Hybrid Unicast and Multicast Flow Control: A Linear Optimization Approach

Homayoun Yousefi'zadeh Fatemeh Fazel Hamid Jafarkhani

Department of Electrical Engineering and Computer Science
University of California, Irvine

***Abstract*— In this paper, we present a solution to the general problem of flow control for both unicast and multicast IP networks. We formulate a convex optimization problem that can be analytically solved with a low complexity. We show that with the proper choice of parameters, our problem can be fine-tuned to reward multicast flows or to provide max-min fairness. Further, our formulation can be deployed in the form of a centralized, decentralized, or quasi-centralized flow control scheme. Utilizing the solution to our optimization problem, we propose flow control algorithms requiring very little or no per flow state information. Our proposed algorithms can be implemented by making use of a simple ECN marking scheme to convey minimum per link or per zone flow fairness information to the end nodes. We also show how our flow control results can be adopted in the context of layered media multicast applications.**

***Index Terms*—Unicast and Multicast IP Networks, Heterogeneity, Flow Control, Flow Priority, Multicast Reward, Max-Min Fairness, Optimality, Layered Media.**

I. Introduction

In the past decade, multicasting techniques have been in widespread use for communication networking applications as efficient means of network resource sharing. However, utilizing multicasting techniques has introduced significant technical challenges at different levels. Enforcing flow (inter-session) fairness among a set of competing flows is one of the most important challenges of utilizing multicasting techniques. For the lack of any built-in flow fairness mechanism in UDP and due to the fact that multicast sessions are typically built on top of UDP, achieving flow fairness in hybrid unicast and multicast networks is in fact a complex task.

In what follows, we briefly review related flow control work in the context of our current research work. The original TCP flow control was discussed by Jacobson [16] and further enhanced by Floyd et al. [9]. In the recent years, the proposition of Explicit Congestion Notification (ECN) marking techniques proposed by Ramakrishnan et al. [25] and by Lapsley et al. [20] has brought the promise of practical deployment of effective flow and congestion control algorithms for the existing Internet infrastructure. In addition, applications of control and optimization theories such as the ones described by [7], [10], [12], [26] have shed light on the general problem of flow control in computer communication networks. Although leading to rather different flow control strategies, the key promise of most of the recent results is to maximize a set of utility functions pertaining to the benefit of various network entities while potentially considering pricing issue. Another closely related literature approach to our current topic advocates a game-theoretic approach as described by [24] and [19] in which reaching a stable Nash equilibrium solution is desired. The main focus of this research work is to develop a generic flow control framework. Our generic framework can be fine tuned to satisfy a variety of fairness measures such as the multicast flow reward metric, i.e., allocating bandwidth proportional to the number of end nodes, or the so-called max-min fairness metric of [4] defined below.

Definition 2.1: A bandwidth allocation scheme among a number of competing flows is max-min fair if no flow can be allocated a higher bandwidth without reducing the allocation of another flow with an equal or a lower rate.

In this study, we pay special attention to the results of Athuraliya et al. [1], Graves et al. [14], Kelly et al. [18], Low et al. [23], Kunniyur et al. [19], Ramakrishnan et al. [25], and Sarkar et al. [28]. Our formulation of the flow control problem is best categorized under the optimization flow control techniques. It aims at maximizing a global, per link, or per zone set of utility functions defined over the complete path of unicast and multicast tree topologies. More specifically, our formulation of the flow control problem is a convex optimization problem defined over a set of piecewise linear utility functions. The main advantage of utilizing such a set of utility functions compared to the previously proposed nonlinear utility functions is simplicity. Not only appealing from the complexity stand point, our technique can also satisfy important characteristics of the well-behaved algorithms such as guaranteed existence, boundedness, stability, and scalability. With respect to practicality, the resulting proposed algorithm can be implemented in real-time by taking advantage of a binary ECN marking mechanism currently under review by IETF [25].

In summary, our solution to the formulation of the flow control problem identifies maximum achievable fair rates for individual unicast and multicast sessions sharing the same network infrastructure. An outline of the paper follows. In Section II, we formulate and analytically solve our generic optimal flow control problem. In this section, we also show how the parameters of the problem can be fine tuned to provide max-min fairness as well as multicast flow reward among other metrics of potential interest. In Section III, we describe the implementation of our flow control algorithm based on the utilization of ECN marks. In Section IV, we apply our flow control schemes to layered and replicated media scenarios. In Section V, we numerically validate our analytical results. Finally, Section VI includes a discussion of concluding remarks and future work.

II. Flow Control Optimization

In this section, we focus on our generic optimal flow control problem categorized under constraint convex optimization problems with piecewise linear objective functions and its solution. We start from a generic formulation of the problem aiming at providing unicast and multicast flow control for a given network topology. We then describe centralized, decentralized, and quasi-centralized versions of our formulation and how they can be related to rewarding multicast flows and/or providing max-min fair algorithms.

Assume f flows are sharing a set of links L over a particular network topology. Further, assume the capacity of link j where $j \in L$ is specified by C_j. Each flow i has a maximum requested bandwidth denoted by X_i. Depending on the characteristics of flow i the term X_i could vary from a minimum guaranteed available bandwidth for a restricted flow to the full capacity of the bottleneck link over a unicast or a multicast path for an unrestricted flow. Hence, assigning a bandwidth higher than the requested value X_i to flow i leads to capacity wastage of the set of links utilized by flow i. In accordance with the latter assumption, we select the following concave utility function to represent the fairness of individual flows.

$$U_i(x_i) \;=\; \min\,(\frac{x_i}{X_i}, 1) \;=\; \left\{ \begin{array}{ccc} \frac{x_i}{X_i} & , & x_i \leq X_i \\ 1 & , & x_i > X_i \end{array} \right. \qquad (1)$$

A. Centralized Flow Control

Assuming an ordered set of bandwidth requirements $X_1, X_2, \cdots, X_f$ such that $X_1 \leq X_2 \leq ... \leq X_f$, our formulation of the centralized flow control problem is now described in the form of the following Linear Programming (LP) problem.

$$\begin{array}{ll} \max\limits_{x_1,\ldots,x_f} & \sum\limits_{i=1}^{f} W_i\,(\frac{x_i}{X_i}) \\ \text{Subject To :} & \sum\limits_{i=1}^{f} w_{ji}\,x_i \leq C_j \quad \forall j \in L \\ & x_i \leq X_i \quad \forall i \in \{1, \cdots, f\} \\ & x_i \leq x_{i+1} \quad \forall i \in \{1, \cdots, f-1\} \end{array} \qquad (2)$$

where f is the total number of flows over a given network topology, C_j is the capacity of link j, and $\{W_1, \cdots, W_f\}$ is the set of weighting functions associated with the flows. Further, w_{ji} are defined as

$$w_{ji} \;=\; \left\{ \begin{array}{ccl} 1 & , & \text{if flow}\, i \,\text{utilizes link}\, j \\ 0 & , & \text{otherwise} \end{array} \right. \qquad (3)$$

indicating whether link j is utilized by flow i. We note that different choices of the weighting functions can lead to implementation of different flow control algorithms. In the theorem below, we investigate the condition under which the solution to the problem formulation of (2) satisfies the max-min fairness metric of [4].

Theorem 2.1: If the solution to the problem formulation of (2) is max-min fair by means of Definition 2.1, then

$$\frac{W_1}{X_1} > \cdots > \frac{W_f}{X_f} \qquad (4)$$

Appendix I proves Theorem 2.1.

Next, we set the weighting functions $\{W_1, \cdots, W_f\}$ with the objective of rewarding multicast flows. We define

$$W_i = \frac{\eta_i}{\sum_{j \in L}\, w_{ji}} \qquad (5)$$

where η_i is the number of flow end nodes and $i \in \{1, \cdots, f\}$. We note that the weighting functions $\{W_1, \cdots, W_f\}$ in Equation (5) are set proportional to the number of the end nodes of a flow and the reverse of the number of links traversed by it. From a practical stand point, the number of end nodes is known in the case of unicast sessions. In the case of multicast sessions, it can be identified from the specification of the multicast group membership. While the choice of the weighting functions in Equation (5) does not necessarily yield a max-min fair solution, it rewards a flow with a larger number of receiving end nodes and smaller number of utilized links. Other choices of the weighting functions $\{W_1, \cdots, W_f\}$ in the formulation of (2) result in implementing other flow priority mechanisms. For example, for a case in which discovering the number of links utilized by a flow introduces a prohibitive overhead, Equation (5) can be simply changed to

$$W_i = \eta_i \qquad (6)$$

The LP problem of (2) can be solved relying on one of the existing methods such as the LU-decomposition method or sparse Bartel-Golub method as described in [13]. Alternatively, the problem may be solved relying on the iterative approximation method of [2]. Depending on the choice of algorithm and numerical applicability, the average complexity of solving the LP problem of (2) can be in the order of $\mathcal{O}((l+2f)^2)$ where l is the number of links over the network topology L and f is the number of flows. The selection of the weighting functions is the design consideration that can be offset by the relative importance of priority over max-min fairness.

B. Decentralized Flow Control

Considering the need for accessing global state information among the set of links of a given network topology as well as the complexity of the solution to the global problem above, we reduce the global problem into a set of per link flow control optimization problems. The set of per link problems can then be solved independently and with a linear complexity for both unicast and multicast flows and without requiring to access any state information among the links of a given topology. Not requiring to access state information, however, comes in exchange for potential estimation of flow fair shares yielding to sub-optimality. The latter is due to the fact that a fair share calculated for a flow at a link may be subject to extra limitations or relaxations imposed by another link. We note that for a set of fair shares calculated over the links utilized by a flow, the overall fair share of the flow is the minimum of those fair shares.

Assume f flows are sharing a link with capacity C and each flow i has a maximum required bandwidth X_i. Relying on the definition of the convex utility function of (1) and assuming an ordered set of bandwidth requirements $X_1, X_2, \cdots, X_f$ such that $X_1 \leq X_2 \leq ... \leq X_f$, our per link formulation of the flow control problem is now described in the form of the following optimization problem.

$$\begin{array}{ll} \max\limits_{x_1,\ldots,x_f} & \displaystyle\sum_{i=1}^{f} \frac{x_i}{X_i} \\ \text{Subject To :} & \displaystyle\sum_{i=1}^{f} x_i \leq C \\ & x_1 \leq x_2 \leq \ldots \leq x_f \end{array} \quad (7)$$

where f is the number of competing flows over a link and C is the capacity of the link. We observe that solving per link optimization problem of (7) does not require accessing any state information. The problem can be solved utilizing a similar approach as the one utilized in the previous subsection and noting the fact that Condition (4) holds. Rather than relying on the approach of the previous subsection, we select water-filling approach in order to find the unique solution of the problem with a lower complexity while satisfying max-min fairness property of Definition 2.1. We express the water-filling solution to the LP problem of (7) as

Case 1: If $C \geq \sum_{j=1}^{f} X_j$

$$x_i = X_i \quad , \quad 1 \leq i \leq f \quad (8)$$

Case 2: If $C < \sum_{j=1}^{f} X_j$

$$x_i = \left\{ \begin{array}{ll} X_i & , \quad 1 \leq i \leq h \\ \frac{C - \sum_{j=1}^{h} X_j}{f-h} & , \quad h+1 \leq i \leq f \end{array} \right. \quad (9)$$

where x_i is the bandwidth assigned to the i-th flow and h is an integer satisfying the following condition

$$\begin{array}{ccc} X_h \leq & \frac{C - \sum_{i=0}^{h} X_i}{f-h} & \leq X_{h+1} \\ 0 \leq & h & \leq f-1 \end{array} \quad (10)$$

for $X_0 \overset{\triangle}{=} 0$. Next, we provide the statement of optimality for our proposed water-filling approach.

Theorem 2.2: the water-filling solution of Equation (9) is the optimal solution to the LP problem of (7).

Appendix II proves Theorem 2.2.

We observe that the water-filling approach of Equation (9) starts by dividing the bandwidth equally among all of the f flows until the first flow reaches its maximum required bandwidth X_1, then it fixes the assigned bandwidth for the first flow to X_1 and divides the remaining bandwidth among the remaining flows equally, and so on. Consequently, the flows that have reached their saturation regions receive their maximum requested bandwidth while the other flows receive equal shares of the remaining bandwidth guaranteed not to be less than the assigned shares of flows in their saturation regions. The method is hence max-min fair. It is worth mentioning that while the decentralized flow control of this section may resemble the work of [5], there are some significant differences. First, we note that our approach does not work with explicit definition of max-min fairness. Rather it relies on an optimization utility function that among other things can lead to max-min fairness. Further, our water filling solution and its optimality proof are independent of the work of [5]. We note that the complexity of solving the LP problem of (2) utilizing our proposed water-filling scheme at an intermediate node accommodating f competing flows, is in the order of $\mathcal{O}(f)$. The complexity is much lower than the complexity of the centralized scheme not only because of the difference between linear and quadric complexities but considering the fact that the calculations are distributed among different intermediate nodes. The lower complexity, however, comes at the expense of introducing a sub-optimal solution compared to the centralized scheme.

C. Quasi-Centralized Flow Control

In this subsection, we envision a quasi-centralized flow control optimization problem that can be solved over local zones. The idea behind proposing such a scenario is to address the tradeoff between accuracy of the centralized scheme of Section II.A and the practicality of the decentralized scheme of Section II.B. In such a scenario, the optimization problem of Section II.A can be solved over the topologies of local zones in which exchanging state information is not overhead prohibitive. The minimum fair share of a flow spanning over multiple zones is the minimum of fair shares calculated over different zones. We note that such a scheme is both effective and accurate specifically for sparse topologies in which flows are typically spread over a local zone rather than the global zone.

We propose the use of designated nodes (DNs) to carry out the calculations of flow control in each zone. While the DN of a zone is typically an intermediate node it can also be an end node. The nodes belonging to each zone are identified using a simple discovery process initiated by the DN of the zone. The discovery process relies on the use of broadcast pilot packets with a specified TTL indicating the depth of a zone centered around the zone's DN node. Knowing the set of nodes belonging to a zone, the DN of the zone can determine per zone fair shares of zone flows using one of the techniques Section II.A. The fair share of the zones can then be announced to the zone members if a distributed enforcement of a flow control protocol is desired. Such a scenario is of interest in highly populated zones where centralized per zone enforcement of flow control can introduce a bottleneck. Otherwise, the DN of a zone can participate on behalf of the zone nodes in a distributed flow control scheme aiming at the enforcement of a flow control protocol. In the next section, we describe the details of our flow control protocol.

The average complexity of the quasi-centralized scheme of this section is in the order of $\mathcal{O}((l_z + 2f_z)^2)$ where l_z is the number of links over the largest zone and f_z is the number of flows in the largest zone. The complexity is better that of the centralized scheme of Section II.A and worse than that of the decentralized scheme of Section II.B. However, the preciseness of the scheme is closer to that of the centralized scheme than that of the decentralized scheme.

III. Utilizing ECN Marks in the Implementation of the Flow Control Protocol

In this section, we describe the implementation of a flow control protocol based on the results of the previous section for both unicast and multicast flows.

A. Flow Control Protocol

In this subsection, we propose our flow control algorithm for a unicast session considering the ECN marking scheme of [25] along with the mark-based bottleneck estimation technique of [14] and the scaled shadow prices of [18]. In the next subsection, we generalize our approach to detect the bottleneck link of a multicast session. Before describing our approach, we make note of the differences between our approach and the literature work cited above. First, the choice of utility functions and the resulting optimization problems are not related. While our linear optimization problem is a generic formulation of the flow control, the nonlinear problems of the literature work above are formulated to provide max-min fairness. Second, we present an exact solution to our problem rather than an iterative solution asymptotically converging to the Lagrange multipliers or the shadow prices of a nonlinear optimization problem. Third, because of introducing an exact solution, we also eliminate the need for providing a discussion of stability regarding the convergence of our approach.

We assume that solving the set of optimization problems of Section II.B or Section II.C yields to an ordered list of per link or per zone fair shares. While in the rest of the discussion of this section we describe our algorithm applied to a set of links, we note that the same exact method can be used to apply our algorithm to a set of zones. Supposing that an ordered list of per link fair shares is identified as $\{l_k^1, l_k^2, \cdots, l_k^{n_k}\}$ with $l_k^1 \leq l_k^2 \leq \cdots \leq l_k^{n_k}$ for a unicast session traversing the set of links $\{l_k^1, l_k^2, \cdots, l_k^{n_k}\}$, we can specify the minimum fair share l_k^1 of the session as follows. Defining $\lambda_k^i \stackrel{\triangle}{=} \frac{1}{l_k^i}$ for $k \in \{1, \cdots, n_k\}$, identifying minimum fair share of a session traversing links $\{l_k^1, l_k^2, \cdots, l_k^{n_k}\}$ is equivalent to specifying λ_k^1 in the ordered list of $\{\lambda_k^1, \lambda_k^2, \cdots, \lambda_k^{n_k}\}$ with $\lambda_k^1 \geq \lambda_k^2 \geq \cdots \geq \lambda_k^{n_k}$. Next, we note that

$$\begin{aligned}\Phi_k \quad &\stackrel{\triangle}{=} [(\lambda_k^1)^{\mathcal{N}} + \cdots + (\lambda_k^{n_k})^{\mathcal{N}}]^{\frac{1}{\mathcal{N}}} \\ &= (\lambda_k^1)[1 + (\frac{\lambda_k^2}{\lambda_k^1})^{\mathcal{N}} + \cdots + (\frac{\lambda_k^{n_k}}{\lambda_k^1})^{\mathcal{N}}]^{\frac{1}{\mathcal{N}}}\end{aligned} \tag{11}$$

can be approximated by λ_k^1 for some large number $\mathcal{N}$ considering the fact that $\frac{\lambda_k^i}{\lambda_k^1} \leq 1$ for $i \in \{2, \cdots, n_k\}$. Hence, identifying the bottleneck link of the session is equivalent to conveying Φ_k to the end nodes of a unicast session or as discussed subsequently to the receivers of a multicast session. Assuming that a packet is marked at link i of the underlying session with probability $1 - \exp(-(\lambda_k^i)^{\mathcal{N}})$, the end nodes of the session can obtain an estimate of the minimum fair share of the session $(-\ln \Xi_k)^{\frac{-1}{\mathcal{N}}}$ at any interval of time by measuring the receiving rate of unmarked packets $\Xi_k = \exp(-\Phi_k^{\mathcal{N}})$ in that interval. Next, we note that numerical implementation of such an algorithm is unstable due to the fact that the value of Ξ_k for large $\mathcal{N}$ can be either very close to 0 or 1 depending on the value of λ_k^i. To overcome the above-mentioned problem, we utilize the following transformation

$$\varphi_k^i \stackrel{\triangle}{=} \log_{b_k^i}(C_k^i\ \lambda_k^i) \tag{12}$$

and apply appropriate coefficients b_k^i, C_k^i to keep the value of φ_k^i close to 1. We note that the value of φ_k^i is guaranteed to satisfy the following inequality

$$1 - \epsilon \leq \varphi_k^i \leq 1 + \epsilon \tag{13}$$

for every link i of the underlying session and some small number ϵ by selecting

$$b_k^i = [\frac{l_{k,max}^i}{l_{k,min}^i}]^{\frac{1}{2\epsilon}} \tag{14}$$

and

$$C_k^i = l_{k,max}^i\, [\frac{l_{k,max}^i}{l_{k,min}^i}]^{\frac{1-\epsilon}{2\epsilon}} \tag{15}$$

where $\lambda_k^i = \frac{1}{l_k^i}$, $l_k^i \in [l_{k,min}^i, l_{k,max}^i]$, $l_{k,min}^i$ indicates minimum guaranteed rate of the session, and $l_{k,max}^i$ indicates the capacity of link i. For clarity, we note that Equation (11) under the transformation of Equation (12) is changed to

$$\begin{aligned}\Phi_k \quad &\stackrel{\triangle}{=} [(\varphi_k^1)^{\mathcal{N}} + \cdots + (\varphi_k^{n_k})^{\mathcal{N}}]^{\frac{1}{\mathcal{N}}} \\ &= (\varphi_k^1)[1 + (\frac{\varphi_k^2}{\varphi_k^1})^{\mathcal{N}} + \cdots + (\frac{\varphi_k^{n_k}}{\varphi_k^1})^{\mathcal{N}}]^{\frac{1}{\mathcal{N}}}\end{aligned} \tag{16}$$

specifying the number of unmarked packets as

$$\Xi_k = \exp(-\Phi_k^{\mathcal{N}}) \tag{17}$$

We now introduce the following pair of flow control algorithms that can be implemented respectively in the intermediate nodes and the end nodes of a unicast or subsequently a multicast session relying on a binary ECN marking scheme.

Flow Control Algorithm: Intermediate Node

- Calculate minimum fair share of the link from one of the algorithms of Section II.B or Section II.C.
- Determine the value of $\lambda_k^i = 1/l_k^i$.
- Compute the values of b_k^i and C_k^i from Equation (14) and Equation (15), respectively.
- Calculate the value of φ_k^i from Equation (12).
- Mark a packet with probability $1 - \exp(-(\varphi_k^i)^{\mathcal{N}})$ for some large $\mathcal{N}$.

We note that marking the packets in the last step of the algorithm above can be done by individual intermediate nodes or only DNs of individual zones depending on whether the scheme of Section II.B or Section II.C are deployed.

Flow Control Algorithm: End Node

- Calculate the rate of receiving unmarked packets from Equation (17) for the previous time interval.
- Approximate minimum fair share of the path from the source as $(C_k^i \;/\; (b_k^i)^{\vartheta_k})$ where $\vartheta_k = [-\ln \Xi_k)]^{1/\mathcal{N}}$.

We note that it is highly likely for an intermediate node to have a set of fixed values for b_k^i and C_k^i over the life time of a slowly varying session and conclude that the first step of the intermediate node algorithm is likely to be taken only once at the time of session establishment. One last consideration is that the proposed flow control algorithms of this section are only effective assuming that the life cycle of a flow is relatively longer than the time required for communicating the information through ECN marks. Practically speaking, the algorithms are effective in environments for which the shortest flow life cycles are at least a multiple of the round trip times between sender and receivers.

B. *Multicasting Implications*

In this subsection, we discuss different aspects of protocol implementation when coping with multicast networks. Without loss of generality, we consider multicast tree architectures with one source and many receivers. We observe that the proposed protocol of the previous subsection can now be utilized for multicast sessions by considering a multicast session as a set of virtual unicast sessions with each virtual session consisting of the source of a multicast session and an individual receiver of the session. It is worth mentioning that protocol implementation of the previous section for a multicast session is subject to the feedback implosion problem. In addressing the implosion problem, we assume that the source of a multicast session initiates the discovery process of identifying the fair shares by sending pilot multicasting packets to the members of a multicast session. The feedback implosion problem can be addressed using one of the following two alternatives. When the objective is to access the bottleneck information of a group of end nodes, the aggregation methods of [21] and [22] can be utilized for aggregating the feedback and sending the response on behalf of the nodes of the zone to the source of the session. This scenario is specifically attractive if used in conjunction with our quasi-centralized algorithm. When the objective is to discover an overall minimum fair share rather than individual minimum fair shares of the receivers of a multicast session, the technique of [8] described below can be used. Upon the receipt of polling packets, receivers of a multicast session set their own timers with a random value. Each receiver reports its minimum fair share after having an expired timer only if it has not seen a smaller fair share value reported by another receiver of the multicast session.

IV. Flow Control for Layered and Replicated Media Systems

We continue our discussion by elaborating on how the current research work fits into the framework of layered or replicated media systems over multicast IP networks. We recall that such systems have strict real-time constraints and hence have a need for low complexity flow control algorithms.

Noting the fact that in a layered or a replicated media system a layer is mapped onto a multicast group, we provide the following briefing to describe a layered media system. Consider a multicast media session with a partitioning of receivers into K groups. For a session with N receivers and K groups, each group $k \in \{1, \cdots, K\}$ consists of N_k receivers such that $N = \sum_{k=1}^{K} N_k$. For such a media session a set $P = \{G_1 | \cdots | G_K\}$ is called a partitioning of the receiver set $\{1, \cdots, N\}$ if P is a decomposition of the set of receivers into a family of disjoint sets. The term group rate is used to denote aggregate receiving rate of a receiver in the group while the term layer rate is used to denote transmission rate to a specific layer. For an ordered partitioning of receivers into K groups with ordered group rates of $g_1, g_2, \cdots, g_K$ such that $g_1 \leq g_2 \leq \cdots \leq g_K$ the layer rates of a layered media session are calculated in the form of

$$g_1, g_2 - g_1, g_3 - g_2, \cdots, g_K - g_{K-1} \qquad (18)$$

A receiver in group k subscribes to layers 1 through k receiving an aggregate rate of g_k. Interpretation of the above formulation in case of replicated media streams is also straight forward. For an ordered partitioning of the receivers into K groups $G_1, G_2, \cdots, G_K$ with ordered group rates of $g_1, g_2, \cdots, g_K$ such that $g_1 \leq g_2 \leq \cdots \leq g_K$ the layer rates are the same as the group rates. A receiver in group k only subscribes to layer k receiving a rate of g_k.

We now note that our formulation of the flow control problem can be applied to a layering architecture described above by treating different multicast groups associated with different layers as independent flows. As the result, we observe the pleasant behavior of our centralized algorithm with a flow priority mechanism implementation based on the number of end nodes associated with a flow and as indicated by Equation (6). Considering Condition (4) and paying attention to the fact that for a layering architecture the relationships $W_1 \geq \cdots \geq W_f$ and $X_1 \leq \cdots \leq X_f$ hold, we conclude that our proposed centralized algorithm never accommodates lower priority higher bandwidth layers before accommodating higher priority lower bandwidth layers. Additionally, we note that the behavior of our decentralized algorithm is also the same considering the fact that the relationships $W_1 = \cdots = W_f = 1$ and $X_1 \leq \cdots \leq X_f$ hold. However, we make note that both of the above-mentioned algorithms may partially accommodate different layers of a media session as the result of total available bandwidth limitations. While this works fine for a replicated media system, a layered media system should allow the receivers to modify their reporting logic such that the bandwidths of lower priority higher bandwidth layers are applied to higher priority lower bandwidth layers resulting in complete fulfillment of the requirements of higher priority layers one layer at a time.

It is also important to note that considering the mapping of the layer rates to the aggregated group rates as indicated by Equation (18), the minimum fair share of a group is the sum of minimum fair shares of the multicast groups representing specific layers of that group in a layered media session.

At the end of this section we point out that for small and large size topologies, the centralized scheme of Section II.A with a priority mechanism based on the number of end nodes and the decentralized scheme of Section II.B can be respectively utilized to specify maximum available bandwidth to individual layers of a media system. For medium size topologies, the best choice is represented by the quasi-centralized scheme of Section II.C. As an example, the flow control work of this research article can be utilized to relate the flow control aspect of our previous research work Layered Media Multicast Control (LMMC) to its rate allocation, partitioning, and error control aspects as discussed in [31] and [32].

V. Numerical Analysis

In this section, we provide two numerical examples to further illustrate centralized, decentralized, and quasi-centralized schemes of Section II. While our first example focuses on the comparison among the three schemes, our second example illustrates the applicability of the schemes to a layered multicast scenario. With the assumption that all of the bandwidth units of the current section are the same, we suppress the units. Additionally for the example of this section, we denote x_i as the rate of the i-th unicast session and x_{ij} as the rate of the j-th virtual session of the i-th multicast session. Accordingly, we assume an unrestricted unicast or multicast session i is requesting a bandwidth of X_i equal to the capacity of the bottleneck link over its path to a source. In the case of multicast session i, we assume a virtual session j is requesting a bandwidth of X_{ij} also equal to the capacity of the bottleneck link over its path to a source. Taking into consideration that the assigned rates of the virtual sessions belonging to the same multicast session are the same, the value of X_i for a multicast session i is related to the values X_{ij} of its virtual sessions as

$$X_i = \min_j X_{ij} \tag{19}$$

Example 5.1 In our first example, we use the sample network topology of Fig. 1 and note that the topology is a slightly modified version of the topology utilized in [14]. The sample topology consists of 6 unrestricted multicast and 5 unrestricted unicast sessions distributed over a total of 21 links. We note that the 6 multicast sessions consist of a total of 14 virtual sessions. Table I provides specifications of the sample network as well as a comparison between the results of our centralized, decentralized, and quasi-centralized schemes. We note that the topology is partitioned into two zones in the case of our quasi-centralized scheme. Designated nodes of the first and the second zones are identified as DN1 and DN2 in Fig. 1, respectively. The first zone with a depth of 2 links includes l_1, l_3, l_4, l_{51}, l_8, l_9, l_{10}, and l_{15}. The second zone with a depth of 3 links includes l_2, l_{52}, l_{53}, l_6, l_7, l_{11}, l_{12}, l_{13}, l_{14}, l_{16}, l_{17}, l_{18}, and l_{19}. The first four columns of Table I respectively show a virtual session, its underlying path, its requested bandwidth, and the resulting requested bandwidth of its flow according to $X_i = \min_j X_{ij}$. While the values of the third and fourth columns are the same in the case of unicast flows, they may differ in the case of multicast sessions. This is due to the fact that the third column value indicates the capacity of the bottleneck link over the path of a specific virtual session while the fourth column value is the capacity of the bottleneck link over all of the virtual sessions of the same multicast session. The middle three columns of Table I respectively show the link number, the link capacity, and the set of corresponding calculated session rates according to our decentralized scheme. Note that the values of the seventh column are sorted in order, corresponding to the value of X_i for their related flows. In Table I, we also compare calculated fair shares of individual flows resulting from utilizing the centralized scheme of Section II.A with those from the decentralized scheme of Section II.B and the quasi-centralized scheme of Section II.C. In order to reward a multicast session, the flows of the centralized and quasi-centralized schemes have been calculated by applying a priority mechanism in which the flow weights are set proportional to the the number of end nodes associated with the flow and inversely proportional to the number of links traversed by a flow in order to discourage heavy utilization of the network resources.

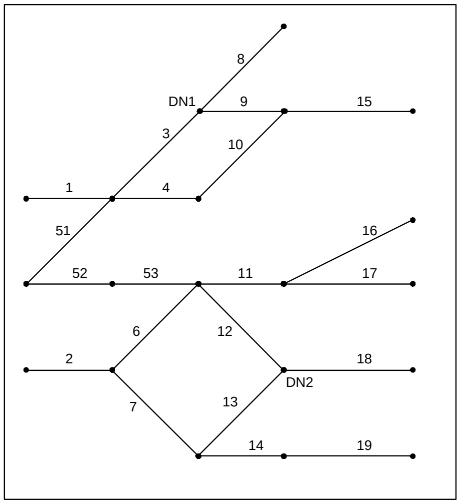

Fig. 1. An illustration of a sample network topology.

We justify the differences between the results of the three methods by considering the fact that the choice of the weighting functions in the centralized and quasi-centralized methods implement closely related flow priority mechanisms while it enforces max-min fairness in the decentralized method of Section II.B. We note that the reduced complexity of implementing our quasi-centralized scheme justifies its suboptimality compared to our centralized scheme. We also argue that from a practical stand point, the use of the decentralized or the quasi-centralized schemes are most probably the preferred choices when dealing with large size networks. □

Example 5.2 In order to show the applicability of our work to layered media scenarios, we utilize the sample network topology of Fig. 2 in our second example. We note that there are four categories of bandwidth in the sample topology of Fig. 2. In the figure, each category is represented by a different link thickness and/or shade. The sample topology consists of six unrestricted multicast and eight unrestricted unicast sessions distributed over a total of 39 links. The six multicast sessions consist of a total of 21 virtual sessions. We observe that besides a stand alone multicast session, the sample topology accommo-

TABLE I

THE PATH OF INDIVIDUAL SESSIONS, LINK CAPACITIES ALONG WITH DECENTRALIZED PER LINK FAIR SESSION RATES, AND A COMPARISON OF PER FLOW MINIMUM FAIR SHARES (MFS) FOR THE SAMPLE NETWORK TOPOLOGY OF FIG. 1.

Virtual Session	Path's Links	X_{ij}	X_i	Link	Capacity	Decentralized Per Link Session Rates	Session x_i	Centralized MFS	Decentralized MFS	Quasi-Central. MFS
x_{01}	$1, 3, 8$	4	4	l_1	14	$2, 2, 2, 2, 2, 2, 2$	x_0	1.25	2	1.25
x_{02}	$1, 3, 9, 15$	4	4	l_2	9	$1.5, 2.5, 2.5, 2.5$				
x_1	$1, 4, 10, 15$	6	6	l_3	6	$2, 2, 2$	x_1	4.75	2	4.75
x_{21}	$1, 51, 52, 53, 11, 16$	5	5	l_4	6	$3, 3$	x_2	1.25	1.25	1.25
x_{22}	$1, 51, 52, 53, 12, 18$	5	5	l_{51}	5	$1.25, 1.25, 1.25, 1.25$				
				l_{52}	6	$1.5, 1.5, 1.5, 1.5$				
				l_{53}	6	$1.5, 1.5, 1.5, 1.5$				
x_3	$1, 51, 52, 53, 11, 17$	4	4	l_6	6	$3, 3$	x_3	1.25	1.25	1.25
x_{41}	$2, 6, 12, 18$	6	5	l_7	6	$1.5, 1.5, 1.5, 1.5$	x_4	1.25	1.5	1.56
x_{42}	$2, 7, 14, 19$	5	5	l_8	4	$2, 2$				
x_5	$2, 7, 13, 18$	1.5	1.5	l_9	4	$2, 2$	x_5	1.25	1.5	1.33
x_6	$1, 3, 9, 15$	4	4	l_{10}	6	$3, 3$	x_6	1.25	2	1.25
x_{71}	$1, 3, 8$	4	4	l_{11}	8	$1.6, 1.6, 1.6, 1.6, 1.6$	x_7	1.25	1.25	1.25
x_{72}	$1, 51, 52, 53, 11, 16$	5	4	l_{12}	7	$2.33, 2.33, 2.33$				
x_{73}	$1, 51, 52, 53, 11, 17$	4	4	l_{13}	1.5	1.5				
x_{81}	$2, 6, 12, 18$	6	5	l_{14}	5	$1.67, 1.67, 1.67$	x_8	1.25	1.5	1.56
x_{82}	$2, 6, 11, 16$	5	5	l_{15}	10	$2.5, 2.5, 2.5, 2.5$				
x_{83}	$2, 7, 14, 19$	5	5	l_{16}	5	$1.67, 1.67, 1.67$				
x_9	$2, 7, 14, 19$	5	5	l_{17}	4	$1.33, 1.33, 1.33$	x_9	2.25	1.5	1.56
x_{A1}	$1, 4, 10, 15$	6	4	l_{18}	9	$1.5, 2.5, 2.5, 2.5$	x_A	1.25	1.25	1.25
x_{A2}	$1, 51, 52, 53, 11, 17$	4	4	l_{19}	5	$1.67, 1.67, 1.67$				

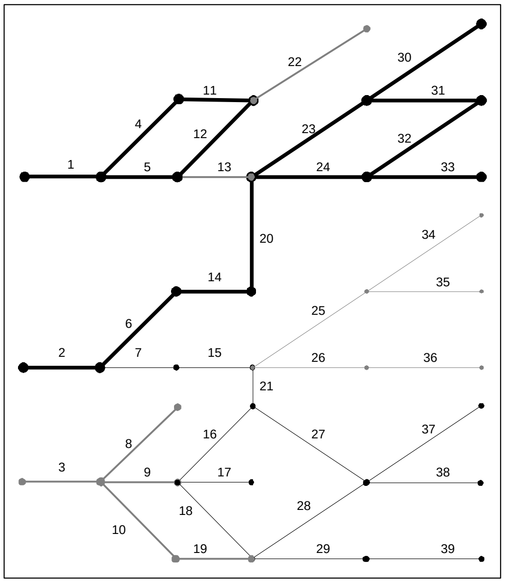

Fig. 2. An illustration of the second sample network topology.

dates two layered media sessions with three and two multicast groups respectively. While the first media session consists of the three multicast groups x_5, x_8, and x_9 as shown in the first column of Table II, the second session includes the two multicast groups x_C and x_D. Table II also provides specifications of the sample network as well as per link results of the decentralized scheme of Section II.B. The first four columns of Table II respectively show a virtual session, its underlying path, its requested bandwidth, and the resulting requested bandwidth of its flow according to (19). The last three columns of Table II respectively show the link number, the link capacity, and the set of corresponding calculated session rates according to our decentralized scheme. Note that the values of the last column are sorted in order, corresponding to the value of X_i for their related flows.

Next, we compare calculated fair shares of individual flows resulting from utilizing the centralized scheme of Section II.A with those from decentralized scheme of Section II.B. Table III includes minimum fair shares of each flow as the result of applying centralized scheme of Section II.A with equal flow weights, centralized scheme of Section II.A with weights set according to Equation (6), and decentralized scheme of Section II.B. Note that Condition (4) only holds in case of assigning equal flow priorities implying that the max-min fairness property of Definition 2.1 is not satisfied under the other centralized scenario.

We anticipate observing similar fair shares but different aggregate utility functions when comparing the results of the first and the second centralized schemes, respectively. The results of Table III are in agreement with our expectations. With the exception of the assignments of x_A and x_B, we observe a close assignment of fair shares when comparing the first and the second centralized schemes. Further, we observe that the assignment of bandwidth to the flows x_5 and x_C are relatively higher in the case of centralized algorithms compared to the decentralized algorithm. Considering the implicit priority mechanism

TABLE II

THE PATH OF INDIVIDUAL SESSIONS AND THE FAIR SHARE OF INDIVIDUAL FLOWS FOR THE SAMPLE NETWORK TOPOLOGY OF FIG. 2.

Virtual						Per Link
Session	Path	X_{ij}	X_i	Link	Capacity	Session Rates
x_0	$1, 5, 13, 24, 33$	38.39	38.39	l_1	99.42	$24.855, 24.855, 24.855, 24.855$
x_1	$1, 4, 11, 22$	31.34	31.34	l_2	99.80	$1.21, 1.21, 1.53, 1.58, 7.15, 84.28$
x_{21}	$1, 5, 12, 22$	31.34	31.34	l_3	39.33	$1.58, 7.15, 7.18, 7.92$
x_{22}	$1, 5, 13, 24, 32$	38.39	31.34	l_4	80.30	31.34
x_3	$1, 5, 13, 23, 31$	38.39	38.39	l_5	95.36	$31.34, 32.01, 32.01$
x_4	$2, 7, 15, 25, 35$	1.58	1.58	l_6	95.77	$1.21, 7.15, 84.28$
x_{51}	$2, 6, 14, 20, 23, 30$	84.28	84.28	l_7	8.14	$1.21, 1.21, 1.53, 1.58, 2.61$
x_{52}	$2, 6, 14, 20, 24, 33$	84.28	84.28	l_8	34.40	$1.58, 7.92$
x_6	$2, 7, 15, 26, 36$	1.21	1.21	l_9	36.83	$1.58, 7.15, 7.18, 7.92$
x_7	$2, 7, 15, 25, 34$	1.53	1.53	l_{10}	32.13	$1.58, 7.92$
x_{81}	$2, 6, 14, 20, 23, 30$	84.28	7.15	l_{11}	88.77	31.34
x_{82}	$2, 6, 14, 20, 24, 33$	84.28	7.15	l_{12}	80.97	31.34
x_{83}	$2, 7, 15, 21, 27, 37$	7.15	7.15	l_{13}	38.39	$12.7967, 12.7967, 12.7967$
x_{84}	$2, 7, 15, 21, 27, 38$	8.14	7.15	l_{14}	84.28	$1.21, 7.15, 75.92$
x_{91}	$2, 6, 14, 20, 23, 30$	84.28	1.21	l_{15}	9.35	$1.21, 1.21, 1.53, 1.58, 3.82$
x_{92}	$2, 6, 14, 20, 24, 33$	84.28	1.21	l_{16}	9.04	$1.58, 7.15$
x_{93}	$2, 7, 15, 21, 27, 37$	7.15	1.21	l_{17}	8.38	$1.58, 6.80$
x_{94}	$2, 7, 15, 21, 27, 38$	8.14	1.21	l_{18}	8.70	7.18
x_{95}	$2, 7, 15, 25, 34$	1.53	1.21	l_{19}	36.29	$1.58, 7.92$
x_{96}	$2, 7, 15, 26, 36$	1.21	1.21	l_{20}	92.87	$1.21, 7.15, 84.28$
x_A	$3, 9, 16, 27, 37$	7.15	7.15	l_{21}	9.62	$1.21, 1.58, 6.83$
x_B	$3, 9, 18, 28, 38$	7.18	7.18	l_{22}	31.34	$15.67, 15.67$
x_{C1}	$3, 8$	34.40	7.92	l_{23}	86.40	$1.21, 7.15, 38.39, 39.65$
x_{C2}	$3, 9, 17$	8.38	7.92	l_{24}	99.20	$1.21, 7.15, 30.28, 30.28, 30.28$
x_{C3}	$3, 10, 19, 29, 39$	7.92	7.92	l_{25}	1.58	$0.3950, 0.3950, 0.3950, 0.3950$
x_{D1}	$3, 8$	34.40	1.58	l_{26}	1.76	$0.88, 0.88$
x_{D2}	$3, 9, 17$	8.38	1.58	l_{27}	9.38	$1.21, 4.0850, 4.0850$
x_{D3}	$3, 10, 19, 29, 39$	7.92	1.58	l_{28}	7.18	7.18
x_{D4}	$3, 9, 16, 21, 25, 35$	1.58	1.58	l_{29}	8.81	$1.58, 7.23$
				l_{30}	94.53	$1.21, 7.15, 84.28$
				l_{31}	88.24	38.39
				l_{32}	94.89	31.34
				l_{33}	85.30	$1.21, 7.15, 38.39, 38.55$
				l_{34}	1.53	$0.765, 0.765$
				l_{35}	1.64	$0.82, 0.82$
				l_{36}	1.21	$0.605, 0.605$
				l_{37}	7.15	$1.21, 2.97, 2.97$
				l_{38}	8.25	$1.21, 3.52, 3.52$
				l_{39}	7.92	$1.58, 6.34$

of a layered media architecture and as proposed in Section IV, the bandwidths of lower priority higher bandwidth layers can be applied to higher priority lower bandwidth layers in order to accommodate the requirements of higher priority layers one layer at a time. □

VI. CONCLUSION

In this paper, we studied the solution to the general problem of flow control for hybrid unicast and multicast IP networks. We aimed at providing centralized, decentralized, and quasi-centralized optimal solutions to address flow control among competing unicast and multicast flows. Relying on the standard linear programming schemes, our solutions to centralized and quasi-centralized formulations of the flow control problem determined maximum allowable rates maximizing a multicast flow reward metric. Relying on the water-filling scheme, our solution to the decentralized formulation of the problem determined the rates satisfying the so-called max-min fairness metric. We pointed out that our low complexity decentralized and quasi-centralized schemes could be implemented with minimal ECN marking support from intermediate network nodes. Further, we noted that our proposed decentralized technique did

TABLE III

A COMPARISON OF PER FLOW MINIMUM FAIR SHARES (MFS) OF THE SAMPLE NETWORK TOPOLOGY OF FIG. 2 AS THE RESULT OF UTILIZING OUR CENTRALIZED AND DECENTRALIZED SCHEMES. THE WEIGHTING FUNCTIONS ARE SET EQUALLY AND BASED ON FLOW END NODES IN CENTRALIZED 1 AND 2 CASES, RESPECTIVELY. THE TABLE ALSO INCLUDES THE AGGREGATE UTILITY OF THE SAMPLE NETWORK FOR THE CENTRALIZED AND DECENTRALIZED SCHEMES.

x_i	Centralized 1 MFS	Centralized 2 MFS	Decentralized MFS
x_0	12.7967	12.7967	12.7967
x_1	12.7967	12.7967	15.6700
x_2	12.7967	12.7967	12.7967
x_3	12.7967	12.7967	12.7967
x_4	0.3950	0.3950	0.3950
x_5	71.7133	68.7308	30.2800
x_6	0.3950	0.3950	0.6050
x_7	0.3950	0.3950	0.3950
x_8	0.3950	3.3775	2.6100
x_9	0.3950	0.3950	0.3950
x_A	6.3600	3.3775	2.9700
x_B	7.1800	4.4775	3.5200
x_C	7.5250	7.5250	6.3400
x_D	0.3950	0.3950	0.3950

not require storing any state information in intermediate network nodes. At the expense of storing some state information, our quasi-centralized scheme provided a solution closer to a global flow control solution. Additionally, we explained how the flow control results of our current work, could be utilized in delivering multimedia content over multicast networks. Finally, we compared the performance of our centralized, decentralized, and quasi-centralized solutions and illustrated their applicability in two sample network topologies. As a general framework of distributing multimedia content over multicast networks, we are currently in process of integrating the results of our flow control work with the rate allocation, partitioning, and error control aspects of our Layered Media Multicast Control (LMMC) work as discussed in [31] and [32]. Relying on the implementation of our flow control scheme, we are further developing a preventive receiver-oriented congestion control scheme that can be applied to various multicasting applications including multimedia content delivery.

APPENDIX I
PROOF OF THEOREM 2.1

In this appendix, we prove Theorem 2.1. The proof proceeds by contradiction. We show that not satisfying relationship (4) can lead to violating Definition 2.1. Specifically, let us assume that

$$\frac{W_j}{X_j} = \max\{\frac{W_1}{X_1}, \cdots, \frac{W_f}{X_f}\} \tag{20}$$

with $X_1 \leq \cdots \leq X_f$. Consequently, in the absence of any prohibiting constraint, assigning all of the available bandwidth to flow j leads to the largest value for the the objective function of the problem formulation of (2). However, the constraint $x_j \leq X_j$ leads to a maximum allocation x_j of X_j. We assume that the allocation of flow j is X_j' with $X_{j-1} \leq X_j' \leq X_j$. If there is not enough bandwidth to assign an allocation of $X_1, \cdots, X_{j-1}$ respectively to flows $1, \cdots, j-1$, an allocation of X_j' to flow j means that a competing flow can be allocated a higher bandwidth as the result of reducing the allocation of other flows with equal or lower rates. It, hence, violates the definition of max-min fairness as provided by Definition 2.1. **QED**

APPENDIX II
PROOF OF THEOREM 2.2

In this appendix, we prove Theorem 2.2. First we note that for the special case of $\sum_{i=1}^{f} X_i < C$ the optimal solution is trivially $(X_1, X_2, \ldots, X_f)$. While such a solution satisfies optimization problem of (7), it introduces an underutilized link. Therefore, in the rest of the proof we assume that $\sum_{i=1}^{f} X_i \geq C$ resulting in a scenario in which at least one of the flows is not in its saturation region. In this case the optimal solution satisfies $\sum_{i=1}^{f} x_i^* = C$, otherwise we can find other solutions yielding larger aggregate utilities according to the following reasoning. Assuming $\sum_{i=1}^{f} x_i = C' < C$, there exists at least one flow j that is not in its saturation region. We note that adding $C - C'$ to the rate of such a flow increases the aggregate utility by a value of $\min(\frac{C-C'}{X_j}, \frac{X_j - x_j}{X_j})$. Therefore, the optimal solution $\underline{x}^*$ must satisfy the condition $\sum_{i=1}^{f} x_i^* = C$.

In what follows we will prove that (9) is the solution to the optimization problem posed in (7) considering $\sum_{i=1}^{f} x_i^* = C$. We denote the solution of Equation (9) by $\underline{x}^* = (x_1^*, \ldots, x_f^*)$ and another feasible solution satisfying the problem constraints by $\underline{x} = (x_1, \ldots, x_f)$. Defining $f(\underline{x}) = \sum_{i=1}^{f} U_i(x_i)$, we show that $f(\underline{x}) \leq f(\underline{x}^*)$ for any $\underline{x} \neq \underline{x}^*$ and hence $\underline{x}^*$ is the optimal solution. Instead of working with the vectors $\underline{x}$ and $\underline{x}^*$, we work with the difference vector $\underline{\Delta} = \underline{x} - \underline{x}^* = (\delta_1, \ldots, \delta_f)$ in which δ_i can be positive or negative corresponding to the deviations from $\underline{x}^*$. Consequently, we can define the following two ordered sets.

$$\{\delta_{i_1}, \ldots, \delta_{i_m}; \quad \delta_{i_p} \leq 0\}; \qquad i_1 < i_2 < \ldots < i_m \tag{21}$$

$$\{\delta_{j_1}, \ldots, \delta_{j_r}; \quad \delta_{j_p} > 0\}; \qquad j_1 < j_2 < \ldots < j_r \tag{22}$$

Combining the facts $\sum_{i=1}^{f} x_i \leq C$ and $\sum_{i=1}^{f} x_i^* = C$, with the expression

$$\sum_{i=1}^{f} x_i = \sum_{i=1}^{f} x_i^* - \sum_{p=1}^{m} |\delta_{i_p}| + \sum_{p=1}^{r} \delta_{j_p} \leq C \tag{23}$$

we conclude that $\sum_{p=1}^{m} |\delta_{i_p}| \geq \sum_{p=1}^{r} \delta_{j_p}$.

Now we will compare the total increase in $f(\underline{x})$ with the total decrease in $f(\underline{x})$ due to the $\underline{\Delta}$ and will show that the total decrease is greater than the total increase. Assuming i_m and j_p are greater than h where h is the index defined by (10), we argue that $i_m < j_p$. Otherwise considering the fact that the elements of $\underline{x}^*$ are identical for $i \geq h$, the second optimization constraint ($x_1 \leq x_2 \leq \ldots x_f$) is violated by considering an increase in $x_{j_p}^*$ by a value of δ_{j_p} where j_p is the last index in the set of positive δ values and a decrease in x_{i_m} by a value of δ_{i_m} where $i_m > j_p$. Now, we can write the maximum total increase in $f(\underline{x})$ due to $\underline{\Delta}$ as

$$\sum_{j_p > i_m} \frac{\delta_{j_p}}{X_{j_p}} + \sum_{j_p \le h} (U_{j_p}(x^*_{j_p} + \delta_{j_p}) - U_{j_p}(x^*_{j_p})) \tag{24}$$

The second term in the above statement is equal to zero because all of the $x^*_{j_p}$'s whose j_p is less than or equal to h, are already in their saturation region and an increase in $x^*_{j_p}$'s will not increase the value of $f(\underline{x})$. The total decrease in $f(\underline{x})$ due to $\underline{\Delta}$ is

$$\sum_{p=1}^{m} \frac{|\delta_{i_p}|}{X_{i_p}} \tag{25}$$

Now, we show that the total decrease (25) is greater than the total increase (24) to $f(\underline{x})$. This can be done by multiplying both Equation (24) and Equation (25) by X_{i_m} and taking into account the fact that $X_{i_1} \le X_{i_2} \le \ldots X_{i_m}$. The result of multiplying (24) by X_{i_m} is

$$\sum_{j_p > i_m} \frac{\delta_{j_p} X_{i_m}}{X_{j_p}} \le \sum_{j_p > i_m} \delta_{j_p} \tag{26}$$

where the inequality holds due to the fact that $X_{i_m} \le X_{j_p}$. The result of multiplying (25) by X_{i_m} is

$$\sum_{p=1}^{m} |\delta_{i_p}| \le \sum_{p=1}^{m} \frac{|\delta_{i_p}| X_{i_m}}{X_{i_p}} \tag{27}$$

Again the inequality holds because $X_{i_m} \ge X_{i_p}$ where $p \in \{1, \ldots, m\}$. Comparing the results of inequalities (26) and (27) and keeping in mind that $\sum_{p=1}^{m} |\delta_{i_p}| \ge \sum_{p=1}^{r} \delta_{j_p}$, we conclude that

$$\sum_{j_p > i_m} \frac{\delta_{j_p} X_{i_m}}{X_{j_p}} \le \sum_{j_p > i_m} \delta_{j_p} \le \sum_{p=1}^{m} |\delta_{i_p}| \le \sum_{p=1}^{m} \frac{|\delta_{i_p}| X_{i_m}}{X_{i_p}} \tag{28}$$

Thus,

$$\sum_{j_p > i_m} \frac{\delta_{j_p}}{X_{j_p}} \le \sum_{p=1}^{m} \frac{|\delta_{i_p}|}{X_{i_p}} \tag{29}$$

Inequality (28) yields that the maximum total increase in $f(\underline{x})$ is less than or equal the total decrease in $f(\underline{x})$ implying that the overall changes in the value of $f(\underline{x})$ is negative. Therefore we conclude $f(\underline{x}) \le f(\underline{x}^*)$. **QED**

References

[1] S. Athuraliya, D. Lapsley, and S. Low, "An Enhanced Random Early Marking Algorithm for Internet Flow Control," In. Proc. of IEEE INFOCOM, 2000.

[2] Y. Bartal, J.W. Byers, D. Raz, "Global Optimization Using Local Information with Applications to Flow Control," In. Proc. of IEEE Symposium on Foundations of Computer Science, 1997.

[3] L. Benmohamed, S.M. Meerkov, "Feedback Control of Congestion in Store-and-Forward Networks: the Case of a Single Congested Node," IEEE/ACM Trans. on Networking, December 1993.

[4] D. Bertsekas, R. Gallager, "Data Networks, Second Edition" Prentice-Hall, 1987.

[5] A. Charny, "An Algorithm for Rate Allocation in A Packet-Switching Network with Feedback," M.S. Thesis, EECS Dept. of MIT, 1994. Available at http://ana-www.lcs.mit.edu/anaweb/pdf-papers/TR_601.pdf

[6] S. Chong, R. Nagarajan, Y.T. Wang, "Designing Stable ABR Flow Control with Rate Feedback and Open Loop Control: First Order Control Case," Performance Evaluation, December 1998.

[7] C. Courcoubetis, V.A. Siris, G.D. Stamoulis, "Integration of Pricing and Flow Control for ABR services in ATM Networks," In Proc. of IEEE GLOBECOM, November 1996.

[8] D. DeLucia, K. Obraczka "Multicast Feedback Suppression Using Representatives," In Proc. of IEEE INFOCOM, April 1997.

[9] S. Floyd, V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Trans. on Networking, August 1993.

[10] R.J. Gibbens, F.P. Kelly, "Distributed Connection Acceptance Control for a Connectionless Network," In Proc. of ITC 16, 1999.

[11] R.J. Gibbens, F.P. Kelly, "Resource Pricing and the Evolution of Congestion Control", Automatica, 1999.

[12] J. Golestani and S. Bhattacharyya, "End-to-End Congestion Control for the Internet: A Global Optimization Framework," In Proc. of IEEE ICNP, 1998.

[13] G. Golub, C. VanLoan, "Matrix Computations, Second Edition," The Johns Hopkins University Press, 1989.

[14] E. Graves, R. Srikant, D. Towsley, "Decentralized Computation of Weighted Max-Min Fair Bandwidth Allocation in Networks with Multicast Flows," In. Proc. of Tyrrhenian IWDC, 2001.

[15] J. M. Jaffe, "Bottleneck Flow Control," IEEE Trans. on Communications, July 1981.

[16] V. Jacobson, "Congestion Avoidance and Control," In Proc. of ACM SIGCOMM, 1988.

[17] F.P. Kelly, "Charging and Rate Control for Elastic Traffic," European Trans. on Telecommunications, 1997.

[18] F.P. Kelly, A. Maulloo, D. Tan, "Rate Control in Communication Networks: Shadow Prices, Proportional Fairness and Stability," Journal of the Operational Research Society, 1998.

[19] S. Kunniyur, R. Srikant, "End-to-End Congestion Control: Utility Functions, Random Losses and ECN Marks," In Proc. of IEEE INFOCOM, 2000.

[20] D.E. Lapsley, M. Rumsewicz, "Improved Buffer Efficiency via the No Increase Flag in EFCI Flow Control," In Proc. of IEEE ATM Workshop, 1996.

[21] B. Levine, D. Lavo, J.J. Garcia-Luna-Aceves, "The Case for Reliable Concurrent Multicasting Using Shared Ack Trees," In Proc. of ACM Multimedia, November 1996.

[22] J.C. Lin, S. Paul, "RMTP: A Reliable Multicast Transport Protocol," In Proc. of IEEE INFOCOM, 1996.

[23] S.H. Low, D.E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence," IEEE/ACM Trans. on Networking, December 1999.

[24] R. Mazumdar, L. Mason, and C. Douligeris, "Fairness in Network Optimal Flow Control: Optimality of Product Forms," IEEE Trans. on Communications, May 1991.

[25] K.K. Ramakrishnan, S. Floyd, "A Proposal to Add Explicit Congestion Notification (ECN) to IP," RFC 2481, January 1999.

[26] G. Ramamurthy and A. Kolarov, "Application of Control Theory for the Design of Closed Loop Rate Control for ABR Service," In Proc. of ITC 15, 1997.

[27] W. Ren, K. Siu, H. Suzuki, "On the Performance of Congestion Control Algorithm for Multicast ABR Service in ATM," In Proc. of IEEE ATM Workshop, 1996.

[28] S. Sarkar, L. Tassiulas, "Distributed Algorithms for Computation of Fair Rates in Multirate Multicast Trees," In Proc. of IEEE INFOCOM, 2000.

[29] H. Tzeng, K. Siu, "On Max-Min Fairness Congestion Control for Multicast ABR Service in ATM," IEEE JSAC, April 1997.

[30] H.A. Wang, M. Schwartz, "Achieving Bounded Fairness for Multicast Traffic and TCP Traffic in the Internet," In Proc. of ACM SIGCOMM, 1998.

[31] H. Yousefi'zadeh, H. Jafarkhani, "Layered Media Multicast Control (LMMC): Rate Allocation and Partitioning," In Proc. of IEEE ICC, 2004. Available at http://www.ece.uci.edu/˜ hyousefi/pub.html/lmmcRAPICC.pdf.

[32] H. Yousefi'zadeh, H. Jafarkhani, A. Habibi, "Layered Media Multicast Control (LMMC): Error Control," In Proc. of IEEE ICC, 2003.