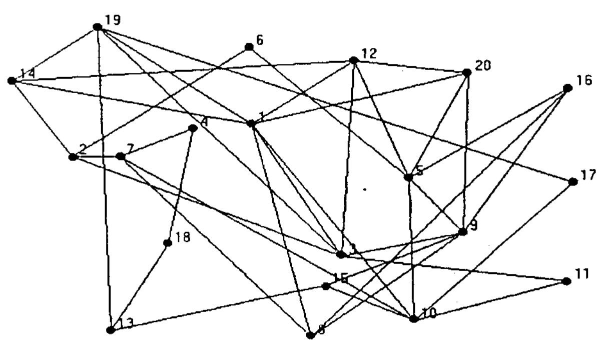

Figure 3:A 20-node random graph with an average degree of four.

Figure 5 shows the effect of mutation rate on the performance of MCMGA. It shows the cost under various mutation rates when the chromosome pool size is set to 80. We can observe that MCMGA yields the best performance when mutation rate is set to 0.4.

Figure 6 shows the effect of the number of generations and the size of routing table on the performance of the algorithm. The chromosome pool size is set to 80 and the mutation rate is set to 0.4. The five curves in Figure 6 show the minimum cost obtained at each generation when the sizes of routing table are set to 40, 60, 80, 100, 120, respectively. One might expect that larger number of routes in routing table leads to better performance, because more diverse routes available in a large routing table can prevent the genetic algorithm from falling into the local optimum. However, we have observed that the smaller the routing table the better the performance. This may due to the slow convergent rate of larger routing table size. That is, the process has not converged to the optimal solution yet. In our later simulations, we set the size of routing table to 40. The number of generations is set to 250 since the cost does not decrease much after that.

5.2 Comparison of Algorithms

In this section, we evaluate the performance of the proposed multicast algorithms through simulations. The average degrees of random graphs we used are shown in Table 1. Each point in following figures is an average of 1000 simulation results from 10 random graphs. Furthermore, group members are randomly selected from the graphs.

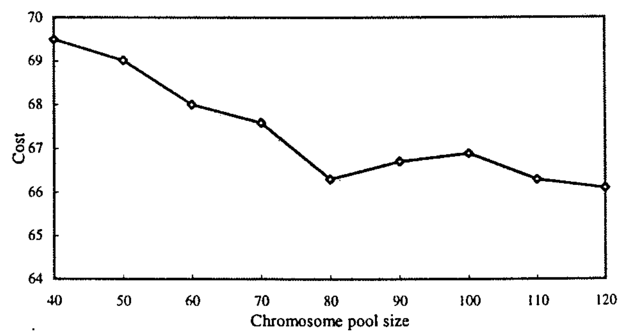

Figure 4: The effect of the chromosome pool size on the performance of MCMGA (200 generations, 60-node graphs).

In our experiments, the delay bound, Δ_1, and loss bound, Δ_2, are set as follows [4, 13]. For each multicast request, we first construct three multicast trees: an unconstrained minimum cost Steiner tree, Z, via the TMR algorithm [9], a shortest path multicast tree, X, via Dijkstra's algorithm [10] using delay as the link cost, and a shortest multicast tree, Y, using loss probability as the link cost. Let the maximum end-to-end delay in Z, X, and Y to be $d_{max}(Z)$, $d_{max}(X)$, and $d_{max}(Y)$, and the maximum end-to-end loss probability to be $\ell_{max}(Z)$, $\ell_{max}(X)$, and $\ell_{max}(Y)$, respectively. We define a delay-loss bound, ρ, as follows:

$$\rho = \frac{\Delta_1 - d_{max}(X)}{d_{max}(Z) - d_{max}(X)},$$

$$\rho = \frac{\Delta_2 - \ell_{max}(Y)}{\ell_{max}(Z) - \ell_{max}(Y)}.$$

As ρ becomes larger, the delay bound and the loss bound become looser. Figure 7 shows the performance of two multicast algorithms under various values of ρ on a 100-node random graph, where the number of destinations is 20. From Figure 7, we see that the MCMGA algorithm outperforms the CSPT algorithm. We also observe that as the delay bound and the loss bound become tighter, the performance difference between them also becomes more significant. For the rest of our simulations, ρ is always set to 0.5 [4].

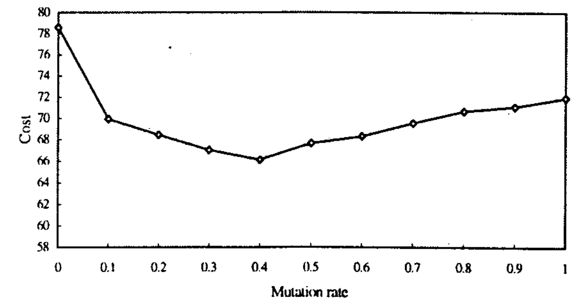

Figure 5: The effect of mutation rate on the performance of MCMGA (200 generations).

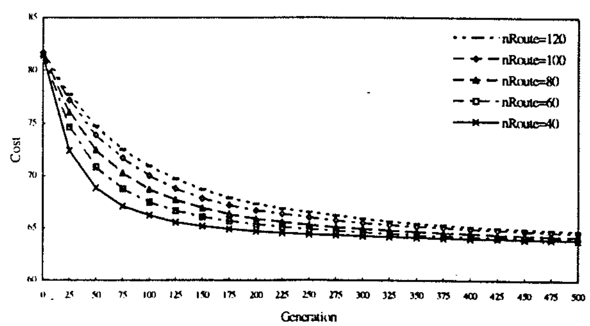

Figure 6: The effect of number of generation and routing table size (60-node graphs, chromosome pool size=80, and mutation rate=0.4).

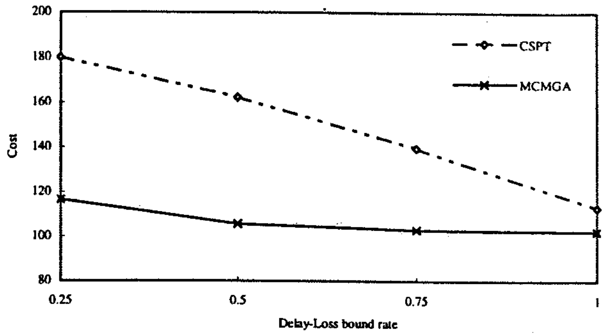

Figure 7: Performance of two multicast algorithms under various delay-loss bounds (100-node graphs, 20 destination nodes).

Figure 8 and 9 compare the performance of the two algorithms under different numbers of multicast group members and different sizes of network. Again, the MCMGA algorithm outperforms the CSPT algorithm significantly.

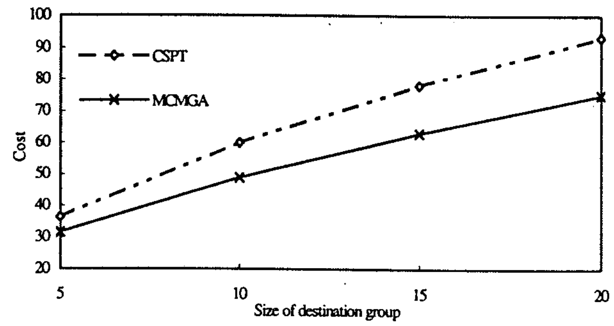

Figure 8: Performance comparison of two multicast algorithms under various size of destination group (20-node graphs, ρ=0.5).

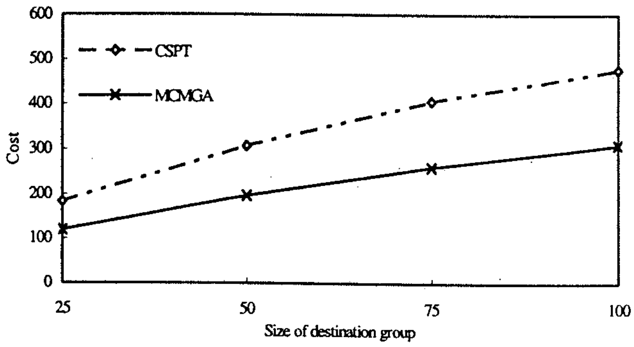

Figure 9: Performance comparison of two multicast algorithms under various sizes of destination group (100-node graphs, ρ=0.5).

Figures 10~12 compare the performance of two multicast algorithms under different network size when the size of destination group is set to 10%, 20%, and 30% of constructing a multicast tree with multiple constraints is NP-complete. In this paper, we have proposed a multiple-constraint multicast algorithm based on genetic algorithms. Our simulation results show that MCMGA yields much better performance than a simple heuristic algorithm, CSPT.

In this paper, we have assumed that the link costs are randomly generated and the bandwidth required on each link of multicast tree is fixed, e.g., obtained based on the effective bandwidth. However, if end-to-end QoS is required for each source-destination pair of a multicast connection, the bandwidth required at each link may not be the same, it would depend on the end-to-end QoS, the length of the path, current resource utilization of each link on the multicast tree, etc. Therefore, how to define link cost and perform QoS allocation and call admission worths further study.

Reference

[1] S. L. Hakimi, "Steiner Problem in Graphs and Its Implications," *Networks*, vol. 1, pp. 113-133, 1971.

[2] R. Karp, "Reducibility Among Combinatorial Problems," *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, Eds. New York: Plenum, 1972, pp. 85-103.

[3] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicast Routing for Multimedia Communication," *IEEE/ACM Trans. on Networking*, vol. 1, no. 3, pp. 286-292, June 1993.

[4] Q. Zhu, M. Parsa, and J. J. Garcial-Luna-Aceves, "A Source-based Algorithm for Delay-constrained Minimum-cost Multicasting," *IEEE INFOCOM'95*, pp. 377-385, Mar. 1995.

[5] Hussein F. Salama, Douglas S. Reeves, and Yannis Viniotis, "Evaluation of Multicast Routing Algorithms for Real-time Communication on High-speed Networks," *IEEE Trans. on Commun.*, vol. 15, pp. 332-344, April 1997.

[6] George N. Rouskas and Ilia Baldine, "Multicast Routing with End-to-End Delay and Delay Variation Constraints," *IEEE Trans. on Commun.*, vol. 15, no. 3, pp. 346-356, April 1997.

[7] D. R. Shier, "On Algorithms for Finding the k Shortest Paths in a Network," *Networks*, vol. 9, pp. 195-214, 1979.

[8] R. E. Tarjan, "Finding Optimum Branchings," *Networks*, pp.25-35, 1977.

[9] H. Takahashi and A. Matsuyama, "An Approximate Solution for the Steiner Problem in Graphs," *Math, Japonica*, Vol. 6, pp. 573-577, 1980.

[10] E. W. Dijkstra, "A Note on Two Problems in Connection with Graphs," *Numer. Math.*, vol. 1, pp. 269-171, 1959.

[11] Michael Garey and David Johuson, *Computers and Intractability: A guide to the Theory of NP-completeness*, New York, NY: Freeman, 1979.

[12] B. Waxman. "Routing of Multipoint Connections," *IEEE J. on Selected Areas in Commun.*, vol. 1, no. 9, pp. 1617-1622, December 1988.

[13] Sung-Jin Chung, Sung-Pil Hong, and Hong-Suk Hub, "A Fast Multicast Routing Algorithm for Delay-Sensitive Applications," *IEEE GLOBECOM'97*, Nov, 1997.

[14] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent Capacity and its Application to Bandwidth Allocation in High-speed Networks," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 968-981, Sept. 1991.

the network size, respectively. As we can see from these graphs, MCMGA still yields better performance than the CSPT algorithm in all cases.

In the following, we study the efficiency of network bandwidth managed by the two multicast algorithms. We assume that any session traversing a link e reserved a fraction of bandwidth equal to the equivalent bandwidth [14] of the traffic it generated. The link cost $C(e)$ was taken equal to the reserved bandwidth divided by the link capacity on that link. Therefore, the cost of a heavily utilized link was larger than the cost of a lightly utilized link. $C(e)$ is dynamic and varied as new sessions were established. A link could accept sessions and reserve bandwidth for them until the sum of the equivalent bandwidths of the sessions traversing that link exceeded 85% of the link capacity, then it got saturated. This admission control policy allowed statistical multiplexing and efficient utilization of the available resources. In our experiments, full duplex ATM networks with homogeneous link capacities of 155 Mb/s (OC3) are used and the equivalent bandwidth is assumed to be 0.5 Mb/s. We start with a completely unloaded network and keep adding multicast sessions for 5,000 times and constructing the corresponding multicast trees.

Figure 13 compares how efficiently these four algorithms manage the network bandwidth under different number of group members in 20-node graphs. As the size of the multicast group increases, the number of multicast sessions that an algorithm can accept before the network saturates decreases. Figure 13 shows that the MCMGA algorithm manage the network bandwidth more efficiently than the CSPT algorithm.

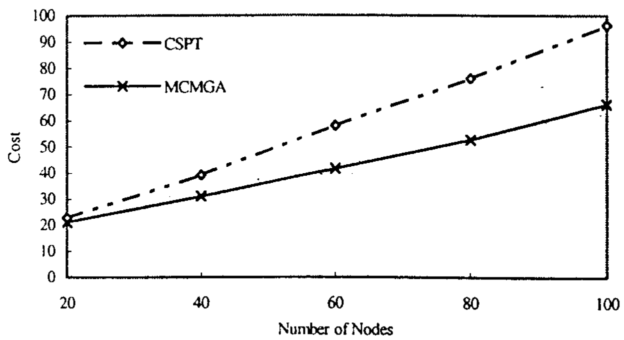

Figure 10: Performance comparison of two multicast algorithms under various network sizes (size of destination group equal to 10% of the number of nodes).

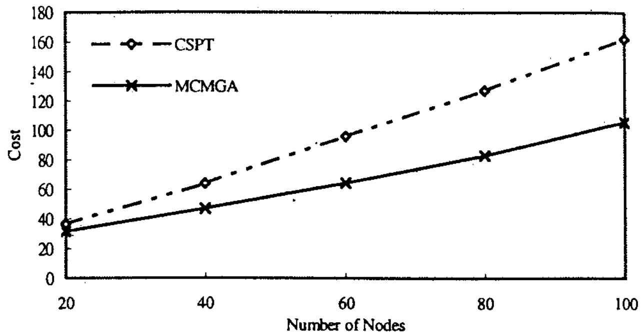

Figure 11: Performance comparison of the two multicast algorithms under various network sizes (size of destination group equal to 20% of the number of nodes).

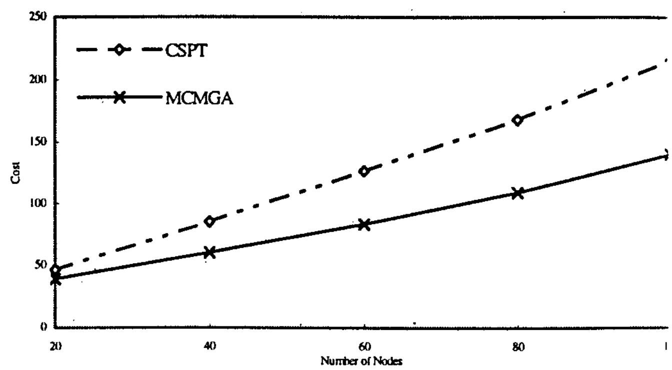

Figure 12: Performance comparison of the two multicast algorithms under various network sizes (size of destination group equal to 30% of the number of nodes).

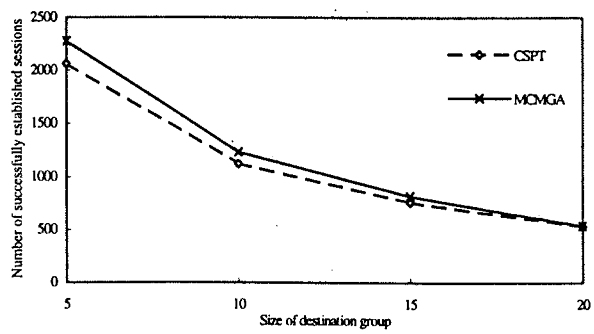

Figure 13: Bandwidth management efficiency of the two multicast algorithms (20-node graphs, ρ =0.5).

6. Conclusion and Future Work

Multimedia applications involved in real-time applications have multiple QoS requirements that must be guaranteed by the underlying network. The problem of