# *An Heuristic for Lower Cost Multicast Routing in the Internet.*

John Crawford and Gill Waters

Computing Laboratory

University of Kent at Canterbury

UK

July 2, 1998

**Abstract**

Some multicast applications require high bandwidth and bounded delay (eg Video-conferencing). The general problem of constructing bounded delay minimal cost multicast trees uses two link metrics (cost and delay) and is NP-complete. A number of heuristics have been developed, most of which tend to have a high order of time complexity. Our heuristics achieve good results with low time-complexity and we believe that they are adaptable to a number of other multicasting scenarios.

Current Internet multicasting protocols use either link-state or distance vector routing to construct multicast trees that have minimum path delays. Tree cost is ignored.

When our heuristics are applied using a single link metric, they achieve minimum delay path multicast trees of lower cost than those constructed using Dijkstra's shortest path algorithm. The cost savings achieved by one of our heuristics is such that it can be considered as a candidate for use in distributed muticast protocols, such as MOSPF, as an alternative to Dijkstra's algorithm.

# 1 Introduction

In their comparison of multicast trees and algorithms Wei and Estrin [17] judge the quality of multicast trees according to three criteria:

1. Low delay
2. Low cost
   - Cost of total bandwidth consumption.
   - Cost of tree state information.
3. Light traffic concentration

Shortest path tree multicasting minimises delay to recipients, but ignores tree cost. Steiner minimal trees attempt to minimise the cost of multicast trees, but their computation is NP-complete [7] which renders them unsuitable for reasonably-sized networks [17].

In [13] [3] respectively, we describe heuristics that offer low cost multicast trees that are constrained by a bounded delay. These heuristics were designed to meet the needs of multicast routing in future high speed networks, such as ATM/B-ISDN, where link costs and delays may be independent functions.

Evaluation of these heuristics [14] [15] indicates that a modified version of the Cheapest Link Multicast (CLM) heuristic may provide lower cost multicast trees than current source based shortest path multicast trees in networks that have only a single metric (delay or cost) per link. With further adaptation this modified heuristic could be integrated into a distributed link-state protocol, such as MOSPF [9] for use in connectionless networks.

Lower cost trees are obtained by restricting CLM to using only a single metric and bounding the multicast tree by the delay to the multicast node furthest from the multicast source.

In section 2 we redefine the original delay bound minimal cost routing problem in terms of the new requirement. Section 3 describes CLM in a form that meets the requirement and gives an example of how it works. Section 4 illustrates some evaluation results. Section 5 outlines how the modified CLM might be adapted for distributed multicast routing protocols and used in connectionless networks.

## 2 Defining the bounded delay, minimum cost multicast routing problem

In our definition of the bounded delay, minimum cost multicast routing problem[13], we assume that all links in a network have independent delay and cost functions. In this definition we use a single edge metric, which can represent delay or cost or any other link metric. For convienience we use the link delay metric.

- 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 function 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 $\sum_{(i,j) \in E_T} d(i,j)$ 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. The addition of the finite delay bound makes the problem harder, and it is still NP-complete.

# 3 Modified CLM ($m$CLM)

CLM was first published in [13] along with some simple preliminary evaluations. The original heuristic uses two link metric functions, $d(i,j)$ and $c(i,j)$. This variant assumes there is only one metric function, $d(i,j)$.

1. Use an extended form of Dijkstra's shortest path algorithm, to find for each $v \in V - \{s\}$ the minimum delay, $dbv$, from $s$ to $v$. As the algorithm progresses, keep a record of all the $dbv$ found so far, and build a matrix $Delay$ such that $Delay(k,v)$ is the sum of the delays on edges in a path from $s$ to $v$, whose final edge is $(k,v)$. When the algorithm is complete, the $dbv$ to the multicast node furthest from the multicast source is the multicast delay bound $dbM$.

2. Set all elements in $Delay(k,v)$ that are greater than $dbM$ to $\infty$. The matrix $Delay$ then represents the edges of a directed graph derived from $G$ which contains all possible solutions to a multicast tree rooted at $s$ which satisfy the delay constraint (i.e. $\Delta = dbM$).

3. Now construct the multicast tree $T$. Start by setting $T = \langle \{s\}, \emptyset \rangle$.

4. Take $v \in V_T$, with the maximum $dbv$ and join this to $T$. Where there is a choice of paths which still offer a solution within the delay bound, choose at each stage the lowest delay edge leading to a connection to the tree. This step involves a depth first search to find a path to the multicast source that does not exceed the delay bound $dbM$. Such a path will always exist, although this step may require backtracking to find it. (Our evaluation of the heuristics indicates that this is not, generally, a problem).

5. Include in $E_T$ all the edges on the path $(s,v)$ not already in $E_T$ and include in $V_T$ all the nodes on the path $(s,v)$ not already in $V_T$.

6. Repeat steps 4 and 5 until $V_T = V$, when the multicas tree will have been built.

7. Prune any unnecessary branches of the tree beyond the multicast recipients.

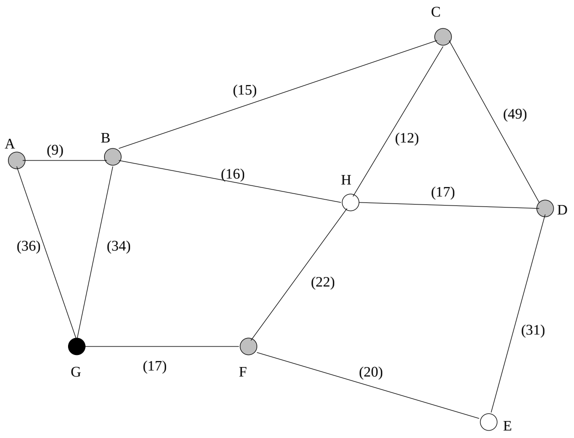


Figure 1: Sample graph to illustrate $m$CLM

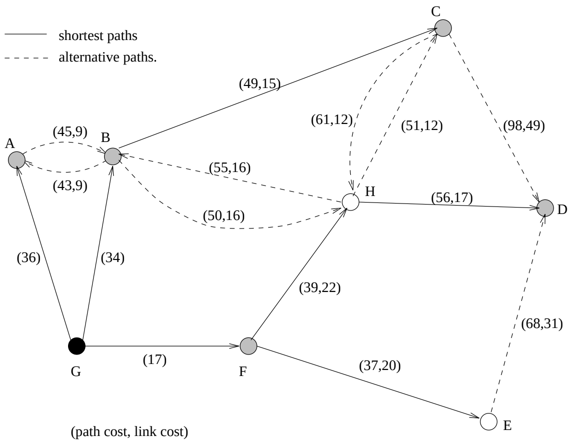


Figure 2: Directed graph of paths within the delay bound

## 3.1 A worked example

To illustrate the working of $m$CLM, we take the graph shown in Fig. 1. The bracketed parameters for each link indicate (*cost*). The example finds the multicast route from source G to destinations A, B, C, D and F.

The application of the extended form of Dijkstra's algorithm results in the directed graph shown in Fig. 2 where parameters shown against each link represent the path cost from the source, G, to the node at the end of that link and the link cost. The multicast bound is the cost to the multicast node furthest from the source, which is node D with cost 56.

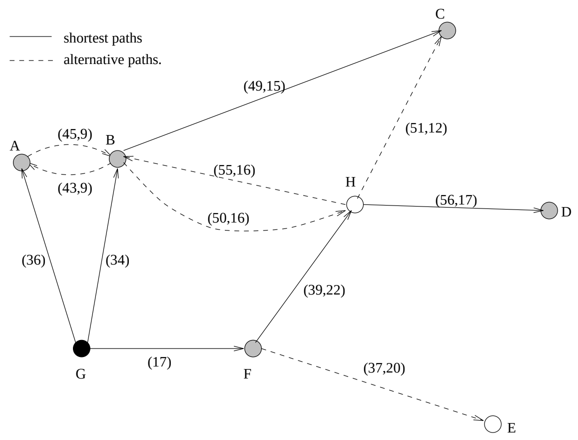


Figure 3: Directed graph of paths within the delay bound

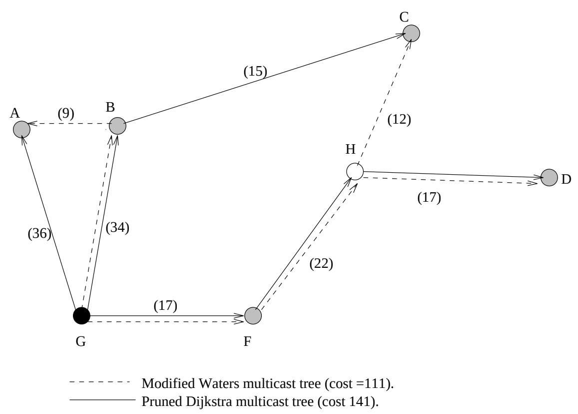


Figure 4: Multicast tree

Fig. 3 illustrates the bounded directed graph, where all edges with a path cost greater than the delay bound $dbM$ have been removed.

Starting with the shortest path node furthest from the source, node D, choose the cheapest link back to the source that remains within the delay bound. There is only one path, to node H. From node H choose the cheapest link back to the source that remains within the delay bound minus the cost of link HD. This is link HF. Repeat this process until the source is reached. The first path constructed will always be a shortest path. Nodes D,H and F are now in the multicast tree. The next furthest node on a shortest path from the source is node C. This is connected to node H without violating the cost bound $dbM$. In this manner nodes E, A and B are added to the multicast tree. Finally, the edge FE is pruned to give the multicast tree of Fig. 4 whose total cost is 111 units and has a cost bound of 56 (to D).

The tree produced by the standard use of Dijkstra's algorithm and then pruned would result in a solution with a cost of 141. This simple example shows that a good compromise between delay bound and overall cost has been found by the our heuristic.

## 3.2 Time complexity of $m$CLM

The first stage of $m$CLM has the same time complexity as Dijkstra's algorithm which is at most $O(n^2)$. The vertices can be put in delay bound order during the construction of the directed graph.

In the second stage, building the multicast tree, requires a constrained depth first search from each leaf node to find a path to the source. As the multicast tree grows the search space for each leaf to source node path becomes smaller. The time complexity of the depth first search is $O((max(n-1)*(max(N,|E|))$ [6] where $N$ is the number of nodes, and $E$ is the set of edges, in the leaf node to source tree. The values of $N$ and $|E|$ depend on the topology of the network and the position of the multicast source node. Our evaluation work has shown that in practice the time required for the multicast tree construction is much closer to $O(n * m)$, where $m$ is the maximum node degree, than $O(n^2)$ although where there are high node degree network clusters connected by very few ($> 1$) links, it can be much greater than $O(n^2)$.

# 4 Performance modelling of the heuristics

## 4.1 Network models

The network model used in this evaluation of the heuristic is attributed to Waxman [16]. The model randomly distributes nodes over a rectangular coordinate grid and uses the Euclidean metric to determine the distance between them. Edges are then introduced into the network using a probability function that depends on their length.

Further extensive evaluation of our heuristics have used network models developed by Crawford which are based on the work of Doar [5].

## 4.2 Performance evaluation

Two hundred Waxman model networks of 100 nodes each were used to obtain the evaluation results presented in fig.5. The average node degree of these networks is 4. The cost of multicast trees constructed using both $m$CLM and Dijkstra's algorithm are compared with the cost of multicast trees built using Prim's minimum cost-spanning tree algorithm. The minimum cost spanning trees are constructed for the whole network then pruned back to the multicast.

For small multicast groups $m$CLM produces, on average, slightly more expensive multicast trees than does Dijkstra's SP algorithm, while for larger multicasts the heuristic achieves

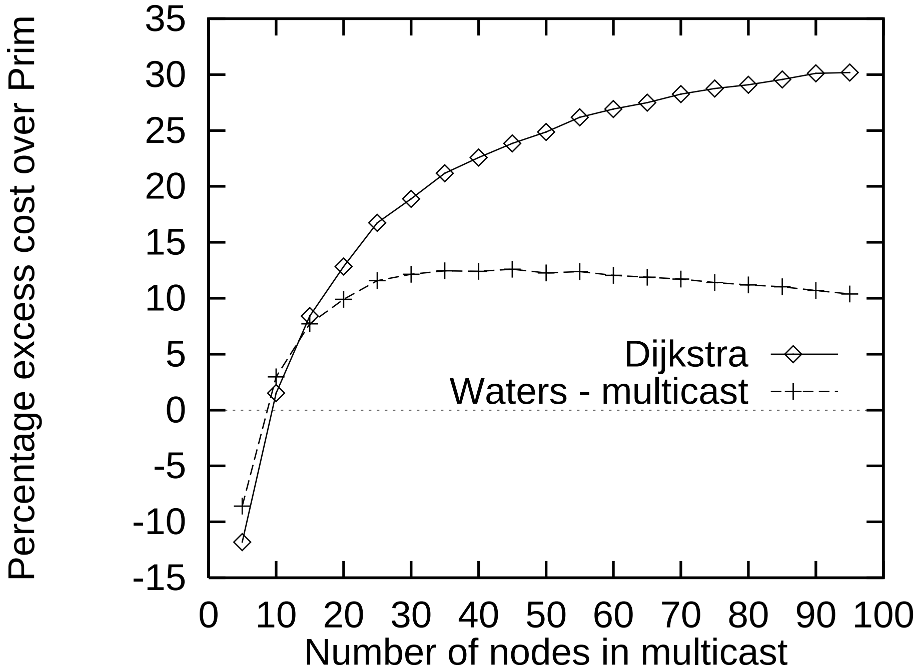


Figure 5: Multicast tree cost of $m$CLM and pruned Dijkstra

much lower cost trees than Dijkstra. $m$CLM solutions are generally cheaper than Dijkstra's because it has a choice of paths back to the source that are within the multicast delay bound. Dijkstra's algorithm only has the shortest path from each node back to the source. As the size of the multicast group increases so does the number of path choices available for $m$CLM to select, hence its improving performance.

## 5 Adpatation to Distributed Routing Protocols

For any network, Dijkstra's algorithm may calculate one of several different shortest path trees routed at the same node, depending on the order in which links are processed. This happens when there is more than one shortest path between the root and any node. The algorithm calculates only one of these paths for inclusion in the tree. In distributed unicast routing protocols such as OSPF [10], where every node calculates a shortest path tree rooted at itself, this is not a problem. Nor is it a problem for source based multicast routing, for much the same reason. In distributed multicast routing protocols this attribute of Dijkstra's algorithm is a problem because every node in the network must calculate exactly the same shortest path tree for each Source/Group in order to create consistent routing tables throughout the network.

The Multicast Extension to OSPF [9] is a distributed routing protocol which uses Dijkstra's algorithm in the "on-demand" construction of multicast routing tables. To ensure that all nodes calculate exactly the same shortest path tree for a Source/Group MOSPF employs a tie-breaker mechanism to select links for inclusion in the tree. For $m$CLM to be used in a distributed multicast routing protocol it must be constrained in much the same manner as Dijkstra's algorithm is in MOSPF.

## 5.1 Synopsis of distributed multicast tree construction using $m$CLM

1. Two primary data structures are required, in addition to the Link-State data. The first, Graph Data, contains successively the unbound directed graph, the bounded directed graph and finally the multicast tree (which could include the shortest path tree). The second is a Vertex Structure that represents a link between a Parent and a Vertex. It holds the Link Cost from the Parent to the Vertex, the Path Cost from the (multicast) Source to the Vertex via the Parent, the upstream Associated Interface with the Vertex, a shortest/alternative path indicator and an in-branch/in-tree/not-in-tree indicator.

2. The directed graph is initialised to contain the Source Vertex.

3. Mark the Parent/Vertex link in the directed graph with the lowest Path Cost that does not have its Vertex in the shortest path, to be in the shortest path tree. Do not remove the Parent/Vertex from the directed graph nor update the forwarding cache.

4. Include all neighbours of the Vertex in the directed graph as alternative links. If the Vertex is the calculating router then set the Associated Interface of each neighbour to the link that connects it with the Vertex. If the Vertex is downstream of the calculating

router then set the Associated Interface of each of its neighbours to that of the Vertex. Otherwise set the associated interface to NULL.

5. The above two steps are repeated until all vertices are marked in the shortest path tree, when the unbounded directed graph will be complete. Note that tie breakers must be used to choose links when there are more than one of the same cost.

6. The "delay bound" for the multicast ($dbM$) is the shortest Path Cost from the Source to the multicast destination Vertex furthest from the Source. Remove all entries from the directed graph with a Path Cost greater than $dbM$. If there are non-multicast vertices with the same Path Cost, remove these also. In the example above, the multicast node furthest from the Source is node $D$, with a path cost of 56. All vertices in the unbound directed graph with a Path Cost greater than this are deleted to give the bounded directed graph. Set RemainingDelay to the bound $dbM$.

7. From the directed graph construct the multicast delivery tree as follows:

   - Starting with the Vertex furthest from the Source, choose the directed graph entry for the Vertex with the lowest Link Cost, that has a Path Cost less than or equal to the RemainingDelay and mark it as "in-branch". If there are two or more links with the same cost then use a tie breaker to decide which link to use. As links are added to the branch decrement RemainingDelay by their Link Cost. If there are no available links, back-track to the previous Vertex to look for an alternative route. Repeat this process recursively, using the Parent of each Vertex as the next Vertex, until either the Source or the existing multicast delivery tree is reached. Once a branch has been established each Vertex in the branch is marked as being "in tree", starting with the Vertex nearest to the Source or the existing tree, moving back down the branch. As each Vertex is marked "in tree" check to see if it must be added to the forwarding cache, i.e. is it downstream of the calculating router and is a multicast destination. If necessary update the forwarding cache with the Associated Interface for the Vertex.

     For example, the bounded directed graph calculated by router H for the network

| Parent | Vertex | Path Cost | Link Cost | Path Type | In-Tree | AI |
|---|---|---|---|---|---|---|
| G | A | 36 | 36 | shortest | no | NULL |
| G | B | 34 | 34 | shortest | in-tree | NULL |
| G | F | 17 | 17 | shortest | in-tree | NULL |
| F | E | 37 | 20 | shortest | no | NULL |
| F | H | 39 | 22 | shortest | in-tree | NULL |
| B | A | 43 | 9 | alternative | in-tree | NULL |
| B | C | 49 | 15 | shortest | no | NULL |
| B | H | 50 | 16 | alternative | no | NULL |
| A | B | 45 | 9 | alternative | no | NULL |
| E | D | 68 | 31 | alternative | no | NULL |
| H | B | 55 | 16 | alternative | no | B |
| H | C | 51 | 12 | alternative | in-tree | C |
| H | D | 56 | 17 | shortest | in-tree | D |
| C | D | 98 | 49 | alternative | no | NULL |
| C | H | 61 | 12 | alternative | no | NULL |

Figure 6: Graph Data for Router H.

illustrated in Fig 1 will contain two entries for routers downstream of H that are "in-branch" which become "in-tree" (Fig 6). When the branch GD is added to the tree, the forwarding cache will be updated to include the Associated Interfaces of these two entries.

- Repeat the previous step for each successive Vertex that is not already in the multicast tree that is currently the furthest from the Source on a shortest path, until all vertices are in the multicast delivery tree.

# 6 Conclusions

The CLM heuristic was not designed for use in connectionless networks that have only single metric links. Rather, it was designed to minimise the cost of multicast delivery trees that are delay bound in connection oriented networks.

Our evaluation work has shown that by making a simple modification to CLM, the cost of multicast trees in single metric networks can be reduced without increasing the cost of any path beyond that to the most distant multicast node. In circumstances where the modified heuristic generates multicast trees that are more expensive than those built using Dijkstra's algorithm, the protocol can simply fall back to the pruned Dijkstra tree, since this will have already been calculated.

The cost saving of *m*CLM multicast trees is sufficiently significant for it to be considered as an alternative to Dijkstra's algorithm in appropriate multicast routing protocols, such as MOSPF [9].

The problem of scaling, and hence tree construction costs, remains with this approach as it does with MOSPF and DVMRP, which makes it unsuitable for sparse multicasting. As the results presented here indicate, *m*CLM generates greater cost savings as the multicast tree size grows, making it ideal for dense multicasting.

Further tree construction costs and bandwidth savings might also be made by considering an approach similar to the ideas behind CBT and PIM applied to *m*CLM dense mode multicasting, where the delay requirement could be relaxed. Some evaluation work in this area has already been done [11] by a collegue at UKC, but we have yet to assess the costs of tree construction verses tree usage.

The directed graph calculated in the first stage of the algorithm contains alternative paths within the cost bound which could be used for load sharing, although there will be an increase in control traffic to do this.

Because the CLM was designed for networks with two metrics per link, the modified version offers an inherent evolutionary path from single metric to two metric multicast routing. Such a requirement might arise where there is a need to minimise the monetary cost of a multicast tree while keeping a bound on its maximum delay. Such a requirement could be specified by combining IP Types of Service in a multicast delivery request (a feature removed by [1]). It would also mean that network routers would have to exchange two metrics per link to construct routing tables for the network.

The need for delay bound minimal cost multicast routing is left for debate (of which there seems to be plenty), although the requirement for it is being taken seriously and several papers that propose solutions have been published, e.g. [15], [8] ,[19], [18] and [12]. What we have shown here is that a solution to a problem that is still only theoretical may have a practical application in today's Internet.

# 7 Acknowledgements

We would like to acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC) for our work on multicast routing (Grant ref. GR/K55837).

This report was presented at the IDMR meeting of the 36th IETF, Montreal, Canada, in June 1996.

# References

[1] P. Almquist. Type of Service in the Internet Protocol Suite. RFC 1349, july 1992.

[2] A.J. Ballardie, P.F. Francis, and J. Crowcroft. Core based trees. *Computer Communications Review*, 23(4):85–95, 1993.

[3] J.S. Crawford. Multicast routing: evaluation of a new heuristic. Master's thesis, University of Kent at Canterbury, September 1994.

[4] S.E. Deering, D. Estrin, D. Farinacci, V. Jacobson, C-G. Liu, and L. Wei. An architecture for wide area multicast routing. *Computer Communications Review*, 24(4):126–135, October 1994.

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

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

[7] R.M. Karp. *Reducibility among combinatorial problems.* Plennum Press, New York, 1972.

[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] J. Moy. OSPF version 2. RFC 1247, march 1994.

[11] S Parkinson. Multicast routing in the internet: evaluating proposed routing mechaisms. Master's thesis, University of Kent at Canterbury, September 1995.

[12] Q. Sun and H. Langendoerfer. Efficient Multicast Routing for Delay-Sensitive Applications. In *Proceedings of 2nd International Workshop on Protocols for Multimedia Systems (PROMS'95)*, pages 452–458, October 1995.

[13] 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.

[14] A.G. Waters. *Multi-party communication over packet networks.* PhD thesis, University of Essex, UK, 1996.

[15] A.G. Waters and J.S. Crawford. Low-cost ATM Multicast Routing with Constrained Delays. *Submitted for publication*, 1996.

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

[17] L. Wei and D. Estrin. The Trade-offs of Multicast Trees and Algorithms. 93-560, University of Southern California Computer Science Department, 1993.

[18] 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.

[19] 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.