Pricing Multicasting in More Practical Network Models

Micah Adler* Dan Rubenstein†

Abstract

The problem of designing efficient algorithms for sharing the cost of multicasting has recently seen considerable attention. In this paper, we examine the effect on the complexity of pricing when two practical considerations are incorporated into the network model. In particular, we study a model where the session is offered at a number of different rates of transmission, and where there is a cost for enabling multicasting at each node of the network. We consider two techniques that have been used in practice to provide multiple rates: using a layered transmission scheme (called the *layered paradigm*) and using different multicast groups for each possible rate (called the *split session paradigm*). We demonstrate that the difference between these two paradigms has a significant impact on the complexity of pricing multicasting.

For the layered paradigm, we provide a distributed algorithm for computing pricing efficiently in terms of local computation and message complexity. For the split session paradigm, on the other hand, we demonstrate that this problem can be solved in polynomial time if the number of possible rates is fixed, but if the number of rates is part of the input, then the problem becomes NP-Hard even to approximate. We also examine the effect of delivering the transmissions for the various rates from different locations within the network. We show that in this case, the pricing problem becomes NP-Hard for the split session paradigm even for a fixed constant number of possible rates, but if layering is used, then it can be solved in polynomial time by formulating the problem as a totally unimodular integer program.

1 Introduction

Multicast transmission offers tremendous savings in network bandwidth over unicast transmission for applications that deliver the same content to multiple customers by allowing these customers to "share" the transmission on common access links [8]. However, this sharing of link bandwidth significantly complicates the issue of pricing [9]. Charging all receivers equally is not an adequate solution, since some receivers might be charged more than they would be willing to pay. If such a receiver drops out of the multicast session, the remaining receivers would be charged more than if a lower, acceptable price had been offered to the exiting receiver. An alternative approach to pricing that has received considerable attention recently [10, 15, 17, 21, 22] is to have each receiver place a bid for the content. The network uses these bids to determine the set of receivers that obtain the content, as well as the price these accepted receivers pay. The price charged to an accepted receiver can be no more than its bid, but for reasons discussed below, it is often advantageous to charge receivers a smaller price. The policy that the network uses to make these decisions is referred to as a *pricing mechanism*. The task of making these decisions using a specific pricing mechanism is referred to as *realizing* that pricing mechanism.

An algorithm for realizing a pricing mechanism in a distributed environment such as the Internet should be efficient in terms of both the computation performed at the distributed nodes of the network, as well as the communication between these nodes. Identifying these types of algorithms was first addressed by Feigenbaum, Papadimitriou and Shenker [10]. They consider two pricing mechanisms: *Marginal Cost* and *Shapley Value*, and provide efficient algorithms for the Marginal Cost mechanism, as well as algorithms and lower bounds on the efficiency of algorithms for the Shapley Value mechanism.

In this paper, we address the model used for the work in this area. In particular, the previous work on pricing mechanisms for multicasting [10, 15, 17, 21] uses a transmission model with a number of simplifying assumptions. We here address two of these assumptions: (1) that there is only one possible rate of transmission of the multicast session, and (2) that multicasting is possible at every node of the network at no cost. Both multiple rate sessions and the fact that not every node of the network is capable of multicasting are very real concerns in the current Internet, and thus the ability to realize pricing mechanisms in models that account for these concerns is central to the task of designing al-

*University of Massachusetts, Department of Computer Science, University of Massachusetts, Amherst, MA 01003-4610. E-mail: micah@cs.umass.edu.

†Columbia University, Department of Electrical Engineering, 500 W. 120th Street, New York, NY 10027. E-mail: danr@ee.columbia.edu.

gorithms for pricing mechanisms. Incorporating these issues into the transmission model makes the task of realizing pricing mechanisms for multicasting considerably more challenging.

In the transmission model of this paper, we assume that a multicast session can be sent at any of a set of ℓ pre-determined rates $\rho_1 \leq \rho_2 \leq \cdots \leq \rho_\ell$. We consider two of the most common techniques for providing multiple rates that are used in practice: using a separate multicast *group* for each possible rate [6], and using a separate *layer* for each rate, where layer 1 has rate ρ_1, layer i, $1 < i \leq \ell$, has rate $\rho_i - \rho_{i-1}$, and to obtain rate ρ_j, a receiver is sent layers $1 \ldots j$ [3, 4, 20, 27]. We refer to the first technique as the *split session* paradigm, and the second as the *layered* paradigm. For both paradigms, each receiver places a bid per rate indicating its willingness to pay for delivery at that rate. In addition to determining the set of accepted receivers for the multicast session, the network must now also determine, for each user, what rate is obtained. We here demonstrate that the seemingly small difference between these paradigms has a significant effect on the complexity of realizing pricing mechanisms.

Our model also assumes that for each node of the network, there is a cost for *enabling* that node, i.e., for making it capable of multicasting. If a node is enabled, it can forward any number of copies of a session it receives; otherwise it cannot forward more copies than it receives. Note that the cost of enabling a node may be offset by the reduced cost of only having to deliver one copy of the session to that node. This aspect of our model incorporates recent approaches to multicasting such as network overlays and application layer multicasting [5, 7, 16, 18, 26]. To realize a pricing mechanism in this model, the network must still determine the set of accepted receivers, but which set is chosen is now influenced by the cost of enabling multicasting at each node. Furthermore, the network must choose the set of nodes of the network to enable.

As in [10], many of our results assume that there is a single directed multicast tree that defines the routes used by all transmissions. [10] demonstrates that without this assumption, even when there is no cost for enabling a node, and when there is only a single possible rate, it becomes NP-Hard to find constant factor approximations to the pricing mechanism we consider here (the problem becomes a version of the Prize Collecting Steiner Tree problem [19, 12]). Sophisticated techniques for approximating other pricing mechanisms when the routes can vary depending on what receivers are accepted are provided in [17]. Incorporating such techniques into the more involved network model we consider here is an interesting open problem. We also point out that while the hardness result of [10] indicates that removing the single tree assumption entirely leads to intractable problems, it does not rule out the possibility of efficient algorithms for a more modest generalization which we describe below.

1.1 Summary of results

In this paper, we focus on the Marginal Cost mechanism [22]. While this is only one of several important mechanisms, it serves as a good test case: we believe that our results for the Marginal Cost mechanism provide important insights on the effect of the practical network considerations we study for pricing mechanisms in general. We start by introducing a straightforward generalization of the Marginal Cost mechanism to scenarios where there are multiple possible rates, and prove that this generalization has the same properties as the mechanism for single rate scenarios. To the best of our knowledge, this is the first pricing mechanism that has been defined for multicast sessions with multiple rates. Auction mechanisms for multiple goods where there is *no* cost for each additional copy of a good delivered have been studied in [13]. We define the new Marginal Cost mechanism mechanism in Section 2, but note here that any algorithm for this mechanism can also be used for the simple mechanism that chooses the network configuration that maximizes the network profit, and every accepted receiver pays exactly what it bid for the service it receives. The profit obtained by such a mechanism is referred to as the *network welfare*.

In Section 3, we consider algorithms for realizing Marginal Cost in the layered paradigm with costs for enabling nodes for multicast. We provide a distributed algorithm for this problem that is efficient in terms of the amount of local computation performed at each node, and only requires three messages per edge of the multicast tree.

We also study realizing the Marginal Cost mechanism for the split session paradigm with costs for enabling nodes. We demonstrate that our algorithm for the layered case can be adapted to provide a solution for this case. However, the computation required of the adapted algorithm becomes proportional to 2^ℓ, and thus, this algorithm is only applicable for the case where ℓ is small. We also demonstrate that we should not expect to find an efficient algorithm for large ℓ. In particular, we demonstrate that it is NP-Hard to determine even any reasonable approximation to the network welfare when the number of possible rates is part of the input. This result holds even in networks without a cost for enabling multicasting costs. This hardness result is significant in that it demonstrates that in terms

of realizing pricing mechanisms, the layered paradigm enjoys a considerable advantage over the split session paradigm.

Finally, in Section 4, we turn to the question of removing the assumption that there is a single fixed multicast tree. We consider the effect of having a single fixed multicast tree for every layer or group comprising the session, but the trees for the different transmissions need not be the same. We demonstrate that in this case, maximizing network welfare for the split session paradigm becomes NP-Hard even for the case of a constant number of possible rates, and no cost for enabling multicasting. Somewhat surprisingly, we find that in the multiple tree case, the Marginal Cost mechanism for the layered paradigm can be realized in polynomial time if there is no cost for enabling multicasting. We demonstrate this by showing that this problem can be expressed as a totally unimodular integer program.

2 Network Model and Optimization Problems

We consider the problem of offering delivery of a single multicast session, in isolation, to a set $R \subset N$ of receivers, over a network modeled as a directed graph $G = (N, E)$. The session emanates from a source $s \in N$, and is delivered to the receivers via the edges and vertices of G. When a node $n \in N$ is enabled, any flow of information entering that node can be forwarded on multiple outgoing edges. If a node is not enabled, then each copy of an incoming flow can only be forwarded on a single outgoing edge. There is a cost $\mathbf{c}_n$ for enabling node n. We can also model nodes that cannot be enabled for multicasting by setting their cost to ∞. We here assume that the source s can be enabled for free (and thus is always enabled), although all of our results can be modified to apply when this is not the case.

There is also a cost for using a directed edge $e \in E$, denoted by $\mathbf{c}_e$, an ℓ-dimensional vector. In the split session paradigm, $[\mathbf{c}_e]_j$ (the jth entry of $\mathbf{c}_e$) is the cost at e of providing the jth group. In the layered paradigm, $[\mathbf{c}_e]_j$ is the cost of providing the jth layer. We assume that receiver r expresses its willingness to pay via a bid, denoted by $\mathbf{b}_r$, an ℓ-component vector such that $[\mathbf{b}_r]_j$ indicates the price that r is willing to pay to receive the jth group or layer. In the split session paradigm, a receiver is sent at most one group. In the layered paradigm, if a receiver is sent layer j, it must also be sent layers 1 through $j-1$.

For most of this paper, we make an assumption analogous to that made in [10]: the multicast session uses a single source, and a unique path from that source to each receiver, regardless of the set of receivers or enabled nodes. With this assumption, we can restrict our attention to the nodes and edges of the tree formed by the union of paths from the source to each of the receivers. We refer to this tree as the *multicast tree*. When considering networks where every node is enabled for multicasting, this assumption is justified by the multicast routing strategy, commonly used in practice [8], consisting of a tree of shortest paths from the receivers to the source. In the case of either the layered or the split session paradigm, this kind of routing may be used with different source nodes for the different groups or layers, since this is an effective way to balance the session load throughout the network. Thus, in Section 4, we consider a model where every layer or group uses a fixed multicast tree, but the trees do not have to be the same.

The fixed multicast tree assumption does limit the practical applicability of our model to some scenarios. However, we consider the model of this paper an important step towards understanding algorithms for pricing in such networks. Furthermore, the hardness result from [10] applies to a network where the routing depends on which nodes are enabled, even if the cost of enabling those nodes is zero. Thus, unless $P = NP$, we must either make some restriction on the choice of routes, or take an approach similar to that of [17], which considers approximation algorithms for other pricing mechanisms in the simpler network model.

For the optimization problems we consider, the input is distributed as follows: each node n is informed of $\mathbf{c}_n$, $\mathbf{c}_e$ for each e incident to n, and $\mathbf{b}_r$ for any r located at n. The simplest problem we consider is maximizing network welfare. In that problem, the network must determine a set $R' \subset R$ of receivers that are sent the multicast session, for each $r \in R'$ a rate of transmission, as well as the set of multicast enabled nodes. Each receiver in R' pays exactly what it bid for the group or subset of layers that it receives, and the other receivers pay nothing. The network welfare is the total payments minus the total costs.

2.1 The Marginal Cost Mechanism

A natural definition of a receiver's *satisfaction* is the utility obtained from the service provided by the network minus the amount it must pay to receive this service. The simple network welfare pricing mechanism described above produces a profit for the network equal to the network welfare, but it also gives a receiver an incentive to bid less than its true utility. By bidding a smaller value, a receiver reduces the cost of receiving the session, thereby increasing its satisfaction and reducing overall network profits. What is needed is a *strategy-proof* mechanism: a mechanism where a receiver maximizes its satisfaction by bidding its true

utility. There are a number of other properties that are desirable in a pricing mechanism, including:

- Efficiency: a configuration that maximizes (total utility minus total cost) is chosen.
- No Positive Transfers (NPT): the price that the receiver pays is not negative.
- Voluntary Participation (VP): receivers that are not admitted are not charged anything.
- Consumer Sovereignty (CS): A receiver is always able to guarantee acceptance of a bid if the bid is increased to a sufficiently large value.

It is shown in [22] that the Marginal Cost pricing mechanism is strategy-proof, efficient, NPT, VP, and CS. A drawback to Marginal Cost is that it does not provide another desirable property, called Budget-balance: the amount paid by receivers exactly equals the cost of transmission. Marginal Cost never runs a budget surplus, but may run a deficit. This is one reason to also consider other mechanisms, such as Shapley Value [24], although that mechanism does not satisfy Efficiency. We refer the reader to [10, 22] for motivation and further explanation of these properties. All of the previous work on Marginal Cost assumes that the service being priced is "all or nothing": each receiver either obtains the service at the single possible rate or it obtains nothing. However, the definition of Marginal Cost can be easily extended to multiple rates.

We next provide such a definition. Consider any network G and set of receivers R that wish to join a session under a network model where each receiver i submits a set of bids, $\mathbf{b}_i = \langle b_i^1, b_i^2, \cdots, b_i^n \rangle$, of which at most one is accepted. In the multiple good Marginal Cost mechanism, the network chooses the configuration of the network that maximizes network welfare. The price charged to a receiver i that has an accepted bid of b_i^k is defined to be $\mathcal{M}_i = b_i^k - (P^*(R) - P^*(R \backslash \{i\}))$, where $P^*(X)$ is the maximum network welfare when restricting admission to receivers within the set X. If no bid from i is accepted, then i is charged 0. In other words, the price that receiver i must pay is the amount of the bid that was accepted, minus the amount that the network welfare is increased by receiver i participating in the multicast session. The fact that this definition of Marginal Cost is strategy-proof follows from the well known Vickrey-Clarke-Groves Theorem in auction theory [25, 22]. In addition, we have a simple proof that more directly solves this problem that appears in a technical report version of this paper [1].

3 Efficient Distributed Algorithms for Marginal Cost

In this section, we present an efficient distributed algorithm for realizing the Marginal Cost mechanism. We first provide an algorithm that maximizes network welfare in the layered paradigm. We then demonstrate that it can be modified to maximize network welfare in the split session paradigm, albeit at the cost of an exponential dependence on the number of groups. We then show that our algorithms for both the layered paradigm and the split session paradigm can be converted into algorithms that compute Marginal Cost.

For any node n of the multicast tree, let $\pi_{1,n}$ be the parent node of n in the tree. Let $\pi_{k+1,n}$ be the parent of node $\pi_{k,n}$. We define h_n to be the value of k such that $\pi_{k,n}$ is the source node of the tree. For any node n, let $D(n)$ be the set of children of n in the tree. An important value computed during the course of our algorithm is $S_{j,k}(n)$, which is computed for $0 \leq j \leq \ell$, $1 \leq k \leq h_n$. $S_{j,k}(n)$ is the maximum network welfare of the subtree rooted at node n, minus the cost of transmitting all necessary layers from node $\pi_{k,n}$ to node n, subject to the following two conditions:

- $\pi_{r,n}$ is not enabled for all $1 \leq r < k$. Nodes n and $\pi_{n,k}$ may or may not be enabled, but only the cost of enabling n counts against $S_{j,k}(n)$.
- At most j layers are transmitted from $\pi_{k,n}$ to n (and thus to any node that is a descendent of n.)

In other words, $S_{j,k}(n)$ is the maximum value of the sum of a set of accepted bids that are in the subtree rooted at n, minus the sum of the costs in the subtree rooted at n to deliver those bids, minus the cost of transmitting the necessary layers from node $\pi_{r,n}$ to node n, subject to the two conditions above. Another set of intermediate values used by our algorithm are represented by $\mathbf{c}_{(\pi_{n,k},n)}$, a vector such that $[\mathbf{c}_{(\pi_{n,k},n)}]_j$ is the cost of transmitting one copy of layer j from $\pi_{n,k}$ to n.

We now describe our algorithm, which we call **Max-Layered-Welfare**. For ease of exposition, we here describe the slightly simpler case where the set of possible receivers is exactly the same as the set of leaves of the multicast tree, but it is not hard to modify this to account for the general case.

Algorithm **Max-Layered-Welfare:**

- The source initiates a phase of the algorithm where every node n of the multicast tree sends each of its children $n_i \in D(n)$ the vector $\mathbf{c}_{(\pi_{n,k},n)}$, for each k, $1 \leq k \leq h_n$. Each child n_i uses these values to compute each $\mathbf{c}_{(\pi_{n_i,k+1},n_i)} = \mathbf{c}_{(\pi_{n,k},n_i)} + \mathbf{c}_{n,n_i}$ (i.e., a vector addition).

- Each leaf node n of the multicast tree computes, for each k and j, $1 \le k \le h_n$, $0 \le j \le \ell$, $S_{j,k}(n)$. Each $S_{0,k}(n) = 0$, and then the remainder are computed in order of increasing j, using the formula $S_{j,k}(n) = \max(\sum_{r=1}^{j}[\mathbf{b}_n]_r - \sum_{r=1}^{j}[\mathbf{c}_{(\pi_{n,k},n)}]_r, S_{j-1,k})$.
- The next phase of the algorithm proceeds from the leaves to the source, and each node n_i sends to its parent n, the value $S_{j,k}(n_i)$, for each k and j, $1 \le k \le h_{n_i}$, $1 \le j \le \ell$. The node n sets each $S_{0,k}(n) = 0$, and then computes all other values of $S_{j,k}(n)$, proceeding from $j = 1$ to $j = \ell$, using the formula

$$S_{j,k}(n) = \max\left\{ \sum_{n_i \in D(n)} S_{j,k+1}(n_i), \sum_{n_i \in D(n)} S_{j,1}(n_i) - \mathbf{c}_n - \sum_{r=1}^{j}[\mathbf{c}_{(\pi_{n,k},n)}]_r,\ S_{j-1,k}(n) \right\}. \tag{3.1}$$

- The source node s returns the value $\sum_{n_i \in D(s)} S_{j,1}(n_i)$.

THEOREM 3.1. *The value returned by Algorithm* **Max-Layered-Welfare** *is the maximum possible network welfare under the layered paradigm. Furthermore, the computation performed by any node n requires time $O(\ell h_n |D(n)|)$, and exactly two messages are communicated between node n and its child $n_i \in D(n)$.*

Proof. It is not difficult to implement this algorithm with the stated computation and communication complexity, and thus we here only describe the proof that the value returned by the algorithm is correct. To do so, we prove that for every n, j, and k, the value of $S_{j,k}(n)$ computed by the algorithm is correct. Assuming this, the correctness of the algorithm follows from the fact that the source determines the welfare obtained by sending the optimal number of layers to each of its children. Also note that it is easy to show that the values of $\mathbf{c}_{(\pi_{n,k},n)}$ computed are correct. Thus, we only need to show the correctness of the $S_{j,k}(n)$. The proof is by a double induction on j and the maximum (simple path) distance from n to a leaf. For the base of the induction, we show that $S_{j,k}(n)$ is correct if either $j = 0$, or n is a leaf. When $j = 0$, $S_{j,k}(n) = 0$, since no receiver in the subtree rooted at n can receive any layers. When n is a leaf, but $j > 0$, we see that $S_{j,k}(n)$ is correct by induction on j, since when up to j layers can be sent to node n, either we send all j layers to n, or we use the best solution using less than j layers.

For the inductive step of the double induction, consider $S_{j,k}(n)$ for any k, any layer $j > 0$, and any non-leaf node n. By the inductive hypothesis, we can assume that for every child $n_i \in D(n)$, both $S_{j,k+1}(n_i)$ and $S_{j,1}(n_i)$ are computed correctly, and also that $S_{j-1,k}(n)$ is computed correctly. There are now four cases: either node $\pi_{n,k}$ transmits layer j to node n or it does not, and either node n is enabled, or it is not. If, in the optimal solution, node $\pi_{n,k}$ does not transmit layer j to node n, then regardless of whether n is enabled or not, $S_{j,k}(n) = S_{j-1,k}(n)$. If the optimal solution transmits all j layers to node n, and n is enabled, then $S_{j,k}(n) = \sum_{n_i \in D(n)} S_{j,1}(n_i) - \mathbf{c}_n - \sum_{j=1}^{r}[\mathbf{c}_{(\pi_{n,k},n)}]_r$. This holds because we maximize the welfare at the subtree rooted at n when n is enabled by maximizing the welfare of the subtrees rooted at each of its children subject to the condition that each of them receives the multicast session directly from n. From this maximization, we subtract the cost of enabling n and of transmitting one copy of layers 1 through j from $\pi_{n,k}$ to n. Finally, if the optimal solution transmits all j layers to node n, but n is not enabled, then all layers are unicast through n, and we see that $S_{j,k}(n) = \sum_{n_i \in D(n)} S_{j,k+1}(n_i)$. Since the algorithm takes the maximum of these possibilities, it does in fact compute the correct value of $S_{j,k}(n)$.

The total number of bits communicated in this algorithm is $O(\ell h K)$, where K is the maximum number of bits required to represent any value $S_{j,k}(n_i)$ or $\mathbf{c}_{(\pi_{n,k},n)}$, and h is the height of the tree. While this is not as small as might be hoped for, we prove in the full version of this paper [1] that $\Omega(hK)$ bits are necessary when considering networks with a cost for enabling multicast, even when there is only a single possible rate of transmission, and that $\Omega(\ell K)$ bits are necessary when there are ℓ layers, even when there is no cost for enabling multicasting at any node. These two bounds provide evidence that the linear dependence of the number of bits communicated on both h and ℓ is reasonable.

We next show that algorithm **Max-Layered-Welfare** can be modified to compute the maximum welfare for the split session paradigm. We provide a brief sketch of how to do so. We define $S_{\int,k}(n)$, where $\int$ is any subset of the ℓ groups, analogously to $S_{j,k}(n)$, except that the second condition becomes

- A group s is transmitted from $\pi_{k,n}$ to n only if $s \in \int$.

The algorithm follows along the same lines as **Max-Layered-Welfare**, with small modifications. To compute $S_{\int,k}(n)$ for all $\int$ and k at a node n, the algorithm starts with $\int$ being the empty set, then considers all subsets of size 1, followed by all subsets of size two, until $\int$ contains all groups. The main formula

of the algorithm (analogous to (3)) is as follows:

$$S_{\int,k}(n) = \max \left\{ \sum_{n_i \in D(n)} S_{\int,k+1}(n_i), \max_{\int' \in m(\int)} S_{\int',k}(n), \sum_{n_i \in D(n)} S_{\int,1}(n_i) - \mathbf{c}_n - \sum_{s \in \int} [\mathbf{c}_{(\pi_{n,k},n)}]_s \right\},$$

where $m(\int)$ is the set of all subsets of $\int$ containing exactly one less element. We call the resulting algorithm **Max-Split-Welfare**. The proof of the following follows along the lines of the proof of Theorem 3.1.

THEOREM 3.2. *The value returned by Algorithm* **Max-Split-Welfare** *is the maximum possible network welfare under the split session paradigm. Furthermore, the computation performed by any node n requires time $O(\ell 2^{\ell} h_n |D(n)|)$ and exactly two messages are communicated between node n and its child $n_i \in D(n)$.*

3.1 Computing the Marginal Cost

We now show how to use algorithms **Max-Layered-Welfare** and **Max-Split-Welfare** to compute Marginal Cost. The following information is sufficient for a receiver r to know what it pays and what it receives: the network welfare, r's accepted bid (if any), as well as the maximum network profit obtainable when r is not admitted. We provide this information to each receiver r using a single downward phase of the algorithm from the root to the leaves of the tree.

The network welfare computed during the upward phase is simply passed back down the tree during the downward phase. To inform every receiver of which bid is accepted, every node stores, during the upward phase, which term of the maximization in (3) provides the largest value. The root informs its children of what configuration they are in, which, combined with the stored information, is sufficient for them to determine the configuration of their children, and so on, until every node knows what configuration it must be in to achieve the maximum network welfare. This informs every node of its closest enabled parent in the multicast tree, whether or not it is enabled, which layers or groups it receives from this enabled parent, and what to forward to its children, if it has any.

More difficult is computing, for each receiver r, the maximum network profit when r is not admitted. This could of course be computed easily using one phase per receiver, but our objective is to provide all the receivers with this information using a single downward phase. We here describe how to achieve this in the layered paradigm, although it is easy to modify what we describe here to work in the split session paradigm. We assume that every node n stores the values $S_{j,k}(n_i)$, for all $n_i \in D(n)$, j and k that it learned during the upward phase. In addition, in the downward phase, each node will compute the quantity $B_{j,k}(n)$, for $0 \le j \le \ell$ and $1 \le k \le h_n$. $B_{j,k}(n)$ is the maximum total network welfare possible, subject to the following conditions:

- The closest ancestor of n that is enabled is $\pi_{n,k}$.
- Node $\pi_{n,k}$ transmits the first j layers to n, but no other layers.
- If node n is an internal node of the tree, no layers are transmitted to any child of node n.
- If node n is a receiver, the profit from that receiver is not included in the network welfare.

Note that the network configurations prescribed by $B_{j,k}(n)$ for $j > 0$ are wasteful in the sense that the layers transmitted to node n are not used by any receiver. Each node n learns every value of $B_{j,k}(n)$ from its parent during the downward phase. Our algorithm for computing Marginal Cost works as follows:

Algorithm **Layered-Marginal-Cost**

- The source node s sends every $n_i \in D(s)$ the value of $B_{j,1}(n_i)$, for $0 \le j \le \ell$, computed using

$$B_{j,1}(n_i) = \left(\sum_{n'_i \in D(s); n'_i \neq n_i} S_{\ell,1}(n'_i) \right) - \sum_{r=1}^{j} [\mathbf{c}_{(s,n_i)}]_r.$$

- Every node n, on receiving $B_{j,k}(n)$, for $0 \le j \le \ell$ and $1 \le k \le h_n$, from its parent, computes $B_{j,k}(n_i)$, for each $n_i \in D(n)$, $0 \le j \le \ell$ and $1 \le k \le h_{n_i}$, and sends these values to n_i. For the case where $2 \le k \le h_{n_i}$, node n uses

$$B_{j,k}(n_i) = \max_{t=j}^{\ell} \left(B_{t,k-1}(n) + \sum_{n'_i \in D(n); n'_i \neq n_i} S_{t,k}(n'_i) \right) - \sum_{r=1}^{j} [\mathbf{c}_{(n,n_i)}]_r.$$

For the case where $k = 1$, node n uses

$$B_{j,1}(n_i) = \max_{t=j}^{\ell} \left(\max_{u=1}^{h_n} B_{t,u}(n) + \sum_{n'_i \in D(n); n'_i \neq n_i} S_{t,1}(n'_i) \right) - \sum_{r=1}^{j} [\mathbf{c}_{(n,n_i)}]_r - \mathbf{c}_n.$$

- Each receiver r returns $\max_{k=1}^{h_r} B_{0,k}(r)$.

THEOREM 3.3. *The value returned by each receiver r in Algorithm* **Layered-Marginal-Cost** *is the maximum possible network welfare obtainable under the layered paradigm, without admitting r. Furthermore, the computation required of any node can be performed in polynomial time and each node n sends only one message to each child $n_i \in D(n)$.*

Proof. Proving the bounds on the amount of computation and communication is straightforward, and thus we here focus on proving correctness. Note first that if receiver r obtains the correct values of $B_{0,k}$, for $1 \leq k \leq h_r$, then it returns the correct value, since in the optimal configuration without r, there is some k such that $\pi_{r,k}$ is enabled (recall that the source is always enabled), and so for some value of k, $B_{0,k}(r)$ contains the maximum welfare.

We prove that all values of $B_{j,k}(n)$ are correct by induction on h_n. For the base case, consider some n that is a child of the source. The maximum network welfare obtainable without any profit provided by the subtree rooted at n is the sum of the maximum welfares obtainable in the subtrees rooted at the other children of the source. Thus, since we require that layers 1 through j are sent to n, the algorithm computes the correct value of $B_{j,k}(n)$ for any n that is a child of the source. For the inductive step, we assume that node n receives the correct values of $B_{j,k}(n)$ from its parent, and show that this implies that for any $n_i \in D(n)$, n sends the correct values of $B_{j,k}(n_i)$ to n_i. For $1 \leq t \leq \ell$, let $T_{t,k}(n_i)$ be the network welfare of the optimal configuration where t layers reach n, $\pi_{n,k-1}$ is the first enabled ancestor of n, and no receiver in the subtree rooted at n_i receives any layers. In the case where $k = 1$, node n is enabled.

We first consider the case where $k > 1$, i.e., when node n is not enabled. In this case, the value of $T_{t,k}(n_i)$ is $B_{t,k-1}(n)$ plus the sum of the maximum welfares possible in the subtrees rooted at each $n'_i \in D(n)$, $n'_i \neq n_i$, when the cost of sending any required layers (up to t) from $\pi_{n,k-1}$ to n'_i is included. The amount added for each n'_i is exactly the value $S_{t,k}(n'_i)$, computed during the up-phase of the algorithm **Max-Layered-Welfare**. Thus, $T_{t,k}(n_i) = B_{t,k-1}(n) + \sum_{n'_i \in D(n); n'_i \neq n_i} S_{t,k}(n'_i)$. The network configuration that achieves the correct value of $B_{j,k}(n_i)$ is the configuration that maximizes $T_{t,k}(n_i)$, subject to the requirement that $j \leq t \leq \ell$. Since we require that j layers travel to n_i, we have $B_{j,k}(n_i) = \max_{t=j}^{\ell} T_{t,k}(n_i) - \sum_{r=1}^{j} [\mathbf{c}_{(n,n_i)}]_r$. In Algorithm **Layered-Marginal-Cost**, since we assume that node n receives the correct values of $B_{j,k}(n)$ from its parent, node n sends the correct values of the $B_{j,k}(n_i)$ to n_i.

We next consider the case where $k = 1$, i.e., when node n is enabled. In this case, $T_{t,1}(n_i) = \max_{u=1}^{h_n} B_{t,u}(n) + \sum_{n'_i \in D(n); n'_i \neq n_i} S_{t,1}(n'_i) - \mathbf{c}_n$, since in the optimal configuration for $T_{t,1}(n_i)$, there is some first ancestor of n, $\pi_{n,u}$, that is enabled (recall that the source is always enabled). The network configuration that achieves the correct value of $B_{1,k}(n_i)$ is the configuration that maximizes $T_{t,1}(n_i)$, subject to $j \leq t \leq \ell$. Since we require that j layers travel to n_i, we have $B_{j,1}(n_i) = \max_{t=j}^{\ell} T_{t,1}(n_i) - \sum_{r=1}^{j} [\mathbf{c}_{(n,n_i)}]_r$. In Algorithm **Layered-Marginal-Cost**, since we assume that node n receives the correct values of $B_{j,k}(n)$ from its parent, node n sends the correct values of the $B_{j,1}(n_i)$ to n_i.

3.2 Hardness of the split session paradigm

We next examine the computational complexity of maximizing network welfare in the split session paradigm. The algorithm **Max-Split-Welfare** has a running time that is exponential in ℓ, but runs in polynomial time for any fixed ℓ. We here demonstrate that if ℓ is given as part of the input, then the problem becomes NP-Hard to even approximate. Thus, we should not expect a significantly more efficient algorithm for maximizing network welfare in the split session paradigm.

THEOREM 3.4. *If ℓ is specified as part of the input, then there is no polynomial time $\ell^{1-\epsilon}$-approximation algorithm, for any $\epsilon > 0$, for the problem of maximizing network welfare in the split session paradigm, unless $NP = ZPP$. This holds even if there is no cost for enabling multicast at the nodes of the network.*

Proof. We show that such an approximation algorithm could be converted into a polynomial time $|V|^{1-\epsilon}$-approximation algorithm for the largest independent set in a graph. Since [14] demonstrates that such an algorithm only exists if $NP = ZPP$, the theorem follows. Given an arbitrary input graph $G = (V, E)$, we construct an input for the network welfare maximization problem. This input can be constructed for any network topology where there is a subset R of $|E| + |V|$ receivers that share the first edge from the source, but each has a distinct last edge from the source. Furthermore, this input can be constructed even if we are subject to the requirement that edge costs are proportional to the rate of the flow crossing that edge. In other words, if for any edge e, there is a constant $c(e)$, such that the cost for any group g to use e is $B(g)c(e)$, where $B(g)$ is the rate of g.

For each vertex $v \in V$, we have one group $g(v)$. All that we require of the group rates is that $B(g(v)) \geq 2 \cdot deg(v) - 1$, where $B(g(v))$ is the rate of $g(v)$ (we assume that there are no degree 0 vertices in V: such vertices can easily be dealt with separately). For each

edge $e = (u, v) \in E$, we have one receiver $r(e) \in R$, and for each vertex $v \in V$, we have one receiver $r(v) \in R$. For each edge of the communication network, the cost of a group using that edge is either equal to the rate of the group, or 0. In particular, the cost for sending $g(v)$ over the shared first edge from the source is $B(g(v))$, the cost of $g(v)$ traversing the last edge from the source to any receiver in R is also $B(g(v))$, and the cost of all other edges in the network is 0. Each receiver $r(u)$ bids $B(g(v))$ for group $g(v)$, for $u \neq v$, and $2B(g(u)) - 2 \cdot deg(u) + 1$ for group $g(u)$. Each receiver $r(e)$, where $e = (u_1, u_2)$, bids $B(g(v))$ for group $g(v)$, for $u_1, u_2 \neq v$, and $B(g(v)) + 2$, for $v \in \{u_1, u_2\}$.

We next show that maximizing the network welfare for the described input is equivalent to finding the largest independent set in the graph G. Assume first that G has an independent set I of size $|I|$. To convert this into a solution to the network welfare maximization problem, the network sends group $g(v)$, for each $v \in I$, to each $r(e)$ such that $e = (u, v)$ for any u. This is a valid solution, since I is an independent set. Also, the network sends such $g(v)$ to each $r(v)$. The revenue from group $g(v)$ for $v \in I$ is $2B(g(v)) - 2 \cdot deg(v) + 1$ from $r(v)$, plus $deg(v)(B(g(v)) + 2)$, from all $r(e)$. The total cost for group $g(v)$ is $B(g(v))(deg(v) + 2)$, for a total welfare of 1 for each such group. Thus, the total welfare is exactly I.

Also, any solution to the network welfare maximization problem with profit P can be converted to an independent set of size P. To do so, we consider only those groups that result in a profit. For any such group $g(v)$, the only way to achieve a profit is if every receiver that bids more than $B(g(v))$ actually receives group $g(v)$. If this is done, then the total profit for group $g(v)$ is exactly 1. No two groups that have a profit of 1 can correspond to vertices that share an edge, and thus the set of groups that achieve a profit must correspond to an independent set of size at least P. The theorem follows.

4 The multiple tree case

In this section, we examine the effect of removing the assumption that there is a single multicast tree that defines the routing for all groups or layers. In particular, we consider the case where for every layer or group, there is a single fixed tree that must be used, but the trees for the different layers or groups do not have the same tree, either because they originate from different sources, or because they use a different routing scheme. We refer to this as the *multiple tree* case. This is actually a property that is desirable in practice, since it helps balance the network load. Unfortunately, many problems become NP-Hard with this relaxation, even if there is no cost for enabling multicasting. Throughout this section we assume a model where there is no such cost.

We first demonstrate that maximizing network welfare becomes more difficult for the split session paradigm in the multiple tree case. Recall that in the single tree case, we have an algorithm for computing Marginal Cost in the split session paradigm that requires computation that is exponential in the number of sessions, but is polynomial for the case of a constant number of sessions.

THEOREM 4.1. *In the multiple tree case, the problem of maximizing network welfare using the split session paradigm is NP-Hard for any fixed constant number of groups larger than 12. This holds even if there is no cost for enabling multicast at any node of the network.*

Proof. We reduce from BOUNDED-3-SAT, the version of 3-SAT where every variable appears at most 5 times, and every clause has exactly 3 literals. This restriction of 3-SAT is still NP-Complete [11]. Given a bounded 3-SAT formula Φ, we first assign a color to every literal, such that if two literals are assigned the same color, then they never appear in the same clause, and they are not negated versions of the same variable. This can be done using 12 colors in a greedy fashion. In particular, for each variable in turn, we assign the colors for both literals of that variable at the same time. Since those literals can appear in a total of at most 5 clauses, at most 10 colors are ruled out by previous assignments to the other two literals in those 5 clauses. Since there are 12 colors, there is always at least two colors available to color the two new literals.

We then construct the welfare maximization problem. We here provide the proof for the case where there is a single source, and each group has an arbitrary multicast tree rooted at that source. However, it is not difficult to modify this reduction so that it holds when the input is required to be a weighted directed graph, the source for each group is placed at some vertex of the graph, and the tree for a group is defined by shortest paths from the source to all the receivers. Furthermore, we here assume that for every edge of the communication network, the cost to use that edge is the same for all groups, although this is also easy to modify so that costs are proportional to rate. In the network, there are two nodes for each variable x_i, labeled x_i^1 and x_i^2. There are also five nodes for each clause c_j, labeled c_j^1 through c_j^5. The edges of the network are as follows. There is an edge from the source node to x_i^1, for each variable x_i. These edges have cost 6. There is an edge from x_i^1 to x_i^2 for each variable x_i. These edges have no cost. For any variable x_i and clause c_j, there is an edge from x_i^1 to c_j^1 if x_i appears in c_j. These edges have cost 2. For

each clause c_j, there is an edge of no cost from c_j^1 to c_j^k, for $k \in \{2,3,4,5\}$. Finally, for each clause c_j, there is an edge from the source to c_j^k, for $k \in \{2,3,4,5\}$. These edges have cost ∞. For each variable x_i, there is an edge from the source to x_i^2 with cost ∞.

There are four receivers for each clause c_j, one at each c_j^k, $k \in \{2,3,4,5\}$. There is also one receiver for each variable x_i at x_i^2. There are no other receivers. The receivers at any c_j^k, for $k \in \{3,4,5\}$ each make a bid of 1 for any group that is successfully delivered to that receiver. The receivers at any c_j^2 each bid 2 for any group. The receivers at any x_i^2 each bid 6 for any group. There are a total of 12 groups, one for each of the colors. We next describe the multicast trees for each group. For group t, the tree is routed from the source to every x_i^1 such that either of the literals x_i or $\bar{x_i}$ is assigned to color t. Note that since x_i and $\bar{x_i}$ are assigned different colors, there are exactly two groups that are routed from the source to each node x_i^1.

From each x_i^1 that group t is routed to, t continues to x_i^2. From x_i^1, t is also routed to every c_j^1 where c_j is a clause that contains the literal of x_i that is assigned to color t. Note that since literals appearing in the same clause are assigned to different groups, the routing for any group t is in fact a tree (even though the underlying network is not a tree). Also note that exactly 3 groups are routed to each c_1. All three of these groups are routed on the edge from c_j^1 to c_j^2. Also from c_j^1, the three groups diverge: one group each goes to c_j^3, c_j^4, and c_j^5. The trees described thus far allow exactly one group to be routed to each node c_j^k, for any c_j and $k \in \{3,4,5\}$ (specifically, the group that is associated with the color of one of the literals within the clause c_j). The remainder of the groups are routed to c_j^k by using the edge directly from the source to c_j^k. Similarly, the remainder of the groups are routed to c_j^2 and x_i^2 using the edge directly from the source. This completes the construction of the multicast problem. The network is depicted in Figure 1. The theorem now follows from the following two claims; their proofs appear in [1].

CLAIM 1. *If the formula Φ is satisfiable, then the resulting multicast problem can achieve network welfare at least C, where C is the number of clauses in Φ.*

CLAIM 2. *If, in the multicast problem resulting from the formula Φ, it is possible to achieve network welfare at least C, where C is the number of clauses in Φ, then the formula Φ is satisfiable.*

4.1 An algorithm for the layered paradigm

The increased difficulty in finding good solutions for the split session paradigm might suggest that maximizing welfare in the layered paradigm in the multiple tree case is also NP-Hard, but this turns out to not be the case.We next provide a polynomial time algorithm for this problem by demonstrating that the welfare maximization problem can be stated as a totally unimodular integer program. Such integer programs can be solved in polynomial time using linear programming techniques (see for example [23]). This leads to a centralized polynomial time algorithm that finds the optimal solution. This may require significant communication overhead in a distributed setting, and thus an interesting open problem is finding a communication efficient distributed algorithm for this problem. Some approaches to solving linear programming problems in a distributed fashion have been studied in [2], but in our scenario, the objective function is distributed across the users, and thus the techniques from [2] do not apply here. We also note that this algorithm can also be used to compute the Marginal Cost allocation in polynomial time by computing the cost for every receiver individually.

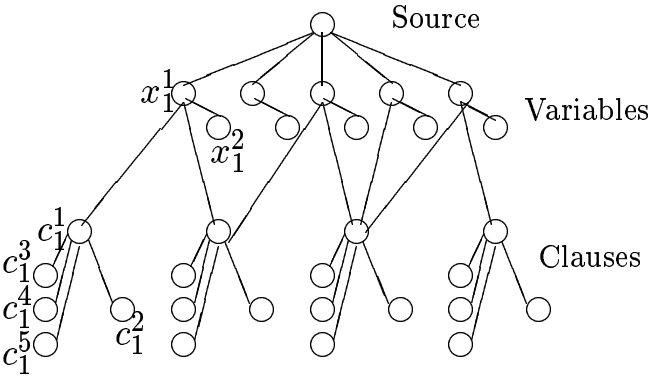

Figure 1: The network used in the proof of Theorem 4.1.

THEOREM 4.2. *There is a polynomial time algorithm that maximizes network welfare in the layered paradigm of the multiple tree case.*

Proof. We show that this problem can be expressed as a totally unimodular integer program. For each edge e and each layer i that can travel on edge e, we have a variable x_{ei}, such that $0 \leq x_{ei} \leq 1$. If $x_{ei} = 1$, this represents that layer i traverses edge e; otherwise $x_{ei} = 0$. To ensure that we have a valid multicast tree, for any edges e and e' such that in the tree for layer i, edge e connects a node n to its parent, and e' connects n to a child, we include the constraint $x_{e'i} \leq x_{ei}$. To ensure that we have a valid layered flow, for any pair of edges e and e' such that e is the last edge that brings a layer i to a receiver in the multicast tree, and e' is the last edge that brings a layer $i+1$ to that same

receiver, we have the constraint $x_{e'(i+1)} \leq x_{ei}$. Any feasible solution to this integer program defines a valid set of layered multicast trees. Our objective function is to maximize

$$\sum_{i,e \in leaf(i)} x_{ei} \cdot p_{e,i} - \sum_{e,i} x_{ei} \cdot c_{e,i}$$

where $c(e,i)$ is the cost of sending layer i on edge e, $leaf(i)$ is the set of edges that bring layer i to its receivers, and $p_{e,i}$ is the profit provided by bringing layer i to the receiver served by e.

Note that this integer program has a constraint matrix A such that all entries are ± 1, each row of A has at most two entries, and if a row contains two entries one is $+1$ and the other is -1. Thus, the determinant of any submatrix of A has either no nonzero terms, a single nonzero term equal to ± 1, or one term equal to 1 and one term equal to -1. Therefore, A is in fact totally unimodular.

References

[1] M. Adler and D. Rubenstein. Pricing Multicasting in More Practical Network Models. Technical report, University of Massachusetts UM-CS-2001-016, April 2001.

[2] Y. Bartal, J. Byers, and D. Raz. Global Optimization Using Local Information with Applications to Flow Control. In 38th IEEE Symp. on Foundations of Computer Science, pages 303–312, 1997.

[3] J. Byers, M. Luby, and M. Mitzenmacher. Fine-Grained Layered Multicast. In *Proceedings of IEEE INFOCOM'01*, Anchorage, AK, April 2001.

[4] J. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A Digital Fountain Approach to Reliable Distribution of Bulk Data. In *Proceedings of SIGCOMM'98*, Vancouver, Canada, September 1998.

[5] Y. Chawathe, S. McCanne, and E. Brewer. RMX: Reliable Multicast for Heterogeneous Networks. In *Proceedings of IEEE INFOCOM'00*, Tel-Aviv, Israel, March 2000.

[6] S. Cheung, M. Ammar, and L. Xue. On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution. In *Proceedings of IEEE INFOCOM'96*, San Francisco, CA, March 1996.

[7] Y. Chu, S. Rao, and H. Zhang. A Case for End System Multicast. In *Proceedings of ACM SIGMETRICS'00*, Santa Clara, CA, May 2000.

[8] S. Deering and D. Cheriton. Multicasting routing in datagram internetworks and extended LANs. *ACM Trans. on Computer Systems*, 8(2):85–110, May 1990.

[9] C. Diot, B. Levine, B. Lyles, H. Kassem, and D. Balensiefen. Deployment Issues for the IP Multicast Service and Architecture. *IEEE Network Magazine*, Jan/Feb 2000.

[10] J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the Cost of Multicast Transmissions. In *Proceedings of the Thirty Second Annual ACM Symposium on Theory of Computing (STOC)*, Portland, Oregon, May 2000.

[11] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, 1979.

[12] M. Goemans and D. Williamson. A General Approximation Technique for Constrained Forest Problems. *SIAM J. Comput.*, 24:296–317, September 1995.

[13] A. Goldberg and J. Hartline. Competitive Auctions for Multiple Digital Goods. Technical report, Technical Report STAR-TR-00,05-02, May 2000.

[14] J. Håstad. Clique is Hard to Approximate Within $n^{1-\epsilon}$. In 37th IEEE Symp. on Foundations of Computer Science, pages 627–636, 1996.

[15] S. Herzog, S. Shenker, and D. Estrin. Sharing the "Cost" of Multicast Trees: An Axiomatic Analysis. *Transactions on Networking*, December 1997.

[16] The Inktomi Overlay Solution for Streaming Media Broadcasts. Inktomi White Paper, 2001.

[17] K. Jain and V. Vazirani. Applications of Approximation Algorithms to Cooperative Games. In *Proceedings of the Thirty ThirdAnnual ACM Symposium on Theory of Computing (STOC)*, 2001.

[18] J. Jannotti, D. Gifford, K. Johnson, M. Kaashoek, and J. O'Toole. Overcast: Reliable Multicasting with an Overlay Network. In *Proceedings of USENIX*, San Diego, CA, October 2000.

[19] D. Johnson, M. Minkoff, and S. Phillips. The Prize Collecting Steiner Tree Problem: Theory and Practice. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, January 2000.

[20] S. McCanne, V. Jacobson, and M. Vetterli. Receiver Driven Layered Multicast. In *Proceedings of SIGCOMM'96*, Stanford, CA, August 1996.

[21] D. Mitzel and S. Shenker. Asymptotic Resource Consumption in Multicast Reservation Styles. In *Proceedings of ACM SIGCOMM'94*, London, UK, August 1994.

[22] H. Moulin and S. Shenker. Strategyproof Sharing of Submodular Costs: Budget Balance versus Efficiency. *Economic Theory*, To appear.

[23] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Dover Publications, 1982.

[24] L. Shapley. A Value for n-Person Games. In *Contributions to the Theory of Games*, pages 31–40, 1953.

[25] Personal communication with Scott Shenker.

[26] I. Stoica, T. Ng, and H. Zhang. REUNITE: A Recursive Unicast Approach to Multicast. In *Proceedings of IEEE INFOCOM'00*, Tel-Aviv, Israel, March 2000.

[27] L. Vicisano, J. Crowcroft, and L. Rizzo. TCP-like Congestion Control for Layered Multicast Data Transfer. In *Proceedings of INFOCOM'98*, San Francisco, CA, March 1998.