A Hybrid Approach to Quality of Service Multicast Routing

John Crawford and Gill Waters

Computing Laboratory
University of Kent at Canterbury
CT2 7NF, UK

e-mail: J.S.Crawford@ukc.ac.uk and A.G.Waters@ukc.ac.uk

Abstract

Several multicast routing heuristics have been proposed to support multimedia services, both interactive and distribution, in high speed networks such as B-ISDN/ATM. Since such services may have large numbers of members and have real-time constraints, the objective of the heuristics is to minimise the multicast tree cost while maintaining a bound on delay. Previous evaluation work has compared the relative average performance of some of these heuristics and concludes that they are generally efficient, although some perform better for small multicast groups and others perform better for larger groups.

We present a detailed analysis and evaluation of some of these heuristics which illustrate that in some situations their average performance is reversed; a heuristic that in general produces efficient solutions for small multicasts may sometimes produce a more efficient solution for a particular large multicast/network combination. Also, in a limited number of cases using Dijkstra's algorithm produces the best result. We conclude that the specific efficiency of a heuristics solution depends on the topology of both the network and the multicast, and that it is difficult to predict.

Because of this unpredictability we propose the integration of two heuristics with Dijkstra's shortest path tree algorithm to produce a hybrid that consistently generates efficient multicast solutions for all possible multicast groups in any network. These heuristics are based on Dijkstra's algorithm which maintains acceptable time complexity for the hybrid, and they rarely produce inefficient solutions for the same network/multicast. The resulting performance attained is generally good and in the rare worst cases is that of the shortest path tree. The performance of our proposal is supported by our evaluation results.

We conclude by discussing the types of networks for which this method is most appropriate and identifying further work.

1 Introduction

Many of the new services envisaged for B-ISDN/ATM high speed networks involve point to multipoint routing. Some of these services, such as video distribution and interactive multimedia communications will require high bandwidths and bounded delays on data delivery. Calculation of multicast routes for these types of applications must take account of their potentially conflicting requirements for efficient network usage and delay bounds on delivery.

The problem of arbitrary delay bound low cost multicasting in networks, where link cost and link delay are different functions, was first addressed by Kompella, Pasquale and Polyzos in [8]. Since then there have been a number of other proposals for solutions to this problem. Previous evaluation work [15] [11] shows that on average these heuristics perform well. Further detailed analysis and evaluation of some of these heuristics has shown that there is a wide variance in the efficiency of their solutions. Whilst on average one heuristic may be more efficient than another, either for all multicast group sizes or for a particular range of multicast group sizes, there are some multicast group and network combinations where this position is reversed. In particular, we have found that as a multicast group grows and dies the heuristic that provides the most efficient multicast solution also changes. The results of our evaluation work indicates that it is difficult to predicit which heuristic provides the most efficient solution for any particular multicast/network combination. The variance in the efficiency of the heuristics solutions is wide enough that on occasions Dijkstra's shortest path algorithm (SPT) calculated on delay is more efficient. By selecting two such heuristics that can be efficently integrated with each other and the SPT algorithm, we propose a hybrid heuristic that produces reasonably consistent and efficient solutions to the multicasting problem, with an acceptable order of time complexity for all possible multicast groups in any network.

The rest of this paper is organised as follows. In section 2 we define the bounded delay minimum cost multicast routing problem. In section 3 we decribe and assess four heuristics, one of which has a variant, as candidates for integration. Sections 4 and 5 decribe the network model, benchmark algorithms and arbitrary delay bound we use to evaluate both the candidate algorithms and the hybrid. The candidate heuristics are evaluated in Section 6. Sections 7 and 8 describe and evaluate the hybrid heuristic. We conclude the paper in Section 9 and identify current and further research.

2 Delay Bound Minimum Cost Multicast Routing

The bounded delay minimum cost multicast routing problem can be stated as follows. Given a connected graph $G = \langle V, E \rangle$ where V is the set of its vertices and E the set of its edges, and the two functions: cost $c(i,j)$ of using edge $(i,j) \in E$ and delay $d(i,j)$ along edge $(i,j) \in E$, find the tree $T = \langle V_T, E_T \rangle$, where $T \subseteq G$, joining the vertices s and $M_{k,k=1,n} \in V$ such that $\sum_{(i,j)\in E_T} c(i,j)$ is minimised and $\forall k, k = 1,n$; $D(s, M_k) \leq \Delta$, the delay bound, where $D(s, M_k) = \sum_{(i,j)} d(i,j)$ for all (i,j) on the path from s to M_k in T. Note that, if the delay is unimportant, the problem reduces to the Steiner tree problem which is well-known to be NP-complete. The addition of the finite delay bound makes the problem harder, and it is still NP-complete, as any potential Steiner solution can be checked in polynomial time to see if it meets the delay bound.

3 Heuristics with an Arbitrary Delay Bound

Several heuristics have been proposed that use arbitrary delay bounds to constrain multicast trees. Kompella, Pasquale, and Polyzos [8] propose a Constrained Steiner Tree (CST_c) heuristic which uses a constrained application of Floyd's algorithm [4]. Widyono [17] proposed four heuristics based on a constrained application of the Bellman-Ford algorithm[1]. Zhu, Parsa and Garcia-Luna-Aceves [18] based their technique on a feasible search optimisation method to find the lowest cost tree in the set of all delay bound Steiner trees for the multicast. Evaluation work carried out by Salama, Reeves, Vinitos and Sheu [11] indicate that Constrained Steiner Tree heuristics have good performance, but are inhibited by high time complexity. The proposals for Constrained Shortest Path Trees by Sun and Langendoerfer [12], which we abbrieviate as $CSPT$ and by Waters [15], which we abbreiviate as $CCET$ (Constrained Cheapest Edge Tree), generally have a lower time complexity than Constrained Steiner Trees but their solutions are not as efficient.

3.1 The CST_c Heuristic

The CST_c algorithm was first published in [8]. and has three main stages [7]. A closure graph (complete graph) of the constrained cheapest paths between all pairs of members of the multicast group is found. This involves stepping through all the values of delay from 1 to the arbitrary delay bound Δ (assuming Δ takes an integer value) and, for each of these values, using a similar technique to Floyd's all-pairs shortest path algorithm. A constrained spanning tree of the closure graph is then found using a greedy algorithm based on cost. An alternative selection mechanism is proposed based on a function of both cost and delay. The edges of the spanning tree are then mapped back onto their paths in the original graph. Finally any loops are removed by using a shortest paths algorithm on the expanded constrained spanning tree [7]. The calculation of the constrained shortest paths during the first stage of the heuristic is the most time consuming, with a complexity of $O(\Delta n^3)$, where n is the number of vertices in the graph. The second stage has a time complexity of $O(m^3)$ where m is the number of nodes in the multicast group. The third stage has a time complexity of at most $O(n^2)$. This gives the algorithm an overall time complexity of $O(\Delta n^3)$.

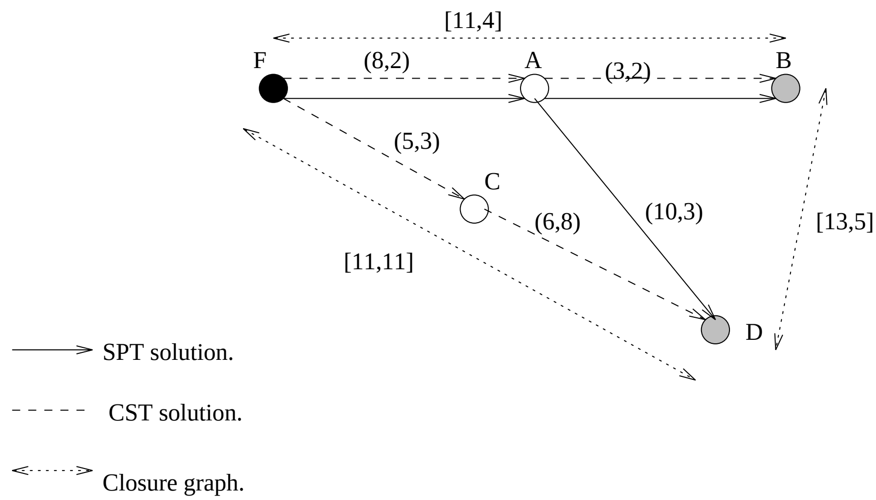

Figure 1: CST_c more expensive than SPT

In most cases CST_c calculates multicast solutions that are cheaper than those produced by SPT, but it does sometimes generate more expensive solutions. Figure 1 illustrates such a case. The multicast is from the source node, F, to the destination nodes, B and D. The arbtrary delay bound is 12. The first stage of CST_c constructs a closure graph from the cheapest contrained paths between the multicast nodes and the source in the underlying graph. From the closure graph CST_c selects the solution. In the example the multicast solution selected will be the closure graph edges FB and FD at a cost of 22 and a delay of 11. The final stage of CST_c maps the closure graph solution back onto the original graph, providing the solution FA,AB and FC,CD. The SPT algorithm will select paths on the basis of the delay only, from the source to each destination node. The solution SPT provides is FA,AB and AD at a cost of 21 and delay 5. By chance the SPT has been able to take advantage of the common edge FA, which was not available in the closure graph for CST_c.

3.2 The $CCET$ Heuristic and Extensions

The $CCET$ heuristic was first published in [14] along with some simple preliminary evaluations.

The original heuristic was bound by either the broadcast delay for all nodes in the network or the multicast delay for the subset of multicast nodes. Here we extend the heuristic such that it is bound by an arbitrary delay, Δ. The extended procedure for the $CCET$ heuristic is as follows. An extended form of the SPT algorithm is used to construct a directed graph that contains the shortest delay path tree and all the forward paths from each node to its neighbours. The last edge of every path from the source that breaks the arbitrary delay bound, Δ is removed from the directed graph. A broadcast tree is constructed by selecting paths between each node and the source from the directed graph. Starting with the node furthest from the source, in terms of delay, the cheapest exit from each successive node is chosen, on a path back to the source that does not break the arbitrary delay bound. Note that the cost of the cheapest exit is in the direction from the source to the destination and not the reverse edge cost. The path is added to a broadcast tree. This process is repeated until all nodes have been included in the broadcast tree. The multicast tree is extracted from the broadcast tree by removing all none multicast nodes that are not on paths between the source and the multicast nodes.

Salama [10] and Crawford [2] proposed similar variants of $CCET$ which use the cheapest path cost as their selection criteria, rather than the cheapest edge cost as in the original. We include this variant, which which we abbrieviate as $CCPT$ (Constrained Cheapest Path Tree), in our evaluation for completeness.

The first stage, determining the directed graph, has the same time complexity as the SPT algorithm, $O(n^2)$. The second stage, building the broadcast tree, requires a depth first search from each node to find a path to the source. The time complexity of the search is $O(max(N, |E|))$ [5] where N is the number of nodes and E is the set of edges in the search tree, which is the tree of all possible routes back to the source through the directed graph. As the multicast tree grows the search tree for each node to source node path becomes smaller, reducing the values of N and $|E|$ for subsequent searches. In practice, an optimal upper bound can be placed on the arbitrary delay to limit the values of N and $|E|$. (see Section 6)

Although the $CCET$ in general performs well, the following anomalies can occur. The cost of multicast trees found using the $CCET$ heuristic can increase when the arbitrary delay bound is relaxed. Such a case is illustrated in Figure 2 where edges are labelled with $(cost, delay)$ and the multicast is from the source node F to the nodes C and H. With a delay bound of $\Delta = 5$ the multicast tree will include the edges FC,FG and GH at a cost of 8 units. This happens because the edge BH will have been excluded as it gives node H a delay of 8 units from the multicast source node F. If Δ is increased to 8 the edge BH is included and will be selected as the "cheapest" return route from node H towards F. The multicast tree then becomes FC, CB, and BH at a cost of 9 units.

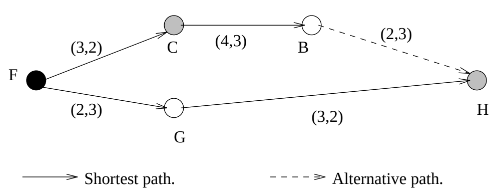

Figure 2: Costs increase as Δ is relaxed.

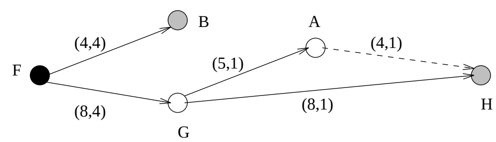

Figure 3: $CCET$ more expensive than SPT.

The cost of solutions found using the SPT algorithm can sometimes be cheaper than those found using the $CCET$ heuristic. The multicast tree found using SPT for the network in Figure 3 includes the edges FB,FG and GH, the shortest paths. The cost of this tree is 20 units. If the $CCET$ heuristic is used to calculate the multicast tree with an arbitrary delay bound of $\Delta = 6$ the solution will include edges FB,FG,GA and AH because AH offers the "cheapest" exit back to the source from node H. The cost of this tree is 21 units.

3.3 The $CSPT$ Heuristic

The $CSPT$ heuristic was first published in [12] and has three steps. Using the SPT algorithm compute a low *cost* spanning tree to as many destination nodes in the multicast as is possible without any path breaking the arbitrary delay bound, Δ. Use the SPT algorithm to compute a shortest *delay* path tree to those multicast nodes not reached in the previous step. Combine the low cost spanning tree from the first step with the shortest delay path tree from the second step making sure that the delay to any destination node does not break the delay bound, Δ, and that all loops are removed.

Each of the first two steps of the heuristic have the time complexity of the SPT algorithm, which is at most $O(n^2)$. The last step has a time complexity of $O(n)$. As with $CCET$, there are also some cases where the cost of solutions found using the SPT algorithm can be cheaper than those found using the $CSPT$ heuristic. In figure 4, for a delay bound greater than 8, to connect the multicast nodes A,G and H to the source F, the $CSPT$ heuristic will use the path FB,BA at cost 18 and path FH,HG at cost 13 because they are the shortest paths based on cost

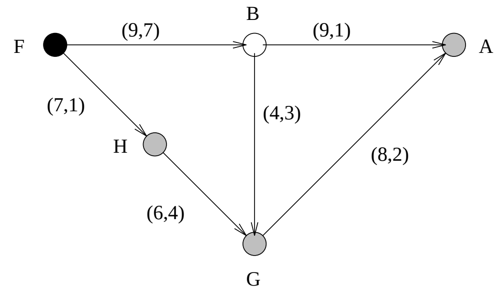

Figure 4: $CSPT$ more expensive than SPT.

between the multicast nodes and the source. This results in a multicast tree of cost 31. The SPT algorithm based on delay will choose the path FH,HG,GA at a cost of 21 to connect all the multicast nodes to the source.

4 Evaluation Networks and Benchmark Algorithms

We use Doar's scaling extention [3] to Waxman's model [16] to generate random networks.

The ideal benchmark algorithm to use would be one that produces optimal delay bound minimum cost multicast trees which, being an NP-complete problem, is impractical for large graphs. Instead we use the Minimum Steiner Tree heuristic (MST) [6] which approaches a minimum cost for multicast trees, although they are of unbound delay. We also use the SPT as a benchmark to evaluate the cost savings made by using the various heuristics.

5 Arbitrary Delay Bounds

We chose the network diameter as the arbitrary delay bound for the evaluation of the multicast algorithms. This purely arbitrary choice provides an evaluation "mid-point" between the multicast delay, which is the tightest bound and the MST which is the delay at which the maximum improvement in network utilisation for each heuristic will be achieved.

6 Evaluation of the Candidate Heuristics

6.1 Performance averages

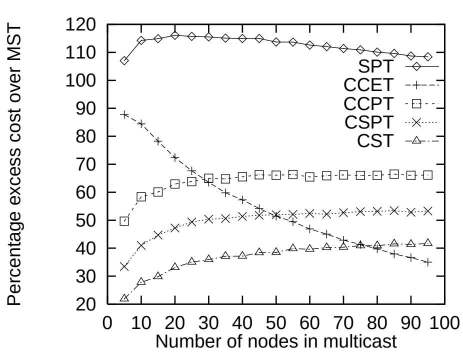

Figure 5: Average comparative costs

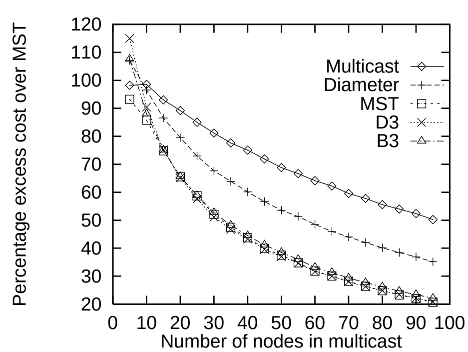

Figure 6: CCET excess costs as Δ increases

For each evaluation 200 networks of 100 nodes of low edge density were used. Multicast groups were selected for sizes from 5 to 95 nodes, at steps of 5. There were 10 multicast samples for each multicast group size, for each network.

Figure 5 illustrates the percentage excess costs of using the four heuristics described above, relative to the MST and SPT benchmarks. For the CST_c heuristic we use a granularity of $\Delta/5$ to step through possible delay values (see Section 3.1).

The algorithm of CST_c generates multicast solutions that are on average cheaper than the other heuristics although, as the size of the multicast group size increases, the $CCET$ heuristic's solutions become cheaper than those of CST_c. The performance of the $CCET$ heuristic is much better than $CSPT$ and $CCPT$ for larger multicasts, but is worse for smaller multicasts. The $CCPT$ heuristic shows poor performance in comparison with that of $CSPT$. The solutions of $CSPT$ and $CCPT$ are similar because they are tightly constrained through their construction of paths using Dijkstra's SPT algorithm. Although $CCET$ uses an extension of the SPT algorithm to construct its search space, it is not constrained by the algorithm when finding its solution. Rather it relies on the chances of the selected edges leading to existing paths of the solution. This chancy approach results in small multicast solutions being relatively expensive, while large multicasts solutions are much cheaper.

We have observed that as the delay bound approaches the MST delay, improvements in solution efficiency of the $CCET$ heuristic become negligible (Figure 6 where D3 = 3*network diameter;B3 = 3*broadcast delay from the source). Up to these delay bound limits the number of nodes visited during the tree search in the heuristic's second stage is of $O(< 2n)$, by observation. If the delay bound goes much beyond these limits the heuristic is occasionally prone to very long execution periods which suggests that either N or $|E|$ (or both) can become unacceptably large.

6.2 Specific multicast comparisons

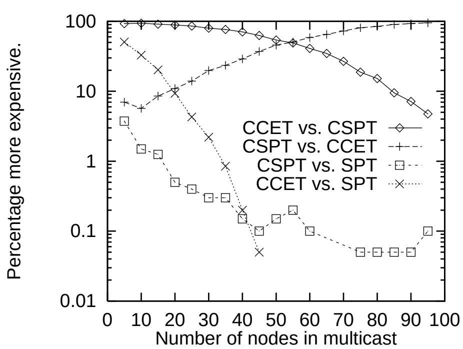

Figure 7: Exceptional comparative costs

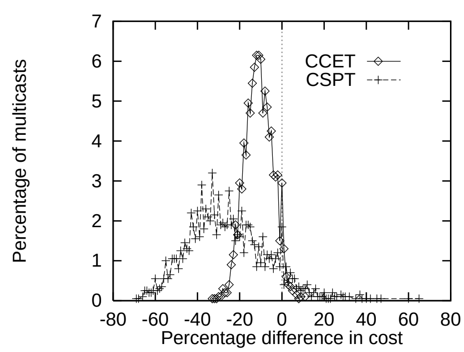

Figure 8: Cost distributions

The $CSPT$ heuristic is generally better for smaller multicast group sizes, while the $CCET$ heuristic is more suited to larger multicasts, although this is not always the case. Figure 7 illustrates a sample of the percentage of times $CCET$ soutions are more expensive than those of $CSPT$, $CSPT$ solutions are more expensive than $CCET$, and when the solutions of both $CSPT$ and $CCET$ are more expensive than the SPT. In nearly 5% of the sample, for multicast groups of 95 nodes, the solutions generated by $CCET$ were more expensive than those generated by $CSPT$. Similarly, in 7% of the sample, for multicast groups of 5 nodes, the solutions generated by $CSPT$ were more expensive than those generated by $CCET$. For smaller multicast groups sizes, both $CSPT$ and $CCET$ generated some solutions that were more expensive than the SPT solutions. For larger multicasts $CSPT$ still generates some solutions that are more expensive than SPT, while $CCET$ does not. Figure 8 indicates just how large and varied these differences can be. The graph for $CSPT$ plots the percentage cost savings of $CSPT$ over $CCET$ for small multicasts. While the majority of $CSPT$ solutions are up to 69% cheaper, some can be up to 65% more expensive. Similarly, for $CCET$ the majority of larger multicasts are up to 33% cheaper than $CSPT$, although some can be as much as 11% more expensive. This behaviour shows, as might be expected from our analysis, that the solutions each heuristic generates depend on the algorithm, the topology of the network and the topology of the multicast. There is also a wide variance in the cost of solutions between the heuristics for the same size multicasts.

7 Hybrid Approach to Heuristic Multicasting

We conclude from our analysis and evaluation work that none of the heuristics we have considered can provide the "cheapest" multicast solutions in all networks for all sizes of multicast groups. They either take too long to find their solutions or are vunerable to generating unacceptable solutions that depend on the network topology and/or the multicast topology. We propose that by combining heuristics of acceptable time complexity that can be efficiently integrated, the resulting hybrid will generate solutions that are predominantly cheaper than SPTs for all network topologies, for all multicast group sizes.

CST_c on average generates good solutions but has an order of time complexity which may be too high for practical use. Also its calculation is based upon a variant of Floyd's All Pair Shortest Paths algorithm [4], making it unsuitable for integration with the other heuristics. We discard $CCPT$ because of its poor overall performance.

The $CCET$ and $CSPT$ heuristics generate the majority of their most efficient multicast solutions at opposite ends of the multicast group size range, and both base their calculations on trees generated by the SPT algorithm. Individually, each is vunerable to generating some inefficient solutions throughout the full range of multicasts, but rarely will both heuristics generate an inefficient solution for the same network/multicast group pair. We combine the $CCET$ and $CSPT$ heuristics to obtain a hybrid of acceptable time complexity that produces solutions of significantly improved efficiency over SPTs. The hybrid will select the "cheapest" tree provided by these combined heuristics as the multicast solution. To guarantee minimal efficiency the SPT algorithm is also included in the hybrid to cater for the rare instances where both $CSPT$ and $CCET$ produce solutions that are more expensive than the SPT. The $CCET$ function, within the hybrid, must place a maximum limit on the delay bound it uses to calculate its multicast solution in order to limit its execution time, as previously suggested. This maximum value does not apply to the $CSPT$ function.

Integration of the three heuristics is simple. All three calculate the shortest path tree for delay, which is extended for the second stage of the $CCET$ heuristic. The $CSPT$ heuristic also calculates the SPT shortest path tree for cost, a task which can be conducted concurrently with the delay calculation. Once the trees have been obtained for each method their costs can be easily calculated and the cheapest tree selected as the solution.

The time complexity of the hybrid is dominated by the $CCET$ function. The first stage of this function has time complexity of at most $O(n^2)$. The second stage, the construction of the broadcast tree, has a time complexity of $O(max(N, |E|))$. In practice, the time taken by this stage is limited by maximum value on the delay bound it uses, as discussed in Section 3.2 and observed in Section 6. The $CSPT$ and SPT functions have a time complexity of $O(n^2)$.

8 Evaluation of Hybrid Heuristic

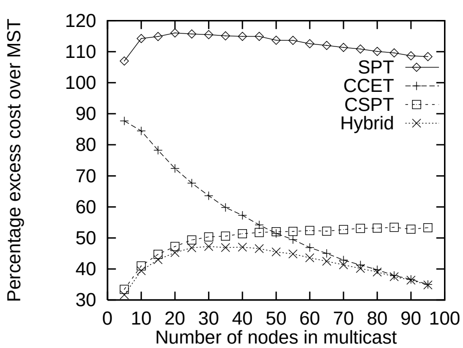

Figure 9: Average comparative costs

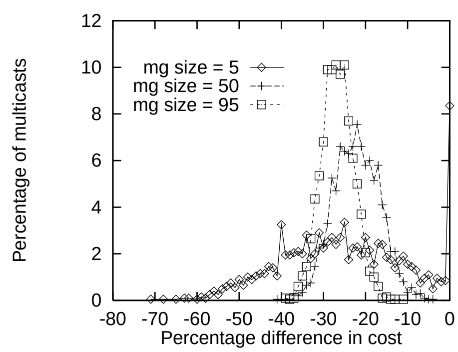

Figure 10: Cost distribution

Figure 9 illustrates the cost performance of the hybrid heuristic in comparison to $CCET$ and $CSPT$. The hybrid outperforms or equals both $CCET$ and $CSPT$. It is interesting to note that for mid-sized multicasts the hybrid is able to provide solutions that are better than either $CSPT$ or $CCET$ can do separately. This occurs because the hybrid is able to choose the most efficient heuristic for each particular multicast. The efficiency of hybrid solutions for small multicasts is still subject to a fairly wide variance as figure 10 shows. These graphs plot the cost savings distributions of the hybrid over SPT for multicast group sizes of 5,50 and 95 respectively. The dominance of $CSPT$ for small multicast groups and $CCET$ for large multicasts is obvious, as is the narrow but sharp intervention of SPT when required.

9 Conclusions and Further Research

We have identified problems of time compelxity and performance variability in heuristics that have been proposed to calculate low-cost multicast trees that are bound by an arbitrary delay. By combining appropriate heuristics we propose a hybrid that produces efficient solutions over all multicast group sizes within an acceptable order of time complexity.

The hybrid heuristic uses metrics for every link in a network to perform its route calculation and so is amenable for implementation in link-state routing protocols such as the Internet's Multicast Open Shortest Path First protocol[9] or that used by the ATM Forum's Private Network-Network Interface [13].

We have not been able to present here all the evaluation work we have carried out. Particularly we have only presented results for fixed size networks and for multicasts with a single arbitrary delay bound. Work not presented includes evaluations using different size single cluster networks, hierarchical networks and a variety of arbitrary delay bounds. These results all confirm the consistent performance of the hybrid. Further research includes the study of join/leave mechanisms and how the heuristic might be implemented as part of a hierarchical routing protocol.

An important result of this work, and a departure from current routing solutions, is the integration of several heuristics which are indivdually unstable (as might be expected in an heuristic approach) into a stable hybrid. Hybrid methods may also have an application in other multicast or load sharing route calculation algorithms.

References

[1] D. Bertsekas and R. Gallager. *Data Networks.* Prentice-Hall,Inc., 1987.

[2] J.S. Crawford. Multicast Routing: Evaluation of a New Heuristic. Master's thesis, University of Kent at Canterbury, 1994.

[3] J.M.S. Doar. Multicast in the Asynchronous Transfer Mode Environment. Technical Report No. 298, University of Cambridge Computing Laboratory, April 1993.

[4] R.W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, 1962.

[5] Alan Gibbons. *Algorithmic Graph Theory.* Cambridge University Press, 1989.

[6] E.N. Gilbert and H.O. Pollack. Steiner Minimal Trees. *SIAM Journal on Applied Mathematics*, 16, 1968.

[7] P. Kompella, V. *Multicast Routing Algorithms for Multimedia Traffic.* PhD thesis, University of California, San Diego, USA, 1993.

[8] V.P. Kompella, J.C. Pasquale, and G.C. Polyzos. Multicast Routing for Multimedia Communications. *IEEE/ACM Transactions on Networking*, 1(3):286–292, 1993.

[9] J. Moy. Multicast Extensions to OSPF. RFC 1584, March 1994.

[10] H.F. Salama, D.S. Reeves, I. Vinitos, and Tsang-Ling. Sheu. Comparison of Multicast Routing Alogrithms for High Speed Networks. Technical report, IBM Technical Report IBM-TR29.1930, September 1994.

[11] H.F. Salama, D.S. Reeves, I. Vinitos, and Tsang-Ling. Sheu. Evaluation of Multicast Routing Algorithms for Real-Time Communication on High-Speed Networks. In *Proceedings of the 6th IFIP Conference on High-Performance Networks (HPN'95)*, 1995.

[12] Q. Sun and H. Langendoerfer. Efficient Multicast Routing for Delay-Sensitive Applications. In *Second Internatiopnal Workshop on Protocols for Multimedia Systems (PROMS'95)*, pages 452–458, 1995.

[13] The ATM Forum Technical Committee. *Private Network-Network Interface Soecification, Version 1.0.* The ATM Forum, March 1996.

[14] A.G. Waters. A New Heuristic for ATM Multicast Routing. In *2nd IFIP Workshop on Performance Modelling and Evaluation of ATM Networks*, pages 8/1–8/9, July 1994.

[15] A.G. Waters and J.S. Crawford. Low-cost ATM Multimedia Routing with Constrained Delays. In *Multimedia Telecommunications and Applications (3rd COST 237 Workshop, Barcelona, Spain)*, pages 23–40. Springer, November 1996.

[16] B.M. Waxman. Routing of Multipoint Connections. *IEEE journal on selected areas in communications*, 6(9):1617–1622, 1988.

[17] R. Widyono. The Design and Evaluation of Routing Algorithms for Real-time Channels. Tr-94-024, University of California at Berkeley and International Computer Science Institute, September 1994.

[18] Q. Zhu, M. Parsa, and J.J. Garcia-Luna-Aceves. A Source-Based Algorithm for Near-Optimum Delay-Constrained Multicasting. In *Proceedings of INFOCOM*, pages 377–385, 1995.