

# A Linear Optimization Approach for Achieving Flow Fairness in Unicast and Multicast Networks

Homayoun Yousefi'zadeh, Member IEEE    Fatemeh Fazel    Hamid Jafarkhani, Senior Member IEEE

**Abstract**—In this paper, we present an analytical solution to the general problem of flow control for both unicast and multicast IP networks. Relying on the so-called max-min flow fairness metric, we formulate a pair of centralized and decentralized convex optimization problems that can be analytically solved with quadratic and linear complexities respectively. Utilizing the solution to the decentralized optimization problem, we then propose a flow control algorithm requiring no per flow state information. The proposed algorithm can be implemented by merely making use of a simple Explicit Congestion Notification (ECN) marking scheme to convey minimum per link flow fairness information to the end nodes. Relying on the results of our current work, we also address the flow control issue of distributing real-time multimedia traffic across multicast IP networks.

**Index Terms**—Unicast IP Networks, Multicast IP Networks, Heterogeneity, Inter-Session Fairness, Flow Control, Max-Min Fairness, Optimality, ECN Marking, Layered Media Systems.

## 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 inter-session (flow) 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 support 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.

Reviewing the literature of multicasting applications reveals that some significant protocols have been proposed within the context of streaming media. Utilizing streaming media over multicast IP networks was first proposed by Deering et al. [9]. Following the work of Deering et al., replicated media streams approach first presented by Cheung et al. [5] within the context of DSG protocol and layered media streams approach first proposed by McCanne et al. [27] in the context of RLM protocol as well as Li et al. [22] in the context of rate control aspect of LVMR protocol are some of the most well-known protocols in this area. While none of these protocols in their original forms dealt with inter-session (flow) fairness, Rubenstein et al. [33] and some follow on research articles later showed that utilizing multiple transmission rates as proposed previously improves network fairness properties within the context of max-min fairness.

With the above introduction, we clarify that the main focus of this research work is to develop inter-session (flow) fairness algorithms relying on the concept of max-min fairness. Such algorithms have to offer low complexity solutions that can be obtained in real time considering the delay constraints of communication networking applications. Further, such algorithms have to address the co-existence issues of unicast and multicast flows.

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. [11]. Addressing flow and rate control problems were also considered in the literature of media multicasting, for example, by Li et al. in [23] and Wang et al. in [37]. In the recent years, proposition of ECN (Explicit Congestion Notification) marking techniques proposed by Ramakrishnan et al. in [28] and by Lapsley et al. in [20] has brought the promise of practical deployment of effective flow and congestion control mechanisms for the existing Internet infrastructure. In addition, applications of control and optimization theories as described by [3], [6], [7], [12], [13], [17], [30], [32] 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 [26] and [19] in which reaching a stable Nash equilibrium solution is desired.

In this study, we pay special attention to the results of Athuraliya et al. [1], [2], Graves et al. [15], Kelly et al. [18], Low et al. [25], Kunniyur et al. [19], Ramakrishnan et al. [28], and Sarkar et al. [34]. Our formulation of the flow control problem is best categorized under the optimization flow control techniques. It is hence aiming at maximizing a global and a per link set of utility functions defined over the complete path of unicast and multicast trees.

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 compare 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 well-behaved algorithms such as guaranteed existence, boundedness, stability, and scalability. Considering practicality, the resulting proposed algorithm can be implemented in real-time by merely taking advantage of a

The authors are with the Center for Pervasive Communications at the Electrical and Computer Engineering Department of University of California, Irvine; e-mail: [hyousefi, fazelsar, hamdi.j]@uci.edu.

simple binary ECN marking mechanism currently under review by IETF [28].

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 underlying network infrastructure. An outline of the paper follows. In Section II, we formulate and analytically solve a pair of global and per link optimal flow control problems relying on the so-called max-min fairness metric. Our solutions to these problems are capable of addressing inter-session fairness issue in order to specify a fair assignment of the available bandwidth among a set of competing unicast or multicast flows. In Section III, we describe a decentralized implementation of our per link flow control solution relying on a simple ECN marking mechanism. We note that the implementation of this section does not require storing any per flow state information in individual links. In this section, we also discuss the implications of applying our flow control problem to both unicast and multicast sessions. In Section IV, we utilize the results of our flow control work to satisfy the real-time requirements of layered and replicated media systems over multicast IP networks. 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 PROBLEM

In this section, we focus on a pair of optimal flow control problems and their solutions. We start from a centralized global formulation of the problem aiming at guaranteeing inter-session fairness among competing unicast and multicast flows utilizing the set of links over an existing network topology. We then move on to a decentralized local formulation of the problem relying on the concept of max-min fairness and compare the two methods regarding complexity and overall fairness issues.

In order to formulate both centralized and decentralized problems, we rely on the max-min fairness concept 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.  $\square$

Intuitively, max-min fair allocation implies that for a number of competing flows over an existing network topology, each flow should be able to receive a “fair share” of the available bandwidth. If a flow cannot fully utilize its fair share due to a limitation imposed by another link, then the residual bandwidth of that flow is split fairly among other flows.

### A. Centralized Flow Control Problem

Using the above notion of max-min fairness, we formulate a convex optimization problem by means of defining a per flow fairness utility with the objective of maximizing the sum of utilities over the set of links of a given network topology.

Assume  $f$  flows are sharing a set of links  $L$  over a particular network topology. Further, assume that the capacity of link  $j$

where  $j \in L$  is specified by  $C_j$ . Each flow  $i$  has a maximum required 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 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  $i$  due to the fact that flow  $i$  cannot utilize more than its maximum required bandwidth. In accordance with the latter assumption, we select the following concave utility function<sup>1</sup> to represent the fairness of individual flows.

$$U_i(x_i) = \min\left(\frac{x_i}{X_i}, 1\right) = \begin{cases} \frac{x_i}{X_i} & , x_i \leq X_i \\ 1 & , x_i > X_i \end{cases} \quad (1)$$

Fig. 1 illustrates sample drawings of such a utility function.

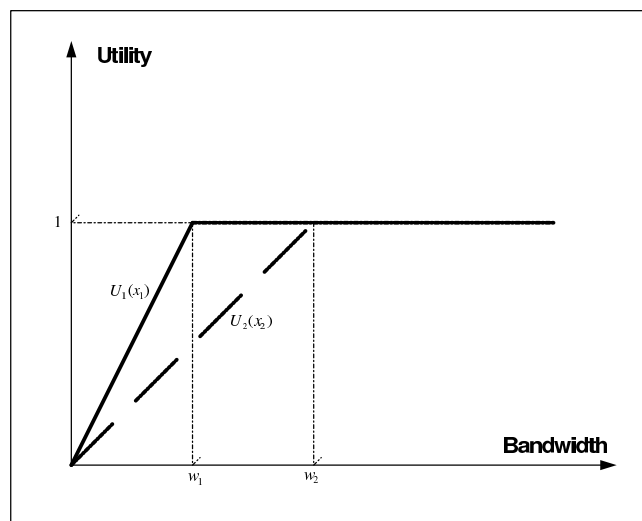


Fig. 1. Sample drawings of the utility function  $U_i(x_i)$  for flows  $i \in \{1, 2\}$  with maximum required bandwidth  $X_i$ .

Assuming an ordered set of bandwidth requirements  $X_1, X_2, \dots, X_f$  such that  $X_1 \leq X_2 \leq \dots \leq X_f$ , our formulation of the global flow control problem is now described in the form of the following optimization problem.

$$\begin{aligned} \max_{x_1, \dots, x_f} \quad & \sum_{j \in L} \sum_{i=1}^f w_{ji} U_i(x_i) \\ \text{Subject To:} \quad & \sum_{i=1}^f w_{ji} x_i \leq C_j \quad \forall j \in L \\ & x_1 \leq x_2 \leq \dots \leq x_f \end{aligned} \quad (2)$$

where  $f$  is the total number of flows over the network topology,  $C_j$  is the capacity of link  $j$ ,  $U_i(x_i)$  is the utility function defined in Equation (1), and  $w_{ji}$  is the weighting function defined below indicating whether link  $j$  is utilized by flow  $i$ .

$$w_{ji} = \begin{cases} 1 & , \text{ if flow } i \text{ utilizes link } j \\ 0 & , \text{ otherwise} \end{cases} \quad (3)$$

<sup>1</sup>The function  $f: \mathcal{C} \rightarrow \mathcal{R}^n$  defined over the convex set  $\mathcal{C} \subseteq \mathcal{R}^n$  is called concave if  $\forall x_1, x_2 \in \mathcal{C}$  and  $0 \leq \alpha \leq 1$  the inequality  $f(\alpha x_1 + (1-\alpha)x_2) \geq \alpha f(x_1) + (1-\alpha)f(x_2)$  holds.

We also note that a flow priority mechanism can be easily implemented by using values other than 1 in Equation (3). Reminding the fact that the summation operation preserves concavity, the problem of (2) is categorized under constraint convex optimization problems with piecewise linear objective functions. We can easily convert the problem of (2) to a standard linear programming (LP) problem as the result of replacing the second linear piece in the saturation area by a new constraint. The equivalent standard linear programming problem is expressed as

$$\begin{aligned} \max_{x_1, \dots, x_f} \quad & \sum_{i=1}^f W_i \left( \frac{x_i}{X_i} \right) \\ \text{Subject To:} \quad & \sum_{i=1}^f w_{ji} x_i \leq C_j \quad \forall j \in L \\ & x_i \leq X_i \quad \forall i \in \{1, \dots, f\} \\ & x_i \leq x_{i+1} \quad \forall i \in \{1, \dots, f-1\} \end{aligned} \quad (4)$$

where  $W_i \triangleq \sum_{j \in L} w_{ji}$  with  $i \in \{1, \dots, f\}$ . We now note the LP problem of (4) can be solved relying on one of the few existing methods such as the revised Simplex method, LU-decomposition method, or sparse Bartel-Golub method as described in [14] and [31]. Depending on the choice of algorithm and numerical applicability, the average complexity of solving the LP problem of (4) is in an order ranging from  $\mathcal{O}((l+2f)^2)$  to  $\mathcal{O}((l+2f)^3)$  where  $l$  is the number of links over the network topology  $L$ . It is in order to mention that the above formulation of the global flow control problem is in fact implementing a priority mechanism in which the flow weights are set proportional to the number of end nodes links traversed by the flow. Despite the fact there may be other considerations for implementing flow priorities such as the number of flow end nodes, the final formulation of the problem nevertheless comes down to (4) for a different choice of the weighting functions  $\{W_1, \dots, W_f\}$ . It is also important to note that the solution to the problem formulation of (4) tends to follow max-min fairness property of definition of 2.1 if

$$\frac{W_1}{X_1} > \dots > \frac{W_f}{X_f} \quad (5)$$

This is a design consideration that can be offset by relative importance of priority over max-min fairness.

### B. Decentralized Flow Control Problem

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 under or over 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. Consequently, the other flows of the first link could be assigned higher or lower fair shares by taking advantage of the unused portion of the bandwidth assigned to the first flow or having to give up a portion of their bandwidth.

We also note that although our simplified approach can be independently utilized for network links accommodating both unicast and multicast sessions, there are additional implementation considerations that need to be addressed in case of multicast networks. We will address the latter issues in the next section. Prior to proceeding with expressing simplified formulation of the flow control problem and its solution, we point out that our simplified approach calls for detecting a per flow bottleneck introducing minimum available fair share to a specific flow. In the next section, we propose a simple ECN marking technique to detect and convey bottleneck link information to the end nodes of a session. With the description provided above, we now focus on the formulation of the per link flow control problem and the corresponding solution.

Assume  $f$  flows are sharing a link with bandwidth capacity  $C$ . 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  $X_1, X_2, \dots, X_f$  such that  $X_1 \leq X_2 \leq \dots \leq X_f$ , our per link formulation of the flow control problem is now described in the form of the following optimization problem.

$$\begin{aligned} \max_{x_1, \dots, x_f} \quad & \sum_{i=1}^f U_i(x_i) \\ \text{Subject To:} \quad & \sum_{i=1}^f x_i \leq C \\ & x_1 \leq x_2 \leq \dots \leq x_f \end{aligned} \quad (6)$$

where  $f$  is the number of competing flows over a link,  $C$  is the capacity of the link, and  $U_i(x_i)$  is the utility function of flow  $i$  as defined in Equation (1). We observe that solving per link optimization problem of (6) does not require accessing any state information. Reminding the fact that the summation operation preserves concavity, the problem of Equation (6) is categorized under constraint convex optimization problems with piecewise linear objective functions. The problem can be solved utilizing a similar approach as the one utilized in the previous subsection and noting the fact that Condition (5) 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 Definition 2.1. We express the water-filling solution to the constraint optimization problem of (6) as follows.

$$\begin{aligned} \text{Case 1: If } C \geq \sum_{j=1}^f X_j \\ x_i = X_i \quad , \quad 1 \leq i \leq f \end{aligned} \quad (7)$$

$$\begin{aligned} \text{Case 2: If } C < \sum_{j=1}^f X_j \\ x_i = \begin{cases} X_i & , \quad 1 \leq i \leq h \\ \frac{C - \sum_{j=1}^h X_j}{f-h} & , \quad h+1 \leq i \leq f \end{cases} \end{aligned} \quad (8)$$

where  $x_i$  is the bandwidth assigned to the  $i$ -th flow and  $h$  satisfies the following condition

$$X_h \leq \frac{C - \sum_{i=0}^h X_i}{f-h} \leq X_{h+1} \quad (9)$$

$$0 \leq h \leq f-1$$

for  $X_0 \triangleq 0$ . Appendix I proves that the water-filling solution of Equation (8) is, in fact, the optimal solution to the constraint optimization problem of (6).

We observe that the water-filling approach of Equation (8) 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 hence satisfies Definition 2.1 of max-min fairness. To clarify the above discussion consider the following examples.

**Example 2.1** Assume three flows are sharing a link with capacity  $C = 3.5 Mbps$ . In addition, assume that the maximum bandwidth requested by each of the three flows is as follows.

$$X_1 = 0.67 Mbps, X_2 = 1 Mbps, X_3 = 2 Mbps$$

According to the algorithm stated above, first we have to find the value of  $h$ . From (9), we observe that the inequality holds for  $h = 2$ . Consequently, Equation (8) introduces the fair bandwidth assignment as

$$x_i = \begin{cases} 0.67 Mbps & , i = 1 \\ 1 Mbps & , i = 2 \\ 1.83 Mbps & , i = 3 \end{cases}$$

We note that the distribution of the bandwidth is max-min fair.  $\square$

**Example 2.2** Assume four flows are sharing a link with capacity  $C = 10 Mbps$ . In addition, assume that the maximum bandwidth requested by each of the four flows is as follows.

$$X_1 = 1 Mbps, X_2 = 2 Mbps, X_3 = 5 Mbps, X_4 = 7 Mbps$$

According to the algorithm stated above, first we have to find the value of  $h$ . From (9), it is not hard to observe that the inequality holds for  $h = 2$ . Consequently, Equation (8) introduces the fair bandwidth assignment as

$$x_i = \begin{cases} 1 Mbps & , i = 1 \\ 2 Mbps & , i = 2 \\ 3.5 Mbps & , i = 3 \\ 3.5 Mbps & , i = 4 \end{cases}$$

Again we note that the distribution of the bandwidth is max-min fair.  $\square$

It is also important to note that there is a simple geometric interpretation for the water-filling approach of Equation (8) specifying the number of flows in their saturation region. In order to explain the geometric interpretation, we consider a coordinate system with its x- and y-axis corresponding to the overall allocated bandwidth and the overall fairness. For such a system, we observe that the aggregate utility function of the optimization problem (6) appears in the form of a piecewise linear function. Further the slopes of consecutive linear pieces along with the associated x-coordinates of the break points are members of the sets

$$\left\{ \sum_{i=1}^f \frac{1}{X_i}, \sum_{i=2}^f \frac{1}{X_i}, \dots, \frac{1}{X_f}, 0 \right\} \quad (10)$$

and

$$\{X_1, \dots, X_f\} \quad (11)$$

respectively. The key observation with regards to the water-filling approach of Equation (8) is that the vertical line passing through the break point  $h$  represents an aggregate link bandwidth assignment of

$$\sum_{i=1}^h X_i + (f-h) X_h \quad (12)$$

to the competing flows. Geometrically speaking, the solution to the constraint optimization problem of (6) lies between two values associated with the bandwidth assignment of Equation (12) for the vertical lines passing through break points  $h$  and  $h+1$ . The vertical line associated with break point  $h$  specifies the number of flows in their saturation region. The following example shows geometric interpretation of Example 2.1.

**Example 2.3** Assume the three flows of Example 2.1 are sharing a link with capacity  $C$ . We discuss max-min fair assignment of the link bandwidth for different values of link capacity,  $C$ .

Fig. 2 shows the drawing of the aggregate utility function for this particular example. From the figure we observe that for the link bandwidths satisfying  $2 \leq C < 2.67 Mbps$  the first flow can receive its maximum requested bandwidth while the other two flows can only receive a portion of their maximum requested bandwidths; for the link bandwidths satisfying  $2.67 Mbps \leq C < 3.67 Mbps$  the first two flows can receive their maximum requested bandwidth while the third flow can only receive a portion of its maximum requested bandwidth; and finally for the link bandwidths satisfying  $C \geq 3.67 Mbps$  all of the three flows can receive their maximum requested bandwidths.  $\square$

Utilizing our water-filling approach, we now formalize our flow control algorithm at an intermediate node with capacity  $C$  accommodating  $f$  competing flows.

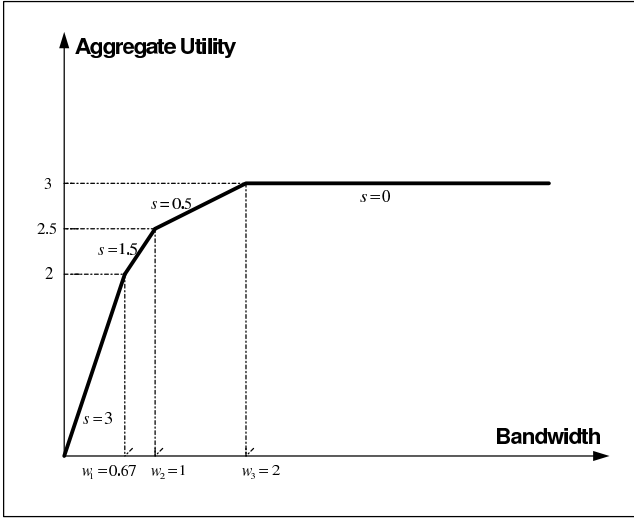


Fig. 2. Drawing of the aggregate utility function for the flows of example 2.1.

### Flow Control Optimization Algorithm

- Initialize  $S = 0$  and  $x_f = 0$ .
- for ( $p = 1$  to  $f - 1$ ) {
  - $S += X_p$
  - if  $\frac{C-S}{f-p} > X_{p+1}$ ,  
then  $x_p = X_p$
  - else {
    - for ( $q = (p + 1)$  to  $f$ ) {
      - $x_q = \frac{C-S}{f-p}$
      - } /\* for ( $q = (p + 1)$  to  $f$ ) \*/
      - break
      - } /\* else \*/
    - } /\* for ( $p = 1$  to  $f - 1$ ) \*/
  - if ( $x_f == 0$ ),  
then  $x_f = \min(X_f, C - S)$

As an important observation, we make note of the fact that the time complexity of “Flow Control Optimization Algorithm”  $\mathcal{O}(f)$  is linear in terms of the number of competing flows  $f$ . In the next section, we utilize a low complexity algorithm to detect the minimum available fair bandwidth for a set of flows relying on the results of the current section.

Before we discuss the implementation of our flow control protocol based on the decentralized solution of this subsection, we note that it is also possible to envision a hybrid flow control optimization problem that can be solved over local zones. The idea behind proposing such a scenario is to address the trade off between accuracy and the practicality of the solution. 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. Applying decentralized approach of this subsection to local zones can then identify the minimum fair share of each flow.

### III. UTILIZING ECN MARKS IN THE IMPLEMENTATION OF THE PER LINK FLOW CONTROL PROTOCOL

In this section, we describe the implementation of a flow control protocol based on the results of Section II.B. We start by de-

scribing utilization of a binary ECN marking technique for both unicast and multicast flows and continue by discussing particular implications of deploying the results of Section II.B in case of multicast flows.

#### A. Decentralized Flow Control Protocol

We start by describing the implementation of the general flow control protocol. Our proposal for implementing the flow control protocol calls for utilization of a binary ECN marking technique as described by [28]. Such a marking technique relies on receiving support from the intermediate nodes to propagate the marks on the paths starting from the source to the receivers of a session. The marking probability is calculated based on the aggregate loads of the intermediate nodes. We note that the marking proposal of [28] is currently under consideration by Internet Engineering Task Force (IETF) as a standardization candidate for next generation IP packet networks. Having solved the per link max-min fairness optimization problem of Section II.B, in this subsection we utilize the mark-based estimation technique of [15] to detect the bottleneck link of a unicast session. In the next subsection, we generalize our approach to detect the bottleneck link of a multicast session.

In what follows we briefly describe the approach of [15] and distinguish the differences between its proposed method and our method. The authors of [15] suggest utilizing a set of per link power rational utility functions forming a per link convex optimization problem. The set of utility functions are defined such that they converge to a max-min fair distribution of available link bandwidth among a set of competing flows. Showing that the choice of utility functions for the limiting case of an infinite power value leads to a max-min fair allocation of the link bandwidth among competing flows, the authors then rely on an approximation of the max-min fair allocation for some large number  $\mathcal{N}$ . Relying on duality theory and making use of the concept of scaled shadow prices of [18], the authors propose an iterative method that will asymptotically converge to the optimal fair share of a link in the limiting case of  $\mathcal{N} \rightarrow \infty$ . We recall that the shadow prices of [18] are defined as the inverse of the Lagrange multipliers in the optimization problem. We note that we are investigating the exact solutions to a related but different primal optimization problem obtained from the definition of max-min fairness instead of looking for approximated solutions to the dual optimization problem of [15]. Consequently rather than recursively calculating scaled shadow prices, we solve the optimization problem of (6) relying on a non-recursive approach. This leads to the introduction of exact per link max-min fair shares at any instant of time. Additionally because of introducing exact solutions relying on a non-recursive approach, we also eliminate the need for providing a stability discussion regarding the convergence of our approach.

Assuming that solving the set of optimization problems of Section II.B yields to the introduction of an ordered list of per link fair shares  $\{l_k^1, l_k^2, \dots, l_k^{n_k}\}$  with  $l_k^1 \leq l_k^2 \leq \dots \leq l_k^{n_k}$  for a unicast session traversing the set of links  $\{l_k^1, l_k^2, \dots, l_k^{n_k}\}$ , we can specify the minimum fair share  $l_k^1$  of the session as follows. Defining  $\lambda_k^i \triangleq \frac{1}{l_k^i}$  -not be mistaken as indicators of Lagrange multipliers- for  $k \in \{1, \dots, n_k\}$ , identifying minimum

fair share of a session traversing links  $\{l_k^1, l_k^2, \dots, l_k^{n_k}\}$  is equivalent to specifying  $\lambda_k^1$  in the ordered list of  $\{\lambda_k^1, \lambda_k^2, \dots, \lambda_k^{n_k}\}$  with  $\lambda_k^1 \geq \lambda_k^2 \geq \dots \geq \lambda_k^{n_k}$ . Next, we note that

$$\begin{aligned} \Phi_k &\triangleq [(\lambda_k^1)^\mathcal{N} + \dots + (\lambda_k^{n_k})^\mathcal{N}]^{\frac{1}{\mathcal{N}}} \\ &= (\lambda_k^1) [1 + (\frac{\lambda_k^2}{\lambda_k^1})^\mathcal{N} + \dots + (\frac{\lambda_k^{n_k}}{\lambda_k^1})^\mathcal{N}]^{\frac{1}{\mathcal{N}}} \end{aligned} \quad (13)$$

can be approximated by  $\lambda_k^1$  for some large number  $\mathcal{N}$  considering the fact that  $\frac{\lambda_k^i}{\lambda_k^1} \leq 1$  for  $i \in \{2, \dots, n_k\}$ . Consequently, identifying the bottleneck link of the session is equivalent to conveying  $\Phi_k$  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  $\Xi_k$  for large  $\mathcal{N}$  can be either very close to 0 or 1 depending on the value of  $\lambda_k^i$ . To overcome the above-mentioned problem, [15] proposes utilizing the following transformation

$$\varphi_k^i \triangleq \log_{b_k^i}(C_k^i \lambda_k^i) \quad (14)$$

and applying appropriate coefficients  $b_k^i, C_k^i$  to keep the value of  $\varphi_k^i$  close to 1. Adopting the proposal of [15] in the context of our work, we note that the value of  $\varphi_k^i$  is guaranteed to satisfy the following inequality

$$1 - \epsilon \leq \varphi_k^i \leq 1 + \epsilon \quad (15)$$

for every link  $i$  of the underlying session and some small number  $\epsilon$  by selecting

$$b_k^i = \left[ \frac{l_{k,max}^i}{l_{k,min}^i} \right]^{\frac{1}{2\epsilon}} \quad (16)$$

and

$$C_k^i = l_{k,max}^i \left[ \frac{l_{k,max}^i}{l_{k,min}^i} \right]^{\frac{1-\epsilon}{2\epsilon}} \quad (17)$$

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 (13) under the transformation of Equation (14) is changed to

$$\begin{aligned} \Phi_k &\triangleq [(\varphi_k^1)^\mathcal{N} + \dots + (\varphi_k^{n_k})^\mathcal{N}]^{\frac{1}{\mathcal{N}}} \\ &= (\varphi_k^1) [1 + (\frac{\varphi_k^2}{\varphi_k^1})^\mathcal{N} + \dots + (\frac{\varphi_k^{n_k}}{\varphi_k^1})^\mathcal{N}]^{\frac{1}{\mathcal{N}}} \end{aligned} \quad (18)$$

specifying the number of unmarked packets as

$$\Xi_k = \exp(-\Phi_k^\mathcal{N}) \quad (19)$$

Having resolved the above-mentioned numerical implementation issue, 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 ‘‘Flow Control Optimization Algorithm’’ of Section II.B.
- Determine the value of  $\lambda_k^i = 1/l_k^i$ .
- Compute the values of  $b_k^i$  and  $C_k^i$  from Equation (16) and Equation (17) respectively.
- Calculate the value of  $\varphi_k^i$  from Equation (14).
- Mark a packet with probability  $1 - \exp(-(\varphi_k^i)^\mathcal{N})$  for some large  $\mathcal{N}$ .

#### Flow Control Algorithm: End Node

- Calculate the rate of receiving unmarked packets from Equation (19) 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.

#### B. Multicasting Implications

In this subsection, we discuss different aspects of protocol implementation when coping with multicast networks. Without loss of generality and consistent with important multicasting applications such as media streaming and bulk data transfer applications, 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. Consequently depending on the requirements of the target applications, different actions may be taken by a group of end-nodes to report minimum fair shares of a multicast session. In this study, we consider two scenarios. While in the first scenario the objective is detect all of the bottlenecks of individual virtual sessions, in the second scenario the objective is to discover the overall minimum fair share of the session. Source centric media streaming and synchronized receiver centric bulk data transfer are among the application examples related to these two scenarios respectively. It is worth mentioning that synchronized protocol implementation of both of the above-mentioned scenarios are subject to feedback implosion problem as first pointed out in [8]. In what follows we adopt some of the multicasting literature techniques to address the problem in each case individually. In both cases, we assume that the source of a multicast session initiates the discovery process by sending pilot multicasting packets to the members of a multicast session.

We start by addressing the first case in which the source of a session or another centralized node may require to access bottleneck information of individual receivers of the session. Considering the fact that aside from feedback implosion accessing

per receiver information may not be a practical way of handling the problem in large scale environments, some feedback aggregation methods have been proposed in the literature of multicasting to cope with the issue. Examples of such techniques can be found in [21], [24], and [35]. We adopt an aggregation method within the lines of the above-mentioned references in which designated receivers in local zones are used for aggregating the feedback and sending the response on behalf of the receivers of the zone to the source of the session. We note that utilizing such a technique imposes an approximation error as the result of reporting the average or the lowest bottleneck to the source of the session. Next, we focus on the second case in which the objective is to discover the overall minimum fair share rather than all of the minimum fair shares of individual receivers of a multicast session. Reviewing the literature of multicasting reveals a rich set of asynchronous receiver-initiated protocols proposed to eliminate feedback implosion. The articles of [29] and [10] are of special interest to us among the set of articles on the topic. In addition, [36] includes a general analysis of such techniques. Considering the fact that any proposed method is expected to guarantee an implosion free way of discovering the overall minimum fair share, the following approach is proposed as a feasible alternative. Upon the receipt of polling packets, receivers of a multicast session set their own timers with a random value calculating and reporting their minimum fair share after having an expired timer only if not having 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  $k \in \{1, \dots, K\}$  consists of  $N_k$  receivers such that  $N = \sum_{k=1}^K N_k$ . For such a media session a set  $P = \{G_1 | \dots | G_K\}$  is called a partitioning of the receiver set  $\{1, \dots, N\}$  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  $g_1, g_2, \dots, g_K$  such that  $g_1 \leq g_2 \leq \dots \leq g_K$  the layer rates of a layered media session are calculated in the form of

$$g_1, g_2 - g_1, g_3 - g_2, \dots, g_K - g_{K-1} \quad (20)$$

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

$G_1, G_2, \dots, G_K$  with ordered group rates of  $g_1, g_2, \dots, g_K$  such that  $g_1 \leq g_2 \leq \dots \leq g_K$  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 priority mechanism implementation based on the number of end nodes associated with a flow. Considering Condition (5) and paying attention to the fact that for a layering architecture the relationships

$$W_1 \geq \dots \geq W_f$$

and

$$X_1 \leq \dots \leq X_f$$

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

$$W_1 = \dots = W_f = 1$$

and

$$X_1 \leq \dots \leq X_f$$

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 (20), 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 algorithm of Section II.A with a priority mechanism based on the number of end nodes and the decentralized algorithm of Section II.B can be respectively utilized to specify maximum available bandwidth to individual layers of a media system. 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 [38], and [39].

#### V. NUMERICAL ANALYSIS

In this section, we provide numerical examples to further illustrate centralized and decentralized algorithms of Section II. With the assumption that all of the bandwidth units of the current section are the same, we do not show the units in the examples of this section. Additionally for both of the examples 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 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 resulting 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

$$X_i = \min_j X_{ij} \quad (21)$$

We point out that the solution to the standard LP problem of (4) is represented in the matrix form

$$\begin{aligned} \max_{\underline{x}} \quad & \underline{c}^T \underline{x} \\ \text{Subject To :} \quad & A \underline{x} \leq \underline{b} \end{aligned} \quad (22)$$

where  $\underline{c} = [\frac{W_1}{X_1} \cdots \frac{W_f}{X_f}]^T$  and  $\underline{x} = [x_1 \cdots x_f]^T$  are  $f \times 1$  coefficients and the flow vectors respectively. Further the constraint matrix  $A$  and the constraint vector  $\underline{b}$  are accommodating the three constraint sets of (4). When comparing the results of the centralized algorithm of Section II.A with those of the decentralized algorithm of Section II.B, we pay attention that while in the former case solving the linear programming problem of (4) directly specifies the fair shares of the flows, finding the fair shares of the flows in the latter case is equivalent to finding the minimum fair share for that flow over the set of links on its path. In addition, the aggregate utility of the decentralized algorithm is obtained by summing up the aggregate utilities of individual links.

**Example 5.1** For our first example, we borrow the sample network topology of Fig. 3 from [15]. The sample topology

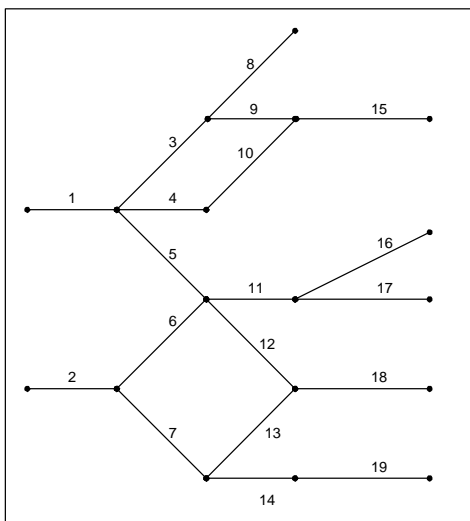


Fig. 3. An illustration of the first sample network topology.

consists of six unrestricted multicast and five unrestricted unicast sessions distributed over a total of 19 links. We note

that the six multicast sessions consist of a total of 14 virtual sessions. Table I provides specifications of the sample network as well as per link results of the decentralized algorithm of Section II.B. 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 (21). While the values of the third and fourth columns are the same in case of unicast flows, they may differ in 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 last 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 algorithm. 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 algorithm of Section II.A with those from the decentralized algorithm of Section II.B. Before proceeding with the review of the results, we recall that the solution to the standard LP problem of (4) in case of the sample network topology of Fig. 3 is represented in the matrix form of (22). Considering 19 link constraints, 11 flows along with 11 upper bound constraints, and 10 ordered set constraints,  $A$  and  $\underline{b}$  are of the size  $40 \times 11$  and  $40 \times 1$  respectively. Table II includes minimum fair shares of each flow and the aggregate utility of the sample network as the result of applying both centralized and decentralized algorithms of Section II. The results of the centralized algorithm have been obtained by applying a priority mechanism in which the flow weights are set proportional to the number of links traversed by the flow and the number of end nodes associated with the flow respectively.

Due to the impact of local and global max-min fairness and due to similar implementation of priority mechanism, we expect observing different fair shares but close aggregate utility functions when comparing the results of the first centralized and the decentralized algorithms. Additionally due to global consideration of max-min fairness and due to applying different weighting functions, we anticipate observing similar fair shares but different aggregate utility functions when comparing the results of the first and the second centralized algorithms. The results of Table II are in agreement with our expectations. While we observe similar aggregate utility functions when comparing the results of the first centralized and the decentralized algorithms, we see close assignment of fair shares when comparing the first and the second centralized algorithms. We argue that considering an aggregate utility offset of 0.7% along with significant lower complexity and the lack of a need to access state information, our decentralized algorithm is a better choice than the first centralized algorithm for this specific example. We also argue that from a practical stand point, the use of the decentralized algorithm is most probably the only choice despite the fact that the accuracy of the results vary from case to case.  $\square$



TABLE I

THE PATH OF INDIVIDUAL SESSIONS AND THE FAIR SHARE OF INDIVIDUAL FLOWS FOR THE SAMPLE NETWORK TOPOLOGY OF FIG. 3.

Virtual Session	Path	$X_{ij}$	$X_i$	Link	Capacity	Per Link Session Rates
$x_{01}$	1, 3, 8	4	4	$l_1$	14	2, 2, 2, 2, 2, 2
$x_{02}$	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
$x_{21}$	1, 5, 11, 16	5	5	$l_4$	6	3, 3
$x_{22}$	1, 5, 12, 18	5	5	$l_5$	5	1.25, 1.25, 1.25, 1.25
$x_3$	1, 5, 11, 17	4	4	$l_6$	6	3, 3
$x_{41}$	2, 6, 12, 18	6	5	$l_7$	6	1.5, 1.5, 1.5, 1.5
$x_{42}$	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
$x_6$	1, 3, 9, 15	4	4	$l_{10}$	6	3, 3
$x_{71}$	1, 3, 8	4	4	$l_{11}$	8	1.6, 1.6, 1.6, 1.6, 1.6
$x_{72}$	1, 5, 11, 16	5	4	$l_{12}$	7	2.3333, 2.3333, 2.3333
$x_{73}$	1, 5, 11, 17	4	4	$l_{13}$	1.5	1.5
$x_{81}$	2, 6, 12, 18	6	5	$l_{14}$	5	1.6667, 1.6667, 1.6667
$x_{82}$	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	1.6667, 1.6667, 1.6667
$x_9$	2, 7, 14, 19	5	5	$l_{17}$	4	1.3333, 1.3333, 1.3333
$x_{A1}$	1, 4, 10, 15	6	4	$l_{18}$	9	1.5, 2.5, 2.5, 2.5
$x_{A2}$	1, 5, 11, 17	4	4	$l_{19}$	5	1.6667, 1.6667, 1.6667

**Example 5.2** In order to show the applicability of our work to layered media scenarios, we utilize the sample network topology of Fig. 4 in our second example. We note that there are four categories of bandwidth in the sample topology of Fig. 4. In the figure, each category is represented by a different link thickness and/or shade. The sample topology consists of six

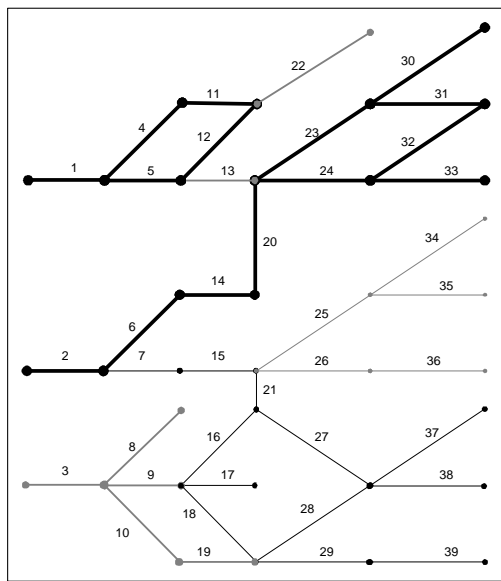


Fig. 4. An illustration of the second sample network topology.

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

accommodates 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 III, the second session includes the two multicast groups  $x_C$  and  $x_D$ . Table III also provides specifications of the sample network as well as per link results of the decentralized algorithm of Section II.B. The first four columns of Table III respectively show a virtual session, its underlying path, its requested bandwidth, and the resulting requested bandwidth of its flow according to (21). The last three columns of Table III respectively show the link number, the link capacity, and the set of corresponding calculated session rates according to our decentralized algorithm. 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 algorithm of Section II.A with those from decentralized algorithm of Section II.B. We recall that the solution to the standard LP problem of (4) in case of the sample network topology of Fig. 4 is represented in the matrix form of (22). Considering 39 link constraints, 14 flows along with 14 upper bound constraints, and 13 ordered set constraints,  $A$  and  $\underline{b}$  are of the size  $66 \times 14$  and  $66 \times 1$  respectively. Table IV includes minimum fair shares of each flow and the aggregate utility of the sample network as the result of applying both centralized and decentralized algorithms of Section II. The results of the centralized algorithm have been obtained by applying priority mechanisms in which the flow weights are set equally, proportional to the number of links traversed by the flow, and the number of end nodes associated with the flow respectively. Note that Condition (5) only holds in case of assigning equal flow priorities implying that the max-min fairness

TABLE III

THE PATH OF INDIVIDUAL SESSIONS AND THE FAIR SHARE OF INDIVIDUAL FLOWS FOR THE SAMPLE NETWORK TOPOLOGY OF FIG. 4.

Virtual Session	Path	$X_{ij}$	$X_i$	Link	Capacity	Per Link 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

property of Definition 2.1 is not satisfied under the other two centralized scenarios. In addition, recall that the implementation of our decentralized algorithm assigns the weighting function of a flow proportional to the number of links traversed by that flow. As the result, the implementation of our decentralized algorithm is closer to the second centralized algorithm than the other two centralized algorithms.

Similar to Example 5.1, we expect observing different fair shares but close aggregate utility functions when comparing the results of the second centralized and the decentralized algorithms respectively. Additionally, we anticipate observing

similar fair shares but different aggregate utility functions when comparing the results of the first, the second, and the third centralized algorithms respectively. The results of Table IV are in agreement with our expectations. While we observe similar aggregate utility functions when comparing the results of the first centralized and the decentralized algorithms, we see close assignment of fair shares when comparing the first, the second, and the third centralized algorithms. Again, considering an aggregate utility offset of 12.9% compare to the second centralized algorithm along with significant lower complexity and the lack of need to access state information, utilizing our

TABLE II

A COMPARISON OF PER FLOW MINIMUM FAIR SHARES (MFS) OF THE SAMPLE NETWORK TOPOLOGY OF FIG. 3 AS THE RESULT OF UTILIZING OUR CENTRALIZED AND DECENTRALIZED ALGORITHMS. THE WEIGHTING FUNCTIONS ARE SET BASED ON THE NUMBER OF FLOW LINKS AND FLOW END NODES IN CENTRALIZED 1 AND CENTRALIZED 2 CASES RESPECTIVELY. THE TABLE ALSO INCLUDES AGGREGATE UTILITY OF THE SAMPLE NETWORK FOR THE CENTRALIZED AND DECENTRALIZED ALGORITHMS.

$x_i$	Centralized 1 MFS	Centralized 2 MFS	Decentralized MFS
$x_0$	1.25	1.25	2
$x_1$	4.75	4.75	2
$x_2$	1.25	1.25	1.25
$x_3$	1.25	1.25	1.25
$x_4$	1.583	1.452	1.5
$x_5$	1.25	1.25	1.5
$x_6$	1.25	1.25	2
$x_7$	1.25	1.25	1.25
$x_8$	1.583	1.649	1.5
$x_9$	1.583	1.649	1.5
$x_A$	1.25	1.25	1.25
Aggregate Utility	22.79	11.22	22.95

decentralized algorithm is justified in case of this example. Further considering implicit priority mechanism 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.  $\square$

## 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 and decentralized optimal solutions to address inter-session fairness issue among competing unicast and multicast flows. Relying on the standard linear programming schemes and water-filling scheme respectively, our solutions to centralized and decentralized formulations of the flow control problem analytically determined maximum allowable rates maximizing a max-min fairness metric. We showed that our low complexity decentralized algorithm could be implemented with minimal ECN marking support from intermediate network nodes. Further, we noted that our proposed decentralized technique did not require storing any state information in intermediate network nodes. Additionally, we explained how the flow control results of our current work, could be utilized in real-time media systems. Taking into consideration the low complexity of our decentralized flow control technique, we argued that our technique could be effectively adopted in different size unicast and multicast networks. We also argued that our decentralized technique is capable of coping with the varying membership dynamics of multicast groups. Finally, we compared the performance of our centralized and decentralized solutions and illustrated their applicability in two sample network topologies. At the end, we

TABLE IV

A COMPARISON OF PER FLOW MINIMUM FAIR SHARES (MFS) OF THE SAMPLE NETWORK TOPOLOGY OF FIG. 4 AS THE RESULT OF UTILIZING OUR CENTRALIZED AND DECENTRALIZED ALGORITHMS. THE WEIGHTING FUNCTIONS ARE SET EQUALLY, BASED ON THE NUMBER OF FLOW LINKS, AND BASED ON FLOW END NODES IN CENTRALIZED 1, 2, AND 3 CASES RESPECTIVELY. THE TABLE ALSO INCLUDES AGGREGATE UTILITY OF THE SAMPLE NETWORK FOR THE CENTRALIZED AND DECENTRALIZED ALGORITHMS.

$x_i$	Centralized 1 MFS	Centralized 2 MFS	Centralized 3 MFS	Decentralized MFS
$x_0$	12.7967	12.7967	12.7967	12.7967
$x_1$	12.7967	12.7967	12.7967	15.6700
$x_2$	12.7967	12.7967	12.7967	12.7967
$x_3$	12.7967	12.7967	12.7967	12.7967
$x_4$	0.3950	0.3950	0.3950	0.3950
$x_5$	71.7133	68.7308	68.7308	30.2800
$x_6$	0.3950	0.3950	0.3950	0.6050
$x_7$	0.3950	0.3950	0.3950	0.3950
$x_8$	0.3950	3.3775	3.3775	2.6100
$x_9$	0.3950	0.3950	0.3950	0.3950
$x_A$	6.3600	3.3775	3.3775	2.9700
$x_B$	7.1800	4.4775	4.4775	3.5200
$x_C$	7.5250	7.5250	7.5250	6.3400
$x_D$	0.3950	0.3950	0.3950	0.3950
Agg. Utility	6.6401	47.0922	19.3802	41.0254

would like to discuss some of the aspects of our future work. We are currently working on the expansion of our flow control results into a general combined framework for congestion and flow control. Relying on the implementation of our flow control algorithm, we are developing a reactive receiver-oriented congestion control scheme that can be applied to real-time layered media systems as well as other multicasting applications.

## APPENDIX I

### PROOF OF OPTIMALITY FOR THE WATER-FILLING APPROACH OF SECTION II.B

In section II.B, we formulated a per link flow control problem as a constraint convex optimization problem. We claimed that the answer to the optimization problem of (6) is given by Equation (8). In this appendix, we prove our claim.

First we note that for the special case of  $\sum_{i=1}^f X_i < C$  the optimal solution is trivially  $(X_1, X_2, \dots, X_f)$ . While such a solution satisfies optimization problem of (6), it introduces an under-utilized 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 (8) is the solution to the optimization problem posed in (6) considering  $\sum_{i=1}^f x_i^* = C$ . We denote the solution of Equation (8) by  $\underline{x}^* = (x_1^*, \dots, x_f^*)$  and another feasible solution satisfying the problem constraints by  $\underline{x} = (x_1, \dots, x_f)$ . 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  $\underline{\Delta} = \underline{x} - \underline{x}^* = (\delta_1, \dots, \delta_f)$  in which  $\delta_i$  can be positive or negative corresponding to the deviations from  $\underline{x}^*$ . Consequently, we can define the following two ordered sets.

$$\{\delta_{i_1}, \dots, \delta_{i_m}; \delta_{i_p} \leq 0\}; \quad i_1 < i_2 < \dots < i_m \quad (23)$$

$$\{\delta_{j_1}, \dots, \delta_{j_r}; \delta_{j_p} > 0\}; \quad j_1 < j_2 < \dots < j_r \quad (24)$$

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

$$\sum_{i=1}^f x_i = \sum_{i=1}^f x_i^* - \sum_{p=1}^m |\delta_{i_p}| + \sum_{p=1}^r \delta_{j_p} \leq C \quad (25)$$

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 (9), 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 ( $x_1 \leq x_2 \leq \dots \leq x_f$ ) is violated by considering an increase in  $x_{j_p}^*$  by a value of  $\delta_{j_p}$  where  $j_p$  is the last index in the set of positive  $\delta$  values and a decrease in  $x_{i_m}$  by a value of  $\delta_{i_m}$  where  $i_m > j_p$ . Now, we can write the maximum total increase in  $f(\underline{x})$  due to  $\underline{\Delta}$  as

$$\sum_{j_p > i_m} \frac{\delta_{j_p}}{X_{j_p}} + \sum_{j_p \leq h} (U_{j_p}(x_{j_p}^* + \delta_{j_p}) - U_{j_p}(x_{j_p}^*)) \quad (26)$$

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

$$\sum_{p=1}^m \frac{|\delta_{i_p}|}{X_{i_p}} \quad (27)$$

Now, we show that the total decrease (27) is greater than the total increase (26) to  $f(\underline{x})$ . This can be done by multiplying both Equation (26) and Equation (27) by  $X_{i_m}$  and taking into account the fact that  $X_{i_1} \leq X_{i_2} \leq \dots \leq X_{i_m}$ . The result of multiplying (26) by  $X_{i_m}$  is

$$\sum_{j_p > i_m} \frac{\delta_{j_p} X_{i_m}}{X_{j_p}} \leq \sum_{j_p > i_m} \delta_{j_p} \quad (28)$$

where the inequality holds due to the fact that  $X_{i_m} \leq X_{j_p}$ . The result of multiplying (27) by  $X_{i_m}$  is

$$\sum_{p=1}^m |\delta_{i_p}| \leq \sum_{p=1}^m \frac{|\delta_{i_p}| X_{i_m}}{X_{i_p}} \quad (29)$$

Again the inequality holds because  $X_{i_m} \geq X_{i_p}$  where  $p \in \{1, \dots, m\}$ . Comparing the results of inequalities (28) and (29) and keeping in mind that  $\sum_{p=1}^m |\delta_{i_p}| \geq \sum_{p=1}^r \delta_{j_p}$ , we conclude that

$$\sum_{j_p > i_m} \frac{\delta_{j_p} X_{i_m}}{X_{j_p}} \leq \sum_{j_p > i_m} \delta_{j_p} \leq \sum_{p=1}^m |\delta_{i_p}| \leq \sum_{p=1}^m \frac{|\delta_{i_p}| X_{i_m}}{X_{i_p}} \quad (30)$$

Thus,

$$\sum_{j_p > i_m} \frac{\delta_{j_p}}{X_{j_p}} \leq \sum_{p=1}^m \frac{|\delta_{i_p}|}{X_{i_p}} \quad (31)$$

Inequality (30) 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  $f(\underline{x}) \leq f(\underline{x}^*)$ . **QED**

## REFERENCES

- [1] S. Athuraliya, D. Lapsley, and S. Low, "An Enhanced Random Early Marking Algorithm for Internet Flow Control," IEEE INFOCOM, 2000.
- [2] S. Athuraliya, S.H. Low, "Optimization Flow Control, II: Implementation," Preprint, <http://netlab.caltech.edu>, May 2000.
- [3] L. Benmohamed, S.M. Meerkov, "Feedback Control of Congestion in StoreandForward Networks: the Case of a Single Congested Node," IEEE/ACM Trans. on Networking, 1(6):693-707, December 1993.
- [4] D. Bertsekas, R. Gallager, "Data Networks," Englewood Cliffs, NJ:Prentice-Hall, 1987.
- [5] S. Cheung, M.H. Ammar, X. Li, "On the Use of Destination Set Grouping to Improve Fairness in Multicast Video Distribution," In Proc. of IEEE INFOCOM, March 1996.
- [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] P.B. Danzig, "Optimally Selecting the Parameters of Adaptive Backoff Algorithms for Computer Networks and Multiprocessors," Ph.D. thesis, UC Berkeley, December 1989.
- [9] S.E. Deering, D.R. Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs," ACM Trans. on Computer Systems, 8:85-110, May 1990.
- [10] D. DeLucia, K. Obraczka "Multicast Feedback Suppression Using Representatives," In Proc. of IEEE INFOCOM 97, Apr. 1997.
- [11] S. Floyd, V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Trans. on Networking, August 1993.
- [12] R.J. Gibbens, F.P. Kelly, "Distributed Connection Acceptance Control for a Connectionless Network," In Proc. of the 16th Int'l Teletraffic Congress, Edinburgh, Scotland, June 1999.
- [13] J. Golestani and S. Bhattacharyya, "EndtoEnd Congestion Control for the Internet: A Global Optimization Framework," In Proc. of IEEE ICNP, October 1998.
- [14] G. Golub, C. VanLoan, "Matrix Computations," 2nd Edition, The Johns Hopkins University Press, Baltimore, MD, 1989.
- [15] E. Graves, R. Srikant, D. Towsley, "Decentralized Computation of Weighted Max-Min Fair Bandwidth Allocation in Networks with Multicast Flows," In Proc. IWDC, September 2001.
- [16] V. Jacobson, "Congestion Avoidance and Control," In Proc. of ACM SIGCOMM, August 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, August 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] X. Li, S. Paul, M.H. Ammar, "Layered Video Multicast with Retransmissions (LVMR): Evaluation of Hierarchical Rate Control," In Proc. of IEEE INFOCOM, 1998.
- [23] X. Li, S. Paul, M. Ammar, "Multi-Session Rate Control for Layered Video Multicast," In Proc. of SPIE Multimedia Computing and Networking, January 1999.
- [24] J.C. Lin, S. Paul, "RMTP: A Reliable Multicast Transport Protocol," In Proc. of IEEE INFOCOM, 1996.
- [25] S.H. Low, D.E. Lapsley, "Optimization Flow Control, I: Basic Algorithm and Convergence," IEEE/ACM Trans. on Networking, 1999.
- [26] R. Mazumdar, L. Mason, and C. Douligeris, "Fairness in Network Optimal Flow Control: Optimality of Product Forms," IEEE Trans. on Communications, May 1991.
- [27] S. McCanne, V. Jacobson, M. Vetterli, "Receiver Driven Layered Multicast," In Proc. of ACM SIGCOMM, September 1996.
- [28] K.K. Ramakrishnan, S. Floyd, "A Proposal to Add Explicit Congestion Notification (ECN) to IP," RFC 2481, January 1999.
- [29] S. Ramakrishnan, B.N. Jain, "A Negative Acknowledgement with Periodic Polling Protocol for Multicast over LANs," In Proc. of IEEE INFOCOM, August 1987.
- [30] 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.
- [31] J. Reid, "A Sparsity-Exploiting Variant of the Bartels-Golub Decomposition for Linear Programming Bases," Mathematical Programming 24, 55-69.
- [32] J. Ros, W.K. Tsai, "A Theory of Maxmin Bottleneck Ordering," Technical Report, UCI, Electrical and Computer Engineering Dept., June 2000.
- [33] D. Rubenstein, J. Kurose, D. Towsley, "The Impact of Multicast Layering on Network Fairness," SIGCOMM '99, Sept 1999.
- [34] S. Sarkar, L. Tassiulas, "Distributed Algorithms for Computation of Fair Rates in Multirate Multicast Trees," In Proc. of IEEE INFOCOM, 2000.
- [35] P. Sharma, D. Estrin, S. Floyd, L. Zhang, "Scalable Session Messages in SRM Using Self-Configuration," USC Technical Report, July 1998.
- [36] D. Towsley, J. Kurose, S. Pingali, "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols," IEEE JSAC, April 1997.
- [37] H.A. Wang, M. Schwartz, "Achieving Bounded Fairness for Multicast Traffic and TCP Traffic in the Internet," In Proc. of ACM SIGCOMM, September 1998.
- [38] H. Yousefi'zadeh, H. Jafarkhani, A. Habibi "Layered Media Multicast Control (LMMC): Rate Allocation and Partitioning," Submitted for Publication. Available at <http://www.ece.uci.edu/~hyousefi/pub.html>.
- [39] H. Yousefi'zadeh, H. Jafarkhani, A. Habibi "Layered Media Multicast Control (LMMC): Error Control," Submitted for Publication. Available at <http://www.ece.uci.edu/~hyousefi/pub.html>.