Layered Multicast Rate Control based on Lagrangian Relaxation and Dynamic Programming

Koushik Kar, Leandros Tassiulas

Abstract— **In this paper, we address the rate control problem for layered multicast traffic, with the objective of solving a generalized throughput/fairness objective. Our approach is based on a combination of lagrangian relaxation and dynamic programming. Unlike previously proposed dual-based approaches, the algorithm presented in this paper scales well as the number of multicast groups in the network increases. Moreover, unlike all existing approaches, our approach takes into account the discreteness of the receiver rates that is inherent to layered multicasting. We show analytically that our algorithm converges and yields rates that are approximately optimal. Simulations carried out in an asynchronous network environment demonstrate that our algorithm exhibits good convergence speed and minimal rate fluctuations.**

I. Introduction

In layered multicasting, data is transmitted in multiple layers. The source encodes the signal (usually an audio or video signal) in layers, and a subset of these layers are sent to the receivers, depending on the receiver requirements, and the congestion of the path from the source to the receiver. Layered multicasting is a form of *multirate multicasting*, since different receivers in the same multicast group can receive traffic at different rates.

Multirate or multilayer transmission is the more preferred form of data delivery when receivers of the same multicast group have different characteristics. Typically, multilayer transmission is achieved through hierarchical encoding of real-time signals. In this approach, a signal is encoded into a number of layers that can be incrementally combined to provide progressive refinement. In layered multicasting, the receivers adapt to congestion by adding or dropping layers. With multilayer transmission, the network can be utilized more efficiently, and receivers can receive data that is more commensurate with their capabilities. For discussions on multirate/multilayer transmission, refer to [11], [12].

Note that in layered multicasting, the granularity at which congestion control can be done is determined by the number of layers. Thus, more fine-grained congestion control is possible with larger number of layers. However, the amount of state maintained as well as the processing complexity at the routers will also typically increase as the number of layers increases.

For efficient use of the network, an effective rate control strategy is necessary. The rate control algorithm should ensure that the traffic offered to a network by different traffic sources remain within the limits that the network can carry. Moreover, it should also ensure that the network resources are shared by the competing flows in some fair manner, and that the throughput achieved is high. It may therefore be desirable that the rate control algorithm would steer the network towards a point where some measure of global fairness is maximized. Throughput and fairness definitions are generalized in a nice way by associating utilities with receivers. Utility of a receiver is a function connecting the bandwidth given to the receiver with the "value" associated with the bandwidth. In this paper, we design rate control algorithms such that they maximize the sum of the utilities over all receivers, subject to the link capacity constraints. This objective was proposed recently by Kelly [8], [9]. Various fairness objectives can be realized within this utility maximization framework for different choices of the utility functions.

Recently, there has been a considerable interest in the problem of fair allocation of resources for multirate multicast sessions. Most of the work in this area is concerned only with the notion of max-min fairness (see [13], [14], [15], [16], [5]). The utility maximization based congestion control problem is addressed in [6], [7], [4]. Whereas [7], [4] take a primal approach, the algorithms in [6] is based on a dual approach. Like [6], the approach that we adopt in this paper is based on the lagrangian dual. However, the algorithm proposed in this paper has several very important advantages compared to the algorithms proposed in [6], as well as the those in [7], [4], as outlined below.

An important aspect in which our approach differs from previously proposed approaches is that it takes into consideration the discreteness of the layer bandwidths. In layered multicast, the receiver rates are constrained to take only a set of discrete values, which are determined by the layer bandwidths. The approaches in [7], [6], [4] approximate the discrete set of rates by a continuous set, and then apply convex programming techniques to develop an iterative rate update procedure. The convergence results obtained in these cases are under the assumption of continuous rates. However, note that in practice, the rate that is computed by the rate update algorithm at each step must be "rounded" to a discrete rate value that corresponds to some layer bandwidth. Such rounding introduces errors at every step of the algorithm, and it is not clear if the rates can be shown to converge to optimality (in an exact or approximate sense) when rounding at every iteration is taken into account. In fact, it is easy to show that if the step-sizes are small enough, then the receiver rates achieved by the algorithms in [6], [4] could be very different from the optimal rates. Note that approximating the discrete rate set by a continuous set of rates may not be a bad approximation if there are many closely spaced discrete rates. However, as mentioned, typically the number of layers is small, and the discrete rate values are widely separated. Therefore, the continuous bandwidth approximation may not be a good approximation in the case of layered multicasting. In our approach, however, the rates are always assumed to be discrete, and so there is no question of rounding

K. Kar is with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA (email: koushik@ecse.rpi.edu).

L. Tassiulas is with the Computer Engineering and Telecommunications Department, University of Thessaly, Greece. (email: leandros@inf.uth.gr).

of the rates. The convergence results that we provide, therefore, directly applies to the algorithm that is implementable in practice.

From a practical perspective too, our algorithm outperforms existing approaches on several aspects. Unlike the algorithms in [6], our algorithm does not require per-group information to be maintained at the network links, and therefore scales well as the number of multicast groups sharing a link increases. Moreover, our algorithm does not suffer from some other drawbacks of the algorithms in [6], like rapid rate fluctuations, two-level convergence etc. The algorithms in [7], [4] can result in constant bandwidth fluctuations, which can lead to rapid adding and dropping of layers. Our algorithm, on the other hand, achieves much smoother convergence. Lastly, it can be intuitively argued that the rate of convergence for our algorithm would typically be much faster than those of the previously proposed algorithms [6], [7], [4], a fact that we have observed in our simulation experiments as well.

In this paper, we take into account the fact that the receivers rates are constrained to take discrete values, and pose the optimal rate control problem as a discrete/integer program. (It is worth noting here that even very simple special cases of the integer program can be shown to be NP-hard.) Dealing with this integer programming directly, and using a combination of *lagrangian relaxation* [18] and *dynamic programming* [1], [3], we show that it is possible to achieve rates that are provably very close to the optimal, without making the approximation that receiver rates take a continuous set of values. Our approach is completely decentralized, and scales well with the size of each multicast group, as well as the number of multicast groups sharing the network. Note that the lagrangian relaxation technique may not yield close-to-optimal solutions for general integer programs. However, we exploit certain special properties of our problem to derive the approximation result in our case. We also identify a nice underlying structure of our problem, which allows us to solve the problem distributedly, using dynamic programming. As we observe later, the efficient use of dynamic programming in this case would not have been possible if the rates were not constrained to take a few discrete values. Thus, even though dealing with the discrete program directly (as opposed to dealing with the convexified version of the program) might seem counterintuitive, we actually exploit the discreteness of the problem to our advantage. For the case of unicast sessions, rate control based on the lagrangian dual was proposed and thoroughly investigated by Low *et al*. [10]. In the unicast case, however, the rates can assumed to be continuous, and convex programming techniques can be directly applied. Our algorithm nicely generalizes the dual-based rate control approach proposed in [10] to the case of layered multicasting. It is also worth noting here that although dual methods have been previously used for addressing the rate control problem, the idea of using dynamic programming for rate control is novel.

II. Problem Statement

First we describe the network model, and formulate the rate control problem as an optimization problem with discreteness constraints on the rates. In the subsequent sections, we will show how we can achieve close-to-optimal rates for this problem.

A. Network Model and Terminology

Consider a network consisting of a set L of unidirectional links, where a link $l \in L$ has capacity c_l. The network is shared by a set of G multicast groups (sessions). Each multicast group is associated with a unique source, a set of receivers, and a set of links that the multicast group uses (the set of links forms a tree)[1]. Thus any multicast group $g \in G$ is specified by $\{s^g, R^g, L^g\}$ where s^g is the source, L^g is the set of links in the multicast tree, and R^g is the set of receivers in group g.

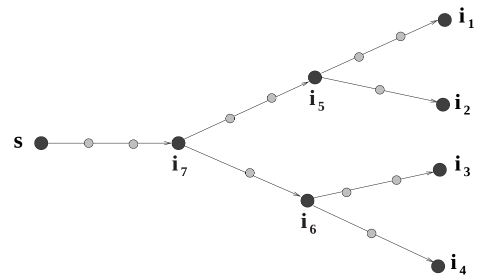

Fig. 1. A multirate multicast tree

Next we introduce some additional terminology that will help us in formulating the rate control problem and describing the algorithms. Consider Figure 1, which shows an example of a multicast tree where s is the *source node* and $\{i_1, i_2, i_3, i_4\}$ is the set of *receiver nodes*. Other than the source and receiver nodes, the nodes that are of particular interest in our case are the forking nodes of the tree, i.e., nodes where the multicast tree "branches off". We refer to these nodes as the *junction nodes*. Thus, in Figure 1, $\{i_5, i_6, i_7\}$ is the set of junction nodes. Source/receiver/junction nodes of different multicast groups are considered to be logically different, even if they are physically located at the same node. In the rest of the paper, we assume that the receivers are only at the leaf nodes of the multicast tree. There is no loss of generality in assuming this, since a receiver at a non-leaf node can be replaced by creating a new leaf node and placing the receiver in it, and connecting the new leaf node to the non-leaf node (where the receiver is actually located) by a link with infinite capacity. Moreover, note that any leaf node must be a receiver node. The *parent* of a receiver/junction node i refers to the closest junction/source node in the upstream path from i towards the source. The *children* of a junction/source node are also defined accordingly. The *branch* of a receiver/junction node i refers to the set of links in the tree between the parent of node i and node i (i.e., the path over which node i receives data from its parent). Note that each junction node receives layered data from its parent node, and forwards them to its children nodes, after possibly dropping some layers. Therefore, the rate at which a junction/receiver node receives data can be no greater than the rate at which its parent receives data. Note that layers

[1] We assume fixed path routing. So the tree associated with each multicast group is fixed.

must be sent *cumulatively*, i.e., no layer between the base layer and the uppermost transmitted layer can be dropped. Thus if a source/junction node intends to send k layers to its child, it must send only the layers $1, 2, ..., k$. The number of layers sent determines the "level" of data transmission, and in the case of audio/video, the perceived transmission quality depends on it. Note that if a node is receiving data at level k from its parent, then it must be receiving data at a rate equal to the sum of the bandwidths of layers $1, 2, ..., k$.

B. Problem Formulation

The utility maximization based rate control problem for multirate multicast traffic can be formulated in two different ways. In the first approach, we associate a rate variable with each receiver, and formulate the optimization problem in terms of the these receiver rate variables. The second approach is to associate a rate variable with each receiver as well as each junction node, and define the problem in terms of all these rate variables. These two formulations are equivalent in the sense that the optimal objective function values of both are the same. Moreover, the optimal receiver rates are also the same for both these formulations. The two "equivalent" representations of the problem, and the relationship between the two representations, are discussed in great detail in [6].

In this paper, we use the second formulation, i.e., the case in which rate variables are associated with both the receiver and the junction nodes. The approach presented in this paper can also be applied to the first formulation. However, the analysis in that case is more complex, and the algorithm derived is very similar to the one derived on the basis of the second formulation.

Let $R = \cup_{g \in G} R^g$ denote the set of all receiver nodes (over all groups). Let J^g denote the junction nodes of any group $g \in G$, and $J = \cup_{g \in G} J^g$ denote the set of all junction nodes (over all groups). Let $I^g = R^g \cup J^g$, and let $I = \cup_{g \in G} I^g$. Therefore, $I = R \cup J$, and denotes the set of all receiver and junction nodes (over all groups). Also, let $S = \{s^g, g \in G\}$ denote the set of all source nodes (over all groups).

Let $I_l \subseteq I$ be the set of receiver/junction nodes whose branches include link $l \in L$. Now associate a rate variable x_i with each receiver/junction node $i \in I$, denoting the rate at which node i receives data from its parent. For the sake of simplicity of exposition, we also introduce a rate variable associated with each source node. Let x_i be the rate variable for any node $i \in S$. Let $\underline{x} = (x_i, i \in I \cup S)$ denote the vector of all rates. Also, for each group $g \in G$, let $\underline{x}^g = (x_i, i \in I^g \cup \{s^g\})$ denote the vector of rates associated with group g. For each node $i \in J \cup S$, let C_i denote the set of children of node i. For any group $g \in G$, let K^g be the number of layers, and let $b_1^g < b_2^g < ... < b_{K^g}^g$ represent the cumulative layer bandwidths (thus the rates of the receivers belonging to group g are constrained to take only these discrete values). Note that for any $k \in \{1, 2, ..., K^g\}$, b_k^g is the sum of the bandwidths of the layers $1, 2, ..., k$. For each receiver $i \in R$, let $U_i : \Re_+ \rightarrow \Re$ denote the utility function (assumed increasing and concave) associated with i. Then the utility maximization based rate control problem is formulated as

$$\mathbf{P}: \quad \text{maximize} \quad \sum_{i \in R} U_i(x_i), \tag{1}$$

$$\text{subject to:} \quad \sum_{i \in I_l} x_i \leq c_l \quad \forall\, l \in L, \tag{2}$$

$$\underline{x}^g \in X^g \quad \forall\, g \in G, \tag{3}$$

where $X^g = Y^g \cap Z^g$, and Y^g and Z^g are defined as

$$Y^g = \{\underline{x} : x_i \geq x_{i'} \ \forall\, i' \in C_i \ \ \forall\, i \in J^g \cup \{s^g\}\}, \tag{4}$$

$$Z^g = \{\underline{x} : x_i \in \{b_1^g, b_2^g, ..., b_{K^g}^g\} \ \forall i \in I^g \cup \{s^g\}\}. \tag{5}$$

Relations (2) represent the link capacity constraints. Relations (4) represent the fact that the rate at which a junction/receiver node receives data can be no greater than the rate at which its parent node receives data. Relations (5) represent the discreteness constraints on the rates. Note that the constraints involving the source rates are redundant. These constraints are introduced here because they yield more concise expressions in the analysis outlined later in this paper.

Our rate control algorithm should therefore achieve the optimal rates for $\mathbf{P}$. In order to be practically viable, the rate control algorithm must be *decentralized*. Moreover, the algorithm should be *scalable*, both in terms of the size of the multicast groups (i.e., number of receivers in a multicast group), and the number of multicast groups sharing the network. We would also prefer to have a solution which has low computational complexity, and converges fast, even in an asynchronous network scenario. The rate control algorithm that we propose in this paper satisfies all of the above criteria.

III. Basic Solution Approach and Convergence Results

In this section, we outline our solution approach. Our approach is based on *lagrangian relaxation*, which is a well-known technique for solving integer programs [18]. Our contribution is that we show how in our case, this technique can help us develop an iterative algorithm that achieves rates that are provably close-to-optimal. Moreover, along with dynamic programming, it leads to an algorithm that is completely distributed in nature. The fact that we can develop a distributed solution that achieves approximately optimal rates, relies heavily on some underlying nice properties of the structure of the problem $\mathbf{P}$. Note that for general integer programs, lagrangian relaxation may not lead to close-to-optimal solutions, and the algorithm may not be distributed. The approach and results presented in this section generalizes those proposed in [10] for the unicast version of our problem.

A. Lagrangian Relaxation

Now let us take a look at the lagrangian dual of the problem $\mathbf{P}$. Let λ_l be the dual variable associated with the link capacity constraint (2) for link $l \in L$. Let $\underline{\lambda} = (\lambda_l, l \in L)$ denote the vector of the dual variables. For any $i \in I$, let

$L_i \subseteq L$ denote the set of links in the branch of node i. Then the lagrangian dual $D(\underline{\lambda})$ can be written as follows [2]:

$$D(\underline{\lambda}) = \sum_{g\in G} \max_{\underline{x}^g \in X^g} \{ \sum_{i\in R^g} U_i(x_i) - \sum_{i\in I^g} (\sum_{l\in L_i} \lambda_l)\, x_i \} + \sum_{l\in L} \lambda_l c_l. \quad (6)$$

The dual minimization problem is $\min_{\underline{\lambda}\geq \underline{0}} D(\underline{\lambda})$, where $D(\underline{\lambda})$ is defined as in (6). Since we are dealing with a discrete program, a duality gap exists, and dualization implicitly involves relaxation of the problem. Note that the dual is convex but non-differentiable (the non-differentiability is due to the presence of the discreteness (integrality) constraints on the rates). We apply a subgradient method [17] (with a constant step-size α) to solve this problem. In this case, each iteration of the subgradient method reduces to two sets of updates: (1) dual variable updates, and (2) rate updates. The dual variable update procedure for any link $l \in L$ at step n is

$$\lambda_l(n+1) = \lambda_l(n) + \alpha(\sum_{i\in I_l} x_i(n) - c_l), \quad (7)$$

where $x_i(n)$ is the value of x_i at the nth iterative step. At the nth step, for any group $g \in G$, the rates of the receiver/junction nodes are updated as follows:

$$\underline{x}^g(n+1) = \arg\max_{\underline{x}^g\in X^g} \{ \sum_{i\in R^g} U_i(x_i) - \sum_{i\in I^g} (\sum_{l\in L_i} \lambda_l(n))\, x_i \}. \quad (8)$$

The update procedures in (7) and (8) have simple intuitive interpretations. Let us interpret the dual variable λ_l as the *price per unit bandwidth* associated with link l. Note that the quantity $(\sum_{i\in I_l} x_i(n) - c_l)$ represents the excess load of the link. Therefore, (7) has a simple economic interpretation: price per unit bandwidth increases if the load (interpreted as demand) is in excess of the available capacity, and decreases otherwise. Now, let us take a detailed look at the expression on the right hand side of (8). Note that the term $\sum_{i\in R^g} U_i(x_i)$ represents the overall utility of group g. Now, assume that each link l charges a price λ_l per unit bandwidth to every group that uses the link. With our interpretation, the term $(\sum_{l\in L_i} \lambda_l(n))$ represents the aggregate price per unit bandwidth charged to group g for using the links in the branch of node i. Therefore, the term $\sum_{i\in I^g}(\sum_{l\in L_i} \lambda_l(n))\, x_i$ can be interpreted as the total price charged by the network (to group g) for network bandwidth used by the group. Therefore, the right hand side of (8) can be interpreted as the *profit* (i.e., utility - price paid) derived by group g. Thus, (8) states that given the link prices, each group chooses the junction/receiver rates so as to maximize its profit function.

B. Convergence Results

In the convergence analysis, we make the following assumption on the utility functions.

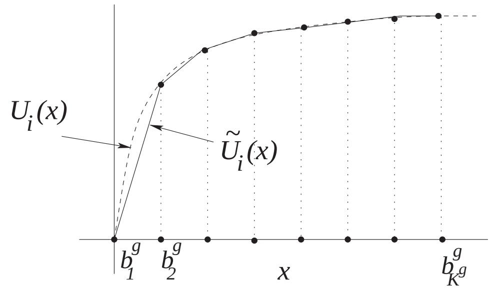

Fig. 2. Relationship between U_i and $\tilde{U}_i$

Assumption 1: (Strict Concavity) The utility functions U_i are increasing, twice continuously differentiable and strictly concave. Thus $-U_i''(x_i) \geq \gamma_i > 0$ for all $x_i \geq 0$, for all $i \in R$.

Now we show that if the rates and the prices are updated iteratively according to (7) and (8), the rate vector "converges" to values that are close-to-optimal. The optimality in this case is not with respect to the optimal rates of the original integer programming problem $\mathbf{P}$, but with respect to a *linearly relaxed* version of it (denoted by $\tilde{\mathbf{P}}$), as we explain below. The problem $\tilde{\mathbf{P}}$ is defined from the problem $\mathbf{P}$ in the following way. Replace the discreteness constraint $x_i \in \{b_1^g, ...b_{K^g}^g\}$ in (5) by the continuous constraint $b_1^g \leq x_i \leq b_K^g$. Thus we "relax" the integrality constraints and assume that x_i can take any value in the continuous set $[b_1^g, b_{K^g}^g]$.

Also, in $\tilde{\mathbf{P}}$, the utility functions are re-defined in the following way. Consider any $x_i \in [b_1^g, b_K^g]$. There are two possible cases:

Case 1: $x_i = b_k^g$ for some $k \in \{1, 2, ..., K^g\}$: In this case, define $\tilde{U}_i(x_i) = U_i(b_k^g)$.

Case 2: $b_k^g < x_i < b_{k+1}^g$ for some $k \in \{1, 2, ..., K^g - 1\}$: In this case, define $\tilde{U}_i(x_i)$ as

$$\tilde{U}_i(x_i) = \frac{(b_{k+1}^g - x_i)U_i(b_k^g) + (x_i - b_k^g)U_i(b_{k+1}^g)}{b_{k+1}^g - b_k^g}. \quad (9)$$

Note that the function $\tilde{U}$ is formed by linearly interpolating the function U between the feasible discrete bandwidth values. The relationship between the functions are shown in Figure 2. The problem $\mathbf{P}$, with these modifications, is a linearly relaxed version of the original discrete programming problem. This problem is denoted by $\tilde{\mathbf{P}}$.

Note that $\tilde{\mathbf{P}}$ is a convex programming problem. Also note that if U^* and $\tilde{U}^*$ be the optimal objective function values of $\mathbf{P}$ and $\tilde{\mathbf{P}}$ respectively, $\tilde{U}^* \geq U^*$. Any solution that is feasible to $\mathbf{P}$ and close-to-optimal to $\tilde{\mathbf{P}}$ must also be a close-to-optimal solution of $\mathbf{P}$. We use this fact to show that our algorithm solves $\mathbf{P}$ approximately, and derive the approximation ratio.

Let $\bar{b}^g = \max_{k\in\{1,...,K^g-1\}}(b_{k+1}^g - b_k^g)$. Also Let $\underline{x}^*$ be *any* optimal solution of $\tilde{\mathbf{P}}$, and x_i^* be the ith component of the vector $\underline{x}^*$. Note that $\tilde{\mathbf{P}}$ can have multiple optimal solutions. However, due to the strict concavity of the utility functions U_i, it can be shown that if $x_{1,i}^*$ and $x_{2,i}^*$ be the ith component ($i \in R^g$) of two optimal solutions $\underline{x}_1^*$ and $\underline{x}_2^*$ of $\tilde{\mathbf{P}}$, then $|x_{1,i}^* - x_{2,i}^*| \leq \bar{b}^g$.

We obtain the following result on the convergence of the receiver rates:

Theorem 1: Assume that the rate and the prices are updated according to (7) and (8). Then there exists an $\bar{\alpha} > 0$ and an integer $\bar{N} > 0$, such that any α satisfying $0 < \alpha < \bar{\alpha}$, the following result holds for all $n > \bar{N}$:

$$|x_i(n) - x_i^*| \leq \bar{b}^g \quad \forall i \in R^g \;\; \forall g \in G.$$

The proof of the above result is provided in the Appendix. To re-emphasize, the above result holds for any optimal solution $\underline{x}^*$ of $\tilde{\mathbf{P}}$. Roughly speaking, the result states that if the step-size is "sufficiently small", then receiver rates "converge" to a neighborhood around $\underline{x}^*$. The notion of "convergence" in this case is approximate: the above result implies that the receiver rate vector is guaranteed to be in a neighborhood around the optimal solution of $\tilde{\mathbf{P}}$. It does not ensure, however, that the rate vector will converge to $\underline{x}^*$. Note that in the relaxed problem, the rates need not correspond to the actual discrete bandwidth values. Thus $\underline{x}^*$ can even be an infeasible to $\mathbf{P}$. However, note that the rates $x_i(n)$ always take values in the discrete bandwidth set (see (8)).

The above result can be strengthened to show that the rate of a receiver r can only achieve values that correspond to the bandwidth levels immediately below or immediately above x_i^*. Thus the achieved rates can be at most "one-layer off" from optimality. In general, finding a closed-form expression of $\bar{\alpha}$ seems difficult.

Let $|U_i(b_{k+1}^g) - U_i(b_k^g)| \leq \bar{u} \; \forall k \in \{1, ..., K^g - 1\} \; \forall i \in R^g \; \forall g \in G$. Thus $\bar{u}$ represents the maximum difference between the receiver utilities at two adjacent discrete bandwidth levels. Let U^* be the optimal value of the objective function of the original problem $\mathbf{P}$. Also, let $U(x(n))$ be the aggregate receiver utility when the rate vector is $\underline{x}(n)$. Thus $U(\underline{x}(n)) = \sum_{i \in R} U_i(x_i(n))$. Then from Theorem 1, and from the continuity of the functions U_i, we obtain the following result.

Corollary 1: Assume that the assumptions of Theorem 1 hold. Then for all $n > \bar{N}$, the following holds:

$$U^* - U(x_i(n)) \leq \bar{u}|R|.$$

The above corollary states that the rates that we achieve are approximately optimal with respect to the original problem $\mathbf{P}$. Note that the error in the achieved utility (with respect to the optimal utility) calculated on a per-receiver basis is at most $\bar{u}$. Therefore, the error would be smaller if the discrete bandwidth levels are closely spaced, as we would intuitively expect.

IV. Group Profit Maximization and Distributed Implementation

Now we show how the group profit maximization problem (as stated in (8)) can be solved in an efficient manner. Note that the group profit maximization problem needs to be solved for each group in every iteration of the rate control algorithm. Therefore, to achieve good convergence speed, the group profit maximization problem must be solved quickly. Moreover, practical considerations dictate that the problem must be solved in a distributed manner. In the following, we show that the group profit maximization problem can be solved in a decentralized manner, within roughly a single round-trip time.

A. Group Profit Maximization

Consider any particular multicast group $g \in G$. Now consider any junction/receiver/source node $i \in I^g \cup \{s^g\}$. Let T_i denote the set of source/junction/receiver nodes that fall within the tree rooted at i (including i). Let $\underline{x}_i = (x_{i'}, i' \in T_i)$ denote the vector of the rate variables associated with the source/junction/receiver nodes in T_i. Let $P_i(\underline{x}_i)$, the tree profit function associated with node i, be defined as follows:

$$P_i(\underline{x}_i) = \sum_{i' \in T_i \cap R^g} U_{i'}(x_{i'}) - \sum_{i' \in T_i \setminus \{i\}} (\sum_{l \in L_{i'}} \lambda_l) x_{i'}. \quad (10)$$

Clearly, P_i denotes the aggregate utility of all receivers in T_i minus the price charged to group g for using the links in tree T_i. Note that for any receiver i, $P_i(\underline{x}_i) = U_i(x_i)$. Also note that the group profit function that we are stated in (8) is P_{s^g}. Next we show that the problem of maximizing the tree profit function associated with any node i can be written in terms of the corresponding problems for its children nodes.

For any node $i \in I^g \cup \{s^g\}$, define $X_i = Y_i \cap Z_i$, where Y_i and Z_i are defined as follows:

$$\begin{aligned} Y_i &= \{\underline{x}_i : x_{i'} \geq x_{i''} \; \forall\, i'' \in C_{i'} \;\; \forall\, i' \in T_i \cap (J^g \cup \{s^g\})\}, \\ Z_i &= \{\underline{x}_i : x_{i'} \in \{b_1^g, b_2^g, ..., b_{K^g}^g\} \; \forall i' \in T_i\}. \end{aligned}$$

Now, for any $i \in I^g \cup \{s^g\}$ and $k \in \{1, 2, ..., K^g\}$, define $X_i(k)$ as follows:

$$X_i(k) = \{\underline{x}_i : \underline{x}_i \in X_i, \; x_i = b_k^g\}. \quad (11)$$

Thus $X_i(k)$ denotes the set of values in which the rates of source/junction/receiver nodes in tree T_i are constrained to lie if node i receives traffic upto layer k from its parent. Then, for any $i \in I^g \cup \{s^g\}$ and $k \in \{1, 2, ..., K^g\}$, define $p_i(k)$, the *conditional maximum profit (CMP)* of node i at level k, as follows:

$$p_i(k) = \max_{\underline{x}_i \in X_i(k)} P_i(\underline{x}_i). \quad (12)$$

Note that $p_i(k)$ denotes the maximum profit derived from the tree T_i if the node i receives traffic upto layer k from its parent. Note that for any receiver node i, $p_i(k) = U_i(b_k^g)$. Also note that maximum group profit that we are interested in obtaining (see (8)) is equal to $p_{s^g}(K^g)$. Next we show how the CMPs of node i can be derived from the CMPs of its children nodes. Thus result will help us in computing the maximum group profit recursively by breaking it up into smaller subproblems.

Consider the constraint $\underline{x}_i \in X_i(k)$, for any $i \in J^g \cup \{s^g\}$ and $k \in \{1, 2, ..., K^g\}$. We first show that this constraint is equivalent to a set of similar constraints involving the subtrees of T_i.

$$\begin{aligned} &\{\underline{x}_i : \underline{x}_i \in X_i(k)\} \\ &= \{\underline{x}_i : \underline{x}_i \in X_i, x_i = b_k^g\} \quad (13) \\ &= \{\underline{x}_i : \underline{x}_{i'} \in X_{i'}, x_{i'} \leq b_k^g \; \forall i' \in C_i, \; x_i = b_k^g\} \quad (14) \\ &= \{\underline{x}_i : \underline{x}_{i'} \in X_{i'}, x_{i'} \in \{b_1, ..., b_k^g\} \; \forall i' \in C_i, \\ &\quad x_i = b_k^g\} \quad (15) \\ &= \{\underline{x}_i : \underline{x}_{i'} \in \cup_{k'=1}^{k} X_{i'}(k') \; \forall i' \in C_i, \; x_i = b_k^g\}. \quad (16) \end{aligned}$$

Relation (13) follows from the definition of $X_i(k)$ (see (11)), while (14) follows easily by expanding the constraints in the set $\underline{x}_i \in X_i(k)$ and using the fact $x_i = b^g_k$.

From (10), it is easy to observe that for any $i \in J^g \cup \{s^g\}$, the following equality holds

$$P_i(\underline{x}_i) = \sum_{i' \in C_i} \{P_{i'}(\underline{x}_{i'}) - (\sum_{l \in L_{i'}} \lambda_l) x_{i'}\}. \quad (17)$$

For any $i \in J^g \cup \{s^g\}$ and $k \in \{1, 2, ..., K^g\}$, we obtain

$$\begin{aligned}
p_i(k) &= \max_{\underline{x}_i \in X_i(k)} P_i(\underline{x}_i) && (18)\\
&= \max_{\underline{x}_i \in X_i(k)} \sum_{i' \in C_i} \{P_{i'}(\underline{x}_{i'}) - (\sum_{l \in L_{i'}} \lambda_l) x_{i'}\} && (19)\\
&= \max_{\substack{\underline{x}_{i'} \in \cup_{k'=1}^{k} X_{i'}(k') \ \forall i' \in C_i, \\ x_i = b^g_k}} \sum_{i' \in C_i} \{P_{i'}(\underline{x}_{i'}) \\
&\quad - (\sum_{l \in L_{i'}} \lambda_l) x_{i'}\} && (20)\\
&= \max_{\underline{x}_{i'} \in \cup_{k'=1}^{k} X_{i'}(k') \ \forall i' \in C_i} \sum_{i' \in C_i} \{P_{i'}(\underline{x}_{i'}) \\
&\quad - (\sum_{l \in L_{i'}} \lambda_l) x_{i'}\} && (21)\\
&= \sum_{i' \in C_i} \max_{\underline{x}_{i'} \in \cup_{k'=1}^{k} X_{i'}(k')} \{P_{i'}(\underline{x}_{i'}) \\
&\quad - (\sum_{l \in L_{i'}} \lambda_l) x_{i'}\} && (22)\\
&= \sum_{i' \in C_i} \max_{k' \leq k} \max_{\underline{x}_{i'} \in X_{i'}(k')} \{P_{i'}(\underline{x}_{i'}) \\
&\quad - (\sum_{l \in L_{i'}} \lambda_l) x_{i'}\} && (23)\\
&= \sum_{i' \in C_i} \max_{k' \leq k} \{ \max_{\underline{x}_{i'} \in X_{i'}(k')} P_{i'}(\underline{x}_{i'}) \\
&\quad - (\sum_{l \in L_{i'}} \lambda_l) \max_{\underline{x}_{i'} \in X_{i'}(k')} x_{i'}\} && (24)\\
&= \sum_{i' \in C_i} \max_{k' \leq k} \{p_{i'}(k') - (\sum_{l \in L_{i'}} \lambda_l) b^g_{k'}\}. && (25)
\end{aligned}$$

Relation (18) follows from (12), and (19) follows from (17). Relation (20) follows from (16). Relation (21) follows from the fact that neither the term $\sum_{i' \in C_i} \{P_{i'}(\underline{x}_{i'}) - (\sum_{l \in L_{i'}} \lambda_l) x_{i'}\}$, nor the other constraints in (20), depends on the variable x_i. Relation (22) follows from the fact that the constraint set $\{\underline{x}_{i'} \in \cup_{k'=1}^{k} X_{i'}(k') \ \forall i' \in C_i\}$ and the objective function $\sum_{i' \in C_i} \{P_{i'}(\underline{x}_{i'}) - (\sum_{l \in L_{i'}} \lambda_l) x_{i'}\}$ are both separable with respect to the variable vectors $\underline{x}_{i'}, i' \in C_i$. Note that in (23) (and subsequently), we represent the constraint $k' \in \{1, ..., k\}$ simply as $k \leq k'$, for the sake of conciseness. Relation (25) follows from the definitions of $p_i(k')$ and $X_i(k')$ (see (12) and (11)).

Relation (25) shows that the CMPs for a source/junction node i can be expressed in terms of the CMPs of its children nodes. This fact allows us to find $p_{s^g}(K^g)$ using a dynamic programming approach. Thus, we can calculate the CMPs with a bottom-up approach: first we compute the CMPs at the receiver nodes, then at junction nodes that are one level above that, and so on, until we reach the source node. Note that the dynamic programming computations at the nodes in any particular level of the tree can be executed simultaneously. This parallelism inherent in the structure of the dynamic program can be utilized to solve the group profit maximization problem in a single round trip time. The practical implementation of this dynamic program is discussed in more detail in the next subsection.

B. Implementation

Firstly, note that the dual variable (price) update procedure (7) can be implemented in a very simple way. To achieve this, each link l can keep track of the price λ_l and periodically update it according to (7).

Now we describe how the rates are updated so that they satisfy (8). Assume that each source/junction/receiver node maintains a CMP table of its own that contains the CMP values for all levels k for that node. Thus the CMP table of node i in group g contains K^g entries, where the kth entry is p^k_i. In order to update the CMP table entries according to (25), a node junction/source i needs to know the following: i) the aggregate price of all links in the branch associated with each child node, and ii) the CMP table of each child node.

Consider a source/junction node i, and a junction/receiver node $i' \in C_i$. Then node i can find the "branch price" associated with i' in the following way. Node i can send a "price packet" downstream to i' while setting the value of a "price field" (included in the price packet) to zero. The subsequent links on the path of the packet add their prices to the price field of the packet. Therefore, when the price packet reaches i', the price field contains the aggregate price of all links on the branch associated with i'. Node i' can then just send the price packet upstream to its parent i.

For any receiver node i, the CMPs are easily calculated as $p_i(k) = U_i(b^g_k)$. Once a receiver has computed its CMP table, it sends the CMP table upstream to its parent. Once a junction node has received the CMP tables from all of its children nodes, it updates its CMP table entries according to (25), and sends its CMP table upstream to its parent node. In this manner, the CMP tables are updated an propagated upstream by each receiver/junction node, till the source node is reached. Once the source node s^g has updated its CMP table, it determines the downstream traffic rates in the following way. For each node $i' \in C_{s^g}$, the source node sends $k_{i'}$ layers to i', where $k_{i'}$ is obtained as

$$k_{i'} = \arg\max_{k' \leq K^g} \{p_{i'}(k') - (\sum_{l \in L_{i'}} \lambda_l) b^g_{k'}\}. \quad (26)$$

A junction node i determines the downstream rates in a similar manner. Let k_i be the number of layers that node i receives from its parent. Then, each node $i' \in C_i$ receives $k_{i'}$ layers from i, where $k_{i'}$ is determined as

$$k_{i'} = \arg\max_{k' \leq k_i} \{p_{i'}(k') - (\sum_{l \in L_{i'}} \lambda_l) b^g_{k'}\}. \quad (27)$$

Thus node i' receives data at a rate $b^g_{k_{i'}}$ from its parent i. From the discussion in Section IV-A, it follows that when the rates chosen as described above, they satisfy (8), i.e., maximize the group profit.

The rate update procedure works in a manner opposite to that of the CMP table update procedure. Thus, the source first determines the rates of its children, and informs each child (by sending a “rate packet” to the child) about the corresponding rate. Each child node then determines the rates of their children nodes, and and this goes on, until the rates of the receivers have been determined.

Therefore, the overall procedure of solving the group profit maximization problem and determining the rates consists of two phases: i) a bottom-up phase to determine the CMP tables, followed by ii) a top-down phase to determine the rates. Note that in each of these phases, the computations at the set of nodes at any particular level of the tree (in this case, a ‘level’ refers to a set of nodes that are at the same hop-distance from the root) can occur in parallel. Thus, if the processing delays are neglected, the total time required to execute the entire procedure is upper bounded by the maximum round-trip delay.

C. Complexity Reduction

Consider a junction/source node i. From (25), we note that the time for computing the CMP for level k at node i is $O(\bar{C}\bar{K})$, where $\bar{C}$ is the maximum number of children of any node, and $\bar{K}$ is the maximum number of layers in any multicast group. Therefore, the time for computing the CMPs for all levels at any node i is $O(\bar{C}\bar{K}^2)$. From (26) and (27), it is easy to observe that the computation time for the rate update procedure at each source/junction node is $O(\bar{C}\bar{K})$. Note that since each receiver/junction node needs to send a CMP table to its parent, the communication complexity is $O(\bar{K})$.

In the following, we show how the worst-case computational complexity for the CMP table update can be improved so that it is only a linear function of $\bar{K}$. Assume that in addition to its CMP table, a source/junction node maintains a *discounted CMP (d-CMP)* table for each of its children. Let i' be a child node of a source/junction node i. Then the d-CMP table for i' maintained at i contains K^g entries, and $\tilde{p}_i(k)$, the kth entry in that table, is defined as

$$\tilde{p}_{i'}(k) \quad = \quad \max_{k' \leq k} \{p_{i'}(k') - (\sum_{l \in L_{i'}} \lambda_l) b^g_{k'}\}. \qquad (28)$$

Let $k \geq 2$. Then, from the definition of $\tilde{p}_{i'}(k)$, we obtain

$$\begin{aligned}\tilde{p}_{i'}(k) &= \max_{k' \leq k} \{p_{i'}(k') - (\sum_{l \in L_{i'}} \lambda_l) b^g_{k'}\} \\ &= \max \{p_{i'}(k) - (\sum_{l \in L_{i'}} \lambda_l) b^g_k, \\ &\qquad \max_{k' \leq k-1} \{p_{i'}(k') - (\sum_{l \in L_{i'}} \lambda_l) b^g_{k'}\}\} \\ &= \max \{p_{i'}(k) - (\sum_{l \in L_{i'}} \lambda_l) b^g_k,\ \tilde{p}_{i'}(k-1)\}. \qquad (29)\end{aligned}$$

Note that $\tilde{p}_{i'}(1) = p_{i'}(1) - (\sum_{l \in L_{i'}} \lambda_l) b^g_1$. Using this fact and (29), we see that if we compute the d-CMP table entries in the order $\tilde{p}_{i'}(1)$, $\tilde{p}_{i'}(2)$, ..., $\tilde{p}_{i'}(K^g)$, we can compute the entire d-CMP table in $O(\bar{K})$ time. Therefore, computing the d-CMP tables for all children of node i takes $O(\bar{C}\bar{K})$ time. Once these table have been computed (which is done once node i has received the CMP tables of its children), the CMP table for node i can be computed in additional $O(\bar{K})$ time, according to the following relation (obtained from (25) and (28)) :

$$p_i(k) \quad = \quad \sum_{i' \in C_i} \tilde{p}_{i'}(k) \ .$$

Therefore, the CMP tables at each node can be computed in $O(\bar{C}\bar{K})$ time.

V. Simulation Results

Simulation experiments carried out on various network topologies/scenarios confirm that our algorithm achieves the optimal rates in an asynchronous slowly time-varying network environment. Next we present a few representative examples to demonstrate this fact.

Figure 3 shows the example network that we consider, which consists of two multicast groups sharing a 11-node 10-link network. We have taken the network topology to be the same as that in [6], [7], so that our simulation results can be easily compared with those of the existing approaches. We assume layered multicasting, and each multicast group can send traffic in 20 layers, each of the layers having a bandwidth of 0.25 MBps. Therefore, the maximum allowed bandwidth is 5 MBps, and bandwidth can be allocated in units of 0.25 MBps. Therefore, to achieve a rate of k*0.25 MBps, the lowest k layers need to be sent. Note that in layered multicasting, each data packet belongs to one particular layer. Therefore, a source/junction node can send traffic to its child at a particular discrete bandwidth level simply by sending/forwarding only those data packets which belong to a corresponding set of cumulative layers.

In our experiments, the link prices (dual variables) are updated at regular intervals of 20 msec, and the receivers send their CMP tables upstream at regular intervals of 50 msec. Thus, the rates are updated (i.e., the group profit maximization problem is solved) once every 50 msec. The step-size of link price updates, α, is kept fixed at 0.005, and the links update the prices based on the estimated (measured) aggregate traffic rate on the link (the estimation time window is 20 msec). All data packets (sent downstream) are 400 bytes long. All control packets (the price packets sent upstream, and the rate packets sent downstream) are 200 bytes long. In all of the simulations described in this paper, maximum utilization of a link is set to 95%. Therefore, a link increases or decreases its price depending on whether the overall estimated traffic on the link exceeds 95% of its capacity or not.

In the network shown, the utility functions of receivers i_4 and i_6 are $0.5\ln(1+x)$, while those of the rest are $\ln(1+x)$ (where x is expressed in MBps). The minimum rate for each receiver is zero, and the maximum rate is the capacity of the link leading to the receiver. Note that since i_5 is connected directly to the source, it behaves essentially like an unicast session. In our simulation scenario, the sequence of arrivals/departures of receivers are as follows. The receivers i_1, i_2, i_3, i_6 and i_7 arrive at time $t = 0$. Receiver i_5 joins at

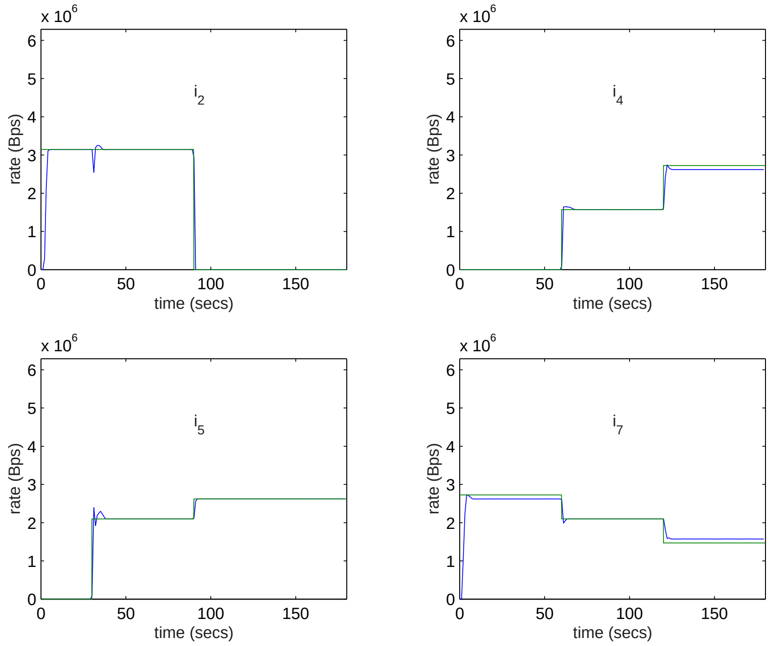

Fig. 4. Convergence of achieved rates. (The straight lines are the optimal (theoretical) rates.)

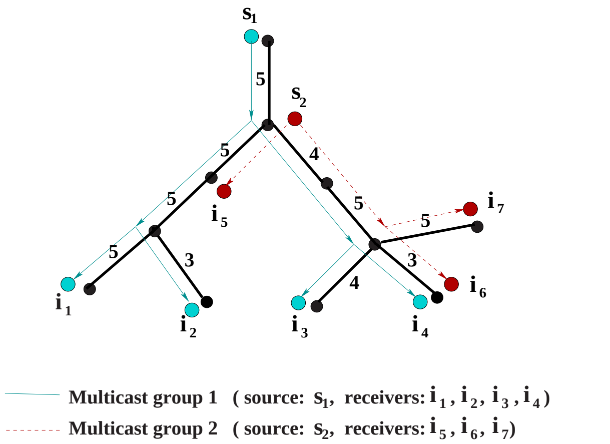

Fig. 3. An example network (The numbers associated with the links are the link capacities (in Mbps). The propagation delay for each link is 1 ms.)

$t = 30$ secs, receiver i_4 joins at $t = 60$ secs, i_2 leaves at $t = 90$ secs, and i_6 leaves at $t = 120$ secs.

Figure 4, which shows some rate plots in the time window 0-180 secs, demonstrates the performance of our algorithm in the particular example considered. Figure 4 shows the rates at which the receivers i_2, i_4, i_5 and i_7 receive data (obtained by measurement at the individual receivers), along with the optimal rates. (These 4 receivers were chosen arbitrarily, and rate plots of the other receivers also exhibit a similar trend.) The plotted rates are computed by averaging the measured rates, where the averaging is done every sec. Note that the sudden changes in the optimal rates at $t = 30, 60, 90, 120$ secs are due to the arrival/departure of receivers. The plots demonstrate that the achieved receiver rates track the optimal rates closely even as the optimal rates change. Note that the optimal rates plotted in the figure are computed based on the relaxed problem $\tilde{\mathbf{P}}$, and not the actual discretized problem $\mathbf{P}$ (which can only be computed by solving a very complex integer program). Therefore, the slight difference between the optimal and the achieved rates, as seen in the figure, is expected. A comparison of Figure 4 with the simulation results in [6], [7] shows that the convergence of our algorithm is much faster, and rate fluctuations significantly lower, as compared to existing approaches.

REFERENCES

[1] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, USA, 1995.

[2] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.

[3] T. H. Cormen, C. E. Leiserson, and R. L.Rivest, *Introduction to Algorithms*, MIT Press, 1990.

[4] S. Deb, R. Srikant,"Congestion Control for Fair Resource Allocation in Networks with Multicast Flows", *Proceedings of Conference on Decision and Control (CDC) 2001*, Orlando, USA, December 2001.

[5] E. Graves, R. Srikant, D. Towsley, "Decentralized Computation of Weighted Max-Min Fair Bandwidth Allocation in Networks with Multicast Flows", *Proceedings of Tyrrhenian International Workshop on Digital Communications (IWDC) 2001*, Taormina, Italy, September 2001.

[6] K. Kar, S. Sarkar, L. Tassiulas, "Optimization Based Rate Control for Multirate Multicast Sessions", *Proceedings of Infocom 2001*, Anchorage, USA, April 2001.

[7] K. Kar, S. Sarkar, L. Tassiulas, "A Scalable, Low-Overhead Rate Control Algorithm for Multirate Multicast Sessions", *IEEE Jounal on Selected Areas in Communications*, Vol. 20, No. 8, October 2002.

[8] F. P. Kelly,"Charging and Rate Control for Elastic Traffic", *European Transactions on Telecommunications*, vol. 8, no. 1, 1997, pp. 33-37.

[9] F. Kelly, A. Maulloo, D. Tan,"Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability ", *Journal of Operations Research Society*, vol. 49, no. 3, 1998, pp. 237-252.

[10] S. Low, D. E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence", *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, December 1999.

[11] X. Li, S. Paul, M. Ammar, "Layered Video Multicast with Retransmission (LVMR): Evaluation of Hierarchical rate control", *Proceedings of IEEE Infocom 1998*, March 1998.

[12] S. McCanne, V. Jacobson, M. Vetterli, "Receiver-Driven Layered Multicast", *Proceedings of ACM Sigcomm '96*, Stanford, CA, September 1996.

[13] D. Rubenstein, J. Kurose, D. Towsley. "The Impact of Multicast Layering on Network Fairness", *Proceedings of ACM Sigcomm '99, Cambridge, MA, September, 1999.*

[14] S. Sarkar, L. Tassiulas, "Fair Allocation of Utilities in Multirate Mul-

ticast Networks", *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*, 1999.
[15] S. Sarkar, L. Tassiulas, "Distributed Algorithms for Computation of Fair Rates in Multirate Multicast Trees", *Proceedings of IEEE Infocom 2000*, Tel Aviv, Israel, March 2000.
[16] S. Sarkar, L. Tassiulas, "Fair Allocation of Discrete Bandwidth Layers in Multicast Networks", *Proceedings of IEEE Infocom 2000*, Tel Aviv, Israel, March 2000.
[17] N. Z. Shor, *Minimization Methods for Non-differentiable Functions*, Springer-Verlag, 1985.
[18] L. A. Wolsey, *Integer Programming*, John Wiley & Sons, 1998.

APPENDIX: PROOF OF THEOREM 1

We first state and prove a few lemmas that will be used in the proof of Theorem 1.

Let $\underline{\lambda}(n) = (\lambda_l(n), l \in L)$ denote the vector of dual variables at the nth iterative step, where $\lambda_l(n)$ is updated according to (7). Let Λ^* denote the set of all optimal solutions of the dual problem $\min_{\underline{\lambda} \geq \underline{0}} D(\underline{\lambda})$ (note the the dual optimal solution can be non-unique). It is easy to show that Λ^* is compact, i.e., closed and bounded. In the following, let $\rho(\underline{\lambda}, \Lambda^*) = \min_{\underline{\lambda}^* \in \Lambda^*} ||\underline{\lambda} - \underline{\lambda}^*||$ denote the Euclidean distance of a point $\underline{\lambda}$ from the set Λ^*. The following lemma states that the dual variables "converge" to a neighborhood around the optimum.

Lemma 1: Choose any $\epsilon > 0$. Then there exists an $\alpha_\epsilon > 0$ and an integer $N_\epsilon > 0$, such that for all α satisfying $0 < \alpha < \alpha_\epsilon$, the following result holds for all $n > N_\epsilon$:

$$\rho(\underline{\lambda}(n), \Lambda^*) \leq \epsilon \; .$$

The lemma can be proved by standard techniques in subgradient optimization theory, and is therefore omitted for brevity. (For instance, the lemma can be proved by proceeding along the same lines as that of Theorem 2.3 of Shor's classic text on this subject [17].)

For each $g \in G$, define the set $\tilde{Z}^g$ as $\tilde{Z}^g = \{\underline{x} : b_1^g \leq x_i \leq b_{K^g}^g \; \forall i \in I^g \cup \{s^g\} \; \forall g \in G\}$. For each $g \in G$, define $\tilde{X}^g = Y^g \cap \tilde{Z}^g$, where Y^g is given by (4). Let $\tilde{X} = \cap_{g \in G} \tilde{X}^g$. Also let $X = \cap_{g \in G} X^g$. Define $\tilde{P}(\underline{x}, \underline{\lambda}) = \sum_{i \in R} \tilde{U}_i(x_i) - \sum_{i \in I} (\sum_{l \in L_i} \lambda_l) x_i$.

Lemma 2: Let $\underline{x}^*$ be any optimal solution of $\tilde{\mathbf{P}}$. Then for any $\underline{\lambda}^* \in \Lambda^*$, the following holds:

$$\tilde{P}(\underline{x}^*, \underline{\lambda}^*) \; = \; \max_{\underline{x} \in \tilde{X}} \tilde{P}(\underline{x}, \underline{\lambda}^*) \; = \; \max_{\underline{x} \in X} \tilde{P}(\underline{x}, \underline{\lambda}^*) \; = \; D(\underline{\lambda}^*) \; .$$

Proof: We prove the lemma in three steps:

(i) $\max_{\underline{x} \in X} \tilde{P}(\underline{x}, \underline{\lambda}^*) = D(\underline{\lambda}^*)$: This follows straightforwardly from the fact that the functions U_i and $\tilde{U}_i$ are equal at the discrete bandwidth levels.

(ii) $\max_{\underline{x} \in \tilde{X}} \tilde{P}(\underline{x}, \underline{\lambda}^*) = \max_{\underline{x} \in X} \tilde{P}(\underline{x}, \underline{\lambda}^*)$: Consider any $\underline{\lambda} \geq \underline{0}$. Let $\underline{\hat{x}} = (\hat{x}_i, i \in I \cup S) \in \tilde{X}$ attain the maximum $\max_{\underline{x} \in \tilde{X}} \tilde{P}(\underline{x}, \underline{\lambda})$. For every $i \in I \cup S$, let $\hat{x}_i^l \leq \hat{x}_i$ represent the largest discrete bandwidth level no larger than $\tilde{x}_i$. Also, for every $i \in I \cup S$, let $\hat{x}_i^u \geq \hat{x}_i$ represent the smallest discrete bandwidth level no smaller than $\hat{x}_i$. Define $\hat{Z} = \{\underline{x} : \hat{x}_i^l \leq x_i \leq \hat{x}_i^u \; \forall i \in I \cup S\}$. Let $\hat{X} = \cap_{g \in G} Y^g \cap \hat{Z}$. Since $\hat{X} \subseteq \tilde{X}$, it follows that $\underline{\hat{x}}$ attains the maximum $\max_{\underline{x} \in \hat{X}} \tilde{P}(\underline{x}, \underline{\lambda})$. Now note that $\hat{X}$ is a polyhedron, and $\tilde{P}(\underline{x}, \underline{\lambda})$ is a linear function of $\underline{x}$ in $\hat{X}$ (this follows from the fact that the functions $\tilde{U}_i$ are linear in between the discrete bandwidth levels). Therefore, one of the "vertices" of $\hat{X}$ must attain the maximum of $\tilde{P}(\underline{x}, \underline{\lambda})$ in $\hat{X}$. It is also easy to see that all the vertices of the polyhedron $\hat{X}$ are elements of X. Therefore, there exists a $\underline{\bar{x}} \in X$ such that $\tilde{P}(\underline{\bar{x}}, \underline{\lambda}) = \max_{\underline{x} \in \hat{X}} \tilde{P}(\underline{x}, \underline{\lambda}) = \max_{\underline{x} \in \tilde{X}} \tilde{P}(\underline{x}, \underline{\lambda})$. Therefore, we obtain

$$\max_{\underline{x} \in \tilde{X}} \tilde{P}(\underline{x}, \underline{\lambda}) = \max_{\underline{x} \in X} \tilde{P}(\underline{x}, \underline{\lambda}) \quad \forall \underline{\lambda} \geq \underline{0} \; . \tag{30}$$

Choosing $\underline{\lambda} = \underline{\lambda}^*$ in (30), we obtain the desired result.

(iii) $\tilde{P}(\underline{x}^*, \underline{\lambda}^*) = \max_{\underline{x} \in \tilde{X}} \tilde{P}(\underline{x}, \underline{\lambda}^*)$: Since $\underline{\lambda}^*$ minimizes $D(\underline{\lambda}) = \max_{\underline{x} \in X} \tilde{P}(\underline{x}, \underline{\lambda})$ over $\underline{\lambda} \geq \underline{0}$, it follows from (30) that $\underline{\lambda}^*$ must also minimize $\max_{\underline{x} \in \tilde{X}} \tilde{P}(\underline{x}, \underline{\lambda})$ over $\underline{\lambda} \geq \underline{0}$. Therefore, $\underline{\lambda}^*$ must be an optimal solution of the dual of problem $\tilde{\mathbf{P}}$. Since $\tilde{\mathbf{P}}$ is a maximization problem with concave objective function and linear constraints, there is no duality gap (from Proposition 5.2.1 of [2]). Then from Propositions 5.1.4 and 5.1.1, we obtain $\tilde{P}(\underline{x}^*, \underline{\lambda}^*) = \max_{\underline{x} \in \tilde{X}} \tilde{P}(\underline{x}, \underline{\lambda}^*)$.

Combining (i), (ii) and (iii), we obtain the desired result. □

Lemma 3: There exists a constant $C < \infty$, such that for every $\underline{\lambda}_1, \underline{\lambda}_2 > 0$, the following holds:

$$||D(\underline{\lambda}_1) - D(\underline{\lambda}_2)|| \quad \leq \quad C||\lambda_1 - \lambda_2|| \; .$$

Proof: For any $\underline{\lambda} \geq \underline{0}$, let $\underline{x}(\underline{\lambda}) = (x_i(\underline{\lambda}), i \in I \cup S)$ represent the point which achieves the maximum in $\max_{\underline{x} \in X} \{ \sum_{i \in R} U_i(x_i) - \sum_{i \in I} (\sum_{l \in L_i} \lambda_l) \, x_i \}$. Define a $|L|$-dimensional vector $\underline{y}(\underline{\lambda})$ as follows: $\underline{y}(\underline{\lambda}) = (y_l(\underline{\lambda}), l \in L)$ where $y_l(\underline{\lambda}) = c_l - \sum_{i \in I_l} x_i(\underline{\lambda})$. It can be shown that $\underline{y}(\underline{\lambda})$ is a subgradient of $D(\underline{\lambda})$ at $\underline{\lambda}$ (see Section 6.1 of [2]). Note that since X is a bounded set, $\underline{x}(\underline{\lambda})$ is bounded. Therefore, there exists a $C < \infty$ such that $||\underline{y}(\underline{\lambda})|| \leq C$ for all $\underline{\lambda} \geq \underline{0}$. From these facts, and the definition of a subgradient (see Section 6.1 of [2]), we obtain the following for any $\underline{\lambda}_1, \underline{\lambda}_2 \geq \underline{0}$:

$$\begin{aligned} D(\underline{\lambda}_1) - D(\underline{\lambda}_2) &\leq \langle \underline{\lambda}_1 - \underline{\lambda}_2, \underline{y}(\underline{\lambda}_1) \rangle \\ &\leq ||\underline{\lambda}_1 - \underline{\lambda}_2|| \, ||\underline{y}(\underline{\lambda}_1)|| \\ &\leq C||\underline{\lambda}_1 - \underline{\lambda}_2||. \end{aligned} \tag{31}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. Similarly, we also obtain

$$\begin{aligned} D(\underline{\lambda}_1) - D(\underline{\lambda}_2) &\geq \langle \underline{\lambda}_1 - \underline{\lambda}_2, \underline{y}(\underline{\lambda}_2) \rangle \\ &\geq -||\underline{\lambda}_1 - \underline{\lambda}_2|| \, ||\underline{y}(\underline{\lambda}_2)|| \\ &\geq -C||\underline{\lambda}_1 - \underline{\lambda}_2||. \end{aligned} \tag{32}$$

Combining (31) & (32), we obtain the desired result. □

Proof of Theorem 1: Let $\hat{b} = \min_{g \in G} \min_{k \in \{1, \ldots, K^g - 1\}} (b_{k+1}^g - b_k^g) > 0$. Therefore, $\hat{b}$ is the minimum difference in bandwidth between two adjacent discrete bandwidth levels, in any multicast group. Also let $B = \max_{g\in G} b^g_{K^g} > 0$ denote the maximum bandwidth level in any multicast group. Let $\underline{x}(n) = (x_i(n), i \in I \cup S)$ denote the rate vector at the nth iterative step, as defined by (8). Let $\gamma = \min_{i\in R} \gamma_i > 0$.

Choose $\epsilon = \frac{\gamma \hat{b}^2}{4(|I||L|B+C)}$ in Lemma 1. Assume that the step-size α used for updating the link prices (as in (7)) satisfies $0 < \alpha < \bar{\alpha} = \alpha_\epsilon$. Then from Lemma 1, there exists a $\underline{\lambda}^* \in \Lambda^*$, such that for all $n > \bar{N} = N_\epsilon$,

$$||\underline{\lambda}(n) - \underline{\lambda}^*|| \le \frac{\gamma \hat{b}^2}{4(|I||L|B+C)} \,. \qquad (33)$$

Consider any $n > \bar{N}$. Let us assume, for the sake of contradiction, that $|x_i^* - x_i(n)| > \bar{b}^g$ for some $i \in R^g$, $g \in G$. Since the difference between adjacent discrete bandwidth levels of group g is at most $\bar{b}^g$, it follows that there exists a $\hat{x}_i \in \{b^g_1, ..., b^g_{K^g}\}$ such that $0 < |x_i^* - \hat{x}_i| < |x_i^* - x_i(n)|$. Thus $\hat{x}_i$ represents some discrete bandwidth level in-between (and excluding) x_i^* and $x_i(n)$, and must satisfy one of the two following conditions: (i) $x_i(n) < \hat{x}_i < x_i^*$, or (ii) $x_i^* < \hat{x}_i < x_i(n)$. Define $\theta_1 = \hat{x}_i - x_i(n)$, and $\theta_2 = x_i^* - \hat{x}_i$. From the above discussion, it follows that $|\theta_1| \ge \hat{b}$, $|\theta_2| > 0$, and $|\theta_1 + \theta_2| > \bar{b}^g \ge \hat{b}$ (by assumption). Define $\underline{\hat{x}} = (\hat{x}_{i'}, i' \in I \cup S)$ as $\underline{\hat{x}} = \frac{\theta_1}{\theta_1+\theta_2}\underline{x}^* + \frac{\theta_2}{\theta_1+\theta_2}\underline{x}(n)$. Geometrically, $\underline{\hat{x}}$ represents the point at which the straight line between $\underline{x}^*$ and $\underline{x}(n)$ cuts the plane $x_i = \hat{x}_i$. Note that since $\underline{x}^*, \underline{x}(n) \in \tilde{X}$, it follows that $\underline{\hat{x}} \in \tilde{X}$.

$$\begin{aligned}
&\tilde{P}(\underline{\hat{x}}, \underline{\lambda}^*) - \frac{\theta_1}{\theta_1+\theta_2}\tilde{P}(\underline{x}^*, \underline{\lambda}^*) - \frac{\theta_2}{\theta_1+\theta_2}\tilde{P}(\underline{x}(n), \underline{\lambda}^*) \\
&= \sum_{i'\in R} \tilde{U}_{i'}(\hat{x}_{i'}) - \frac{\theta_1}{\theta_1+\theta_2}\sum_{i'\in R}\tilde{U}_{i'}(x^*_{i'}) - \\
&\quad \frac{\theta_2}{\theta_1+\theta_2}\sum_{i'\in R}\tilde{U}_{i'}(x_{i'}(n)) && (34)\\
&\ge \tilde{U}_i(\hat{x}_i) - \frac{\theta_1}{\theta_1+\theta_2}\tilde{U}_i(x_i^*) - \frac{\theta_2}{\theta_1+\theta_2}\tilde{U}_i(x_i(n)) && (35)\\
&\ge U_i(\hat{x}_i) - \frac{\theta_1}{\theta_1+\theta_2}U_i(x_i^*) - \frac{\theta_2}{\theta_1+\theta_2}U_i(x_i(n)) && (36)\\
&\ge \frac{\gamma\theta_1\theta_2}{2} \,. && (37)
\end{aligned}$$

Relation (34) follows from the fact that the term $\sum_{i\in I}(\sum_{l\in L_i} \lambda_l(n))\, x_i$ in $\tilde{P}(\underline{x}, \underline{\lambda})$ is linear in $\underline{x}$. Relation (35) follows from the fact that the term $\sum_{i'\in R\setminus\{i\}} \tilde{U}_{i'}(x)$ is a concave function of $\underline{x}$. Relation (36) is obtained using the facts $\tilde{U}_i(\hat{x}_i) = U_i(\hat{x}_i)$, $\tilde{U}_i(x_i^*) \le U_i(x_i^*)$ and $\tilde{U}_i(x_i(n)) = U_i(x_i(n))$, which directly follow from (9), the definition of $\tilde{U}_i$. Relation (37) follows from the strict concavity of U_i (Assumption 1) and the fact $\gamma_i \ge \gamma$.

$$\begin{aligned}
\tilde{P}(\underline{x}(n), \underline{\lambda}^*) - \tilde{P}(\underline{x}(n), \underline{\lambda}(n)) &= \sum_{i'\in I} x_{i'}(n) \sum_{l\in L_{i'}} (\lambda_l^* - \lambda_l(n)) \\
&\le \frac{|I||L|B\gamma\hat{b}^2}{4(|I||L|B+C)}. && (38)
\end{aligned}$$

Relation (38) follows from (33), and from the fact that $|x_{i'}| \le B \;\forall i' \in I$.

$$\begin{aligned}
\tilde{P}(\underline{x}(n), \underline{\lambda}(n)) - \tilde{P}(\underline{x}^*, \underline{\lambda}^*) &= D(\underline{\lambda}(n)) - D(\underline{\lambda}^*) && (39)\\
&\le C||\underline{\lambda}(n) - \underline{\lambda}^*|| && (40)\\
&\le \frac{C\gamma\hat{b}^2}{4(|I||L|B+C)}. && (41)
\end{aligned}$$

Note that (39) is obtained using Lemma 2, the definition of $\underline{\lambda}(n)$ (see (8)), and the fact that the functions U_i and $\tilde{U}_i$ are equal at the discrete bandwidth levels. Relation (40) follows Lemma 3, and relation (41) follows from (33). From (38) and (41), we obtain

$$\tilde{P}(\underline{x}(n), \underline{\lambda}^*) - \tilde{P}(\underline{x}^*, \underline{\lambda}^*) \;\le\; \frac{\gamma\hat{b}^2}{4} \,. \qquad (42)$$

Combining (37) and (42), we obtain

$$\begin{aligned}
\tilde{P}(\underline{\hat{x}}, \underline{\lambda}^*) - \tilde{P}(\underline{x}^*, \underline{\lambda}^*) &\ge \frac{\gamma\theta_1\theta_2}{2} - \frac{\gamma\hat{b}^2\theta_2}{4(\theta_1+\theta_2)} \\
&\ge \frac{\gamma\hat{b}\theta_2}{2} - \frac{\gamma\hat{b}^2\theta_2}{4\hat{b}} && (43)\\
&= \frac{\gamma\hat{b}\theta_2}{4} \\
&> 0 \,. && (44)
\end{aligned}$$

Relation (43) follows from the fact $(\theta_1 + \theta_2) \ge \theta_1 \ge \hat{b}$, and (44) follows from the fact $\gamma, \hat{b}, \theta_2 > 0$. From (44), it follows that $\underline{x}^*$ cannot attain the maximum of $\tilde{P}(\underline{x}, \underline{\lambda}^*)$ over $\tilde{X}$, which contradicts Lemma 2. Therefore, our assumption that there exists some $i \in R^g$, $g \in G$ such that $|x_i^* - x_i(n)| > \bar{b}^g$, was incorrect, thus proving Theorem 1. □