A Linear Integer Programming Approach to Analyze P2P Media Streaming

Fulu Li * **Jianwei Huang** + **Andrew Lippman** *

MIT, Cambridge, MA USA *
Email: fulu@mit.edu

Chinese Univ. of Hong Kong, Hong Kong +
jwhuang@ie.cuhk.edu.hk

Abstract

Recent advancement in peer-to-peer (P2P) technologies has enabled a wide range of new applications. In this paper, we present mixed integer programming (MIP) formulations to analyze multiple interior-disjoint-tree-based and mesh-based P2P media streaming for application level multicast (ALM). Network optimization is the key to simultaneously satisfy a large group (potentially millions) of peers' needs given limited network resources. The key to our analytical approach is to cast the P2P media streaming problem as a constraint system. We intend to answer the following question: given a source node, a group of intended destination peer nodes with heterogeneous network resources, and an objective function to optimize, what is the best way to distribute information among these peer nodes? To the best of our knowledge, it is the first time that mixed integer programming (MIP) formulations in the framework of multiple interior-disjoint trees and mesh-based P2P streaming are presented to provide analytical insight and better understanding of the P2P streaming problem. Methods to obtain the optimal solutions for the presented formulations are also discussed.

1. Introduction

With the growing popularity of the Internet, the expansion of wireless local and personal area connectivity, the ubiquity of mobile devices, there is a tremendous social phenomena of online sharing, either sharing video, audio, pictures, webcam, or even bandwidth (like BitTorrent [8]) for the benefits of a large group (potentially millions) of peers (viewers). The underlying P2P (peer-to-peer) technologies are the key to accelerate this growing trend of the new Internet era in the 21st century.

In P2P streaming, the media streaming is accomplished in a peer-to-peer (P2P) fashion in the sense that each receiver node does not have to get the media content from the original media source, which may be far away geographically. Instead, each receiver node just needs to find if one of its peer neighbors is currently tuning in to receive the same media stream. Recursively, its neighbor will find its neighbor's neighbor all the way up to the original media source node.

Notably, P2P protocols/applications have received a great deal of attention recently in both academia and industry [3,8,9,11,12-13,15,17]. BitTorrent-like protocols [8] do not address the real-time needs of streaming as it is mainly designed to redistribute the file downloading and uploading cost among a group of peer nodes to achieve more robustness and scalability than traditionally centralized client-server model. P2P media streaming can be roughly classified into two categories: *(1)* P2P *on-demand* media streaming with *known* traffic profiles *a priori*; *(2)* P2P *live* media streaming with *uncertainty* of the *live* media stream traffic. The work in [9,15] assumes stored videos where the video traffic is known *a priori*. In live P2P media streaming, the dynamic traffic of the live media stream cannot be predicted accurately and one has to resort to *stochastic* models for robust network optimization [4].

The work in [9,15] on P2P video streaming do not consider application level multicast. In this paper, we focus on application level multicast (ALM) [1,2,6,7,16,17] for P2P media streaming from one source to multiple intended peer destination nodes. Each peer node is equipped with network connectivity capability, which allows them to receive the data stream over the network and feeds the signal into the display devices.

In this paper, we present mixed integer programming formulations in the framework of multiple interior-disjoint trees and mesh-based P2P streaming for application level multicast. The key to our analytical approach is to cast the P2P media streaming problem as a constraint system. We intend to answer the following question: given a source node, a group of intended destination peer nodes with heterogeneous network resources, and an objective function to optimize, what is the best way to distribute information among these peer nodes? Without an optimal solution, it is hard to judge the efficiency of existing P2P streaming protocols.

This paper is organized as follows: we discuss the related work in Section 2; we present integer programming formulations for P2P media streaming in n Section 3; we discuss methods to solve the presented integer programs in Section 4; the conclusions are presented in Section 5.

2. The Related Work

In the following, we give a general overview on application level multicast (ALM), SplitStream [6] and mesh-based P2P streaming, which are the basis for our integer programming formulations in Section 3.

2.1 Application Level Multicast

While application level multicast (ALM) is a well-studied research area [1,2,6,7,16], it has to have some new ingredients in the realization of P2P streaming to satisfy the real-time needs in a distributed P2P fashion.

First, the ALM tree has to be constructed in such a way that media quality fluctuation is *minimized* among all peer viewers given limited network resources at each peer node. This is a particular challenge in the face of the uncertainty of the live media stream traffic.

Secondly, as discussed in [1], the ALM protocol has to operate well in *adversarial* scenarios. For example, the protocol must be able to deal with frequent node failures, rapid node joining and leaving (also called *churn*), denial of service (DoS) attacks, uncooperative peers, etc.

Thirdly, it is essential that the ALM system is contribution-aware as the contributions from the peer nodes are most likely to be *heterogeneous* due to the heterogeneous environments at each peer node in the real world. Contribution-awareness is also needed to establish meaningful charging models and incentive mechanisms in P2P media streaming.

2.2 SplitStream

The SplitStream scheme in [6] provides a multi-tree data delivery framework, which can leverage the distribution of the streaming data with *interior-node-disjoint* trees to better use the outgoing bandwidth among participating peer nodes. If $T1$ and $T2$ are two spanning trees which have no interior nodes in common then $T1$ and $T2$ are said to be *interior-node-disjoint*. Notably, each node in a set of trees is interior node in at most one tree and leaf node in the other trees [6].

Notably, high bandwidth rate for watching media stream may not be available consistently to some P2P clients from a set of peers. Using this SplitStream framework [6], peer nodes self-organize themselves into a forest of S trees, all rooted at media source node. The media source node encodes media content with source rate R evenly into S stripes of size R/S [16], each of which is distributed along a different tree. The low-rate stripes are re-combined upon playing-out at each destination node to obtain a high fidelity copy of the media content. The more stripes one receives, the better media quality it plays out. A layered codec based on multiple description coding (MDC) is typically used to realize this goal [5,16]. For example, Fine-Grained Scalable Coding (FGSC) is widely available in current video codec and is now part of the MPEG4 standard. FGSC is being increasingly used to encode videos in P2P networks [15].

2.3 Mesh-based P2P Streaming

As an alternative to the tree-based approaches, recent work in [12-13,17] present a mesh-based P2P streaming framework to better utilize outgoing bandwidth among participating peers. As the name suggests, participating peers initially form a directed mesh in mesh-based P2P streaming. Each peer node can have multiple parents and multiple child peers. Each peer node maintains a sufficient number of parents that can collectively fill its incoming link bandwidth, where each parent node provides a specific sub-stream of the content. In [12-13], the content delivery combines two phases: the diffusion phase along a diffusion tree (the black arrows), e.g., the push-based streaming from parents at lower level to child nodes at higher level (see Fig. 1), and the swarming phase (the red arrows), e.g., the pull-based streaming from parents at higher level to child nodes at lower level in a mesh-based overlay. As we can see from Figure 1 that the swarming phase maximizes the utilization of the outgoing bandwidth of the leaf nodes in the diffusion tree.

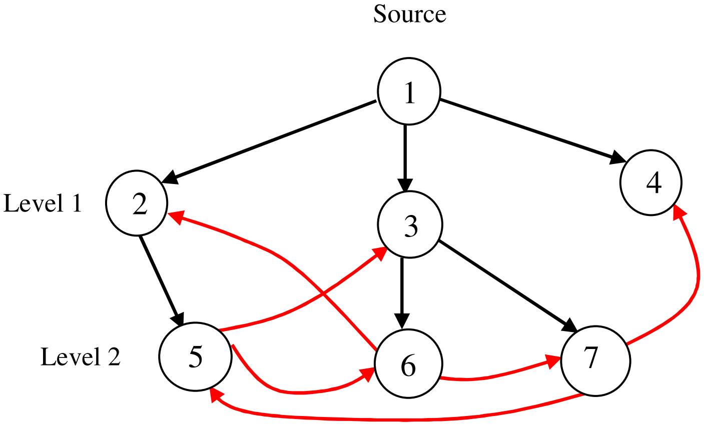

Figure 1: An Illustration of mesh-based P2P streaming.

3. Integer Programming Formulations

In this section, we present mixed integer programming formulations to analyze P2P media streaming for application level multicast (ALM).

Let us define t_k as the k^{th} epoch in time ($0 \le k \le N$). Let $s_{i,k}$ be the number of stripes used at peer i in the playout of the k^{th} epoch t_k ($0 \le i \le n-1, 0 \le k \le N$). Let b_i^I be the incoming bandwidth limit at node i ($0 \le i \le n-1$) and

b_i^O be the outgoing bandwidth limit at node i ($0 \le i \le n-1$). Let $f_{i,j,k}$ denote the number of stripes transmitted from node i to node j (no other peer nodes inbetween) during the k^{th} epoch t_k. Without loss of generality, we assume that the media source node is *Node* 0. We further define $x_{i,j,k}$ as the decision variable to indicate if there is a transmission from node i to node j (no other peer nodes inbetween) during the k^{th} epoch t_k. The variable $x_{i,j,k}$ is equal to one if there is a transmission from node i to node j (no other peer nodes inbetween) during the k^{th} epoch t_k, and zero otherwise. Let S denote the number of stripes when encoding the media source and R_k be the media source rate during the k^{th} epoch t_k. Let us define a decision variable x_e, which is equal to one if edge e is included in the tree, and zero otherwise. Let $abs(f(x))$ denote the absolute value of a function $f(x)$.

3.1 Multiple Interior-Disjoint Trees

In this section, we present a network optimization formulation for the construction of multiple interior-disjoint multicast trees. The SplitStream scheme in [6] uses multiple *interior-node-disjoint* trees to leverage the distribution of P2P streaming to better use the outgoing bandwidth among participating peers.

NOTATIONS	
t_k	the k^{th} epoch in time
$s_{i,k}$	the number of stripes used at peer i in the playout of the k^{th} epoch t_k
b_i^I	the incoming bandwidth limit at node i
b_i^O	the outgoing bandwidth limit at node i
$f_{i,j,k}$	the number of stripes transmitted from node i to node j (no other peer nodes inbetween) during the k^{th} epoch t_k
$x_{i,j,k}$	the decision variable to indicate if there is a *direct* transmission from node i to node j during the k^{th} epoch t_k
S	the number of stripes when encoding the media source
x_e	the decision variable to indicate if edge e is included in the tree
R_k	the media source rate during the k^{th} epoch t_k

Table 1: Some notations for the MIP formulation of multiple interior-node-disjoint trees.

In Table 1, we summarize some of the notations used in the MIP (mixed integer programming) formulation of the multiple interior-node-disjoint trees.

In the scenario of multiple interior-disjoint trees, $f_{i,j,k}$ takes a value of either one or zero as each stripe of the media source is distributed along a different interior-disjoint tree. We need to define two additional variables to distinguish constraints among different trees. We assume that variable $x_{i,j,k,s}$ is equal to one if there is a transmission from node i to node j (no other peer nodes inbetween) during the k^{th} epoch t_k along the s^{th} tree, and zero otherwise. Let us define a decision variable $x_{e,s}$, which is equal to one if edge e is included in the s^{th} ($1 \le s \le S$) tree, and zero otherwise.

Given a directed network graph $G = (V, E)$, where V is the vertex set and E is the edge set, the network optimization problem can be formulated as a mixed integer programming (MIP) problem:

$$\text{maximize} \quad \sum_{i=1}^{n-1}\sum_{j=1}^{N} s_{i,j} \tag{1}$$

subject to:

$$\frac{R_k}{S} \times \sum_j f_{i,j,k} \le b_i^O \text{, for } 0 \le i \le n-1,\ 0 \le k \le N \tag{2}$$

$$\frac{R_k}{S} \times \sum_i f_{i,j,k} \le b_j^I \text{, for } 0 \le j \le n-1,\ 0 \le k \le N \tag{3}$$

$$s_{i,k} = \sum_j f_{j,i,k} \le S \text{, for } 0 \le i \le n-1,\ 0 \le k \le N \tag{4}$$

$$f_{i,j,k} \le 1 \text{, for } 0 \le i, j \le n-1,\ 0 \le k \le N \tag{5}$$

$$\sum_{i=0}^{n-1} x_{i,0,k,s} = 0 \text{, for } 0 \le k \le N, 1 \le s \le S \tag{6}$$

$$x_{i,i,k,s} = 0 \text{, for } 0 \le i \le n-1,\ 0 \le k \le N,\ 1 \le s \le S \tag{7}$$

$$\sum_{j=1}^{n-1} x_{0,j,k,s} \ge 1 \text{, for } 0 \le k \le N,\ 1 \le s \le S \tag{8}$$

$$f_{i,j,k} \le f_{m,i,k} \text{ if } x_{m,i,k,s} = 1 \text{, for } 1 \le j \ne i \le n-1, \quad 0 \le k \le N, 1 \le s \le S \tag{9}$$

$$\sum_{i=0}^{n-1}\sum_{j=1}^{n-1} x_{i,j,k,s} = n-1 \text{, for } 0 \le k \le N, 1 \le s \le S \tag{10}$$

$$\sum_{s=1}^{S}(\min(\sum_{j=1}^{n-1} x_{i,j,k,s}, \quad 1)) = 1 \text{, for } 0 \le i \le n-1, 0 \le k \le N \tag{11}$$

Now, let us define A_s as a non-empty set ($A_s \subset V$) and $E(A_s) = \{(i,j) \in E \mid (i,j) \in A_s\}$. We have the following constraint to eliminate all cycles:

$$\sum_{e\in E(A_s)} x_e \le |A_s| - 1, \text{ for } A_s \subset V,\ A_s \ne \Phi, V \qquad (12)$$

$$\sum_{i=1}^{n-1}\sum_{j=1}^{N} abs(s_{i,j-1} - s_{i,j}) \le \eta \qquad (13)$$

Our objective is to maximize the overall received stream data among all peers with constrained media playout quality fluctuation (Constraint 13) given limited network resources with multiple interior-disjoint trees. Constraint (2) indicates that the outgoing bandwidth has to be limited by the outgoing link capacity at each peer node. Likewise, Constraint (3) requires that the incoming bandwidth have to be limited by the incoming link capacity at each peer node. Constraint (4) and Constraint (5) indicate the number of stripes constraint for playout and traffic flow, respectively. Constraint (6) ensures that there is no flow to the root node (the media source node, *Node* 0) in each of the interior-disjoint trees. Constraint (7) indicates that no node transmits to itself. Constraint (8) means that at least there is one outgoing flow from the media source node (*Node* 0) in each of the interior-disjoint trees. Constraint (9) indicates that the downstream flow from a node is upper-bounded by incoming flow from its parent node. Constraint (10) means that there are exactly $n-1$ edges in each of the interior-disjoint trees (we have n nodes). Constraint (11) ensures that the final trees are interior-disjoint. Constraint (12) guarantees that there is no cycle in each of the final interior-disjoint trees. This is due to the fact that this constraint *recursively* guarantees that the number of included edges is less or equal to the number of included vertices minus one, which eliminates any cycles in the tree [4]. Finally, Constraint (13) ensures that the media playout quality fluctuation among all peers is constrained and the parameter η can be set empirically based on user experiences.

3.2 Mesh-based P2P Streaming

As illustrated in Figure 1 that in mesh-based P2P streaming each participating peer node can have multiple parent nodes, which collectively fill its incoming link bandwidth, delivering as much sub-streams as possible to the given node. As described in [12-13], media data delivery is accomplished within two phases: the push-based diffusion phase and the pull-based swarming phase. We first consider the formation of the diffusion tree, rooted from the source node to reach every other participating peer node. Then we construct the swarming relationship among those peer nodes, where the swarming delivery operations only occur from a leaf node to the intermediate nodes or other leaf nodes in the diffusion tree.

Besides the notations at the beginning of Section 3, we have the following additional definitions for mesh-based P2P streaming. Let $f^T_{i,j,k}$ denote the number of stripes transmitted from node i to node j (no other peer nodes inbetween) during the k^{th} epoch t_k along the diffusion tree. Likewise, let $f^W_{i,j,k}$ denote the number of stripes transmitted from node i to node j (no other peer nodes inbetween) during the k^{th} epoch t_k in the swarming delivery phase of operations. We assume that variable $x^T_{i,j,k}$ is equal to one if there is a transmission from node i to node j (no other peer nodes inbetween) during the k^{th} epoch t_k along the diffusion tree (see Fig. 1), and zero otherwise. Likewise, we assume that variable $x^W_{i,j,k}$ is equal to one if there is a transmission from node i to node j (no other peer nodes inbetween) during the k^{th} epoch t_k in the swarming delivery phase of operations (see Fig. 1), and zero otherwise. Let us define a decision variable $x_{e,T}$, which is equal to one if edge e is included in the diffusion tree, and zero otherwise.

To facilitate our discussion, we summarize some of the notations in the following table.

NOTATIONS	
$f^T_{i,j,k}$	the number of strips directly transmitted from node i to node j during the k^{th} epoch t_k along the diffusion tree
$f^W_{i,j,k}$	the number of strips directly transmitted from node i to node j during the k^{th} epoch t_k in the swarming delivery phase
$x^T_{i,j,k}$	the decision variable to indicate if there is a direction transmission from from node i to node j during the k^{th} epoch t_k along the diffusion tree
$x^W_{i,j,k}$	the decision variable to indicate if there is a direction transmission from from node i to node j during the k^{th} epoch t_k in the swarming delivery phase

Table 2: Some notations for the MIP formulation of mesh-based P2P streaming.

For the mesh-based P2P streaming, we have the MIP formulation as follows:

$$\text{maximize} \quad \sum_{i=1}^{n-1}\sum_{j=1}^{N} s_{i,j} \qquad (14)$$

subject to:

$$\frac{R_k}{S}\times\sum_{j}(f_{i,j,k}^{T}+f_{i,j,k}^{W})\le b_i^O \text{ , for } 0\le i\le n-1,\ 0\le k\le N \quad (15)$$

$$\frac{R_k}{S}\times\sum_{i}(f_{i,j,k}^{T}+f_{i,j,k}^{W})\le b_j^I \text{ , for } 0\le j\le n-1,\ 0\le k\le N \quad (16)$$

$$s_{i,k}=\sum_{j}(f_{j,i,k}^{T}+f_{j,i,k}^{W})\le S \text{ , for } 0\le i\le n-1,\ 0\le k\le N \quad (17)$$

$$f_{i,j,k}^{T}+f_{i,j,k}^{W}\le S \text{ , for } 0\le i,j\le n-1,\ 0\le k\le N \quad (18)$$

$$\sum_{i=0}^{n-1}(x_{i,0,k}^{T}+x_{i,0,k}^{W})=0 \text{, for } 0\le k\le N \quad (19)$$

$$x_{i,i,k}^{T}+x_{i,i,k}^{W}=0 \text{ , for } 0\le i\le n-1,\ 0\le k\le N \quad (20)$$

$$\sum_{j=1}^{n-1}x_{0,j,k}^{T}\ge 1 \text{, for } 0\le k\le N \quad (21)$$

$$f_{i,j,k}^{T}\le f_{m,i,k}^{T} \text{ if } x_{m,i,k}^{T}=1 \text{, for } 1\le j\ne i\le n-1,$$
$$0\le k\le N \quad (22)$$

$$\sum_{i=0}^{n-1}\sum_{j=1}^{n-1}x_{i,j,k}^{T}=n-1 \text{ , for } 0\le k\le N \quad (23)$$

Now, let us define A_T as a non-empty set ($A_T\subset V$) and $E(A_T)=\{(i,j)\in E\mid(i,j)\in A_T\}$. We have the following constraint to eliminate all cycles in the diffusion tree:

$$\sum_{e\in E(A_T)}x_e\le\mid A_T\mid-1 \text{, for } A_T\subset V \text{ , } A_T\ne\Phi \text{ ,} V \quad (24)$$

$$\text{if } x_{i,j,k}^{W}=1 \text{ , then } \sum_{j}x_{i,j,k}^{T}=0 \text{ , } 1\le j\ne i\le n-1,$$
$$0\le k\le N \quad (25)$$

$$\text{if } x_{i,j,k}^{W}=1 \text{ , then } x_{j,i,k}^{T}=0 \text{ , } 1\le j\ne i\le n-1,$$
$$0\le k\le N \quad (26)$$

For the bandwidth per flow ratio constraint for each connection, we have:

if $x_{i,j,k}^{W}=1$ or $x_{i,j,k}^{T}=1$, then

$$abs\left(\frac{b_i^O}{\sum_{l}(x_{i,l,k}^{T}+x_{i,l,k}^{W})}-\frac{b_j^I}{\sum_{l}(x_{l,j,k}^{T}+x_{l,j,k}^{W})}\right)\le\gamma,$$
$$1\le j\ne i\le n-1,\ 0\le k\le N, \quad (27)$$

where γ is a given parameter.

$$\sum_{i=1}^{n-1}\sum_{j=1}^{N}abs(s_{i,j-1}-s_{i,j})\le\eta \quad (28)$$

Our objective is to maximize the overall received stream data among all peers with constrained media playout quality fluctuation (Constraint 28) given limited network resources in a mesh-based P2P streaming framework. Constraint (15) indicates that the outgoing bandwidth has to be limited by the outgoing link capacity at each peer node. Likewise, Constraint (16) requires that the incoming bandwidth have to be limited by the incoming link capacity at each peer node. Constraint (17) and Constraint (18) indicate the number of stripes constraint for playout and traffic flow, respectively. Constraint (19) ensures that there is no flow to the root node (the media source node, *Node* 0) in both diffusion and swarming phases. Constraint (20) indicates that no node transmits to itself. Constraint (21) means that at least there is one outgoing flow from the media source node (*Node* 0) in the diffusion tree. Constraint (22) indicates that the downstream flow from a node is upper-bounded by incoming flow from its parent node in the diffusion tree. Constraint (23) means that there are exactly $n-1$ edges in the diffusion tree (we have n nodes). Constraint (24) guarantees that there is no cycle in the diffusion tree, which is part of the final mesh-based overlay. This is due to the fact that this constraint *recursively* guarantee that the number of included edges is less or equal to the number of included vertices minus one, which eliminates any cycles in the tree. Constraint (25) indicates that the swarming delivery operation only occurs from a leaf node to other leaf nodes or intermediate nodes in the diffusion tree. Constraint (26) ensures that there is no swarming delivery operation from a child node to its parent node in the diffusion tree. Constraint (27) means that the bandwidth per flow ratio has to be roughly the same for each connection in the final mesh-based overlay. Finally, Constraint (28) ensures that the media playout quality fluctuation among all peers is constrained.

3.3 Contribution-Awareness

To provide tangible incentives to encourage peer nodes to increase their contributions, in [16] a contribution-aware overlay broadcast framework is presented to ensure that it distributes more bandwidth to nodes that contribute more.

$$\alpha\le\left.\sum_{k=0}^{t}s_{i,k}\middle/\sum_{j=0}^{n-1}\sum_{k=0}^{t}f_{i,j,k}\right.\le\beta,\qquad 0\le i\le n-1,0\le t\le N \quad (29)$$

where α and β are threshold parameters to ensure contribution-awareness and the variable k denotes the k^{th} epoch in time.

4. Methods to Solve MIP

The most widely-used approaches to solve the integer programs (IPs) is to intelligently and efficiently search the solutions to the related linear programs (LPs) and check if the integer conditions are satisfied.

With appropriate transformations, the presented mixed integer programming formulations can be converted into the following standard matrix forms:

$$\max\ c'x+h'y$$

subject to: $Ax + By = b$

$$x \in Z_+^n,\ y \geq 0 \qquad (30)$$

where x and y are the decision vectors, matrix A, B and vector b are determined by the constraints in the formulations.

After the transformation to the above standard matrix form, Gomory cutting plane algorithm or branch and bound method can be used to obtain the optimal solution based on the solutions for the corresponding linear programming (LP) problem.

An elastic constraint approach was presented in [14], aiming at increasing the possibility of finding the feasible solution of the problem. In [10], Glover outlined some key areas for integer programming including controlled randomization, learning strategies, induced decomposition and tabu search.

5. Conclusions

In this paper, we present mixed integer programming (MIP) formulations to analyze multiple interior-disjoint-tree-based and mesh-based P2P media streaming for application level multicast (ALM). Network optimization is the key to simultaneously satisfy a large group (potentially millions) of peers' needs given limited network resources. The key to our analytical approach is to cast the P2P media streaming problem as a constraint system. P2P media streaming may give rise to a wide range of new and exciting applications such as party media streaming, sport media streaming, lecture media streaming, etc. We also discuss the means to obtain the optimal solution based on the MIP formulations.

Acknowledgement

Fulu Li would like to thank the Digital Life consortium at MIT Media Lab for the support.

Reference

[1] Badishi G., I. Keidar, R. Melamed, "Towards Survivability of Application-Level Multicast", in *FuDiCo II*, June 2004.

[2] Banerjee, S., B. Bhattacharjee and C. Kommareddy, "Scalable Application Layer Multicast", in *SIGCOMM '2002*.

[3] Baset S., H. Schulzrinne, "An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol", in IEEE *INFOCOM '2006*.

[4] Bertsimas, D., J. Tsitsiklis, "Introduction to Linear Optimization", *Athena Scientific Press*, 1997.

[5] Byers J., J. Considine, M. Mitzenmacher, S. Rost, "Informed Content Delivery Across Adaptive Overlay Networks", in IEEE *Trans. on Networking*, Oct. 2004.

[6] Castro M., P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, A. Singh, "SplitStream: High-Bandwidth Content Distribution in Cooperative Environments", in *SOSP '2003.*

[7] Chu Y., S. G. Rao, H. Zhang, "A Case for End System Multicast", "IEEE *JSAC*, 2002.

[8] Cohen B., "Incentives Build Robustness in BitTorrent", in Workshop on Economics of Peer to Peer Systems 2003.

[9] Decuetos P. and K.W. Ross, "Unified Framework for Optimal Video Streaming", in IEEE *INFOCOM '2004*.

[10] Glover F., "Future Paths for Integer Programming and Links to Artificial Intelligence", in *Computers and Operations Research*, 1986.

[11] Jennigs C., D. Bryan, "P2P For Communications: Beyond File Sharing", in Business Communication Review, Feb. 2006.

[12] Magharei N., R. Rejaie, Y. Guo, "Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches", in the proceeding of *IEEE INFOCOM* '2007.

[13] Magharei N., R. Rejaie, "PRIME: Peer-to-Peer Receiver-drIven Mesh-based Streaming", in *IEEE INFOCOM* '2007.

[14] Nauss R.M., "Bond Portfolio Analysis using Integer Programming", in *Financial Optimiz*ation, 1993.

[15] Rajendran R., D. Rubenstein, "Optimizing the Quality of Scalable Video Streams on P2P Networks", in *Globecom '2004*.

[16] Sung Y., M. Bishop, S. Rao, "Enabling Contribution Awareness in an Overlay Broadcasting System", in the proceeding of ACM *SIGCOMM '2006*.

[17] Zhang X., J. Liu, B. Li, T. Yum, "Coolstreaming: A Data-driven Overlay Network for Live Media Streaming", in the proceeding of IEEE *INFOCOM '2005*.